sciendo

TATRA MOUNTAINS
Mathematical Publications

DOI: 10.2478/tmmp-2019-0007
Tatra Mt. Math. Publ. **73** (2019), 83–96

A CRYPTOGRAPHIC SYSTEM BASED ON A NEW CLASS OF BINARY ERROR-CORRECTING CODES

Pál Dömösi*—Carolin Hannusch**—Géza Horváth*

Faculty of Informatics, University of Debrecen, Hungary

ABSTRACT. In this paper we introduce a new cryptographic system which is based on the idea of encryption due to [McEliece, R. J. *A public-key cryptosystem based on algebraic coding theory*, DSN Progress Report. **44**, 1978, 114–116]. We use the McEliece encryption system with a new linear error-correcting code, which was constructed in [Hannusch, C.—Lakatos, P.: *Construction of self-dual binary* $2^2k, 2^2k-1, 2^k$*-codes*, Algebra and Discrete Math. **21** (2016), no. 1, 59–68]. We show how encryption and decryption work within this cryptosystem and we give the parameters for key generation. Further, we explain why this cryptosystem is a promising post-quantum candidate.

1. Introduction

In 2016 it was stated in the report of the NIST (National Institute of Standards and Technology, US Department of Commerce) [7] that we probably can not consider encryption by RSA [28] or by ECDH [11] (cryptography based on elliptic curves) to be still secure in some decades. Thus it is topical to think about new principles of encrypting and decrypting information and to develop new cryptographic systems. One possibility for developing such systems, which also have a public key (public-key cryptography was invented by Diffie and Hellmann [8]), is cryptography based on linear codes. Such a scheme was published by McEliece [19] in 1978. Its vulnerability is discussed in [6, 16, 26]. A detailed overview of the McEliece cryptosystem and its security can be found in [27].

2010 Mathematics Subject Classification: 94A60, 11T71.
Keywords: public key encryption; error-correcting codes; McEliece encryption; post-quantum cryptography.
*This work was supported by the National Research, Development and Innovation Office of Hungary under the Grant no. TÉT 16-1-2016-0193.
**This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.

In addition, the McEliece system has two further big disadvantages. One is its key size, which is much bigger than in the case of other cryptosystems, like RSA.

The other disadvantage is that McEliece encryption uses Goppa codes. The generation of a Goppa code inquires computing with primitive elements of a finite field, which is a long and complicated computation for efficient code length.

Wang [30] could improve the McEliece system in such a way that it works with a random linear code instead of a Goppa code. Even before Wang, Niederreiter [23] suggested the usage of generalized Reed-Solomon codes and Berger and Loidreau [3] suggested the usage of a subcode of generalized Reed-Solomon codes. Sidelnikov [29] suggested Reed-Muller codes for use in the McEliece cryptosystem and Janwa and Moreno [18] suggested algebric-geometric codes. Baldi, Botrado and Chiaraluce [1] worked on a McEliece cryptosystem based on LDPC codes, Misoczki, Tillich, Sendrier and Barreto [22] on MDPC codes. Löndahl and Johansson [17] suggested convolutional codes. Berger, Cayrel, Gaborit, and Otmani [2] as well, as Misoczki and Barreto [20] suggested quasi-cyclic and quasi-diadic structures for a compact version of the McEliece cryptosystem. Most of these cryptosystems were attacked successfully. Cryptosystems among these which were recognized until very recently as secure systems are those based on MDPC (Moderate Density Parity Check)-codes [22], QC-LDPC (Quasi-Cyclic Low Density Parity Check)-codes [1] and the original system based on binary Goppa codes. On the other hand, Guo, Johansson, and Stankovski [14] showed a key recovery attack on MDPC with CCA security using decoding errors in 2016. In addition, Fabšič, Hromada, Stankovski, Zajac, Guo, Johansson [12] also showed a reaction attack on the QC-LDPC McEliece Cryptosystem in 2017. Therefore, these systems can not be considered secure anymore.

In contrast to the systems mentioned above, the cryptosystem due to Wang [30] is considered to be secure.

In the current paper we introduce an encryption scheme [9] based on linear error-correcting codes, which were introduced in [15]. Computing a generator matrix for this code inquires only binary multiplication and addition. Thus operations are very easy to perform, which is an advantage of our cryptosystem.

2. The encryption scheme

We denote the field of two elements (namely $\{0,1\}$) by $\mathbb{F}_2$. Further, $\mathcal{M}_{k\times m}(\mathbb{F}_2)$ denotes the set of matrices with k rows and m columns over $\mathbb{F}_2$.

Let $u = (u_0, \ldots, u_{2047})$ be a message, where $u_i \in \mathbb{F}_2$ for $i \in \{0, \ldots, 2047\}$. Furthermore, let $S \in \mathcal{M}_{2048\times 2048}(\mathbb{F}_2)$ be a nonsingular matrix, i.e. its determinant is not zero.

Let G be a generator matrix of a binary $(4096, 2048, 64)$-code, which was constructed in [15]. Such a code is not unique, therefore we will introduce this class of codes (also called HL-codes) in the next section in detail. The code generated by G can correct t errors, if $t \leq \lfloor \frac{64-1}{2} \rfloor = 31$.

Let $P \in \mathcal{M}_{4096\times 4096}(\mathbb{F}_2)$ be a permutation matrix, i.e., it has exactly one 1 in each row and it has no coinciding rows. We denote the usual matrix multiplication by $*$.

Public key: $(S * G * P, 31)$.

Private key: (S, G, P).

Encrypting a plaintext u:

$$c = u * S * G * P + e,$$

where $*$ denotes the usual matrix multiplication and e is a binary random vector with at most 31 non-zero elements. Thus the encrypted message c is a vector of length 4096.

3. Construction of the HL-code

In Theorem 1 of [15] there is constructed a class of binary self-dual codes of length 2^m with dimension 2^{m-1} and with minimum distance $2^{\frac{m}{2}}$. In this section we show the construction of one such code for $m = 12$.

Now we explain how a matrix G can be generated. The generator matrix G consists of 2048 rows and 4096 columns. The first 13 rows are the vectors v_i, for $i \in \{0, \ldots, 12\}$, defined by the following:

$$v_0 = (\underbrace{1, 1, 1, 1, 1, 1, 1, 1, \ldots, 1, 1, 1, 1, 1, 1, 1, 1}_{4096}),$$

$$v_1 = (0, 1, 0, 1, 0, 1, 0, 1, \ldots, 0, 1, 0, 1, 0, 1, 0, 1),$$

$$v_2 = (0, 0, 1, 1, 0, 0, 1, 1, \ldots, 0, 0, 1, 1, 0, 0, 1, 1),$$

$$v_3 = (0, 0, 0, 0, 1, 1, 1, 1, \ldots, 0, 0, 0, 0, 1, 1, 1, 1),$$

$$v_4 = (\underbrace{0,\ldots,0}_{8},\underbrace{1,\ldots,1}_{8},\ldots,\underbrace{0,\ldots,0}_{8},\underbrace{1,\ldots,1}_{8}),$$
$$v_5 = (\underbrace{0,\ldots,0}_{16},\underbrace{1,\ldots,1}_{16},\ldots,\underbrace{0,\ldots,0}_{16},\underbrace{1,\ldots,1}_{16}),$$
$$v_6 = (\underbrace{0,\ldots,0}_{32},\underbrace{1,\ldots,1}_{32},\ldots,\underbrace{0,\ldots,0}_{32},\underbrace{1,\ldots,1}_{32}),$$
$$v_7 = (\underbrace{0,\ldots,0}_{64},\underbrace{1,\ldots,1}_{64},\ldots,\underbrace{0,\ldots,0}_{64},\underbrace{1,\ldots,1}_{64}),$$
$$v_8 = (\underbrace{0,\ldots,0}_{128},\underbrace{1,\ldots,1}_{128},\ldots,\underbrace{0,\ldots,0}_{128},\underbrace{1,\ldots,1}_{128}),$$
$$v_9 = (\underbrace{0,\ldots,0}_{256},\underbrace{1,\ldots,1}_{256},\ldots,\underbrace{0,\ldots,0}_{256},\underbrace{1,\ldots,1}_{256}),$$
$$v_{10} = (\underbrace{0,\ldots,0}_{512},\underbrace{1,\ldots,1}_{512},\ldots,\underbrace{0,\ldots,0}_{512},\underbrace{1,\ldots,1}_{512}),$$
$$v_{11} = (\underbrace{0,\ldots,0}_{1024},\underbrace{1,\ldots,1}_{1024},\underbrace{0,\ldots,0}_{1024},\underbrace{1,\ldots,1}_{1024}),$$
$$v_{12} = (\underbrace{0,\ldots,0}_{2048},\underbrace{1,\ldots,1}_{2048}).$$

In order to construct all rows of G, we will take the products of the vectors v_i according to some conditions. We multiply two vectors by multiplying its coordinates. The next rows will be the following in lexicographical order:

- all possible products of two different v_i vectors $(v_1v_2, v_1v_3, \ldots)$,
- all possible products of three different v_i vectors $(v_1v_2v_3, v_1v_2v_4, \ldots)$,
- all possible products of four different v_i vectors $(v_1v_2v_3v_4, v_1v_2v_3v_5, \ldots)$,
- all possible products of five different v_i vectors $(v_1v_2v_3v_4v_5, v_1v_2v_3v_4v_6, \ldots)$.

Remark 1. So far, we know 1586 rows of G. The code generated by these rows is a Reed-Muller code $RM(5, 12)$.

In order to compute the last 462 rows of the generator matrix G, we need the following definition.

Definition 2. Let X be the following set

$$X = \{a = (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12})|,$$
$$a_i \in \mathbb{F}_2, \sum_{i=1}^{12} a_i = 6, i \in \{1, \ldots, 12\}\}.$$

We call the binary tuple $\mathbf{1} - a$ the *complement* of a, where

$$\mathbf{1} = (\underbrace{1, \ldots, 1}_{12}).$$

Further we say that a subset Y of X is *complement-free*, if each $a \in Y$ implies $\mathbf{1} - a \notin Y$.

Remark 3. The order of such a complement-free set Y is $\frac{1}{2}\binom{12}{6} = 462$.

The last 462 rows of G are constructed according to the complement-free set Y. For each element $(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}) \in Y$ we add the row

$$v_1^{a_1} v_2^{a_2} v_3^{a_3} v_4^{a_4} v_5^{a_5} v_6^{a_6} v_7^{a_7} v_8^{a_8} v_9^{a_9} v_{10}^{a_{10}} v_{11}^{a_{11}} v_{12}^{a_{12}},$$

where $v_i^0 = v_0$ and $v_i^1 = v_i$.

Theorem 4. *The linear code generated by G is an error-correcting binary self--dual $(4096, 2048, 64)$-code, which can correct 31 errors.*

Proof. Theorem 4 is a special case of Theorem 1 in [15] for $m = 12$. □

4. Creating a complement-free set

Imagine we write down all binary 12-tuples with exactly six 1-s into a table, such that on the left-hand side $a_1 = 1$ for all elements a. Then we write down the complement of each 12-tuple on the righthand side of the table:

Table 1. This table has 462 rows (see Remark 3).

Row	a	$\mathbf{1} - a$
1	$(1,1,1,1,1,1,0,0,0,0,0,0)$	$(0,0,0,0,0,0,1,1,1,1,1,1)$
2	$(1,1,1,1,1,0,1,0,0,0,0,0)$	$(0,0,0,0,0,1,0,1,1,1,1,1)$
⋮	⋮	⋮
462	$(1,0,0,0,0,0,0,1,1,1,1,1)$	$(0,1,1,1,1,1,1,0,0,0,0,0)$

Now, we take a random binary string with 462 coordinates

$$r = (r_1, r_2, \ldots, r_{462}),$$

where each

$$r_i \in \{0, 1\} \quad \text{for all} \quad i \in \{1, \ldots, 462\}.$$

If $r_i = 0$, then we choose the element a in the ith row. If $r_i = 1$, then we choose the element $\mathbf{1} - a$ in the ith row for all $i \in \{1, \ldots, 462\}$. By this method, we get a randomly created complement-free set Y.

Remark 5. Since the first 1586 rows of G are fixed, it is enough to share r in the private key.

5. Key generation

The public key is the matrix $S * G * P \in \mathcal{M}_{2048\times 4096}$, i.e., the size of the key is 8 Mbit. The key generation works in the following steps:

(1) Build up the first 1586 rows of G as described in Section 3. These can be saved for further computations.

(2) Create a complement-free set Y as described in Section 4.

(3) Construct the last 462 rows of G according to the complement-free set Y. For each element

$$(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}) \in Y$$

we add the row

$$v_1^{a_1} v_2^{a_2} v_3^{a_3} v_4^{a_4} v_5^{a_5} v_6^{a_6} v_7^{a_7} v_8^{a_8} v_9^{a_9} v_{10}^{a_{10}} v_{11}^{a_{11}} v_{12}^{a_{12}},$$

where

$$v_i^0 = v_0 \qquad \text{and} \quad v_i^1 = v_i.$$

(4) Take a nonsingular matrix $S \in \mathcal{M}_{2048\times 2048}(\mathbb{F}_2)$.

(5) Take a permutation matrix $P \in \mathcal{M}_{4096\times 4096}$.

(6) Compute $S * G * P$.

6. The decryption scheme

Notice that we know the matrices S, G and P.

Decrypting a ciphertext c

Consider

$$w = c * P^{-1} = (u * S * G * P + e) * P^{-1}$$

and apply the Reed-type majority logic decoding scheme [25] deriving $u * S$ from w. Then compute the original plaintext as

$$u = u * S * S^{-1}.$$

EXAMPLE 6. We show an example for the decoding scheme of Reed for $m = 4$. For simplicity, we suppose S is the identity matrix $I_{8\times 8}$ and P is the identity matrix I_{16x16} in this example.

Let G be the following generator matrix of a $(16, 8, 4)$-code due to the complement-free set

$$Y = \{(0,0,1,1),(0,1,0,1),(1,0,0,1)\} :$$

$$G = \begin{pmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\ 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\ 0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1\\ 0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1 \end{pmatrix} = \begin{pmatrix} v_0\\ v_1\\ v_2\\ v_3\\ v_4\\ v_3v_4\\ v_2v_4\\ v_1v_4 \end{pmatrix}.$$

(Then $m = 2$, and thus there are no rows of G with products of $2, 3. \ldots, m-1$ different v_i vectors in $\{v_1v_2, v_1v_3, v_4\}$. Moreover, by

$$(0,0,1,1) \in Y, \quad v_1^0 v_2^0 v_3^1 v_4^1 = v_3v_4,$$

by

$$(0,1,0,1) \in Y, \quad v_1^0 v_2^1 v_3^0 v_4^1 = v_2v_4,$$

and by

$$(1,0,0,1) \in Y, \quad v_1^1 v_2^0 v_3^0 v_4^1 = v_1v_4.)$$

We encode the message $u = (u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7)$ into

$$w = u * G = (w_0, w_1, w_2, \ldots, w_{15}).$$

Decoding due to the majority logic principle is a useful tool in this case, see [21]. In the case $m = 4$ the code is able to correct 1 error. We will see how. First we have:

$$u * G = u_0g_0 + \cdots + u_7g_7,$$

where $g_0, \ldots g_7$ are the column vectors of G. In more details,

$$u * G = \begin{pmatrix} u_0 \\ u_0 + u_1 \\ u_0 + u_2 \\ u_0 + u_1 + u_2 \\ u_0 + u_3 \\ u_0 + u_1 + u_3 \\ u_0 + u_2 + u_3 \\ u_0 + u_1 + u_2 + u_3 \\ u_0 + u_4 \\ u_0 + u_1 + u_4 + u_7 \\ u_0 + u_2 + u_4 + u_6 \\ u_0 + u_1 + u_2 + u_4 + u_6 + u_7 \\ u_0 + u_3 + u_4 + u_5 \\ u_0 + u_1 + u_3 + u_4 + u_5 + u_7 \\ u_0 + u_2 + u_3 + u_4 + u_5 + u_6 \\ u_0 + u_1 + u_2 + u_3 + u_4 + u_5 + u_6 + u_7 \end{pmatrix} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \\ w_9 \\ w_{10} \\ w_{11} \\ w_{12} \\ w_{13} \\ w_{14} \\ w_{15} \end{pmatrix}. \quad (1)$$

If $u = (1,1,0,0,0,0,0,0)$, then $w = (1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0)$. Imagine that 1 error occurred in the second position, thus we receive

$$w^\star = (1,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0).$$

Since

$$w = u_0v_0 + u_1v_1 + u_2v_2 + u_3v_3 + u_4v_4 + u_5v_3v_4 + u_6v_2v_4 + u_7v_1v_4, \quad (2)$$

we will be able to recover the symbols u_5, u_6 and u_7. We get from the equation (1) and from the relations proved in [25]

$$\begin{array}{rcccccccc} u_5 &=& w_0 &+& w_4 &+& w_8 &+& w_{12} &=& 0, \\ u_5 &=& w_1 &+& w_5 &+& w_9 &+& w_{13} &=& 1, \\ u_5 &=& w_2 &+& w_6 &+& w_{10} &+& w_{14} &=& 0, \\ u_5 &=& w_3 &+& w_7 &+& w_{11} &+& w_{15} &=& 0. \end{array} \quad (3)$$

Since we have four possibilities for the value of u_5, but each w_i is only appearing once for all $i \in \{0,\dots,15\}$, we can decode the right value by the majority principle, i.e., $u_5 = 0$.

We recover the values for u_6 and u_7 similarly:

$$\begin{array}{rcccccccc} u_6 &=& w_0 &+& w_2 &+& w_8 &+& w_{10} &=& 0, \\ u_6 &=& w_1 &+& w_3 &+& w_9 &+& w_{11} &=& 1, \\ u_6 &=& w_4 &+& w_6 &+& w_{12} &+& w_{14} &=& 0, \\ u_6 &=& w_5 &+& w_7 &+& w_{13} &+& w_{15} &=& 0. \end{array} \quad (4)$$

Thus $u_6 = 0$ and by

$$\begin{array}{rcccccccccc} u_7 & = & w_0 & + & w_1 & + & w_8 & + & w_9 & = & 1, \\ u_7 & = & w_2 & + & w_3 & + & w_{10} & + & w_{11} & = & 0, \\ u_7 & = & w_4 & + & w_5 & + & w_{12} & + & w_{13} & = & 0, \\ u_7 & = & w_6 & + & w_7 & + & w_{14} & + & w_{15} & = & 0. \end{array} \tag{5}$$

we have $u_7 = 0$. Now we subtract

$$u_5 v_3 v_4 + u_6 v_2 v_4 + u_7 v_1 v_4$$

from w and we get

$$w - u_5 v_3 v_4 + u_6 v_2 v_4 + u_7 v_1 v_4 = u_0 v_0 + u_1 v_1 + u_2 v_2 + u_3 v_3 + u_4 v_4 = w'. \tag{6}$$

We denote $w' = (w'_0, w'_1, \ldots, w'_{15})$. In our example $w' = w$. By the equation (6) and the relations proved in [25] we obtain eight possibilities for each value of u_1, u_2, u_3, and u_4 :

$$\begin{array}{rcccccc} u_1 & = & w_0 & + & w_1 & = & 0, \\ u_1 & = & w_2 & + & w_3 & = & 1, \\ u_1 & = & w_4 & + & w_5 & = & 1, \\ u_1 & = & w_6 & + & w_7 & = & 1, \\ u_1 & = & w_8 & + & w_9 & = & 1, \\ u_1 & = & w_{10} & + & w_{11} & = & 1, \\ u_1 & = & w_{12} & + & w_{13} & = & 1, \\ u_1 & = & w_{14} & + & w_{15} & = & 1. \end{array} \qquad \begin{array}{rcccccc} u_2 & = & w_0 & + & w_2 & = & 0, \\ u_2 & = & w_1 & + & w_3 & = & 1, \\ u_2 & = & w_4 & + & w_6 & = & 0, \\ u_2 & = & w_5 & + & w_7 & = & 0, \\ u_2 & = & w_8 & + & w_{10} & = & 0, \\ u_2 & = & w_9 & + & w_{11} & = & 0, \\ u_2 & = & w_{12} & + & w_{14} & = & 0, \\ u_2 & = & w_{13} & + & w_{15} & = & 0. \end{array}$$

$$\begin{array}{rcccccc} u_3 & = & w_0 & + & w_4 & = & 0, \\ u_3 & = & w_1 & + & w_5 & = & 1, \\ u_3 & = & w_2 & + & w_6 & = & 0, \\ u_3 & = & w_3 & + & w_7 & = & 0, \\ u_3 & = & w_8 & + & w_{12} & = & 0, \\ u_3 & = & w_9 & + & w_{13} & = & 0, \\ u_3 & = & w_{10} & + & w_{14} & = & 0, \\ u_3 & = & w_{11} & + & w_{15} & = & 0. \end{array} \qquad \begin{array}{rcccccc} u_4 & = & w_0 & + & w_8 & = & 0, \\ u_4 & = & w_1 & + & w_9 & = & 1, \\ u_4 & = & w_2 & + & w_{10} & = & 0, \\ u_4 & = & w_3 & + & w_{11} & = & 0, \\ u_4 & = & w_4 & + & w_{12} & = & 0, \\ u_4 & = & w_5 & + & w_{13} & = & 0, \\ u_4 & = & w_6 & + & w_{14} & = & 0, \\ u_4 & = & w_7 & + & w_{15} & = & 0. \end{array}$$

Thus major logic decision will also lead us to the right values of $u_1 = 1, u_2 = 0, u_3 = 0$ and $u_4 = 0$. Afterwards, we subtract

$$u_1 v_1 + u_2 v_2 + u_3 v_3 + u_4 v_4$$

from

$$w - u_5 v_3 v_4 + u_6 v_2 v_4 + u_7 v_1 v_4$$

and we get

$$w'' = u_0 v_0 + e,$$

where e is the error vector, whose coordinates are all 0, except at most one coordinate, which may be 1.

In our case $w - v_1 = (1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1) = u_0 v_0 + e$. Thus we obtain $u_0 = 1$ and $e = (0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$.

In general, the value of u_0 depends on the number of 1-s in w''.

The decoding principle works similarly for our $C - (4096, 2048, 64)$-code.

In [13, G o p a l a n, K l i v a n s and Z u c k e r m a n] showed that list decoding is possible for Reed-Muller codes with polynomial complexity of n^3, where n is the code length. In [10, D u m e r, K a b a t i a n s k y and T a v e r n i e r] improved this bound to $n \ln^{s-1} n$ for the Reed-Muller code $RM(s,m)$ with code length $n = 2^m$. In our case $m = 2k$ and the code C generated by G is between two RM-codes

$$RM(k-1,m) \subseteq C \subseteq RM(k,m).$$

We denote the lower complexity order for the full list decoding of C by λ. Then

$$n \ln^{k-2} n \leq \lambda \leq n \ln^{k-1} n.$$

Further decoding algorithms for Reed-Muller codes with low complexity were given in [24]. It is a question of implementation which decoding scheme invented for Reed-Muller codes will be used for the HL-code.

7. Performance of the cryptosystem

Regarding the key generation of the classical McEliece cryptosystem, we can see the following performance results in [5]: The key generation of mceliece--8192128 software took billions of cycles, with medians of 4,010,278,828 cycles, 6,008,245,724 cycles (about 2 seconds), and 4,005,886,024 cycles. Each key-generation attempt took about 2 billion cycles.

We have already implemented a key generation software in C++ for our proposed novel cipher. It took 0.5 second running time on Dell Lattitude PC having a single processor with performance of 1.6 GHz. Of course, further investigations are necessary for the correct comparison.

Unfortunately, we have no performance results for coding and decoding this time.

The keyspace is appropriate, because the size of the keyspace is bigger than 2^{128}. So it cannot be attacked by a brute-force algorithm.

Since there exist 2^{462} complement-free sets Y (see Remark 3), there exist 2^{462} different generator matrices G, which all generate a binary self-dual $(4096, 2048, 64)$-code. Thus if someone wants to recreate G and attack the system, up to 2^{462} tries are needed.

The only requirement for the matrix S is to be nonsingular, thus we have $|GL(2048, 2)|$ possibilities for the choice of S, where $GL(n, q)$ denotes the general linear group of dimension n over $\mathbb{F}_q$.

In [19, McEliece] described the following attack on his cryptosystem: We choose k columns of the generator matrix G randomly (i.e., as many as the number of rows in G) and we try to build up the message from these columns. The probability that there is no error in the set of these k columns is

$$\frac{\binom{n-t}{k}}{\binom{n}{k}},$$

where n is the number of rows in G and t is the number of errors. In order to thwart such attack McEliece advised the use of $n = 1024$,$k = 524$, $t = 50$.

Bernstein, Lenge and Peters [4] advise other parameters:

(1) $n = 1362, \quad k = 1269, \quad t = 34 \quad$ or

(2) $n = 2048, \quad k = 1751, \quad t = 27.$

The ratio, in the case which McEliece advised, is $\frac{1}{1.37 \cdot 10^{16}}$ and in the other two cases, it is $\frac{1}{5.46 \cdot 10^{22}}$ and $\frac{1}{1.24 \cdot 10^{23}}$. In our case, the ratio is $\frac{1}{2.421 \cdot 10^{9}}$ which is greater than in the two mentioned cases. In all cases the ratio is greater than the expected ratio of

$$c = \frac{1}{2^{128}} = \frac{1}{3.402 \cdot 10^{38}}.$$

The situation can be radically improved by applying the cryptographic system again, regarding the ciphertext as a message to be encrypted.

Of course, after the first turn the ciphertext will be two times longer than the plaintext. Thus, using the same public key in the second turn, we should encrypt again the prefix of length n of the ciphertext of the first turn and also the suffix of length n of it. Then the ciphertext of the second turn will be the concatenation of the two further encryptions generated after the second turn. (During the decryption procedure, first we decrypt the prefix of length $2n$ of the ciphertext of the second turn and also the suffix of length $2n$ of it. Then we concatenate the generated plaintext of the prefix and the generated plaintext of the suffix and then we apply the decryption procedure again for this concatenation.)

Applying the mentioned attack, we should repeat it three times because we can use three binary random error vectors which are independent of each other. Therefore, the probability that there is no error in the set of 3 portions of the considered k columns is

$$\left(\frac{\binom{n-t}{k}}{\binom{n}{k}} \right)^3.$$

For example, in our case the ratio is

$$r = \left(\frac{1}{2.421 \cdot 10^9}\right)^3 = \frac{1}{1.419 \cdot 10^{729}}, \quad \text{and we have} \quad r < c.$$

We remark that the size of the ciphertext grows in our cryptosystem, as well as in the McEliece cryptosystem, by applying the algorithm more than once. This improves the security of our cryptosystem, therefore it is a useful property. When applying our cryptosystem for key exchange, the larger ciphertext does not pose a problem since symmetric keys are small in general (128-256 bits). The growth ratio of the cipher is in McEliece cryptosystem 1.95, in the other two mentioned systems 1.28 and 1.16 respectively. In our cryptosystem the growth ratio is 2.

8. Conclusions and Further Research

In this paper we proposed a novel McEliece type cipher based on Hannusch-Lakatos error correcting codes. We consider this paper as the first step of our investigations in this subject. Therefore, several questions and problems remain for our future investigations.

A further challenge in research is to give an exact comparison of our discussed cryptosystem with the original McEliece system and its variants. Moreover, we would like to overcome the following shortcomings of this paper:

In this paper we considered only information set decoding attacks. Of course, investigations regarding more state-of-the-art attacks (ISD attacks including quantum attacks, structural attacks, etc.) are necessary. In addition, measurable improvements are discussed (partly) only for the key generation in comparison to Goppa codes based McEliece ciphers. We should give a more exact comparison for key generation, and detailed investigations are necessary for the decoding procedures. (It seems that there are no measurable differences in coding procedures.) Implementation comparisons and deep theoretical complexity analysis would also be necessary.

REFERENCES

[1] BALDI, M.—BODRATO, M.—CHIARALUCE, F.: *A new analysis of the Mceliece cryptosystem based on qc-ldpc codes*. In: *Security and Cryptography for Networks*, Springer-Verlag, Berlin, 2008, pp. 246–262.

[2] BERGER, T. P.—CAYREL, P.-L.—GABORIT, P.—OTMAN, A.: *Reducing key length of the Mceliece cryptosystem*, In: *Progress in Cryptology–AFRICACRYPT*, Springer-Verlag, Berlin 2009, pp.77–97.

[3] BERGER, T. P., LOIDREAU, P.: *How to mask the structure of codes for a cryptographic use. Designs*, Codes and Cryptography, **35** (2005), no. 1, 63–79.

[4] BERNSTEIN, D.,LANGE, T., PETERS, C.: *Attacking and defending the McEliece cryptosystem*. In: Proc. 2nd Internationsl Workshop on Post-Quantum Cryptography, PQ Crypto 2008, Cincinnati, OH, USA, October 17–19; *Lecture Notes in Comput. Sci.*, Vol. 299, Springer-Verlag, 2008, pp. 31-46.

[5] BERSTEIN, D. J.—CHOU, T.—LANGE, T.—VON MAURICH, I.— MISOCZKI, D.—NIEDERHAGEN, D.—PERSICHETTI, E.—PETERS, S.— SCHSWABE, P.– –SENDRIER, N.—SZEFER, J.—WANG, W.: *Classic McEliece: conservative code-based cryptography*, The First NIC PQC Workshop, 2018, April 11–13, 2018; `https://classic.mceliece.org/nist/mceliece-20171129.pdf`

[6] CAYREL, P.-L., GUEYE, C. T., NDIAYE, O., NIEBUHR, R.: *Critical attacks in code-based cryptography*, Int. J. Information and Coding Theory, **3** (2015), no. 2, 158–176.

[7] CHEN L.—LIU Y-K.—JORDAN S.—MOODY, D.—PERALTA, R.—PERLNER, R.– –SMIDT-TONE, D.: *Report on Post-Quantum Cryptography*. NISTIR 8105, U.S. Department of Commerce, 2016.

[8] DIFFIE, W.—HELLMAN, M. E.: *New directions in cryptography*, IEEE Transactions on Information Theory, **IT-22** (1976), 644–654.

[9] DM̈ÖSI, P.—HANNUSCH, C.—HORVÁTH, G.: *Public Key Cryptographic Method and Apparatus for Data Encryption and Decryption Based on Error-Correcting Codes*. Hungarian Intellectual Property Office, Budapest, 2018, Patent Application, P1800038.

[10] DUMER, I.—KABATIANSKY, G.—TAVERNIER, C.: *On complexity of decoding Reed-Muller codes within their code distance*. In: Proc. Eleventh International Workshop on Algebraic and Combinatorial Coding Theory 2008, pp. 82–85

[11] ELGAMAL, T.: *A public key cryptosystem and a signature scheme based on discrete logarithms*. a.) In: *Advances in cryptology*: Proceedings of CRYPTO 84. *Lecture Notes in Comput. Sci.* Vol. 196. Springer-Verlag, Santa Barbara, California, United States, pp. 10–18; b.) IEEE Trans. on Inf. Theory **31** (1985), 469–472.

[12] FABŠIČ, T.—HROMADA, V.—STANKOVSKI, P.—ZAJAC, P.—GUO, Q.—JOHANSSON, T.: *A reaction attack on the QC-LDPC McEliece Cryptosystem*. (T. Lange and T. Takagi, eds.), *Post-Quantum Cryptography*, Proc. 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, *Lecture Notes in Comput. Sci.* Vol. 10346, Springer-Verlag, Berlin, 2017, pp.51–68.

[13] GOPALAN, P.—KLIVANS, A. R.—ZUCKERMAN, D.: *List-decoding Reed-Muller codes over small fields*. In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, ACM, 2008, pp. 265–274.

[14] GUO, Q.—JOHANSSON, T.—STANKOVSKI, P.: *A key recovery attack on MDPC with CCA security using decoding errors*. In: (J. H. Cheon and T. Takagi, eds.) *Advances in Cryptology*. ASIACRYPT 2016, Proc. 22nd Int. Conf. on the Theory and Appplication of Cryptology and Information Security, Hanoi, Vietnam, December 4-8-2016, Part 1, *Lecture Notes in Comput. Sci.* Vol. 10031, Springer-Verlag, Berliln, 2016, pp. 789–815.

[15] HANNUSCH, C.—LAKATOS, P.: : *Construction of self-dual binary* $2^2k, 2^2k-1, 2^k$*-codes*, Algebra and Discrete Math. **21** (2016), no. 1, 59–68.

[16] KOBARA, K.—IMAI, H.: *Semantically secure Mceliece public-key cryptosystems-conversions for Mceliece PKC*. In:*Public Key Cryptography*, Cheju, Island, 2001, (K. Kim, ed.), *Lecture Notes in Comput. Sci.* Vol. 1992, Springer-Verlag, Berlin, 2001, pp. 19–35.

[17] LÖNDAHL, C.—JOHANSSON, T.: *A new version of Mceliece PKC based on convolutional codes*. In: *Information and Communications Security*, International Conference on Information and Communications Security, ICICS 2012; *Lecture Notes in Comput. Sci.* Vol. 7618, Springer-Verlag, Berlin, 2012, pp. 461–470.

[18] JANWA, H.—MORENO, O.: *Mceliece public key cryptosystems using algebraic-geometric codes*. Designs, Codes and Cryptography, **8** (1996), no. 3, 293–307.

[19] MCELIECE, R. J.: *A public-key cryptosystem based on algebraic coding theory*, In: *The Deep Space Network Progress Report*, DSN PR 42-44, January and February 1978, pp. 114-116; `https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF`

[20] MISOCZKI, R.—BARRETO, P.: *Compact Mceliece keys from Goppa codes*. In: *Selected Areas in Cryptography*, Springer-Verlag, Berlin, 2009, pp. 376–392.

[21] MACWILLIAMS, F.J.—SLOANE, N. J. A.: *The Theory of Error-Correcting Codes*. Elsevier, 1977.

[22] MISOCZKI, R.—TILLICH, J.-P.—SENDRIER, N.—BARRETO, P.: *MDPC-McEliece, New McEliece variants from moderate density parity-check codes*). In: *Information Theory Proceedings (ISIT), 2013*, pp. 2069–2073.

[23] NIEDERREITER, H.: *Knapsack-type cryptosystems and algebraic coding theory*, Problems Control Inform. Theory/Problemy Upravlen. Teor. Inform. **15** (1986), no. 2, 159–166.

[24] PATERSON, K. G.—JONES, A. E.: *An efficient decoding algorithms for generalized Reed-Muller codes*, IEEE Transactions on Communications, **48** (2000) no. 8, (2000) 1272–1285.

[25] REED, I. S.: *A Class of Multiple Error Correcting Codes and the Decoding Scheme*, Massachusettts Institute of Technology, Lincoln Laboratory, 1953, Technial Report no. 44.

[26] REPKA, M.: *McEliece PKC Calculator*, Journal Electr. Eng. **65** (2014), no. 6, 342–348.

[27] REPKA, M.—ZAJAC, P.: : *Overview of the McEliece cryptosystem and its security*, Tatra Mt. Math. Publ. **60** (2014), 57–83.

[28] RIVEST, R. L.—SHAMIR, A.—ADLEMAN, L. M.: *Cryptographic Communications System and Method*, Patent US 4405829 A, Massachusetts Institute of Technology, September 20, 1983; `https://patentimages.storage.googleapis.com/49/43/9c/b155bf231090f6/US4405829.pdf`

[29] SIDELNIKOV, V. M.: *A public-key cryptosystem based on binary Reed-Muller codes*, Discrete Math. Appl. **4** (1994), no. 3, 191–208.

[30] WANG, J.: *Quantum resistant random linear code based public key encryption scheme RLCE*, Cryptology ePrint Archive, **298** (2015), 1–12.

Received October 26, 2018

Department of Computer Science
Faculty of Informatics
University of Debrecen
H-4002 Debrecen
Pf.: 400
HUNGARY

E-mail: domosi@unideb.hu
hannusch.carolin@inf.unideb.hu
horvath.geza@inf.unideb.hu