Cryptogr. Commun. (2018) 10:159–175
DOI 10.1007/s12095-017-0239-4

On generating invertible circulant binary matrices with a prescribed number of ones

Tomáš Fabšič[1] · Otokar Grošek[1] · Karol Nemoga[2] · Pavol Zajac[1]

Received: 28 November 2016 / Accepted: 30 June 2017 / Published online: 11 July 2017

Abstract We study the problem how to efficiently generate circulant binary matrices with a prescribed number of ones which are invertible over $\mathbb{Z}_2$. A natural method to generate such matrices consists of two steps. Firstly, a circulant binary matrix with the prescribed number of ones is generated. Afterwards, it is tested for invertibility and if needed the process is repeated. To increase the efficiency of the process, we are interested in generating the matrices directly, without the need for the additional invertibility testing. We propose algorithms which fulfill this task for a wide range of parameters. Furthermore, we propose algorithms to construct matrices S and Q in the QC-LDPC McEliece cryptosystem. Matrices S and Q have to be composed of blocks of circulant matrices and they have to be invertible. In addition, S has to be dense and Q has to have a prescribed number of ones in a row. To avoid known attacks on the QC-LDPC McEliece cryptosystem, our algorithms generate S and Q with blocks of an odd size.

Keywords Circulant matrices · Invertible matrices · Binary matrices with a prescribed number of ones · QC-LDPC McEliece cryptosystem

This article is part of the Topical Collection on *Recent Trends in Cryptography*

This work is a partial result of the Research and Development Operational Programme for the project International centre of excellence for research of intelligent and secure information-communication technologies and systems, ITMS 26240120039, co-funded by the ERDF.

✉ Tomáš Fabšič
tomas.fabsic@stuba.sk

[1] Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Ilkovičova 3, 81219 Bratislava, Slovakia

[2] Slovak Academy of Sciences, Mathematical Institute, Štefánikova 49, 814 73 Bratislava, Slovakia

Mathematics Subject Classification (2010) 94A60

1 Introduction

In [1], Baldi and Chiaraluce proposed a version of the McEliece cryptosystem which uses quasi-cyclic low-density parity-check codes (QC-LDPC codes). Compared to the original version of the McEliece cryptosystem, their proposal's advantage was in the reduced size of public and private keys. However, in [7], Otmani et al. showed that the proposal had serious vulnerabilities. In [2], Baldi et al. proposed an amended version of the cryptosystem which was immunized against the attacks of Otmani, Tillich and Dallot. A related cryptosystem was proposed in [6] by Misoczki et al..

Generation of public and private keys for the QC-LDPC McEliece cryptosystem requires generating circulant binary matrices with a prescribed number of ones which are invertible over $\mathbb{Z}_2$. In [10] the problem of generating such matrices was solved as follows. Firstly a circulant binary matrix with a prescribed number of ones was generated. Afterwards the matrix was tested for invertibility and if it turned out to be singular, the process was repeated. The number of all invertible circulant matrices of a given size over $\mathbb{Z}_2$ can be computed by a formula (see Section 2 for details). Therefore, if a circulant binary matrix with a random number of ones was generated, the probability of the matrix being invertible could be computed. However, to the best of authors knowledge, no formula for computing the number of invertible circulant binary matrices of a given size and with a prescribed number of ones exists. Therefore the expected number of repeated generations cannot be directly computed and can be only estimated by simulations. These extra generations and the associated extra invertibility tests can be costly in terms of time and in terms of entropy needed to generate extra random bits.

In the present paper, we study the problem of constructing invertible circulant binary matrices with a prescribed number of ones directly, that means without the need for subsequent invertibility testing. We propose algorithms that achieve this task and thus avoid the additional costs mentioned above. On the other hand, the disadvantage of our algorithms is that they generate matrices from a smaller pool. For each of our algorithms a formula for the size of the pool is derived.

Subsequently, we investigate the application of our algorithms in the QC-LDPC McEliece cryptosystem. We focus on the problem how to construct matrices S and Q which form a part of the private key in the cryptosystem. Both S and Q have to be composed of blocks of circulant matrices and they both have to be invertible. In addition, S has to be dense and Q has to have a prescribed number of ones in a row. In [2] it was shown how matrices satisfying these requirements can be constructed in the case when the size of blocks is a power of 2. However, in [8] it was demonstrated that an even value of the block size allows an attacker to build a more efficient information-set decoding attack on the cryptosystem. In the present paper, we therefore propose methods how to construct S and Q when the size of the block is odd.

The paper is structured as follows. In Section 2, we review preliminary facts on circulant matrices. In Section 3, we study how to construct invertible circulant binary matrices with a prescribed number of ones. In Section 4, we investigate how our constructions from Section 3 can be used in the QC-LDPC McEliece cryptosystem. Finally, we conclude the paper in Section 5.

2 Circulant matrices

All questions on circulant matrices can be formulated in terms of polynomials, as the following well known fact explains.

Fact 1 *[Proposition 1.7.1 in [4]] Consider the mapping τ which sends the circulant binary $(n \times n)$-matrix with the first row $(c_0, c_1, c_2, \ldots, c_{n-1})$ onto the polynomial $c(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1}x^{n-1}$. Then the mapping τ is an isomorphism between the ring of circulant binary $(n \times n)$-matrices and the ring $\mathbb{Z}_2[x]/(x^n + 1)$.*

In order to construct an invertible circulant matrix, the following result is useful.

Fact 2 *[Lemma 1.7.2 and Observation 1.7.3 in [4]] A circulant binary $(n \times n)$-matrix C is invertible over $\mathbb{Z}_2$ if and only if $\tau(C)$ is relatively prime to $x^n + 1$ in $\mathbb{Z}_2[x]$.*

Thus in order to study invertible circulant matrices, it is useful to know the factorisation of the polynomial $x^n + 1$ in $\mathbb{Z}_2[x]$. This leads us to the notion of cyclotomic polynomials.

Definition 3 Let k be odd and let ζ be a primitive k-th root of unity over $GF(2)$. Then the polynomial

$$Q_k(x) = \prod_{s:\ \gcd(s,k)=1 \wedge s \le k} (x - \zeta^s)$$

is called the k-th cyclotomic polynomial over $GF(2)$.

Next, we review some fundamental facts about cyclotomic polynomials over $GF(2)$ and the factorisation of $x^n + 1$. Here and in the rest of the paper, $o_k(2)$ denotes the order of 2 in the group $\mathbb{Z}_k^*$ and $\phi(d)$ denotes the Euler function.

Fact 4 *[Theorem 2.45 in [5]] For odd n we have:*

1. $x^n + 1 = \prod_{k|n} Q_k$ *in* $\mathbb{Z}_2[x]$,
2. *the coefficients of Q_k belong to $GF(2)$.*

Fact 5 *[Theorem 2.47 in [5]] Let $d = o_k(2)$. Then Q_k factors into $\phi(k)/d$ distinct monic irreducible polynomials of degree d in $\mathbb{Z}_2[x]$.*

Let f be a polynomial in $\mathbb{Z}_2[x]$, and denote by $\psi(f)$ the number of polynomials of smaller degree which are relatively prime to f in $\mathbb{Z}_2[x]$. In [4] it is shown that if $\gcd(f(x), g(x)) = 1$, then $\psi(fg) = \psi(f)\psi(g)$. By Fact 2, the number of all invertible circulant binary $(n \times n)$-matrices is equal to $\psi(x^n + 1)$. Using Facts 4 and 5, a formula for $\psi(x^n + 1)$ can be derived, as demonstrated in [4].

Fact 6 *Let $n = 2^\alpha m$, where m is odd.*

Then we have

$$\psi(x^n + 1) = 2^n \prod_{k|m} \left(1 - 2^{-o_k(2)}\right)^{\phi(k)/o_k(2)} . \tag{1}$$

We finish the section with two well known observations concerning circulant matrices with an even number of ones in a row. Here and in the rest of the paper, the weight of a polynomial stands for the number of its nonzero coefficients.

Fact 7 *Let $f \in \mathbb{Z}_2[x]/(x^n + 1)$. Then f has an even weight iff f is a multiple of $x + 1$.*

Proof A polynomial f has an even weight iff $f(1) = 0$. $f(1) = 0$ iff f is a multiple of $x + 1$. □

Fact 8 *Every circulant $(n \times n)$-matrix over $\mathbb{Z}_2$ with an even number of ones in a row is singular.*

Proof If a circulant matrix C has an even number of ones in a row, then its corresponding polynomial $\tau(C)$ has an even weight. Thus by Fact 7, $\tau(C)$ is a multiple of $x+1$. The result follows by Fact 2. □

3 Constructing invertible circulant matrices with a prescribed number of ones

3.1 Special cases

In this section, we show that constructing invertible circulant binary $(n \times n)$-matrices with a prescribed number of ones is easy for some choices of n.

3.1.1 Case when n is prime and $o_n(2) = n - 1$

Let us suppose that n is prime such that $o_n(2) = n - 1$. Then the facts above imply that the polynomial $x^n + 1$ has only two irreducible factors: Q_1 and Q_n. Thus by Fact 7 all polynomials with odd weight except of Q_n correspond to invertible circulant matrices, as is observed in [3]. Thus we have the following simple algorithm to generate an invertible circulant binary $(n \times n)$-matrix with w ones in a row uniformly from a set of $\binom{n}{w}$ such matrices.

Algorithm 1

INPUT: n prime such that $o_n(2) = n - 1$, w odd and $w < n$
OUTPUT: a polynomial f corresponding to an invertible circulant binary $(n \times n)$-matrix with w ones in a row under the isomorphism τ from Fact 1

1. Choose values $t_0, \ldots, t_{w-1}$ from $\{0, \ldots, n-1\}$ uniformly at random without replacement.
2. Return the polynomial $f(x) = \sum_{i=0}^{w-1} x^{t_i}$.

We note that according to Artin's conjecture for primitive roots, approximately 37% of primes p satisfy $o_p(2) = p - 1$.

3.1.2 Case when n is a power of 2

Let us suppose that n is a power of 2. Then we have $x^n + 1 = (x + 1)^n$ and the only irreducible factor of $x^n + 1$ is $Q_1 = x + 1$. Thus by Fact 7 all polynomials with odd weight

correspond to invertible circulant matrices [2]. Thus for odd w we again have a simple algorithm to generate an invertible circulant binary $(n \times n)$-matrix with w ones in a row uniformly from a set of $\binom{n}{w}$ such matrices.

Algorithm 2

INPUT: n a power of 2, w odd and $w \leq n$
OUTPUT: a polynomial f corresponding to an invertible circulant binary $(n \times n)$-matrix with w ones in a row under the isomorphism τ from Fact 1

1. Choose values $t_0, \ldots, t_{w-1}$ from $\{0, \ldots, n-1\}$ uniformly at random without replacement.
2. Return the polynomial $f(x) = \sum_{i=0}^{w-1} x^{t_i}$.

3.2 First method for general *n*

Let us consider the irreducible factorisation of $x^n + 1$ in $\mathbb{Z}_2[x]$. Consider the set of polynomials other than $x + 1$ appearing in this factorisation. Let $p(x)$ be a polynomial with the smallest degree in this set. Let $d = \deg(p(x))$.

Let $n = 2^i m$, where m is odd. Using Fact 4, Fact 5 and the well known fact that for $f \in \mathbb{Z}_2[x]$ we have $f(x^2) = (f(x))^2$, we can determine the value of d as

$$d = \min_{k:\, k|m \,\wedge\, k>1} \{o_k(2)\} \quad . \tag{2}$$

Since $a|b$ implies that $o_a(2)|o_b(2)$, we can simplify the (2) as

$$d = \min_{p:p \text{ is prime} \,\wedge\, p|m} \left\{o_p(2)\right\} \quad . \tag{3}$$

In the QC-LDPC McEliece cryptosystem, the size of circulant blocks is typically below 10^5. For such values of n, the value of d can be computed instantly on a standard PC.

It is immediate that if a polynomial has odd weight and degree less than d, than it is relatively prime to x^n+1. We have seen in the previous section that if n is prime and $o_n(2) = n-1$, then constructing invertible circulant binary $(n \times n)$-matrices with a prescribed number of ones is easy. In all other cases we have that $d \leq \frac{n-1}{2}$. Thus in this section we will assume that $d \leq \frac{n-1}{2}$.

We can use the following simple algorithm to generate an invertible circulant binary $(n \times n)$-matrix with w ones in a row.

Algorithm 3

INPUT: n, d as above and $d \leq \frac{n-1}{2}$, w odd and $w \leq d$
OUTPUT: a polynomial f corresponding to an invertible circulant binary $(n \times n)$-matrix with w ones in a row under the isomorphism τ from Fact 1

1. Choose values $t_1, \ldots, t_{w-1}$ from $\{1, \ldots, d-1\}$ uniformly at random without replacement.
2. Choose $k \in \{0, \ldots, n-1\}$ uniformly at random.
3. Return the polynomial $f(x) = x^k + \sum_{i=1}^{w-1} x^{k+t_i} \mod (x^n + 1)$.

Proposition 9 *Algorithm 3 generates an invertible circulant binary $(n \times n)$-matrix with w ones in a row uniformly from a set of $n \binom{d-1}{w-1}$ such matrices.*

Proof Let us consider a polynomial $h(x) = x^k + \sum_{i=1}^{w-1} x^{k+t_i}$ for some choice of $t_1, \ldots, t_{w-1} \in \{1, \ldots, d-1\}$ and $k \in \{0, \ldots, n-1\}$. There are $n \binom{d-1}{w-1}$ different ways how to choose $t_1, \ldots, t_{w-1}$ and k, each of them equally probable.

Let $\tilde{h}(x) = x^{\tilde{k}} + \sum_{i=1}^{w-1} x^{\tilde{k}+\tilde{t}_i}$ for some other choice of $\tilde{t}_1, \ldots, \tilde{t}_{w-1}$ from $\{1, \ldots, d-1\}$ and $\tilde{k} \in \{0, \ldots, n-1\}$. In order to prove the proposition, we need to show that $h(x)$ and $\tilde{h}(x)$ remain different after modulation $\mod (x^n + 1)$.

Without loss of generality, suppose $k \geq \tilde{k}$. Suppose $h(x) = \tilde{h}(x) \mod (x^n + 1)$. Then we have $x^{k-\tilde{k}} \left(1 + \sum_{i=1}^{w-1} x^{t_i}\right) = 1 + \sum_{i=1}^{w-1} x^{\tilde{t}_i} \mod (x^n + 1)$. If $k - \tilde{k} = 0$, this is only possible if $\{t_1, \ldots, t_{w-1}\} = \{\tilde{t}_1, \ldots, \tilde{t}_{w-1}\}$, in which case $h(x) = \tilde{h}(x)$. If $k - \tilde{k} > 0$, then $x^{k-\tilde{k}} \left(1 + \sum_{i=1}^{w-1} x^{t_i}\right) \mod (x^n + 1)$ has to contain the absolute term 1. Since every t_i is less than $\frac{n-1}{2}$, this implies that $k - \tilde{k} > \frac{n+1}{2}$. Let j, $j < n$, be such that $1 = x^{k-\tilde{k}} x^j \mod (x^n + 1)$, i.e. $j = n - k + \tilde{k}$. If $j \in \{t_1, \ldots, t_{w-1}\}$, then $x^{k-\tilde{k}} \left(1 + \sum_{i=1}^{w-1} x^{t_i}\right)$ $\mod (x^n + 1)$ contains a term $x^{k-\tilde{k}}$, which has degree $k - \tilde{k}$, $\frac{n+1}{2} < k - \tilde{k} < n$. But this is not possible since $1 + \sum_{i=1}^{w-1} x^{\tilde{t}_i}$ contains no term of such degree. Therefore $j = 0$. Thus $k - \tilde{k}$ would have to be equal to n, which is not allowed, since $k, \tilde{k} \in \{0, \ldots, n-1\}$. □

3.3 Second method for general *n*

The method from the previous section allows us to construct matrices with at most d ones in a row. The following observation can be used to construct matrices with a higher number of ones.

Proposition 10 *Let t be odd. Let $r_t(x) = \sum_{i=0}^{t-1} x^i$. If t and n are coprime, then also $r_t(x)$ and $x^n + 1$ are coprime.*

Proof We have that $r_t(x) = \frac{x^t+1}{x+1}$. Thus by Fact 4 we have $r_t = \prod_{k|r \wedge k>1} Q_k$. By the same fact $x^n + 1$ is a product of cyclotomic polynomials, none of which appear in the cyclotomic factorisation of $r_t(x)$. Since distinct cyclotomic polynomials are coprime, the result follows. □

The above proposition gives us a recipe to construct invertible circulant $(n \times n)$-matrices with t ones in a row, when t is odd and coprime to n. The next observation will allow us to construct more such matrices.

Proposition 11 *Let f be a polynomial of odd weight and such that $\deg(f) < t$. Then the polynomial $g(x) = f(x) r_t(x)$ has t terms.*

Proof Let $f(x) = x^{k_0} + x^{k_1} + x^{k_2} + \cdots + x^{k_m}$, $k_m < t$, and let the weight of f, $w(f)$, be odd. Without loss of generality we can assume that $k_0 = 0$. We can think of the product $f(x) r_t(x)$ as of the summation of polynomials $x^{k_i} r_t(x)$. We illustrate this summation in Table 1 for polynomials $r_7(x)$ and $f(x) = 1 + x^2 + x^3 + x^5 + x^6$. We will focus on the grey area in the table. Imagine the corresponding table for the polynomials r_t and f. Symbols

Table 1 Multiplication of $r_7(x)$ with $\tilde{f}(x) = 1 + x^2 + x^3 + x^5 + x^6$

	1	x	x^2	x^3	x^4	x^5	x^6	x^7	x^8	x^9	x^{10}	x^{11}	x^{12}
$1 \times r_7(x)$	1	1	1	1	1	1	1						
$x^2 \times r_7(x)$			1	1	1	1	1	1	1				
$x^3 \times r_7(x)$				1	1	1	1	1	1	1			
$x^5 \times r_7(x)$						1	1	1	1	1	1	1	
$x^6 \times r_7(x)$							1	1	1	1	1	1	1
$\Sigma\ (= \tilde{f} \times r_7)$	1	1		1	1		1			1			1

Coefficients of the product are obtained by summing the terms "1" in the grey area along the columns

"1" in the grey area form columns of length at most $w(f)$. Coefficients of the resulting product $f(x)r_t(x)$ are obtained by summing the ones in the grey area along these columns. Thus, only columns of an odd length will give rise to nonzero coefficients in $f(x)r_t(x)$.

The number of columns of the length $w(f)$ is $t - \deg(f)$. Since the second term of f is x^{k_1}, the second row of ones in the grey area is shifted by k_1 positions to the right with respect to the first row. As a result, k_1 columns with length 1 (the columns $1, x, \ldots, x^{k_1-1}$) and k_1 columns with length $w(f) - 1$ (the columns $x^t, \ldots, x^{t+k_1-1}$) are created. Similarly, the j-th row is shifted by $k_{j-1} - k_{j-2}$ positions with respect to the $(j-1)$-th row. This creates $k_{j-1} - k_{j-2}$ columns of length $j-1$ and $k_{j-1} - k_{j-2}$ columns of length $w(f) - j + 1$. Since $w(f)$ is odd, the lengths $j-1$ and $w(f) - j + 1$ are of different parity. Thus, if we consider the columns $1, x, \ldots, x^{\deg(f)-1}$ together with the columns $x^t, \ldots, x^{t+\deg(f)-1}$, exactly half of them will have an odd length. Apart from these columns, there are $t - \deg(f)$ columns of length $w(f)$, which is odd. Thus, the total number of columns of an odd length is t. □

Proposition 11 shows that if t is odd and coprime with n, then we can construct more polynomials of weight t coprime with $x^n + 1$. Indeed, if one takes any polynomial f satisfying the constraints

1. $\gcd(f(x), x^n + 1) = 1$,
2. $\deg(f) < t$,
3. $\deg(f) + (t-1) < n$,

then the product $f(x)r_t(x)$ will be coprime with $x^n + 1$ and will have the weight t. (We note that the condition $\gcd(f(x), x^n + 1) = 1$ implies that the weight of f is odd, as required by Proposition 11. This follows by Fact 7.) Thus, one can construct more invertible circulant $(n \times n)$-matrices with t ones in a row. We state this formally as a corollary.

Corollary 12 *Let t be odd and such that $t < n$ and $\gcd(t, n) = 1$. Let the polynomial f satisfy:*

1. $\gcd(f(x), x^n + 1) = 1$,
2. $\deg(f) < t$,
3. $\deg(f) + (t-1) < n$.

Let $k \in \{0, \ldots, n-1\}$. Then $\tau^{-1}\left(x^k f(x) r_t(x)\right)$ is an invertible circulant binary $(n \times n)$-matrix with t ones in a row.

We further remark that the multiplication of polynomials f and r_t from Corollary 12 can be performed efficiently by the following algorithm with linear complexity.

Algorithm 4

INPUT: n, t as in Corollary 12, polynomial $f(x) = \sum_{i=0}^{t-1} a_i x^i$ with $a_i \in \{0, 1\}$ and with properties as in Corollary 12
OUTPUT: a polynomial $h(x) = f(x)r_t(x)$

1. Set $b_{-1} = 0$.
2. For k from 0 to $t-1$ do: if $a_k = 1$ set $b_k = b_{k-1} + 1 \mod 2$, else set $b_k = b_{k-1}$.
3. For k from t to $2t-2$ do: if $a_{k-t} = 1$ set $b_k = b_{k-1} + 1 \mod 2$, else set $b_k = b_{k-1}$.
4. Return the polynomial $h(x) = \sum_{i=0}^{2t-2} b_k x^k$.

Proof of correctness of Algorithm 4 Suppose that we substitute the steps of Algorithm 4 with the following set of steps:

1. Set $h(x) = 0$.
2. For k from 0 to $t-1$ do: if $a_k = 1$ do $h(x) = h(x) + \sum_{i=k}^{2t-2} x^i$.
3. For k from t to $2t-2$ do: if $a_{k-t} = 1$ do $h(x) = h(x) + \sum_{i=k}^{2t-2} x^i$.
4. Return the polynomial $h(x)$.

Then it is easy to see that the changed algorithm produces the same output as the original one. Realizing that we can merge steps 2 and 3 in the changed algorithm into a single step, we arrive at the algorithm with the following steps:

1. Set $h(x) = 0$.
2. For k from 0 to $t-1$ do: if $a_k = 1$ do $h(x) = h(x) + \sum_{i=k}^{2t-2} x^i + \sum_{i=k+t}^{2t-2} x^i$.
3. Return the polynomial $h(x)$.

But $\sum_{i=k}^{2t-2} x^i + \sum_{i=k+t}^{2t-2} x^i = x^k r_t(x)$ and thus the algorithm above corresponds to the standard algorithm for multiplication of $f(x)$ and $r_t(x)$. □

Corollary 12 tells us that if we want to generate invertible circulant binary $(n \times n)$-matrix with t ones in a row, it suffices to generate a polynomial f satisfying the three conditions in the corollary. To increase the efficiency of generating, it might be desirable if we did not need to check the condition $\gcd(f(x), x^n + 1) = 1$. To this end, we may use the knowledge from Section 3.2. Let d be as in Section 3.2, i.e. d is the degree of a smallest (in terms of degree) polynomial other than $x + 1$ appearing in the irreducible factorisation of $x^n + 1$. Again we assume $d \leq \frac{n-1}{2}$. Then we know that any polynomial f of odd weight and of degree $\deg(f) < d$ satisfies $\gcd(f(x), x^n + 1) = 1$. Using this fact, we can employ the following efficient algorithm to generate an invertible circulant binary $(n \times n)$-matrix with t ones in a row.

Algorithm 5

INPUT: n, t as in Corollary 12 and $2t - 2 < n$, d as above and $d \leq t$ and $d \leq \frac{n-1}{2}$
OUTPUT: a polynomial g corresponding to an invertible circulant binary $(n \times n)$-matrix with t ones in a row under the isomorphism τ from Fact 1

1. For k from 1 to $d-2$ do: choose $a_k \in \{0, 1\}$ uniformly at random.
2. If $\sum_{k=1}^{d-2} a_k$ is odd, set $a_{d-1} = 1$, otherwise $a_{d-1} = 0$.
3. Compute $h(x) = r_t(x)\left(1 + \sum_{k=1}^{d-1} a_k x^k\right)$ by Algorithm 4.
4. Choose $s \in \{0, \ldots, n-1\}$ uniformly at random.
5. Return $g(x) = x^s h(x) \mod (x^n + 1)$.

Proposition 13 *Algorithm 5 generates an invertible circulant binary $(n \times n)$-matrix with t ones in a row uniformly from a set of $n2^{d-2}$ such matrices.*

Proof We proceed analogously to the proof of Proposition 9.

Let $g_1(x) = x^s r_t(x)\left(1 + \sum_{k=1}^{d-1} a_k x^k\right)$ for $a_1, \ldots, a_{d-1}$ and s selected as in Algorithm 5. There are $n2^{d-2}$ different ways how to choose $a_1, \ldots, a_{d-1}$ and j, each of them equally probable.

Let $g_2(x) = x^{\tilde{s}} r_t(x)\left(1 + \sum_{k=1}^{d-1} \tilde{a}_k x^k\right)$ for some other choice of $\tilde{a}_1, \ldots, \tilde{a}_{d-1}$ and $\tilde{s}$. In order to prove the proposition, we need to show that $g_1(x)$ and $g_2(x)$ remain different after modulation $\mod (x^n + 1)$.

Without loss of generality, suppose $s \geq \tilde{s}$. Suppose $g_1(x) = g_2(x) \mod (x^n + 1)$. Then, since $r_t(x)$ is relatively prime to $x^n + 1$, we have

$$x^{s-\tilde{s}}\left(1 + \sum_{k=1}^{d-1} a_k x^k\right) = 1 + \sum_{k=1}^{d-1} \tilde{a}_k x^k \mod (x^n + 1) \ .$$

If $s - \tilde{s} = 0$, this is only possible if $a_k = \tilde{a}_k$ for every k, which would mean that $g_1 = g_2$. If $s - \tilde{s} > 0$, then $x^{s-\tilde{s}}\left(1 + \sum_{k=1}^{d-1} a_k x^k\right) \mod (x^n + 1)$ has to contain the absolute term 1. Since $d - 1 < \frac{n-1}{2}$, this implies that $s - \tilde{s} > \frac{n+1}{2}$. Let i, $i < n$, be such that $1 = x^{s-\tilde{s}} x^i \mod (x^n + 1)$, i.e. $i = n - s + \tilde{s}$. If $i > 0$, then $x^{s-\tilde{s}}\left(1 + \sum_{k=1}^{d-1} a_k x^k\right) \mod (x^n + 1)$ contains a term $x^{s-\tilde{s}}$, which has degree $s - \tilde{s}$, $\frac{n+1}{2} < s - \tilde{s} < n$. But this is not possible since $1 + \sum_{k=1}^{d-1} \tilde{a}_k x^k \mod (x^n + 1)$ contains no term of such degree. Therefore $i = 0$. Thus $s - \tilde{s}$ would have to be equal to n, which is not allowed, since $s, \tilde{s} \in \{0, \ldots, n-1\}$. □

4 Application to the QC-LDPC McEliece cryptosystem

In [1], Baldi et al. proposed a variant of the McEliece cryptosystem based on LDPC codes - 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. The matrix H is formed by a row $\{H_0, \ldots, H_{n_0-1}\}$ of $n_0 = n/(n-k)$ binary circulant blocks with 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:

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

The remaining part of the private key is formed by two other matrices: a $k \times k$ invertible matrix S and a sparse $n \times n$ invertible matrix Q. S and Q are formed by blocks of $p \times p$ circulant matrices. In addition, Q has a fixed row weight m. The public key is then computed as follows:

$$G' = S^{-1} \cdot G \cdot Q^{-1} \ .$$

Strongly related to the QC-LDPC McEliece cryptosystem is the QC-MDPC McEliece cryptosystem which was proposed in [6]. The QC-MDPC McEliece cryptosystem again

uses a secret parity-check matrix H formed by a row of circulant blocks $\left\{H_0, \ldots, H_{n_0-1}\right\}$. However, in the QC-MDPC McEliece cryptosystem the blocks contain slightly more ones compared to the QC-LDPC variant. Another distinguishing feature is that the QC-MDPC McEliece cryptosystem does not employ matrices S and Q.

In [1], Baldi et al. proposed the following values for the parameters of the QC-LDPC McEliece cryptosystem: $n_0 = 4$, $w = 13$, $p = 4032$ and $m = 7$. They chose S and Q to be sparse and they proposed the following block-diagonal form for Q:

$$Q = \begin{bmatrix} Q_0 & 0 & 0 & \ldots & 0 \\ 0 & Q_1 & 0 & \ldots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & Q_{n_0-1} \end{bmatrix}.$$

In [7], Otmani et al. demonstrated that this cryptosystem is vulnerable to attacks which exploit the fact that Q is block-diagonal and S is sparse. In order to immunize their cryptosystem against these attacks, Baldi et al. proposed a version of the QC-LDPC McEliece cryptosystem with the matrix S dense and the matrix Q no longer block-diagonal in [2]. In [2], they proposed two variants of their cryptosystem: the first with parameters $n_0 = 4$, $w = 13$, $p = 4096$ and $m = 7$, and the second with parameters $n_0 = 3$, $w = 13$, $p = 8192$ and $m = 11$. They further suggested to choose 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. As for the matrix Q, Baldi et al. suggest to obtain Q in the first variant by constructing a matrix of 4×4 circulant blocks with the blocks on the diagonal having rows of weight 1 and the blocks away from the diagonal having rows of weight 2, and by randomly permuting its block rows and columns. Similarly, in the second variant, they suggest to obtain Q by constructing a matrix of 4×4 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 columns.

Baldi et al. claim that if S and Q are constructed in the above manner, then they are invertible. This is true only because in both variants p was selected as a power of 2. To see why this selection implies that the matrix Q is invertible, let us replace every circulant block in Q by its corresponding polynomial and let us think of Q as of an $n_0 \times n_0$ matrix over the ring $\mathbb{Z}_2[x]/(x^p + 1)$. A square matrix over a commutative ring is invertible if and only if its determinant is invertible over the ring. From the construction of Q and from Fact 7 it follows that in the Leibniz formula for the determinant of Q only one of the summands is not divisible by $x + 1$. Therefore the whole determinant is not divisible by $x + 1$. As explained in Section 3.1, when p is a power of 2, the non-divisibility by $x + 1$ guarantees that the determinant of Q is invertible over $\mathbb{Z}_2[x]/(x^p + 1)$. By the same argument, the matrix is S is invertible too.

Thus the choice of a power of 2 for p allows an easy construction of matrices S and Q which are invertible. However, in [8] it was demonstrated that an even value of p allows an attacker to build a more efficient information-set decoding attack on the QC-LDPC cryptosystem. Therefore it may be desirable to use the QC-LDPC cryptosystem with p odd. However, when p is odd, the construction from [2] does no longer guarantee that the resulting matrix will be invertible. This leads to the question: How to construct matrices S and Q when p is odd?

4.1 Constructing matrices S and Q when the block size is a prime p with $o_p(2) = p - 1$

Based on the analysis from Section 3.1.1, it appears that the best choice for the block size p would be to make p equal to a prime such that $o_p(2) = p - 1$. This choice guarantees that every $p \times p$ circulant matrix with an odd weight is invertible, with exception of the matrix of weight p. According to Artin's conjecture for primitive roots, approximately 37% of primes p satisfy $o_p(2) = p - 1$.

4.1.1 Constructing the matrix S

When the block size p is a prime such that $o_p(2) = p - 1$, we propose to use the same construction for S as in [2]. This means, that we propose to construct S by the following algorithm.

Algorithm 6

INPUT: k_0, p
OUTPUT: a binary matrix consisting of $k_0 \times k_0$ circulant blocks of size $p \times p$

1. Generate each block on the diagonal independently uniformly at random from the set of all circulant $p \times p$ matrices with an odd weight.
2. Generate each block away from the diagonal independently uniformly at random from the set of all circulant $p \times p$ matrices with an even weight.
3. Permute the block rows of the matrix by a permutation selected uniformly at random from the space of all permutations of k_0 elements.

Proposition 14 *Let p be a prime such that $o_p(2) = p - 1$. Let S be a matrix produced by Algorithm 6. Then*

$$P(S \text{ is invertible}) \geq \left(1 - \frac{1}{2^{p-1}}\right)^{k_0}.$$

Proof Let $\tilde{S}$ be a matrix generated by first 2 steps of Algorithm 6. We will again think of $\tilde{S}$ as of a matrix over $\mathbb{Z}_2[x]/(x^p+1)$. The probability that $\tilde{S}$ is invertible over $\mathbb{Z}_2[x]/(x^p+1)$ is greater or equal to the probability that $\tilde{S}$ can be turned into the identity matrix by doing Gaussian elimination over $\mathbb{Z}_2[x]/(x^p+1)$. Let $\tilde{s}_{ij}(x)$ be the polynomial in the i-th row j-th column of $\tilde{S}$. The probability that the polynomial $\tilde{s}_{11}(x)$ can be inverted is $1 - \frac{1}{2^{p-1}}$. After inverting $\tilde{s}_{11}(x)$, Gaussian elimination will multiply the first row of $\tilde{S}$ by $\tilde{s}_{11}^{-1}(x)$. Afterwards, it adds suitable multiples of the first row to other rows to nullify entries in the first column. Then it will try to invert the second term on the diagonal of the matrix. This term is now equal to $\tilde{s}_{22}(x) + \tilde{s}_{21}(x)\tilde{s}_{11}^{-1}(x)\tilde{s}_{12}(x)$. The polynomial $\tilde{s}_{21}(x)\tilde{s}_{11}^{-1}(x)\tilde{s}_{12}(x)$ has an even weight. The polynomial $\tilde{s}_{22}(x)$ was generated independently from the polynomials $\tilde{s}_{21}(x)$, $\tilde{s}_{11}(x)$ and $\tilde{s}_{12}(x)$. Therefore we can think of the polynomial $\tilde{s}_{22}(x) + \tilde{s}_{21}(x)\tilde{s}_{11}^{-1}(x)\tilde{s}_{12}(x)$ as if it was generated uniformly at random from the set of all polynomials in $\mathbb{Z}_2[x]/(x^p+1)$ with an odd weight and independently of $\tilde{s}_{11}(x)$. Thus the probability that Gaussian elimination will be able to invert the first two polynomials on the diagonal is $\left(1 - \frac{1}{2^{p-1}}\right)^2$. Similarly, it can be shown that the probability that Gaussian elimination will be able to invert all polynomials on the diagonal is $\left(1 - \frac{1}{2^{p-1}}\right)^{k_0}$. □

In variants of QC-LDPC and QC-MDPC McEliece cryptosystem [2, 6], the value of k_0 is not larger then 3 and the value of p is at least 4096. For such values, the probability $\left(1-\frac{1}{2^{p-1}}\right)^{k_0}$ is very close to 1. Therefore Algorithm 6 produces an invertible matrix with very high probability.

4.1.2 Constructing the matrix Q

Now, we propose an algorithm to construct the matrix Q. The matrix Q has to have a constant row weight m, and it has to be invertible and composed of $n_0 \times n_0$ circulant blocks of size $p \times p$. We suppose that m is odd and that it can be written as $m = u(n_0 - 1) + v$, where $u \geq 2$ is even. Then v must be odd. Again, we propose a similar construction to the one in [2], with blocks on the diagonal having weight v and blocks away from the diagonal having weight u.

Algorithm 7

INPUT: $n_0, p, u, v \in \mathbb{N}$, $u \geq 2$ even, v odd
OUTPUT: matrix Q with rows with weight $m = u(n_0 - 1) + v$ and composed of $n_0 \times n_0$ circulant blocks of size $p \times p$

1. For $i = 1$ to n_0 do:
2. Generate numbers from $\{1, \ldots, p\}$ uniformly independently at random until you obtain v different numbers $d_1^i, d_2^i, \ldots, d_v^i$.
3. Create a circulant $(p \times p)$-matrix D^i with first row having symbols 1 only at positions $d_1^i, d_2^i, \ldots, d_v^i$.
4. End for.
5. For $i = 1$ to $n_0(n_0 - 1)$ do:
6. Generate numbers from $\{1, \ldots, p\}$ uniformly independently at random until you obtain u different numbers $b_1^i, b_2^i, \ldots, b_u^i$.
7. Create a circulant $(p \times p)$-matrix B^i with first row having symbols 1 only at positions $b_1^i, b_2^i, \ldots, b_u^i$.
8. End for.
9. Create a matrix B composed of $n_0 \times n_0$ circulant blocks of size $p \times p$ by placing blocks D^i along the diagonal and blocks B^i away from the diagonal.
10. Create a matrix Q from B by permuting block rows of B by a permutation selected uniformly at random from the space of all permutations of n_0 elements.

Proposition 15 *Let p be a prime such that $o_p(2) = p - 1$. Suppose that values n_0, p, u, and v satisfy*

$$n_0! \times (\max\{u, v\})^{n_0} < p \quad . \tag{4}$$

Then Algorithm 7 always produces an invertible matrix.

Proof Consider the Leibniz formula for the determinant of Q. The formula implies that the weight of the determinant is at most $n_0! \times (\max\{u, v\})^{n_0}$. If inequality (4) holds, then the determinant of Q must be an odd polynomial other than $\sum_{i=0}^{p-1} x^i$. Thus the determinant is invertible. □

We note that the inequality (4) is comfortably met by both variants of the QC-LDPC McEliece cryptosystem presented in [2].

4.2 Constructing matrices S and Q for an arbitrary odd block size

4.2.1 Constructing the matrix S

If the block size is an arbitrary odd number, we can still use Algorithm 6 to construct the matrix S. The lower bound on the probability that the resulting matrix will be invertible, however, decreases.

Proposition 16 *Let p be an arbitrary odd number. Let S be a matrix produced by Algorithm 6. Then*

$$P(S \text{ is invertible}) \geq \left(\frac{\psi(x^p+1)}{2^{p-1}}\right)^{k_0}.$$

Proof Same as the proof of Proposition 14. □

We remind that the value of $\psi(x^p+1)$ can be computed by formula (1) in Section 2. By choosing the number p so that the polynomial $\frac{x^p+1}{x+1}$ factors into few irreducible polynomials of high degrees in $\mathbb{Z}_2[x]$, we can make the value of $\psi(x^p+1)$ close to 2^{p-1}. Since k_0 is typically not larger than 3 in variants of QC-LDPC and QC-MDPC McEliece cryptosystems, the probability that Algorithm 6 produces an invertible matrix can be made large by a proper choice of p.

4.2.2 Constructing the matrix Q

If p is an arbitrary odd number, Algorithm 7 no longer guarantees the invertibility of the resulting matrix. This is because the weight below p of the determinant of Q no longer implies the invertibility of Q. What is more, the approach used in Proofs of Propositions 14 and 16 cannot be used to estimate the probability that Algorithm 7 produces an invertible matrix. Therefore, we propose a different algorithm to generate the matrix Q when p is an arbitrary odd number. The algorithm places blocks of an odd weight on the diagonal and blocks of an even weight away from the diagonal as was the case in the previous algorithms. However, this time the algorithm generates blocks by using a modified version of Algorithm 3. Recall that in Algorithm 3 d represented the degree of a smallest (in terms of degree) polynomial other than $x+1$ appearing in the irreducible factorisation of x^p+1. Firstly, we state a modified version of Algorithm 3.

Algorithm 8

INPUT: p, n_0, d as above, $w \leq \frac{d}{n_0}$
OUTPUT: a polynomial f with weight w

1. Choose values $t_1, \ldots, t_w$ from $\left\{0, \ldots, \left\lfloor \frac{d}{n_0} \right\rfloor - 1\right\}$ uniformly at random without replacement.
2. Return the polynomial $f(x) = \sum_{i=1}^{w} x^{t_i}$.

Next, we state the algorithm to generate the matrix Q.

Algorithm 9

INPUT: p, n_0, d as above, $u \geq 2$ even, v odd, $u, v \leq \frac{d}{n_0}$
OUTPUT: invertible matrix Q with rows with weight $m = u(n_0 - 1) + v$ and composed of $n_0 \times n_0$ circulant blocks of size $p \times p$

1. For $i = 1$ to n_0 do:
2. Generate polynomial $f_i(x)$ by running Algorithm 8 with inputs p, n_0, d and $w = v$.
3. End for.
4. For $i = 1$ to $n_0(n_0 - 1)$ do:
5. Generate polynomial $g_i(x)$ by running Algorithm 8 with inputs p, n_0, d and $w = u$.
6. End for.
7. Choose $k \in \{0, \ldots, p - 1\}$ uniformly at random.
8. Create a $n_0 \times n_0$ matrix $\tilde{Q}$ over the ring $\mathbb{Z}_2[x]/(x^p+1)$ by placing polynomials $x^k f_i(x)$ on the diagonal and polynomials $x^k g_i(x)$ away from the diagonal.
9. Create a matrix Q from $\tilde{Q}$ by permuting rows of $\tilde{Q}$ by a permutation selected uniformly at random from the space of all permutations of n_0 elements.

Proposition 17 *Let p be an arbitrary odd number. Then Algorithm 9 always produces an invertible matrix.*

Proof Again, we think of Q as of a matrix over $\mathbb{Z}_2[x]/(x^p + 1)$. From the Leibniz formula for the determinant of Q we see that the determinant is of the form

$$\det(Q) = \sum_{\sigma \in S_{n_0}} x^{n_0 \times k} h_\sigma(x) \mod x^p + 1 \ ,$$

where the polynomials $h_\sigma(x)$ have degrees less than d. Thus the determinant is of the form

$$\det(Q) = x^{n_0 \times k} z(x) \mod x^p + 1 \ ,$$

where the polynomial $z(x)$ has degree less than d. Therefore the determinant is invertible. □

When designing a variant of a QC-LDPC McEliece cryptosystem it is necessary to guarantee that the number of potential candidates for the matrix Q is greater than the proposed security level. For this reason, we compute the number of matrices that can be generated by Algorithm 9.

Proposition 18 *Let p be an arbitrary odd number. The number of different matrices that can be generated by Algorithm 9 is*

$$n_0! p \left(\binom{\lfloor \frac{d}{n_0} \rfloor}{v}^{n_0} \binom{\lfloor \frac{d}{n_0} \rfloor}{u}^{n_0(n_0-1)} - \binom{\lfloor \frac{d}{n_0} \rfloor - 1}{v}^{n_0} \binom{\lfloor \frac{d}{n_0} \rfloor - 1}{u}^{n_0(n_0-1)} \right).$$

Proof If $k = 0$, then the algorithm can produce

$$\Sigma_0 = n_0! \binom{\lfloor \frac{d}{n_0} \rfloor}{v}^{n_0} \binom{\lfloor \frac{d}{n_0} \rfloor}{u}^{n_0(n_0-1)}$$

different matrices. If $k = 1$, then the algorithm can produce

$$\Sigma_1 = n_0! \left(\binom{\lfloor \frac{d}{n_0} \rfloor}{v}^{n_0} \binom{\lfloor \frac{d}{n_0} \rfloor}{u}^{n_0(n_0-1)} - \binom{\lfloor \frac{d}{n_0} \rfloor - 1}{v}^{n_0} \binom{\lfloor \frac{d}{n_0} \rfloor - 1}{u}^{n_0(n_0-1)} \right)$$

new matrices (i.e. matrices which could not be produced by $k = 0$). If $k = 2$, again Σ_1 new matrices can be generated. Consider the sum

$$\Sigma = \Sigma_0 + (p-1)\Sigma_1 \ .$$

Then Σ counts every possible matrix that can be generated by the algorithm, but it double counts matrices in which every block begins with a row which has all its ones distributed among the first $\left\lfloor \frac{d}{n_0} \right\rfloor - 1$ positions. Thus the total number of matrices which can be generated by the algorithm is

$$\Sigma - n_0! \binom{\lfloor \frac{d}{n_0} \rfloor - 1}{v}^{n_0} \binom{\lfloor \frac{d}{n_0} \rfloor - 1}{u}^{n_0(n_0-1)} = p\Sigma_1 \ .$$

□

4.3 How to construct the matrix H_{n_0-1}?

It is tempting to consider using Algorithm 3 to generate the last block H_{n_0-1} in the secret parity-check matrix H in the QC-LDPC and QC-MDPC McEliece cryptosystems. However, we demonstrate that this is not a good idea, as the special form of matrices generated by Algorithm 3 allows a more efficient information set decoding attack on the cryptosystem.

In particular, we consider using Algorithm 3 to generate the matrix H_{n_0-1} in the QC-MDPC McEliece cryptosystem with parameters recommended for 80-bit security in [6]. In the considered version the matrix H consists of 2 blocks of size 4801×4801 and with 45 ones in a row. 4801 is prime and $o_{4801}(2) = 1200$. Thus Algorithm 3 generates a polynomial $f(x) = x^k + \sum_{i=1}^{44} x^{k+t_i} \mod (x^n + 1)$, where $t_1, \ldots, t_{44} \in \{1, \ldots, 1199\}$ and $k \in \{0, \ldots, 4800\}$. Before modulation, the degrees of the terms in $f(x)$ all lie in an interval of length 1199. Therefore the circulant matrix corresponding to the polynomial $f(x)$ must contain a row in which all ones will be placed within the first 1200 positions. Thus the matrix H contains a row with 90 ones which ends with at least 3601 zeros.

One of the most efficient known attacks against the QC-MDPC McEliece cryptosystem is to use information set decoding algorithms to search for low-weight words in the dual code to the code generated by the public generator matrix G and thus reveal the secret matrix H. Knowing that a low-weight word ends with a large number of zeros, allows an attacker to perform a more efficient attack to recover this word. For example, an attacker aiming to recover a word of length 9802, weight 90 which ends with at least 3601 zeros can use a slightly modified Stern's algorithm [9] and is expected to find the word after approximately 2^{54} bit operations.

A matrix generated by Algorithm 3 will always contain a row ending with a large number of zeros. Therefore we do not recommend using Algorithm 3 to generate the matrix H_{n_0-1}. Instead, we recommend to generate H_{n_0-1} by repeated generation of random circulant matrices with the prescribed number of ones until an invertible matrix is obtained, as was proposed in [10].

5 Conclusion

We studied the problem how to generate invertible circulant binary matrices with a prescribed number of ones. In [10] this problem was solved by repeatedly generating random circulant matrices with the prescribed number of ones until an invertible matrix was obtained. We proposed two alternative algorithms - Algorithms 3 and 5 - which fulfill this task for a wide range of parameters. Compared with the approach from [10], our algorithms have the advantage that they generate matrices satisfying all the requirements on the first attempt. On the other hand, their disadvantage is that they generate matrices from a smaller pool. For each of our algorithms a formula for the size of the pool was derived. Thus a user is allowed to evaluate whether the size of the pool is sufficient for his/her application. The size of the pool depends on the degree of a smallest polynomial other than $x + 1$ appearing in the irreducible factorisation of $x^n + 1$ (n represents the size of the matrix). In order to achieve a large pool, the value of n should be chosen so that this degree is large.

Furthermore, we proposed algorithms to construct matrices S and Q in the QC-LDPC McEliece cryptosystem. Our algorithms assume that the size of blocks in S and Q is odd. Previously, constructions of S and Q were proposed for the case when the size of blocks was a power of 2 [2]. Choosing the size of blocks to be odd has the advantage that it prevents the recently developed attack on the QC-LDPC McEliece cryptosystem [8]. Again, the size of the pool (and also the efficiency of the algorithm for S) depends on the irreducible factorisation of $x^p + 1$ (here p represents the size of blocks). With this in mind, the best choice for the size of the block appears to be a prime p such that the multiplicative order of 2 modulo p is equal to $p - 1$. According to Artin's conjecture for primitive roots, approximately 37% of primes satisfy this condition.

In addition, we studied the possibility of using Algorithm 3 to generate the parity-check matrix H in the QC-LDPC McEliece cryptosystem and the QC-MDPC McEliece cryptosystem. However, we concluded that this would result in a significant decrease in the security of the cryptosystem.

Acknowledgements The authors wish to thank the anonymous referees for the conference ArcticCrypt 2016 for remarks which helped to improve the quality of the paper.

References

1. Baldi, M., Chiaraluce, F.: Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes. In: Proceedings of the IEEE ISIT 2007, pp. 2591–2595. Nice, France, (2007)
2. Baldi, M., Bodrato, M., Chiaraluce, F.: A new analysis of the McEliece cryptosystem based on QC–LDPC 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)
3. Chen, C.L., Peterson, W.W., Weldon, E.J.: Some results on quasi-cyclic codes. Inf. Control. **15**(5), 407–423 (1969)
4. Jungnickel, D.: Finite Fields: Structure and Arithmetics. B.I. Wissenschaftsverlag (1993)
5. Lidl, R., Niederreiter, H.: Finite fields. Encycl. Math. Appl. **20** (1983)
6. 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 (ISIT'2013), pp. 2069–2073. Istanbul (2013)
7. Otmani, A., Tillich, J.P., Dallot, L.: Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes. In: Proceedings of the First International Conference on Symbolic Computation and Cryptography (SCC 2008). Beijing, China (2008)

8. Koochak Shooshtari, M., Ahmadian-Attari, M., Johansson, T., Reza Aref, M.: Cryptanalysis of McEliece cryptosystem variants based on quasi-cyclic low-density parity check codes, vol. 10 (2016)
9. Stern, J.: A Method for Finding Codewords of Small Weight. In: Wolfmann, J., Cohen, G. (eds.) Coding Theory 1988. LNCS, vol. 388, pp. 106–113. Springer, Heidelberg (1989)
10. von Maurich, I., Guneysu, T.: Towards side-channel resistant implementations of QC-MDPC McEliece encryption on constrained devices. In: Mosca, M. (ed.) Post-Quantum Cryptography, LNCS, vol. 8772, pp. 266–282. Springer International Publishing (2014)