A Reaction Attack on the QC-LDPC McEliece Cryptosystem

Tomáš Fabšič[1⋆], Viliam Hromada[1⋆], Paul Stankovski[2], Pavol Zajac[1⋆], Qian Guo[2], Thomas Johansson[2]

[1] Slovak University of Technology in Bratislava
Faculty of Electrical Engineering and Information Technology
Ilkovičova 3, 81219 Bratislava, Slovak Republic
{tomas.fabsic, viliam.hromada, pavol.zajac}@stuba.sk
[2] Department of Electrical and Information Technology,
Lund University, Lund, Sweden
{qian.guo, thomas.johansson, paul.stankovski}@eit.lth.se

Abstract. Guo et al. recently presented a reaction attack against the QC-MDPC McEliece cryptosystem. Their attack is based on the observation that when a bit-flipping decoding algorithm is used in the QC-MDPC McEliece, then there exists a dependence between the secret matrix H and the failure probability of the bit-flipping algorithm. This dependence can be exploited to reveal the matrix H which constitutes the private key in the cryptosystem. It was conjectured that such dependence is present even when a soft-decision decoding algorithm is used instead of a bit-flipping algorithm.
This paper shows that a similar dependence between the secret matrix H and the failure probability of a decoding algorithm is also present in the QC-LDPC McEliece cryptosystem. Unlike QC-MDPC McEliece, the secret key in QC-LDPC McEliece also contains matrices S and Q in addition to the matrix H. We observe that there also exists a dependence between the failure probability and the matrix Q. We show that these dependences leak enough information to allow an attacker to construct a sparse parity-check matrix for the public code. This parity-check matrix can then be used for decrypting ciphertexts.
We tested the attack on an implementation of the QC-LDPC McEliece using a soft-decision decoding algorithm. Thus we also confirmed that soft-decision decoding algorithms can be vulnerable to leaking information about the secret key.

Keywords: QC-LDPC McEliece cryptosystem, reaction attack, soft-decision decoding.

1 Introduction

In 1978, R. J. McEliece proposed a public key cryptosystem based on coding theory [8], now called the McEliece cryptosystem. The cryptosystem has never

⋆ Support by grant VEGA 1/0159/17 is acknowledged.

been adopted widely, mainly due to the large size of the public key. The interest in the McEliece cryptosystem has, however, risen recently, since it has become a candidate for post-quantum cryptography.

In [2], Baldi and Chiaraluce proposed a variant of the McEliece cryptosystem based on quasi-cyclic low-density parity-check codes (QC-LDPC codes). Their cryptosystem is now known as the QC-LDPC McEliece cryptosystem. The use of quasi-cyclic codes in this cryptosystem allows to reduce the size of the public key. However, in [10], Otmani et al. showed that the proposed system had serious vulnerabilities. In [3], Baldi et al. proposed an amended version of the cryptosystem which was immunized against the attacks from [10]. An important role in the cryptosystem is played by matrices which are formed by blocks of circulant matrices. In [12], it was demonstrated that when the block size is chosen to be an even number a more efficient information-set decoding attack on the cryptosystem can be executed. However, this attack is not applicable when the block size is odd.

A cryptosystem related to the QC-LDPC McEliece cryptosystem, the QC-MDPC McEliece cryptosystem, was proposed by Misoczki et al. in [9]. Both QC-LDPC McEliece and QC-MDPC McEliece use an iterative decoding algorithm in their decryption procedure. Two types of iterative decoding algorithms are proposed in the literature; bit-flipping algorithms and soft-decision decoding algorithms. Both types of algorithms fail with some small probability. In [5], Guo et al. demonstrated that when the QC-MDPC McEliece cryptosystem is implemented with a bit-flipping algorithm, there exists a dependence between the secret matrix H and the failure probability of the bit-flipping algorithm. They further demonstrated that this dependence allows an attacker to recover the secret matrix H very efficiently. They conjectured that such dependence is present when a soft-decision decoding algorithm is used, as well.

In the present paper, we show that a similar dependence between the secret matrix H and the failure probability of a decoding algorithm is also present in the QC-LDPC McEliece cryptosystem. Unlike in QC-MDPC McEliece, the secret key in QC-LDPC McEliece also contains matrices S and Q in addition to the matrix H. We observe that there also exists a dependence between the failure probability and the matrix Q. We show that these dependences leak enough information to allow an attacker to construct a sparse parity-check matrix for the public code. This parity-check matrix can then be used for decrypting ciphertexts.

For our experiments we used an implementation of the QC-LDPC McEliece cryptosystem which uses a soft-decision decoding algorithm. Thus, apart from showing that an attack similar to the one in [5] can be mounted on the QC-LDPC McEliece cryptosystem, we also confirm the conjecture from [5] that these types of attacks are also possible when a soft-decision decoding algorithm is used instead of a bit-flipping algorithm.

The paper is structured as follows. In Section 2, we review the QC-LDPC McEliece cryptosystem, the QC-MDPC McEliece cryptosystem and the attack on the QC-MDPC McEliece from [5]. In Section 3, we describe a new attack on

the QC-LDPC McEliece. Finally, in Section 4, we summarize our results and conclude the paper.

2 Preliminaries

2.1 The QC-LDPC McEliece Cryptosystem

In [2], Baldi et al. proposed a variant of the McEliece cryptosystem based on LDPC codes – the QC-LDPC McEliece cryptosystem. A part of the private key in this cryptosystem is formed by an $(n-k) \times n$ parity-check matrix H of an LDPC code able to correct t errors. The matrix H is formed by a row $\{H_0, \ldots, H_{n_0-1}\}$ of $n_0 = n/(n-k)$ binary circulant blocks of size $p \times p$, where $p = n - k$. Each block has a row weight (i.e. the number of ones in a row) equal to a number w which is small compared to p. If H_{n_0-1} is invertible, a generator matrix G for the code can be obtained as

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

The remaining part of the private key is formed by two other matrices; an invertible $k \times k$ matrix S and a sparse invertible $n \times n$ matrix Q. The matrices S and Q are formed by blocks of circulant $p \times p$ matrices. In addition, Q has a fixed row weight m. The public key is then computed as $G' = S^{-1} \cdot G \cdot Q^{-1}$.

Encryption is done as follows. Let the original message be u. Alice encrypts u as $x = u \cdot G' + e$, where e is a randomly generated error vector of length n and Hamming weight $w_{\mathrm{H}}(e) = t' \leq \frac{t}{m}$.

When Bob receives the encrypted message x, he first computes

$$x' = x \cdot Q = u \cdot S^{-1} \cdot G + e \cdot Q.$$

The vector x' is a codeword of the LDPC code chosen by Bob (corresponding to the information vector $u' = u \cdot S^{-1}$), affected by the error vector $e \cdot Q$, whose maximum weight is t. Bob is able to correct all the errors with very high probability by means of LDPC decoding, thus recovering u', and then u through a post-multiplication by S.

In [10], Otmani et al. demonstrated that this cryptosystem is vulnerable to attacks which exploit the facts that Q is block-diagonal and S is sparse. In order to immunize their cryptosystem against these attacks, Baldi et al. proposed versions of the QC-LDPC McEliece cryptosystem with the matrix S dense and the matrix Q no longer block-diagonal in [3].

In [12], it was demonstrated that when the value of the block size is chosen to be an even number, a more efficient information-set decoding attack on the cryptosystem can be executed. However, this attack is not applicable when the block size is odd.

2.2 The QC-MDPC McEliece Cryptosystem

The QC-MDPC McEliece cryptosystem was proposed in [9]. This cryptosystem uses moderate density parity check (MDPC) codes, which are codes that admit a parity check matrix H^{MDPC} which is sparse, but not as sparse as in LDPC codes. The matrix H^{MDPC} again has to be quasi-cyclic, i.e. it has to be formed by a row of circulant blocks $\left\{H_0^{\mathrm{MDPC}}, \ldots, H_{n_0-1}^{\mathrm{MDPC}}\right\}$. The matrix H^{MDPC} forms the whole private key in the QC-MDPC McEliece cryptosystem. If $H_{n_0-1}^{\mathrm{MDPC}}$ is invertible, a generator matrix G^{MDPC} for the code can be obtained by the same calculation as in QC-LDPC McEliece. The matrix G^{MDPC} forms the public key for the cryptosystem.

Encryption is done as follows. Let the original message be u. Alice encrypts u as $x = u \cdot G^{\mathrm{MDPC}} + e$, where e is a randomly generated vector with the Hamming weight equal to a number of errors t^{MDPC} that the MDPC code can correct.

When Bob receives the encrypted message x, he is able to correct all the errors with very high probability by means of an LDPC decoding algorithm, thus recovering the message u.

2.3 Previous Attack on the QC-MDPC McEliece Cryptosystem

In [5], Guo et al. presented a reaction attack on the QC-MDPC McEliece cryptosystem. They demonstrate that if the QC-MDPC McEliece cryptosystem employes a bit-flipping decoding algorithm in its decryption procedure, then there exists a dangerous dependence between the probability of decoding error and the secret key.

Guo et al. demonstrate their attack on a version of the cryptosystem with two blocks in the secret parity check matrix H^{MDPC}. Since the blocks are circulant, the block H_0^{MDPC} is determined by its first row h_0^{MDPC}. They show that an attacker who sends a large number of messages encrypted by the public key and for each message learns whether it was successfully decrypted can learn distances between ones in h_0^{MDPC}. The distance between two ones in positions p_1 and p_2, $p_2 > p_1$, in h_0^{MDPC} is defined as $\min\{p_2 - p_1, p - (p_2 - p_1)\}$, where p is the length of h_0^{MDPC} (i.e. the distance is computed cyclically). With the knowledge of distances in h_0^{MDPC}, the attacker can reconstruct h_0^{MDPC} and recover the private key.

Guo et al. consider two different scenarios in their paper. In the first scenario, the attacker is allowed to choose the error vector e that is added to the message during encryption. In the second scenario, the attacker has no such freedom and the error vector is always chosen at random. Here we focus on the second scenario.

In the second scenario, the attacker sends a large number of messages containing a randomly generated error vector. The attacker then groups the messages into sets Σ_d, $d \in \{1, \ldots p/2\}$ by the following principle: a message belongs to the set Σ_d if its error vector contains the distance d. Guo et al. observe that if d is present in h_0^{MDPC}, then the estimate for the probability of decoding failure based on the set Σ_d is smaller than the estimate obtained from Σ_d when d is

not present in h_0^{MDPC}. Thus, the attacker is able to learn which distances are present in h_0^{MDPC}.

3 The Attack

As in [5], we also consider an attacker who sends a large number of messages encrypted by the public key and for each message learns whether it was successfully decrypted. Similarly to the more restrictive attack scenario in [5], we assume that the attacker has no freedom to choose the error vector e that is added to the message during encryption, i.e. the error vector is always generated randomly. We will demonstrate that the attacker can learn information about the matrices H and Q which allow him to construct a sparse parity check matrix for the public code. Using this matrix, the attacker can then decrypt ciphertexts encrypted by the cryptosystem.

Similarly to [5], a special role in our attack is played by distances between ones in matrices H and Q. Following [5], we define the distance between two ones in positions p_1 and p_2, $p_2 > p_1$, in a vector of length p as $\min\{p_2 - p_1, p - (p_2 - p_1)\}$ (i.e. the distance is computed cyclically).

3.1 Learning Distances in the Matrix H - Intuition

The key observation from [5] can be loosely rephrased as: "Let e be an error vector divided into blocks of length p. Suppose that a block of e contains the distance d. If the distance d is also present in the corresponding circulant block of the matrix H^{MDPC}, then a bit-flipping algorithm fails to decode a message with error vector e less frequently."

We now analyze whether this behaviour could be utilized in attacking the QC-LDPC McEliece cryptosystem. In QC-LDPC McEliece, the decoding algorithm is not applied to e but to eQ, where Q is secret. Thus, we face the question: can the attacker learn whether a given distance d is present in eQ?

The answer to this question is positive. Suppose that the attacker knows that e has the distance d in its first block of p digits. We can think of the multiplication of e and Q as an addition of those rows of Q for which the corresponding entries in e are one. Thus, if distance d is present in e, two rows of Q, q_i and $q_{i+d \mod p}$, will be added together in the multiplication process. Since the distance d is present in a block of length p in e and since Q is composed of circulant blocks of dimension $p \times p$, the blocks of length p in $q_{i+d \mod p}$ are cyclic shifts of the corresponding blocks in q_i. The row q_i has m ones, with m being a small number. Thus, the vector $q_i + q_{i+d \mod p}$ contains m pairs of ones separated by the distance d, unless an unlikely cancellation occurs. Since all the rows of Q are sparse, we can expect these pairs to remain in eQ, undisturbed by additions of further rows of Q. The attacker therefore knows that the distance d will be present in eQ.

Note that the distance d will always appear in all blocks of eQ. This means that when the attacker estimates the decoding error probability, he can only hope

to learn whether the distance d is present in one of the blocks of H. Unlike the QC-MDPC case, the attacker will not learn whether d is present in one particular block of H. This could potentially make the subsequent reconstruction of H more involved. However, later we show that this is not a serious issue and that H can still be reconstructed efficiently.

These ideas give us hope that reconstruction of H is possible in the QC-LDPC McEliece cryptosystem with a bit-flipping decoding algorithm. Also, similarities between bit-flipping algorithms and soft-decision decoding algorithms give us further hope that this reconstruction is possible even for QC-LDPC McEliece with a soft-decision decoding algorithm.

3.2 Learning Distances in the Matrix Q - Intuition

The matrix H, however, forms only a part of the private key. The rest of the private key is formed by matrices S and Q. Here we argue that the attacker can even learn information about distances in the matrix Q.

Let q_i be the i-th row of Q. Suppose that the row q_1 contains a distance d in one of its blocks of length p. Suppose that the attacker knows that the error vector contains distance d in its first block of length p. Then two rows q_i and $q_{i+d \mod p}$ will be added together during the multiplication of e and Q. Since Q is composed of circulant blocks of size $p \times p$, both rows q_i and $q_{i+d \mod p}$ will contain the distance d in the same block of length p. Suppose that q_i contains the ones separated by the distance d in positions $j \times p + s$ and $j \times p + (s + d \mod p)$. Then $q_{i+d \mod p}$ will contain ones in positions $j \times p + (s + d \mod p)$ and $j \times p + (s + 2d \mod p)$. Thus, the ones in the position $j \times p + (s + d \mod p)$ will cancel in $q_i + q_{i+d \mod p}$. Since the matrix Q is very sparse, we normally expect $w_{\mathrm{H}}(eQ) = m \times w_{\mathrm{H}}(e)$. The cancellation described above will decrease the Hamming weight of eQ below its standard Hamming weight. Consequently, the decoding algorithm in the QC-LDPC McEliece will have to correct fewer errors than normally. Therefore we can expect the probability of the decoding error to decrease severely when e contains the distance d in its first block of length p. We can expect this effect to be present in both bit-flipping and soft-decision decoding algorithms. Thus, observing the probability of the decoding error, the attacker can learn whether the distance d is present in one of the blocks of length p of the row q_1. Again, the attacker can not learn exactly which block the distance is present in. Similarly, the attacker can learn about the presence of a distance d in rows $q_{p+1}, q_{2p+1}, \ldots, q_{(n_0-1)p+1}$.

3.3 Learning Distances - Experiments

Below, we present results of our experiments, confirming the intuition from sections 3.1 and 3.2. We used a version of the QC-LDPC McEliece cryptosystem with the following parameters: $n_0 = 3$, $w = 13$, $p = 8192$ and $m = 11$.[3] These

[3] These parameters were selected because they were proposed in [3]. The attack presented in this paper is equally feasible for other sets of parameters, including parameters with p odd.

values were suggested in [3] for 80-bit security. We increased the value of t' to 48 from 40 in the original suggestion to increase the decoding error probability and make it easier to estimate. We discuss the relevance of this change in the conclusion. We constructed matrices S and Q as suggested in [3]. Thus, we constructed the matrix S so that every block in S has rows with weight approximately equal to $p/2$, with blocks along the diagonal having rows with an odd weight and blocks away from the diagonal having rows with an even weight. We obtained the matrix Q by constructing a matrix of 3×3 circulant blocks with the blocks on the diagonal having rows of weight 3 and the blocks away from the diagonal having rows of weight 4, and by randomly permuting its block-rows and block-columns.

Our implementation is based on the project BitPunch [4], which is a free standalone cryptographic library containing implementations of various variants of the McEliece cryptosystem. In our implementation, we used a soft-decision decoding algorithm from [11].

We conducted an experiment to learn what distances are present in the circulant blocks of matrices H and Q. Since the value of p in our cryptosystem was 8192, we were only interested in distances from 1 to 8192/2=4096. To learn the distances, we used a slight variation of Algorithm 4 in [5]. Our variation of the algorithm is presented here as Algorithm 1.

Algorithm 1

INPUT: number N of ciphertexts to generate
OUTPUT: vectors a, b, u and v

1. $a \leftarrow$ zero-initialized vector of length $p/2$
2. $b \leftarrow$ zero-initialized vector of length $p/2$
3. $u \leftarrow$ zero-initialized vector of length $p/2$
4. $v \leftarrow$ zero-initialized vector of length $p/2$
5. $i \leftarrow 0$
6. **while** $i < N$ **do**:
   (a) generate a ciphertext c with a random error vector e
   (b) $s \leftarrow$ distances present in at least one block of length p in e
   (c) $r \leftarrow$ distances present in the first block of length p in e
   (d) $l \leftarrow 1$ if the decoding failure occurs, 0 otherwise
   (e) **for** d from 1 to $p/2$ **do**:
      i. **if** $s[d] \geq 1$ **then**:
         A. $a[d] \leftarrow a[d] + l$
         B. $b[d] \leftarrow b[d] + 1$
      ii. **if** $r[d] \geq 1$ **then**:
         A. $u[d] \leftarrow u[d] + l$
         B. $v[d] \leftarrow v[d] + 1$
   (f) $i \leftarrow i + 1$

The algorithm decrypts a large number of messages with randomly generated error vectors. The algorithm uses two vectors of counters: a and b. Each vector of

counters has length 4096 and is initialized as the zero vector. After the algorithm decrypts a ciphertext c with an error vector e, the algorithm computes distances between ones in every block of length p in e. If a distance d is present in one of the blocks of e, the value of $b[d]$ is increased by 1. If a distance d is present in one of the blocks of e and there occurred a decoding error when decrypting c, the value of $a[d]$ is increased by 1. Thus, after a large number of ciphertexts is processed, the ratio $\frac{a[d]}{b[d]}$ estimates the probability of the decoding failure for ciphertexts with error vectors containing a distance d.

Our variation of the algorithm in addition uses two other vectors of counters: u and v. They again have length 4096 and are initialized as zero vectors. Vectors u and v are useful for reconstruction of the first block-row of Q. Similarly as a and b, they are updated every time the algorithm decrypts a new ciphertext. If a distance d is present in the first block of the error vector e, the value of $v[d]$ is increased by 1. If a distance d is present in the first block of e and there occurred a decoding error when decrypting the ciphertext, the value of $u[d]$ is increased by 1. Thus, after a large number of decryptions, the ratio $\frac{u[d]}{v[d]}$ estimates the probability of the decoding failure for ciphertexts with error vectors containing a distance d in its first block.

We decrypted 103 million ciphertexts. The resulting probability estimates $\frac{a[d]}{b[d]}$ are presented in Fig. 1.

If d was present in one of the circulant blocks of Q the estimates ranged from 0.095 to 0.109. If d was present in one of the circulant blocks of H the estimated probability typically ranged from 0.110 to 0.118. For four distances in H the probability was below this range but this was due to the fact that these distance were present in Q at the same time. If a distance d was present neither in Q nor in H, the estimated probability ranged from 0.115 to 0.122. Thus, our experiment confirms the expectation that the lowest probabilities are obtained for distances in Q and that probabilities for distances in H are on average lower than probabilities for distances which are neither in Q nor in H.

3.4 Distance Spectrum Reconstruction Problem

In order to explain how the attacker can reconstruct the secret matrices H and Q, we need to consider the problem of recovering a circulant matrix C, provided we only know the distances in C. This problem was already introduced in [5]. However, here we present a different approach to the problem, translating the problem into a graph problem.

Let us consider a circulant matrix C of the dimension $p \times p$. Let $P = \{p_0, p_1, \ldots, p_{w-1}\}$ be the ordered sequence of positions of ones in the first row of C. We define the distance spectrum of P as the set

$$DS(P) = \{p_i - p_j \mod p; p_i, p_j \in P\}.$$

Suppose we know the distance spectrum D and we want to learn the matrix C. Since every row of C gives rise to the same distance spectrum, we can only

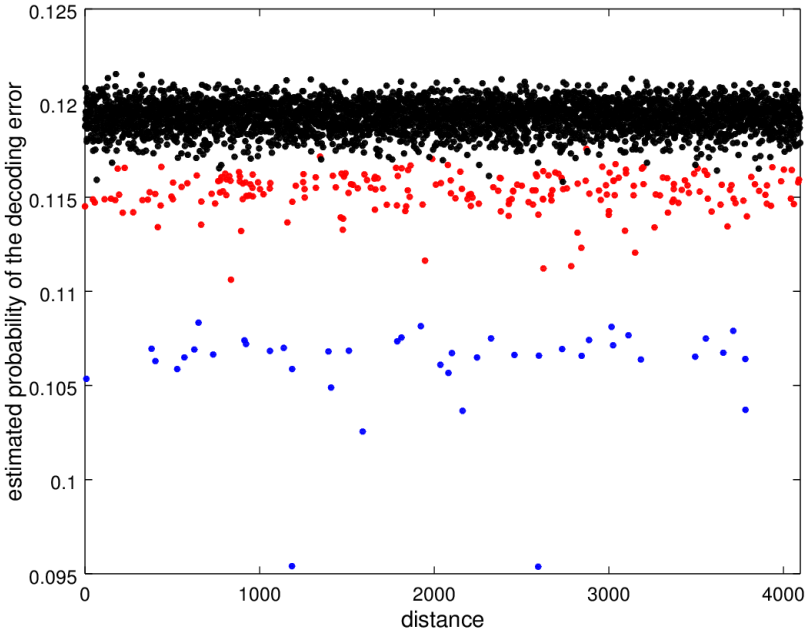

Fig. 1. Estimates of the probability of the decoding error from the experiment in Section 3.3. Distances in one of the circulant blocks of Q are marked in blue. Distances in one of the circulant blocks of H are marked in red. Distances which are present neither in Q nor in H are marked in black.

hope to learn C up to a shift of rows. Thus we can look for all sets P such that $DS(P) = D$ with the additional constraint that $p_0 = 0$. In addition, we know that the smallest distance in D must correspond to a distance between two cyclically consecutive ones. Thus, we can add the additional constraint that $p_1 = \min(D)$.

Definition 1. *Distance spectrum reconstruction (DSR) problem: Given a set D, find all P such that $DS(P) = D$, $p_0 = 0$ and $p_1 = \min(D)$.*

It is easy to show that if a set $P = \{p_0, p_1, p_2, \ldots, p_{w-1}\}$ is a solution to the DSR problem, then so is the set

$$P' = \{p_0, p_1, p - p_{w-1} + p_1, p - p_{w-2} + p_1, \ldots, p - p_2 + p_1\}.$$

Given the spectrum D, let us define the graph G_D as follows: a set of vertices is given by the set D. Edge (d_i, d_j) exists, if and only if $d_i - d_j \mod p \in D$. If $DS(P) = D$, the induced subgraph $G_D[P]$ is a complete graph.

We will change the DSR problem into a graph problem: Given graph G_D, find a clique of w vertices, that contains vertices $\{p_0 = 0, p_1 = \min(D)\}$. From each clique, we obtain a candidate for a solution P of the DSR problem. The candidate sequence P can be verified by checking whether $DS(P) = D$ holds.

It is well known that the clique problem is NP-hard in general. In our experiments, we exploit the fact that the spectrum D and the graph G_D are sparse. In sparse graphs, we expect to find only a small number of possible w-cliques.

Instead of looking for w-cliques directly, we filter potential sets of positions with the following algorithm:

Algorithm 2

INPUT: set of distances D, size of cliques w
OUTPUT: set of candidates for w-cliques

---

1. (*Identify 3-cliques*) Find a set of candidates A: $\forall p_2 \in A$: $\{0, p_1, p_2\}$ is a 3-clique. This can be checked by testing for each $p_2 \in D \setminus \{0, p_1\}$ whether $p_2 - p_1 \in D$.
2. (*Combine 3-cliques*) Set $E = \emptyset$. For each pair (p_1, p_2), $p_2 \in A$:
   (a) Construct set B: $\forall p_3 \in B$: $\{0, p_1, p_2, p_3\}$ is a 4-clique. This can be checked by testing each $p_3 \in A \setminus \{p_2\}$, whether $p_3 - p_2 \in D$.
   (b) (*Filter 1*) If $|B| < w - 3$, try another pair (p_1, p_2).
   (c) Repeat: Remove from $C = \{0, p_1, p_2\} \cup B$ all elements that are not connected to at least $w - 1$ until either $|C| < w$, or no more elements can be removed.
   (d) (*Filter 2*) Remove all C's with $|C| < w$.
   (e) Set $E = E \cup \{C\}$.
3. Return E.

After the algorithm finishes, E contains sets of positions, that can contain an original position sequence P. Clearly, if some set $C \in E$ contains exactly w

elements, it must form a w-clique: there are exactly w vertices in the induced subgraph, and each is connected to $w-1$ other vertices. If the size of C is greater than w, we can apply further clique finding algorithms to this set.

We implemented a controlled experiment, where we tried to reconstruct a randomly generated sequence of positions. We used parameters $n = 8192, w = 13$. Out of 1000 experiments, Algorithm 2 reported surplus results (4 or 3 sets instead of the expected 2) only in 10 cases. The set of 1000 experiments with software written in Python took 558s on Intel i7-3820 CPU @ 3.60GHz.

3.5 Reconstructing the Matrix H

Suppose that the attacker performed the experiment from Section 3.3. Due to the intuition presented in Section 3.1 he expects that if a distance d is present in one of the circulant blocks of the matrix H, then the estimate $\frac{a[d]}{b[d]}$ of the probability of the decoding error will be lower than normal. Thus he might select distances for which the estimated probability in the experiment was below some threshold and try to reconstruct the matrix H from these distances. Let D'_T be the set of distances for which the estimated probability in the experiment was below a threshold T. The attacker can create a set $D_T = \{d : d \in D'_T \text{ or } p - d \in D'_T\}$. Let P_i be the ordered sequence of positions of ones in the first row of H_i. Assuming that $DS(P_i) \subset D_T \ \forall i$, the attacker can try to solve the following variation of the DSR problem:

Problem 1. Given a set D_T, find all P such that $|P| = w$, $DS(P) \subset D_T$, $p_0 = 0$ and $p_1 = \min(DS(P))$.

If P satisfies all the conditions in the problem, it becomes a candidate for a row in one of the blocks H_i. Similarly as in the DSR problem, if a set $P = \{p_0, p_1, p_2, \ldots, p_{w-1}\}$ satisfies the conditions in Problem 1, then so does the set $P' = \{p_0, p_1, p - p_{w-1} + p_1, p - p_{w-2} + p_1, \ldots, p - p_2 + p_1\}$.

We attempted to solve Problem 1 using the data presented in Fig. 1 and the threshold $T = 0.118$. Running a variant of Algorithm 2 on a standard PC[4], we instantly obtained $n_0 = 3$ pairs of solutions (P, P'). Upon observing such result, the attacker knows that with a very high probability only one sequence in each pair (P, P') represents a row in one of the blocks H_i and for every two different pairs these sequences correspond to rows in distinct blocks H_i and H_j. Let P_1 be the set of positions of ones in the first row of H_1. If we reorder rows of H by a cyclical shift, the resulting matrix will still be a parity check matrix for the private code composed of circulant blocks. Thus the attacker can assume that the first position in P_1 is 0 and that the second position is equal to $\min(DS(P_1))$. Therefore, upon observing solutions to Problem 1 to be n_0 pairs (P, P'), the attacker obtains $(n_0!) \times 2^{n_0} \times p^{n_0-1}$ candidates for the matrix H. For the parameters from Section 3.3 this means obtaining approximately 2^{32} candidates.

[4] In particular, we ran Algorithm 2 with inputs $D = D_{0.118}$ and $w = 13$ for all possible values of p_1. We tested candidates for p_1 in ascending order. After a candidate for p_1 was tested, it was removed from $D_{0.118}$.

3.6 Reconstructing the Matrix Q

Due to the intuition presented in Section 3.2, the attacker expects distances present in circulant blocks in the first block-row of the matrix Q to give the smallest ratios $\frac{u[d]}{v[d]}$ in the experiment from Section 3.3. This was the case in our experiment, where for distances present in circulant blocks in the first block-row of Q the ratio was always below 0.085, whereas for other distances it was always above 0.105. The graph of the ratios $\frac{u[d]}{v[d]}$ is presented in Fig. 2. Thus the attacker

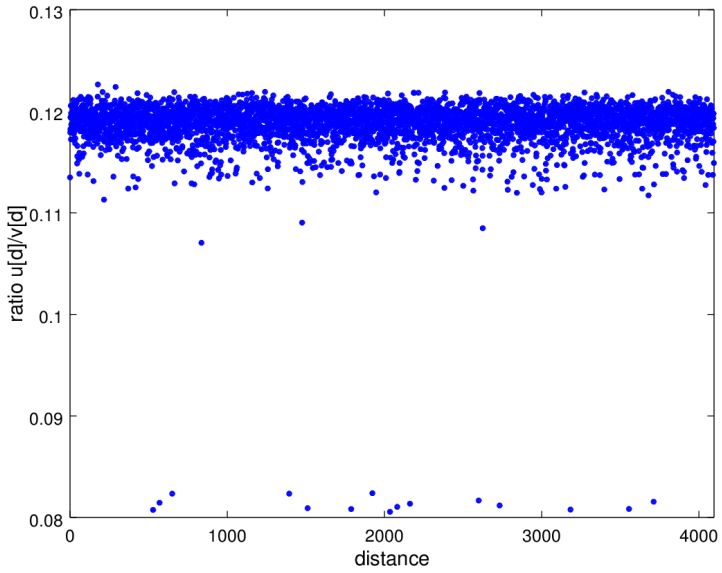

Fig. 2. Ratios $\frac{u[d]}{v[d]}$ from the experiment in Section 3.3. The ratios below 0.09 correspond precisely to the distances present in circulant blocks in the first block-row of Q.

can select distances for which the ratio $\frac{u[d]}{v[d]}$ in the experiment was below some small threshold L and try to reconstruct the first block-row of the matrix Q from these distances. Let D'_L be the set of distances for which the ratio $\frac{u[d]}{v[d]}$ in the experiment was below a threshold L. Suppose that the attacker knows that the Hamming weight of rows in circulant blocks of Q is either w_1 or w_2. (this was the case in the cryptosystems proposed in [3]). Then the attacker can try to solve the following problem:

Problem 2. Given a set $D_L = \{d : d \in D'_L \text{ or } p - d \in D'_L\}$, find all P such that $|P| \in \{w_1, w_2\}$, $DS(P) \subset D_L$, $p_0 = 0$ and $p_1 = \min(DS(P))$.

If P satisfies all the conditions in the problem, it becomes a candidate for a row in one of the blocks in the first block-row of Q. Again, if a set $P =$

$\{p_0, p_1, p_2, \ldots, p_{w-1}\}$ satisfies the conditions in Problem 2, then so does the set $P' = \{p_0, p_1, p - p_{w-1} + p_1, p - p_{w-2} + p_1, \ldots, p - p_2 + p_1\}$.

We attempted to solve Problem 2 for the set $D_{0.085}$ derived from the data from the experiment in Section 3.3. For the cryptosystem used in Section 3.3, it is a public knowledge that every block-row of Q contains two blocks with rows with the Hamming weight 4 and one block with rows with the Hamming weight 3. We found 2 pairs of sequences (P, P') of length 4. For the length 3 we found 5 pairs (P, P') which were not derived from the solutions for the length 4. This result allows the attacker to build a set of $3! \times 2^2 \times 5 \times 2 \times p^3 \approx 2^{47}$ candidates for the first block-row of Q.

Provided that suitable counters are added to Algorithm 1, the attacker can analogously build sets of candidates for other block-rows of Q. However, if the attacker wanted to combine these sets to produce one set of candidates for Q, the resulting set would be too large.

3.7 Learning to Decrypt

Instead of reconstructing the private key $\{H, S, Q\}$, the attacker can try to construct the matrix $\tilde{H} = H \times Q^T$. The matrix $\tilde{H}$ is a parity check matrix of the public code since $G' \cdot \tilde{H}^T = S^{-1} \cdot G \cdot Q^{-1} \cdot Q \cdot H^T = S^{-1} \cdot G \cdot H^T = S^{-1} \cdot 0 = 0$. The matrix $\tilde{H}$ contains at most $n_0 \times w \times m$ ones in a row. Due to the sparsity of the matrix $\tilde{H}$, the attacker can hope to use an LDPC decoding algorithm with $\tilde{H}$ to decrypt an arbitrary message encrypted by the cryptosystem.

The attacker can try to construct the first block of the matrix $\tilde{H}$. For the block $\tilde{H_0}$ it holds that $\tilde{H_0} = \sum_{i=0}^{n_0-1} H_i \left(Q_{0i}\right)^T$. For each H_i, the set of solutions to Problem 1 contains a sequence P^i which represents a row in H_i. Since the first column of a circulant matrix is equal to its last row reversed, the transpose of a circulant matrix generates the same distance spectrum as the original matrix. Therefore, for every $\left(Q_{0i}\right)^T$, the set of solutions of Problem 2 contains a sequence $P^{Q,i}$ which represents a row in $\left(Q_{0i}\right)^T$. For the sequences P^i and $P^{Q,i}$ we consider polynomials $p^i(x)$ and $p^{Q,i}(x)$ obtained as follows: to a sequence $P = \{p_0, p_1, \ldots, p_{s-1}\}$ we allocate the polynomial $p(x) = \sum_{j=0}^{s-1} x^{p_j}$.

Next, we will use the fact that the ring of circulant binary matrices of dimension $p \times p$ is isomorphic to the ring $\mathbb{Z}_2[x]/(x^p + 1)$. The isomorphism maps a circulant matrix with the first row $(c_0, c_1, c_2, \ldots, c_{p-1})$ onto the polynomial $c(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{p-1} x^{p-1}$. Therefore for the polynomial $\tilde{h}_0(x)$ corresponding to the block $\tilde{H_0}$ we have $\tilde{h}_0(x) = \sum_{i=0}^{n_0-1} \left(x^{\alpha_i} p^i(x)\right) \left(x^{\beta_i} p^{Q,i}(x)\right) \pmod{x^p + 1}$ for some $\alpha_i, \beta_i \in \{0, 1, \ldots, p-1\}$. Thus we have $\tilde{h}_0(x) = \sum_{i=0}^{n_0-1} x^{\gamma_i} p^i(x) p^{Q,i}(x) \pmod{x^p + 1}$ for some $\gamma_i \in \{0, 1, \ldots, p-1\}$. If we reorder rows of $\tilde{H}$ by a cyclical shift, the resulting matrix will still be a parity check matrix for the public code. Thus it suffices the attacker to look for the polynomial $\tilde{h}_0(x)$ with $\gamma_0 = 0$.

Suppose that the attacker attacks the cryptosystem which we used in Section 3.3 and suppose that he obtains the same number of solution to Problem 1 and Problem 2 as we obtained in sections 3.5 and 3.6. Then the attacker can create $3! \times 2^3 \times 2^2 \times 5 \times 2 \times p^2 \approx 2^{37}$ candidates for $\tilde{h}_0(x)$.

Having obtained a number of candidates for the first row of $\tilde{H}_0$, the attacker can proceed to create a set of candidates for the first row of $\tilde{H}$. Let V be the set of candidates for the first row of $\tilde{H}_0$. For every $v \in V$, the attacker will look for words in the dual code to G' starting with v and having the Hamming weight at most $n_0 \times w \times m$. Thus the attacker can look for vectors $u^1, \ldots, u^{n_0-1} \in \mathbb{Z}_2^p$ satisfying

$$\begin{pmatrix} G'_{00} & G'_{01} & \ldots & G'_{0,n_0-1} \\ \vdots & \vdots & \ddots & \vdots \\ G'_{n_0-2,0} & G'_{n_0-2,1} & \ldots & G'_{n_0-2,n_0-1} \end{pmatrix} \begin{pmatrix} v \\ u^1 \\ \vdots \\ u^{n_0-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

The equation can be rewritten as

$$\begin{pmatrix} G'_{01} & \ldots & G'_{0,n_0-1} \\ \vdots & \ddots & \vdots \\ G'_{n_0-2,1} & \ldots & G'_{n_0-2,n_0-1} \end{pmatrix} \begin{pmatrix} u^1 \\ \vdots \\ u^{n_0-1} \end{pmatrix} = \begin{pmatrix} G'_{00} \\ G'_{10} \\ G'_{20} \end{pmatrix} (v). \tag{1}$$

For the cryptosystem from Section 3.3 the matrix on the left-hand side of the equation (1) had a full rank. Therefore, for the cryptosystem from Section 3.3, the equation (1) has at most one solution for a given v. In Appendix, we consider a scenario when the matrix on the left-hand side of the equation (1) has each of its circulant blocks generated uniformly independently at random. We argue that for values of n_0 and the block length p relevant for the QC-LDPC McEliece cryptosystem the probability that the rank of the matrix is close to the full rank is always nontrivial. Thus it is reasonable to expect that the equation (1) will with a nontrivial probability have only a small number of solutions.

Note that the attacker needs to put the matrix on the left-hand side of the equation (1) in the reduced upper echelon form only once and can use the reduced upper echelon form for every $v \in V$. The attacker will keep only those solutions with $w_{\mathrm{H}}((v, u^1, \ldots, u^{n_0-1})) \leq n_0 \times w \times m$. Each solution fully determines a candidate for the matrix $\tilde{H}$. If the resulting set of candidates for $\tilde{H}$ contains more than one element, the correct candidate can be determined by checking against a plaintext-ciphertext pair.

For the cryptosystem from Section 3.3, we have verified that $\tilde{H}$ can be used in a LDPC decoding algorithm to successfully decrypt ciphertexts.

4 Conclusion

We have presented a reaction attack on the QC-LDPC McEliece cryptosystem. Our attack is based on ideas from [5], where the attack on the closely related QC-MDPC McEliece cryptosystem was described. Compared to the recent attack on the QC-LDPC McEliece presented in [12], our attack has the advantage that it is feasible even when the size of circulant blocks in the cryptosystem is chosen to be odd.

We have verified the attack ideas on a version of QC-LDPC McEliece cryptosystem with parameters as proposed in [3], except for the parameter t' which we increased from 40 to 48. The parameter t' represents the number of errors added to an encoded message. Its increase resulted in the cryptosystem's probability of the decoding error to increase to approximately 0.1. This allowed us to estimate the probability of the decoding error using fewer decryptions. Consequently, we were able to break the cryptosystem after running 103 million decryptions.

In real applications the probability of the decoding error of around 0.1 would be very impractical. Thus, one would expect the QC-LDPC cryptosystem to be used with a value of t' which makes the probability of the decoding error significantly smaller. If this is the case, and if the attacker cannot inject into encoded messages a number of errors higher than t', then the attacker would need significantly more decryptions to estimate the probability of the decoding error and execute the attack. For instance, results of simulations presented in [1] (Fig.6.1. on p.88 in [1]) indicate that if the original value $t' = 40$ was used in the cryptosystem considered in this paper, then the probability of the decoding error would be of order 10^{-5}. Therefore, we expect that the attacker who can only send messages with $t' = 40$ errors would need 10^4 times more decryptions in order to break the cryptosystem.

In the experiments presented in this paper, we always assumed that the attacker does not have the freedom to choose what error vector is added to the message during encryption. Although we omitted the results from this paper, we also conducted experiments for the scenario where the attacker is free to choose the error vector. Similarly as in [5], we considered an attacker who for every possible distance d constructs error vectors with many pairs of ones separated by the distance d. In this case, it turns out that the attacker can break the same cryptosystem with $t' = 48$ with only 4 million decryptions.

The version of the QC-LDPC McEliece cryptosystem we used to verify our attack ideas employed a soft-decision decoding algorithm. Thus our results also confirm the conjecture from [5] that soft-decision decoding algorithms can be vulnerable to leak information about the secret parity-check matrix.

References

1. Baldi, M.: QC-LDPC code-based cryptography. Springer Science & Business, (2014)
2. Baldi, M., Chiaraluce, F.: Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes. In: Proc. IEEE ISIT 2007, Nice, France, June 2007, pp. 2591-2595 (2007)
3. Baldi, M., Bodrato, M., Chiaraluce, F.: A new analysis of the McEliece cryptosystem based on QCLDPC codes. In: Ostrovsky, R., De Prisco, R., Visconti, I. (eds.) 6th International Conference on Security and Cryptography for Networks (SCN 2008). LNCS, vol. 5229, pp. 246-262. Springer, Berlin (2008)
4. BitPunch, https://github.com/FrUh/BitPunch

5. Guo, Q., Johansson, T. and Stankovski, P.: A key recovery attack on MDPC with CCA security using decoding errors. In Advances in CryptologyASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22 (pp. 789-815). Springer Berlin Heidelberg (2016)
6. Hill, R.: A first course in coding theory. Oxford University Press (1986)
7. Jungnickel, D.: Finite Fields: Structure and Arithmetics, B.I. Wissenschaftsverlag, (1993)
8. R.J. McEliece: A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report, 44:114-116 (1978)
9. Misoczki R., Tillich J-P., Sendrier N., Barreto P.S.L.M.: MDPC-McEliece: new McEliece variants from moderate density parity-check codes. In: IEEE International Symposium on Information Theory (ISIT2013), pp. 2069-2073. Istanbul (2013)
10. Otmani, A., Tillich, J.P., Dallot, L.: Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes. In: Proc. First International Conference on Symbolic Computation and Cryptography (SCC 2008), Beijing, China (2008)
11. Radford M. N.: Software for Low Density Parity Check (LDPC) codes, http://www.cs.utoronto.ca/ radford/ldpc.software.html
12. M. Koochak Shooshtari, M. Ahmadian-Attari, T. Johansson, M. Reza Aref: Cryptanalysis of McEliece cryptosystem variants based on quasi-cyclic low-density parity check codes. in IET Information Security, vol. 10, no. 4, pp. 194-202, 7 (2016)

Appendix: On the rank of a randomly generated block-circulant matrix

In this appendix we study the rank over GF(2) of a matrix composed of $n_0 \times n_0$ randomly generated circulant blocks, the blocks being of size $p \times p$. We focus on the case when p is odd, since this ensures that the QC-LDPC McEliece cryptosystem is immune against the attack presented in [12].

Firstly, we recall some well-known facts about circulant matrices.

Fact 1 *[Proposition 1.7.1 in [7]] Consider the mapping τ which sends the circulant binary $(p \times p)$-matrix with the first row $(c_0, c_1, c_2, \ldots, c_{p-1})$ onto the polynomial $c(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{p-1} x^{p-1}$. Then the mapping τ is an isomorphism between the ring of circulant binary $(p \times p)$-matrices and the ring $\mathbb{Z}_2[x]/(x^p + 1)$.*

Fact 2 *[p. 42 in [7]] The inverse of a non-singular circulant matrix is again circulant. A circulant binary $(p \times p)$-matrix C is non-singular if and only if $\tau(C)$ is relatively prime to $x^p + 1$.*

Let f be a polynomial in $\mathbb{Z}_2[x]/(x^p + 1)$ and let $f(x) = g(x)h(x)$ where $g(x) = \gcd(f(x), x^p + 1)$. Then $\tau^{-1}(f) = \tau^{-1}(g)\tau^{-1}(h)$. By Fact 2, $\tau^{-1}(h)$ is non-singular. Therefore $\tau^{-1}(f)$ has the same rank as $\tau^{-1}(g)$. It is well-known (e.g. Theorem 12.12 in [6]) that $\tau^{-1}(g)$ generates a cyclic code of dimension $p-d$ where d is the degree of g. Thus we have:

Fact 3 *The rank of a circulant binary $(p \times p)$-matrix C is equal to $p - d$ where d is the degree of* $\gcd(\tau(C), x^p + 1)$.

Let f and g be polynomials in $\mathbb{Z}_2[x]$, and denote by $\psi(f)$ the number of polynomials of smaller degree which are relatively prime to f in $\mathbb{Z}_2[x]$.

Fact 4 *[Theorem 1.7.5 in [7]] If* $\gcd(f(x), g(x)) = 1$*, then* $\psi(fg) = \psi(f)\psi(g)$

Fact 5 *[Theorem 1.7.6 in [7]] Let p be odd. Then we have*

$$\psi(x^p + 1) = 2^p \prod_{j|p} \left(1 - 2^{-o_j(2)}\right)^{\phi(j)/o_j(2)} .$$

Here $o_j(2)$ denotes the order of 2 in the group $\mathbb{Z}_j^*$ and $\phi(j)$ denotes the Euler function.

It follows that the number of $p \times p$ circulant matrices with full rank is $\psi(x^p + 1)$. Circulant $p \times p$ matrices with rank $p - 1$ are precisely the matrices whose corresponding polynomial is a product of $x+1$ and a polynomial coprime to $\frac{x^p+1}{x+1}$ with degree less than $p - 1$. If p is odd, then $x + 1$ appears in the irreducible factorization of x^p+1 only once. Thus it follows that the number of $p \times p$ circulant matrices with rank $p - 1$ is $\psi(\frac{x^p+1}{x+1}) = \psi(x^p + 1)/\psi(x + 1) = \psi(x^p + 1)$.

Now we turn to block-circulant matrices. Let $\rho(p) = \psi(x^p + 1)/2^p$.

Proposition 1. *Let p be odd. Let B be a matrix composed of $(n_0 - 1) \times (n_0 - 1)$ circulant blocks of size $p \times p$. Suppose that the blocks in B were generated uniformly and independently at random from the space of all binary circulant $p \times p$ matrices. Then*

$$P(rank(B) \geq (n_0 - 1) \times (p - 1)) \geq \prod_{i=1}^{n_0-1} \left(1 - (1 - \rho(p))^i + \rho(p)^i\right).$$

Proof. Let B_{ij} be the $p \times p$ block present in the i-th block-row and j-th block-column of B. Let $b_{ij}(x) = \tau(B_{ij})$. With probability $1-(1 - \rho(p))^{n_0-1}+\rho(p)^{n_0-1}$ it holds that either one of the blocks in the first block-column is invertible or all blocks in the first block-column have rank $p - 1$.

Firstly, we look at the case when there exists an invertible block in the first block-column. Without loss of generality we can assume that this block is B_{11} (if not, we can swap block-rows of B). For every $i \in \{2, \ldots, n_0 - 1\}$ we can erase the block B_{i1} by adding to the i-th block-row the first block-row multiplied by $\left(B_{i1} \times B_{11}^{-1}\right)$. This corresponds to multiplying B from the left by the matrix $M_i = I_{p(n_0-1)\times p(n_0-1)}+\tilde{M}_i$, where $\tilde{M}_i$ is the matrix composed of $(n_0-1)\times(n_0-1)$ blocks of size $p \times p$ with the block $B_{i1} \times B_{11}^{-1}$ in the i-th block-row and the first block-column and with zero blocks everywhere else. Thus the resulting matrix has the same rank as B. We obtain a matrix of the form

$$\begin{pmatrix} B_{11}\ B_{12} \ldots B_{1,n_0-1} \\ 0 \\ \vdots \qquad \tilde{B} \\ 0 \end{pmatrix}, \tag{2}$$

where $\tilde{B}$ is a matrix composed of $(n_0 - 2) \times (n_0 - 2)$ circulant blocks of size $p \times p$. Let $\tilde{B}_{ij}$ be the $p \times p$ block present in the i-th block-row and j-th block-column of $\tilde{B}$. Then $\tilde{B}_{ij} = B_{i+1,1} \times B_{11}^{-1} \times B_{1,j+1} + B_{i+1,j+1}$. The block $B_{i+1,j+1}$ was generated independently from all other blocks in B, hence we can see $\tilde{B}_{ij}$ as a sum of $B_{i+1,j+1}$ and an independent circulant matrix. Since $B_{i+1,j+1}$ was generated uniformly at random from the space of circulant $p \times p$ matrices, $\tilde{B}_{ij}$ will, like $B_{i+1,j+1}$, have the property that each bit in its first row will be 1 with probability $1/2$ independently of other bits in its first row. Thus we can think of $\tilde{B}_{i,j}$ as of another uniformly randomly generated matrix from the space of circulant $p \times p$ matrices. Moreover, $\tilde{B}_{i,j}$ is independent of other blocks in $\tilde{B}$ and it is also independent of blocks in the first block-column of the original matrix B.

Now we consider the case when all blocks in the first block-column of B have rank $p - 1$. Then for every $b_{i1}(x)$ there exists $r_i(x) \in \mathbb{Z}_2[x]/(x^p + 1)$ such that $b_{i1}(x)r_i(x) = x + 1 \mod (x^p + 1)$ (the polynomial $r_i(x)$ can be found by the extended Euclidean algorithm). Thus for every $i \in \{2, \ldots, n_0 - 1\}$ we can erase the block B_{i1} by adding to the i-th block-row the first block-row multiplied by $\tau^{-1}\left(\frac{b_{i1}(x)}{x+1}\right) \times \tau^{-1}(r_1(x))$. By the same argument as in the previous case, this will not change the rank of B. We obtain a matrix of the form (2), where $\tilde{B}$ is again composed of $(n_0 - 2) \times (n_0 - 2)$ circulant blocks of size $p \times p$. Now we have $\tilde{B}_{ij} = \tau^{-1}\left(\frac{b_{i+1,1}(x)}{x+1}\right) \times \tau^{-1}(r_1(x)) \times B_{1,j+1} + B_{i+1,j+1}$. By the same argument as in the previous case, we can again think of $\tilde{B}_{i,j}$ as of a uniformly randomly generated matrix from the space of circulant $p \times p$ matrices. In addition, $\tilde{B}_{i,j}$ is independent of other blocks in $\tilde{B}$ and it is also independent of blocks in the first block-column of the original matrix B.

Thus in both cases we were able to transform the matrix B to a matrix of the form (2), while preserving its rank. The submatrix $\tilde{B}$ in (2) has the same properties as the original matrix B except it contains $(n_0 - 2) \times (n_0 - 2)$ blocks instead of $(n_0 - 1) \times (n_0 - 1)$ blocks. In addition, the submatrix $\tilde{B}$ is independent of blocks in the first block-column of the original matrix B. Proceeding inductively, the statement of the proposition follows.

In the QC-LDPC McEliece cryptosystem n_0 is typically small (3 or 4, for example). Let $\alpha(p, n_0)$ be the lower bound from Proposition 1, i.e.

$$\alpha(p, n_0) = \prod_{i=1}^{n_0 - 1} \left(1 - (1 - \rho(p))^i + \rho(p)^i\right).$$

In Figure 3 we present values of $\alpha(p, 4)$ for all odd p in the range from 1 to 20000. The smallest value of $\alpha(p, 4)$ in the figure is 0.11. Thus the figure shows that if $n_0 = 4$ then the probability that the rank of B is close to the full rank is nontrivial for all odd p below 20000.

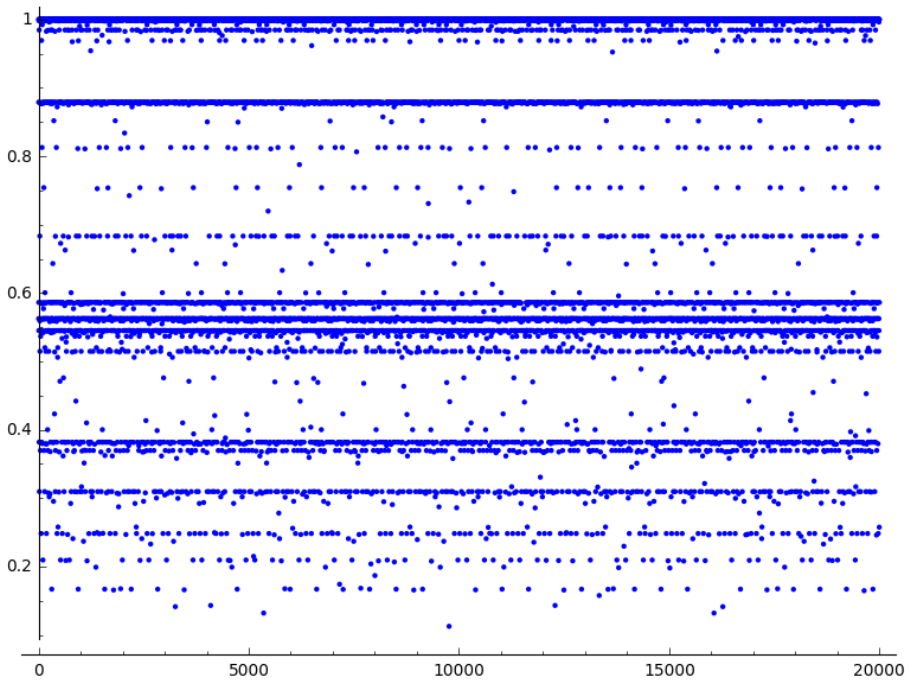

Fig. 3. Values of the lower bound $\alpha(p, 4)$ for the probability that a matrix composed of 3×3 circulant blocks of size $p \times p$ which are generated uniformly and independently at random has rank at least $3 \times (p - 1)$ for all odd p in the range from 1 to 20000.