Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1085–1089
DOI: 10.15439/2018F99
ISSN 2300-5963 ACSIS, Vol. 15

MDPC decoding algorithms and their impact on the McEliece cryptosystem

Artur Janoska
Military University of Technology
Institute of Mathematics and Cryptology
ul. Urbanowicza 2, 00-908 Warsaw, Poland
Email: artur.janoska@wat.edu.pl

***Abstract*—In recent years, research has been conducted aimed at finding alternative asymmetric systems other than traditional systems such as RSA (Rivest–Shamir–Adleman algorithm) and ECC (Elliptic-curve cryptography). One of the most promising is code-based cryptosystems since their security is based on well-known NP-hard problems. Especially, the most interesting cryptosystem is system proposed by Misoczki et al. based on QC-MDPC codes which use the modified BitFlip algorithm as the decoding algorithm. This work presents a comparison of different variants of MDPC decoding algorithms and their impact on the cryptosystem. We present a complete analysis of modification of this algorithm and new results of the likelihood of correct word decoding for security systems which ensure security level 2^{128} and 2^{256}.**

I. Introduction

ASYMMETRIC cryptography is one of the most important cryptographic mechanism currently in use. It provides a number of options such as secure cryptographic keys exchanging over a public channel, digital signature and secure messaging. Most of those systems are based on number theory's problems, such as factorization of a large number (RSA) and the discrete logarithms in an elliptic curve. an important limitation of the indicated security systems is the fact that using Shor's algorithm [10] can solve these problems on the quantum computer so it means that they are not safe in the long-term security.

If a large-scale quantum computer is built for, that algorithm will become one of the most useful tools in cryptoanalysis, especially in public key cryptography. Nowadays there is a lot of interest in alternative systems that are not based on numerical problems. From that reason, NIST has initiated a process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms.

Promising alternative is code-based crypto algorithms those security is based on the General Decoding Problem were extensively studied for their usage in small and embedded systems. The first cryptosystems relying on coding theory was proposed by J.R. McEliece in 1978 [8]. It uses binary Goppa codes as a basis for the construction. This system presents many advantages: it is very fast for both encryption and decryption and the best-known attacks are exponential in the length of the code. Although it has proved resistance against all known attacks, it has one big flaw: the size of the public key. The public key size for the original parameters that provide security 2^{80} proposed by McEliece has 67072 bytes, against 256 bytes of a 1024-bit modulus instance of RSA.

In order to reduce key sizes, several alternative approaches for code-based cryptography were proposed. One idea to shorten key is to use codes with particular structures.Unfortunately, most of the modifications of the code structure result in the cryptosystem sensitivity to the so-called structural attacks. Such attacks aim to exploit the hidden structure, in order to recover the private key.

Another idea is to replace the binary Goppa codes with another linear code which generates matrix rolled representation and has no vulnerability to structural attacks. One of the promising algorithms was proposed by Misoczki et al. [9] They proposed to modify the McEliece cryptosystem by using a quasi-cyclic Moderate Parity Check (QC-MDPC). They achieve a very fast cryptosystem with a relatively small public key size (about 4801 b with secure level 2^{80}), lightweight implementation and still preserving the security properties of the cryptosystem. Unfortunately, in 2016 a very powerful attack was shown. The attack can recover a secret key for a system with security 2^{80} using a chosen ciphertext attack [4]. It means that this system is impractical for the encryption scheme, but still can be used as secure Key Encapsulation Mechanism.

The next section will present the motivation for the research. In the third section basic definitions and necessary statements will be presented. In the 4th section, the McEliece algorithm based on the QC-MDPC codes will be presented and in the fifth section, the decoding algorithms will be presented. Section 6 contains a description of the assumptions during the research. Section 7 contains the obtained results and their analysis. The 8th section contains a summary of completed research.

II. Motivation

The McEliece system based on QC-MDPC codes is a very interesting contribution because it has a very good performance on embedded systems and limited resources. MDPC code extends the concept a low-density parity-check (LDPC) codes [3] by using the parity check matrix with moderated sparse. Unfortunately, this leads to a significantly degraded error correction performance. However, in cryptography, We

are not interested in correcting a large number of errors, but only the number ensuring an adequate level of security.

The probabilistic decoding algorithm used in the cryptosystem is known as the modified bit flipping algorithm. During the last few years this probabilistic aspect was very intensively examined by community [1], [7]. The results of this research increased error-correcting capability for the tested systems for security 2^{80}. The quality of the solution was measured by the decoding failure rate and the number of iterations required to decrypt the message.

In the submission for the NIST standardization project, an IND-CPA[1] secure ephemeral Key Encapsulation Mechanism (KEM) based on the Quasi-Cyclic Moderate Density Parity-Check (QC-MDPC) McEliece encryption knows as QC-MDPC KEM was presented. In this work, we investigated several ways to efficiently decode erroneous MDPC codewords, especially for such codes as were proposed for the McEliece cryptosystem with security 2^{128} and 2^{256}. Additionally, we have proposed and evaluated a new way of choosing the parameter b for systems with security 2^{256}. This optimization leads to reduced decoding failure probability and fewer decoding iterations.

III. Preliminaries

In order to unify the notations and definitions, we will present a few definitions of the necessary concepts. All considerations will be conducted on finite field $\mathcal{F}_2$

The **Hamming weight** (or simply weight) of vector $x \in \mathcal{F}_2^n$ is the number of nonzero components denoted as $wt(x)$

A **binary** (n, r)**-linear code** $\mathcal{C}$ of length n and dimension r is an r-dimensional vector subspace of $\mathcal{F}_2^n$. It is spanned by the rows of a matrix $\mathbf{G} \in \mathcal{F}_2^{r\times n}$, called a generator matrix of $\mathcal{C}$. Also, it is the kernel of a matrix $\mathbf{H} \in \mathcal{F}_2^{(n-r)\times n}$ called a parity-check matrix of $\mathcal{C}$. The codeword $c \in \mathcal{C}^n$ of a vector $m \in \mathcal{F}_2^r$ is $c = mG$. The syndrome $s \in \mathcal{F}_2^{n-r}$ of a vector $\epsilon \in \mathcal{F}_2^n$ is $s = \mathbf{H}e^T$

An (n, r)-linear code is a **quasi-cyclic code** (QC) if there is some integer n_0 such that every cyclic shift of a codeword by n_0 places is again a codeword. Additionally when $n = n_0 p$ for some integer p, it is possible to have generator and parity check matrices composed by $p \times p$ circulant blocks which are completely described by their first row (or column).

An (n, r, w)-LDPC or MDPC code is a linear code of length n, dimension r which admits a parity-check matrix of constant row weight w. LDPC and MDPC codes differ in the magnitude of the row weight w. We assume for MDPC codes row weight whose scale is $O(\sqrt{n \log n})$. On the other hand, the constant row weight is usually less than 10 for LDPC.

A. MDPC and QC-MDPC code

An random (n, r, w)-MDPC code is easily generated by selecting a random parity-check matrix $\mathbf{H} \in \mathcal{F}_2^{r\times n}$ of row weight w. We only have to check that the rightmost $r \times r$ block is full rank. If not, wecan swap a few columns to get a full rank matrix. The general definition of MDPC codes can be found in [9]. For the purpose of this article, construction using $n_0 = 2$ will be discussed.

The (n, r, w)–QC-MDPC codes where $n = 2p$ and $r = p$. So then the parity check matrix has the form

$$H = [H_0|H_1]$$

where H_i is a $r \times r$ circulant block. To define the parity-check matrix $\mathbf{H}$ we pick up a random first row of weight w and the other $r - 1$ rows are obtained from $r - 1$ quasi-cyclic shift of this first row.

A generator matrix $\mathbf{G}$ in the row reduced echelon form can be easily derived from the H_i's blocks. Assuming that the block H_1 is non-singular (which particularly implies row h_i of matrix H_1 has $wt(h_i)$ odd) we construct a generator-matrix

$$G = \left[\quad I \quad \middle| \left(H_1^{-1} \cdot H_0\right)^T \right]$$

IV. QC-MDPC McEliece variant

In order to define a McEliece variant based on t-error correcting (n, r, w)–QC-MDPC code we need to fix some MDPC decoding algorithm equipped with the knowledge of $\mathbf{H}$ (denoted as Ψ_H). Encryption, decryption and key generation for the QC-MDPC McEliece variant cryptosystem are defined as follows.

Key Generation. The key Generation procedure consists of two steps. First we generate a parity-check matrix $\mathbf{H} \in \mathcal{F}_2^{r\times n}$ of a t-error-correcting(r, r, w)–QC-MDPC code by choosing the first row of the parity-check matrix. The second step is to generate the corresponding generator matrix $\mathbf{G} \in \mathcal{F}_2^{r\times n}$ in the row reduced echelon form.

The public-key of this system is the tuple $(\mathbf{G}, t)$ and the private-key is matrix $\mathbf{H}$.

Encryption. In order to encrypt message $m \in \mathcal{F}_2^r$ we need to generate random vector $\epsilon \in \mathcal{F}_2^n$ of $wt(\epsilon) \leq t$ and compute

$$x \leftarrow m\mathbf{G} + \epsilon$$

where $x \in \mathcal{F}_2^n$ is a ciphertext.

Decryption. To decrypt $x \in \mathcal{F}_2^n$ into $m \in \mathcal{F}_2^r$ we compute

$$m\mathbf{G} \leftarrow \Psi_H(x)$$

If the generated matrix $\mathbf{G}$ is in the row reduced echelon form, we extract the message m from the first r position of $m\mathbf{G}$.

A. Security and Parameter Selection

Theoretical security of the QC-MDPC McElice cryptosystem has been presented [9]. In particular, an analysis of the safety and impact a quasi-cyclic structure on the security was presented. Recently, new attacks on system using those codes have been proposed. The most important is very powerful attack using a quasi-cyclic form of the parity check matrix [4]. The attack leverages the fact that there is some probability, termed the Decoding Failure Rate (DFR), that the decoding may fail to compute the errors.

Parameters for the examined systems are based on analyzes carried out in the work [11]. The suggested parameters are presented in Table I. The tests have been carried out using these values.

[1]Indistinguishability under chosen-plaintext attack

Table I
SUGGESTED PARAMETER SETS FOR CLASSIC SECURITY AND QUANTUM SECURITY [11]

System	Classical security	Quantum Security	n	r	w	t	Public Key size
McEliece	80	58	9602	4801	90	84	4801
McEliece	128	86	19714	9857	142	134	9857
McEliece	256	154	65542	32771	274	264	32771

V. DECODING ALGORITHMS

Decoding in the McEliece cryptosystem is a more complex operation than encryption. For lightweight embedded systems, the best solution seems to be the variant of the Gallager's bit flipping algorithm [3], dedicated to the LDPC code. Positive aspects of this solution are its simplicity and lack of floating-point arithmetic. On the other hand, the disadvantage of this solution is that we find the codeword with some probability determined by threshold b, which will be later discussed.

The algorithm works as follows. At each iteration, the number of unsatisfied parity-check equations associated to each bit of the message is computed. Each bit associated with more than b unsatisfied equations is flipped and the syndrome is recomputed. This process is repeated until either the syndrome becomes zero or after a maximum number of iteration is reached. We name this algorithm Algorithm 1.

The algorithm has complexity $O(nwI)$, where I stands for the average number of iterations. The most important difference from the algorithm proposed by Gallager is how threshold b is determined. The first proposition was to precompute thresholds for each iteration i. The threshold is set as the maximum number of unsatisfied parity-check equations $b = max(\sigma_i)$. In [9] the authors suggest to use $b = max(\sigma_i) - \delta$, for some small δ (suggested in [9] is $\delta \approx 5$). In [7] the authors propose incrementing the precomputed thresholds by $\Delta = 1$ and in the case of a decoding failure increase it to $\Delta = \Delta + 1$. Decoding is restarted with the adapted Δ until reaching a predefined $\Delta_{max} = 5$. The survey of this optimalization is included in Table II

Algorithm 2 Modified Gallager's bit flipping algorithm

Input: $x \in \mathrm{F}_2^n$
Require: $H \in M_r^n, r_{max} \in \mathcal{Z}_+$
Output: $m \in C$ lub $error$
    $s \leftarrow xH^T$
2: **for** $r \in \{0, \ldots, r_{max-1}\}$ **do**
        for $i \in \{0, \ldots, n-1\}$ **do**
4: $\sigma_i \leftarrow \langle s, h_i \rangle \in Z$ $\star$
            if $\sigma_i \geq b$ **then**
6: $x_i \leftarrow x_i \oplus 1$ $\star\star$
                $s = s \oplus h_i$
8: **end if**
        end for
10: **if** $s = 0^r$ **then**
            return x
12: **end if**
    end for
14: **return** $error$

$\star$ $\langle \cdot, \cdot \rangle$ – means scalar product of two vectors, h_i – means i-th column of matrix H
$\star\star$ x_i – means i-th position in vector x

A. Effective implementation

Code-based systems allow lightweight and effective implementations in devices with limited hardware resources and all operations are much less complex when compared to the other post-quantum systems.

Encryption involved simple operations such as vector-matrix multiplication followed by an addition.

Decoding is a more complex operation, but it is possible to reduce the cost of this operation by some improvements. In [7] they propose to improve the syndrome computation. They observe that if i-the bit of ciphertext is flipped, the new syndrome is equal to the old syndrome accumulated with row h_i of the parity check matrix. The authors suggest updating the syndrome directed after flipped i-th bit. This modified algorithm is presented as algorithm 2.

Key generation is a very simple operation, as it mainly uses a pseudorandom generator. In this paper, we do not investigate selecting pseudorandom generators appropriate for the code-based systems. Some research focused on software implementations in this area is presented in [2].

VI. EXPERIMENTAL SETUP

In this work we focus on the parameter selection of decoding algorithms for the proposed McEliece systems based on QC-MDPC codes and corresponding to other security levels according to the Table II a total 10000 random decoding trials were evaluated on a computer equipped with an Intel Core i7 2670QM running at 2.20 GHz.

The research was conducted to investigate the impact of choosing the parameter value of the Decoding Failure Rate (DFR) and to examine the number of rounds needed for correct decryption of the ciphertext. The generation of plaintext and error pattern was based on a uniform distribution.

As part of the study, particular attention was paid to decoding algorithms that use the maximum value of the coefficient $sigma_i$ to calculate the b parameter. In this work the results for the following algorithms will be presented: Decoder:

Decoder A Algorithm 1, threshold b value chose using Misoczki method,

Decoder B Algorithm 2, threshold b value chose using Misoczki method,

Decoder C Algorithm 2 with method choosing threshold b proposed in this work with increment δ by two (starting from 5 to 9) every two iterations,

Table II
PROPOSED METHOD OF CHOOSING THRESHOLD b

	Proposed by	Year	Method	Comments
1	Galager [3]	1962	$b_i - const, i \in \{1, \ldots, r\}$	$b_i \in < 28, 26, 24, 22, 20 >$
2	Huffman and Pless [6]	2010	$b = \#max$	$\#max$ means the maximal value σ_i
3	Misoczki [9]	2013	$b = \#max - \delta$	Decrement δ from 5 to 0 for every incorrect decoding
4	Heyse [5]	2013	$b = b_i + \delta$	Decrement δ from 5 to 0 for every incorrect decoding
5	This paper	2018	$b = \#max - \delta$	Increment δ every two iterations by two starting from 5 to 9
6	This paper	2018	$b = \#max - \delta$	similarly to the Misoczki proposal, only the starting point is different for each level of security, starting from 5 to 7 respectively

Decoder D Algorithm 2 with method choosing threshold b proposed in this work similar to the Misoczki proposal, only the starting point is different for each level of security, starting from 5 to 7 respectively.

The method proposed by Huffman and Pless is a special case of the method proposed by Misoczki et al. and, therefore, it will not be considered in the study. [7]

VII. RESULTS

In the beginning, we focused on the analysis of the average distribution of σ_i coefficients depending on the chosen security level. The graph for these values is presented in Figure 1. The average distribution of the parameter σ_i represents the first distribution before any coding algorithm is used. As can be seen, the values of the parameter σ_i for systems with security 2^{80} and 2^{128} are similar to each other while for the 2^{256} security system, the average values σ_i are much higher and more intense.

These are the first symptoms that the decoding parameter δ for the safest system will need to be modified. An interesting relationship is that if we take the average distribution for σ_i for the files corresponding to the word correctly decoded, we can say that it is different depending on the choice of the decoding algorithm. an example of a distribution for the system with security level 2^{128} is shown in Figure 2. The given property can be used to distinguish, which decoding algorithm was used.

The Decode Failure Rates for the tested algorithms are listed in Table III for all levels of security considered in the table I.

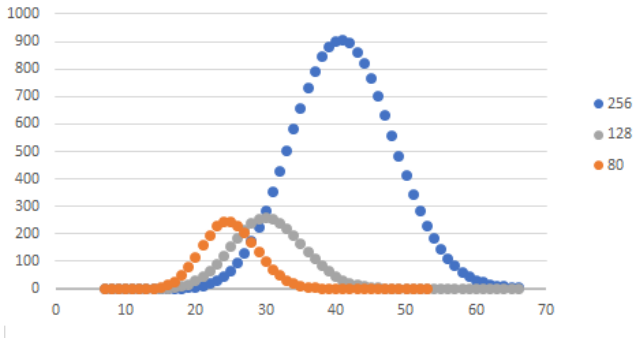

Figure 1. The average distribution of σ_i coefficients depending on the chosen security level.

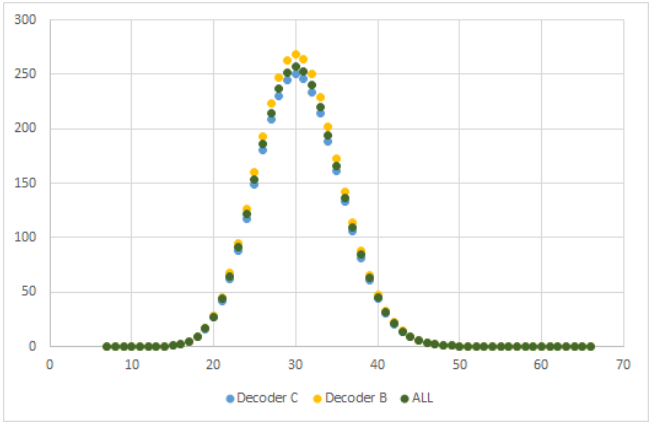

Figure 2. The average distribution of σ_i coefficients depending on the chosen decoding algorithm.

Analysing the results obtained for the Decoder A, which was proposed in the original work, it can be seen that for higher security levels it is not useful. Especially for the level 256, wherein any of the examined cases, the message was properly decoded.

Comparing the two decoders from literature (Decoder a and Decoder B), in Decoder B provides much better decoding failure rate for higher security. However, it still has too high DFR to be practically used. Our proposition to change the δ value for higher security level has a very strong impact on the decoding failure rate. In addition, our solution led to a smaller average number of decoding iterations, respectively 13% and 26% less.

Comparing decoders B, C, D we can see that the decoding failure rate and the number of decoding iterations are not strongly correlated. The improvement of the DFR level does not always result in an improvement in the number of decoding iterations. If we properly manipulate the parameters of decoding algorithm, we can get properties adapted to the specific application.

Additionally, as part of the study, the decoding failure rate was analyzed depending on the distribution of words with the desired Hamming weight. The test assumes that the codewords have the Hamming weight equal to $r/2$. It was noted that the DFR, as well as the number of rounds, increased slightly.

Table III
EVALUATION OF THE PERFORMANCE AND ERROR CORRECTION CAPABILITY OF THE TESTED ALGORITHMS. NOTE, A DFR OF 0 MEANS THAT NO DECODING ERROR OCCURRED DURING OUR EVALUATIONS BUT THE DECODERS ARE STILL PROBABILISTIC.

Name	80		128		256	
	DFR	Round Number	DFR	Round Number	DFR	Round Number
Decoder A	0.00000*	6.30000	0.21400	9.65600	1.00000	–
Decoder B	0.00000*	3.02850	0.00499	6.31087	0.16400	8.69617
Decoder C	0.01600	3.98831	0.07304	5.45662	0,00300	6,72700
Decoder D	0,00000*	3.02850	0.00440	5.54209	0,00000*	6.87500

VIII. CONCLUSIONS

In this work, we examined various variants of the decoding algorithm depending on the choice of the threshold. Additionally, we presented the method of selecting the threshold for codes used in high-security levels systems.

Additionally, as part of the work, an analysis of the possibility of improving the bit-flipping algorithm in applications to MDPC codes was presented.

An interesting fact is that if we use algorithms with a relatively high DFR coefficient, we can distinguish these algorithms based on the analysis of histograms discussed in Section VII. However, it should be noted that when using low-DFR decoding algorithms, the corollary analysis does not apply.

In the light of the achieved results, it can be concluded that the modified Bit Flipping algorithm can be successfully applied to various types of key encapsulation mechanism based on QC-MDPC codes. Especially, for the QC-MDPC-KEM algorithm reported to a process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms organized by NIST.

In further work I will focus on compare and analysis of other algorithms for decoding QC-MDPC codes, in particular the "One-round Bit Flipping" algorithm and their use in the KEM QC-MDPC system.

REFERENCES

[1] Julia Chaulet and Nicolas Sendrier. “Worst case QC-MDPC decoder for McEliece cryptosystem”. In: *Information Theory (ISIT), 2016 IEEE International Symposium on*. IEEE. 2016, pp. 1366–1370.

[2] Nir Drucker and Shay Gueron. “A toolbox for software optimization of QC-MDPC code-based cryptosystems”. In: (2017).

[3] Robert Gallager. “Low-density parity-check codes”. In: *IRE Transactions on Information Theory* 8.1 (1962), pp. 21–28.

[4] Qian Guo, Thomas Johansson, and Paul Stankovski. “A key recovery attack on MDPC with CCA security using decoding errors”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2016, pp. 789–815.

[5] Stefan Heyse, Ingo Von Maurich, and Tim Güneysu. “Smaller keys for code-based cryptography: QC-MDPC McEliece implementations on embedded devices”. In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2013, pp. 273–292.

[6] W. Cary Huffman and Vera Pless. *Fundamentals of Error-correcting Codes*. Cambridge University Press, 2010.

[7] Ingo Von Maurich, Tobias Oder, and Tim Güneysu. “Implementing QC-MDPC McEliece Encryption”. In: *ACM Transactions on Embedded Computing Systems (TECS)* 14.3 (2015), p. 44.

[8] Robert J McEliece. “A public-key cryptosystem based on algebraic”. In: *Coding Thv* 4244 (1978), pp. 114–116.

[9] Rafael Misoczki et al. “MDPC-McEliece: New McEliece variants from moderate density parity-check codes”. In: *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE. 2013, pp. 2069–2073.

[10] Peter W Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”. In: *SIAM review* 41.2 (1999), pp. 303–332.

[11] Atsushi Yamada et al. “QC-MDPC KEM: A Key Encapsulation Mechanism Based on the QC-MDPC McEliece Encryption Scheme - First Round Submission in NIST Post-Quantum Cryptography Standardization”. In: (2017).