ISSN 0146-4116, Automatic Control and Computer Sciences, 2019, Vol. 53, No. 7, pp. 719–729. © Allerton Press, Inc., 2019.
Russian Text © The Author(s), 2018, published in Modelirovanie i Analiz Informatsionnykh Sistem, 2018, No. 3, pp. 276–290.

On the Support Splitting Algorithm for Induced Codes

Yu. V. Kosolapov[a, *] **and A. N. Shigaev**[a, **]

[a]*Southern Federal University, Rostov-on-Don, 344016 Russia*
**e-mail: itaim@mail.ru*
***e-mail: aleksejshig@gmail.com*

Received February 12, 2018

Abstract—As shown by N. Sendrier in 2000, if a $[n, k, d]$-linear code $C(\subseteq \mathbb{F}_q^n)$ with length n, dimensionality k and code distance d has a trivial group of automorphisms $\mathrm{PAut}(C)$, it allows one to construct a determined support splitting algorithm in order to find a permutation σ for a code D, being permutation-equivalent to the code C, such that $\sigma(C) = D$. This algorithm can be used for attacking the McEliece cryptosystem based on the code$C$. This work aims the construction and analysis of the support splitting algorithm for the code $\mathbb{F}_q^l \otimes C$, induced by the code C, $l \in \mathbb{N}$. Since the group of automorphisms $\mathrm{PAut}(\mathbb{F}_q^l \otimes C)$ is nontrivial even in the case of that trivial for the base code C, it enables one to assume a potentially high resistance of the McEliece cryptosystem on the code $\mathbb{F}_q^l \otimes C$ to the attack based on a carrier split. The support splitting algorithm is being constructed for the code $\mathbb{F}_q^l \otimes C$ and its efficiency is compared with the attack to a McEliece cryptosystem based on the code $\mathbb{F}_q^l \otimes C$.

Keywords: group codes, induced group codes, support splitting algorithm, McEliece cryptosystem
DOI: 10.3103/S0146411619070125

1. INTRODUCTION

In the post-quantum era, the McEliece cryptosystems [1] are considered as possible alternatives to asymmetric cryptosystems whose resistance is currently based on factorization complexity of large integers or discrete logarithmization in a finite group [2]. A prerequisite for constructing the McEliece cryptosystems based on linear codes is the existence of effective (polynomial) decoding algorithms for these codes. Meanwhile, this condition is not sufficient. Although the Reed–Solomon and Reed–Maller codes possess fast decoding algorithms [3], the related McEliece cryptosystems are shown to give way to structural attacks [4, 5]. As found for the McEliece cryptosystems, the more the code is structurally similar to a random code, the more difficult is the analysis of the corresponding McEliece cryptosystem. Among the feasible ways to construct a robust McEeliece cryptosystem, there is the search or the construction of a code with an available effective decoder and a random-like structure.

It is shown in work [6] that for a base code Cdisposing the effective majority decoder, one can also construct a decoder for the induced code $\mathbb{F}_q^l \otimes C$, $l \in \mathbb{N}$. In connection with this, the McEliece cryptosystem was developed on the base of an induced code $\mathbb{F}_q^l \otimes C$ [7]. If a McEliece cryptosystem based on the code C is unstable to attacks on keys, one can carefully select the induced code parameters so that the attack on the key of a related cryptosystem based on the induced code $\mathbb{F}_q^l \otimes C$ will fail.

This work aims at the development of support splitting algorithms for induced codes and the estimation of its efficiency in finding a secret key of the McEeliece cryptosystem based on the induced code $\mathbb{F}_q^l \otimes C$. The monograph has the following structure. The second section provides information on codes, support splitting algorithms and preliminary results on induced codes. The support splitting algorithm for induced codes is considered, as well. The third section gives the example of applying this algorithm in the determination of a secret permutation of a McEliece cryptosystem on the induced code, and its efficiency is compared with that obtained in work [7]. The feasible use of induced codes in the identification algorithm is also within the scope of this study.

2. SUPPORT SPLITTING ALGORITHM FOR INDUCED CODES

2.1. Preliminary Results

Let $\mathbb{F}_q$ be the Galois field with a strength q, where q is the degree of a prime number. For a vector $\mathbf{x}$ from a space $\mathbb{F}_q^n$ with dimensionality n, the weight $\mathrm{wt}(\mathbf{x})$ can be found as the power of a set of nonzero coordinates of the vector $\mathbf{x}$. Consider a $[n,k,d]$-code C with a dimensionality k, a length n and a code distance d in a space $\mathbb{F}_q^n$. Let $G(C)$ be a code generator matrix, $C \subseteq \mathbb{F}_q^n$. Codes C and D with dimensionality k and length n are called permutation-equivalent, if there is a permutation σ from a symmetric group S_n, acting on the elements of a set $I_n = \{1,\dots,n\}$, so that

$$D = \{(c_1,\dots,c_n) | (c_{\sigma^{-1}(1)},\dots,c_{\sigma^{-1}(n)}) \in C\}.$$

Hence, one uses the common designation $D = \sigma(C)$. The next step is to determine the invariant and the signature from [8]. For some subset $J(\subseteq I_n)$, designate a set of vectors, obtained from those of the code C by zeroing the coordinates with numbers from J, by C_J. Let $\mathcal{L}_n$ be a set of all codes with a length n, $\sigma'(C) \neq D$. The mapping $\mathcal{V} : \mathcal{L} \to E$ is called the *invariant* over a set E, if any two permutation-equivalent codes C and D obey the equality: $\mathcal{V}(C) = \mathcal{V}(D)$. A *signature* over a set F is the mapping $\mathcal{S} : \mathcal{L}_n \times I_n \to F$, so that any permutation $\sigma(\in S_n)$ and any code $C \in \mathcal{L}_n$ are referred to the equality: $\mathcal{S}(C,i) = \mathcal{S}(\sigma(C),\sigma(i))$. Below we consider only the signatures based on the invariant that meets the following rule:

$$\mathcal{S}(C,i) = \mathcal{V}(C_i), \tag{1}$$

where $C_i = C_{\{i\}}$. A discriminant of the code C is a signature S, for which i and j from I_n result in $\mathcal{S}(C,i) \neq \mathcal{S}(C,j)$. Then a *full discriminant* for the code C is a signature cal $\mathcal{S}$, so that $\mathcal{S}(C,i) \neq \mathcal{S}(C,j)$ for all different i and j from I_n. The known fact can be summarized by the lemma below.

Lemma 1. Let C be a $[n,k,d]$-code, $\sigma \in S_n$, $D = \sigma(C)$. The equality $D = \gamma(C)$ is valid if and only if $\gamma \in \sigma\mathrm{PAut}(C)$, where $\sigma\mathrm{PAut}(C)$ is a factor-class from the factor-set $S_n/\mathrm{PAut}\,(C)$.

Proof. It is evident, if $\gamma \in \sigma\mathrm{PAut}(C)$, then $D = \gamma(C)$. Let us prove in the opposite direction. Assume the equality $D = \gamma(C)$, where $\gamma \notin \sigma\mathrm{PAut}(C)$. Since $\gamma(C) = \sigma(C)$, then $\gamma^{-1}\sigma \in \mathrm{PAut}(C)$. Hence $\gamma^{-1} = \phi\sigma^{-1}$, $\phi \in \mathrm{PAut}(C)$, and consequently $\gamma \in \sigma\mathrm{PAut}(C)$.

Consider the SSA algorithm that finds a permutation σ' for two permutation-equivalent codes C and $D = \sigma(C)$ using $\mathcal{S}$, so that $D = \sigma'(C)$. Notice that $\sigma \neq \sigma'$ in the general case, but $\sigma'^{-1} \in \mathrm{PAut}(C)$ by **Lemma 1.** The permutation σ', returned by the SSA algorithm, will be called suitable. If $\mathcal{S}$ is the total discriminant, then $\sigma = \sigma'$, and the permutation σ will be found at the first iteration of the cycle from this algorithm. As follows from statement 8 of the monograph [8], the complete discriminant fulfills for the code C, when the group of automorphisms $\mathrm{PAut}(C)$ of the code C is trivial. Mention that codes with a trivial group of automorphisms exist [9].

According to work [8], even if the complete discriminant of the code C exists, its calculation may be a computationally difficult task. In this respect, there was proposed the approach [8] that ensures the construction of computationally simple complete discriminants based on incomplete discriminants. Since it considers only signatures based on invariants (see Eq. (1)), the latter have to be computationally simple.

```
The initial parameters: $C \in \mathcal{L}_n$, $D = \sigma(C)$, $\mathcal{S}$
Result: $\sigma' : \mathcal{D} = \sigma'(C)$
1. Calculate $\mathcal{D} = (\mathcal{S}(D, i))_{i=1}^{n}$
2. Calculate$\mathcal{C} = (\mathcal{S}(, i))_{i=1}^{n}$
3. $\Sigma = \varnothing$, exit $=$ false;
Run until exit! $=$ true
    Select $\sigma'$ from $S_n \setminus \Sigma$
    if $\sigma'(C) \neq D$ than
        $\Sigma = \Sigma \cup \{\sigma'\}$
     end
    else
        exit $=$ true
    end
end
return $\Omega$
```

Algorithm 1: SSA

An example of a computationally simple invariant for low-dimensionality codes is the mapping $\mathcal{V}^W : \mathcal{L} \to \mathbb{Z}[X]$ that assigns the code C to its weight numerator $\mathcal{W}(C) = \sum_{i=0}^{n} W_i X^i$, where W_i is the number of vectors with weights i in the code C, $\mathbb{Z}[X]$ is the set of polynomials from one variable with coefficients from $\mathbb{Z}$. Using this invariant, one can construct a signature $\mathcal{S}^W : \mathcal{L}_n \times I_n \to \mathbb{Z}[X]$, determined from the rule $\mathcal{S}^W(C, i) = \mathcal{V}^W(C_i) = \mathcal{W}(C_i)$. Mention that the computation complexity of the invariant $\mathcal{V}^W(C_i)$ increases in a nonpolynomial order with dimensionality of the code C_i. Hence a discriminant in work [8] is based on the computation of weight numerators of the code hull. A hull of the code C [8] implies the intersection of the code C with its dual code $C^{\perp}$:

$$\mathcal{H}(C) = C \cap C^{\perp}. \tag{2}$$

A choice of this characteristic is due to the fact that the hull dimensionality is typically much less than the dimensionality of the code C, which enables one to efficiently calculate the numerators, as well as to plot computationally simple discriminant even in case of the large dimensionality of the code C.

2.2. Induced Codes and Their Properties

Let C^i be a $[n_i, k_i, d_i]$-code with a generator matrix $G(C^i)$ and a check matrix $H(C^i)$, $i = 1, 2$. The Cartesian product $C^1 \times C^2$ of codes C^1 and C^2 is assumed to be a set in the following form:

$$C^1 \times C^2 = \{(\mathbf{a} \| \mathbf{b}) : \mathbf{a} \in C^1, \mathbf{b} \in C^2\},$$

where $\mathbf{a} \| \mathbf{b}$ is the concatenation of vectors $\mathbf{a}$ and $\mathbf{b}$. It is easily to see that the generator and check matrices of the code $C^1 \times C^2$ can be presented as

$$G(C^1 \times C^2) = \begin{pmatrix} G(C^1) & O_{k_1 \times n_2} \\ O_{k_2 \times n_1} & G(C^2) \end{pmatrix},$$

$$H(C^1 \times C^2) = \begin{pmatrix} H(C^1) & O_{n_1 - k_1 \times n_2} \\ O_{n_2 - k_2 \times n_1} & H(C^2) \end{pmatrix},$$

where $O_{a \times b}$ is a zero $(a \times b)$-matrix. As follows from definition (2):

$$\mathcal{H}(C^1 \times C^2) = \mathcal{H}(C^1) \times \mathcal{H}(C^2). \tag{3}$$

Lemma 2. Let C^i be a $[n_i, k_i]$-code, $\mathcal{W}(C^i) = \sum_{j=0}^{n_i} W_j^{(i)} X^{(i)}$ is a numerator of the code C^i, $i = 1, 2$. Then the numerator of the code $C^1 \times C^2$ takes the form

$$\mathcal{W}(C^1 \times C^2) = \mathcal{W}(C^1) \cdot \mathcal{W}(C^2).$$

Proof. Each code vector $\mathbf{c}$ from $C^1 \times C^2$ can be presented by a concatenation $(\mathbf{a} \| \mathbf{b})$ of vectors $\mathbf{a}$ and $\mathbf{b}$ from codes C^1 and C^2, respectively. Find the number of weight vectors $j (0 \le j \le n_1 + n_2)$ in the code $C^1 \times C^2$. For this, consider all kinds of ordered pairs (j_1, j_2) of nonnegative integers, so that $j_1 + j_2 = j$. Each pair (j_1, j_2) in the code $C^1 \times C^2$ has a set from $W_{j_1}^{(1)} \cdot W_{j_2}^{(2)}$ of weight vectors j. These sets do not intersect for various pairs. Hence, the code $C^1 \times C^2$ contains

$$\sum_{(j_1, j_2): j_1 + j_2 = j} W_{j_1}^{(1)} \cdot W_{j_2}^{(2)}$$

weight vectors j. Then

$$\mathcal{W}(C^1 \times C^2) = \sum_{j=0}^{n_1+n_2} \left(\sum_{(j_1, j_2): j_1 + j_2 = j} W_{j_1}^{(1)} \cdot W_{j_2}^{(2)} \right) X^j = \mathcal{W}(C^1) \cdot \mathcal{W}(C^2).$$

A tensor product $A \otimes B$ of matrices $A = (a_{i,j})_{i=1,m; j=1,l}$ and B is implied to be a matrix

$$A \otimes B = \begin{pmatrix} a_{1,1}B & \dots & a_{1,l}B \\ \dots & \dots & \dots \\ a_{m,1}B & \dots & a_{m,l}B \end{pmatrix}.$$

Let C be a $[n, k, d]$-code with a generator matrix $G(C)$, E_l is the unit matrix of the order l. A subspace, generated by the lines of a matrix $E_l \otimes G(C)$, will be designated by $\mathbb{F}_q^l \otimes C$ and called the induced code (or the code induced by the code C) [7]. A generator matrix of this code $G(\mathbb{F}_q^l \otimes C) = E_l \otimes G(C)$ has a block structure

$$G(\mathbb{F}_q^l \otimes C) = \underbrace{\begin{pmatrix} G(C) & O_{k\times n} & \dots & O_{k\times n} \\ O_{k\times n} & G(C) & \dots & O_{k\times n} \\ \dots & \dots & \dots & \dots \\ O_{k\times n} & O_{k\times n} & \dots & G(C) \end{pmatrix}}_{l \text{ bloks}}, \tag{4}$$

where each block line has $l - 1$ nonzero matrices $O_{k\times n}$ and one matrix $G(C)$. Since

$$\mathbb{F}_q^l \otimes C = \underbrace{C \times \dots \times C}_{t \text{ times}},$$

then **Lemma 2** yields

Corollary 1. Let C be a $[n, k]$-code with a generator matrix $G(C)$ and $\mathcal{W}(C)$ be a numerator of the code C. Then

(1) $\mathcal{W}(F_q^l \otimes C) = \underbrace{\mathcal{W}(C) \cdot \dots \cdot \mathcal{W}(C)}_{l \text{ times}}$;

(2) $\forall i \in \{1, \dots, ln\} \exists j \in \{1, \dots, n\}$: $\mathcal{W}((\mathbb{F}_q^l \otimes C)_i) = \mathcal{W}(C_j) \cdot \underbrace{\mathcal{W}(C) \cdot \dots \cdot \mathcal{W}(C)}_{l-1 \text{ times}}$.

It also appears that a check matrix of the code $\mathbb{F}_q^l \otimes C$ can be written as $H(\mathbb{F}_q^l \otimes C) = E_l \otimes H(C)$, where $H(C)$ is the check matrix of the code C. As follows from Eq. (3), the hull of the code $\mathbb{F}_q^l \otimes C$ takes the form

$$\mathcal{H}(\mathbb{F}_q^l \otimes C) = \mathbb{F}_q^l \otimes \mathcal{H}(C). \tag{5}$$

Consider the induced code $\mathbb{F}_q^l \otimes C$, $l \in \mathbb{N}$. A group of automorphisms $\mathrm{PAut}(\mathbb{F}_q^l \otimes C)$ of this code is nontrivial. Indeed, the generator matrix of the code $\mathbb{F}_q^l \otimes C$ is presented by a block diagonal structure (4), and any permutation of blocks of this matrix results in a generator matrix of the same code. The permutation of block columns of this matrix is equivalent to the permutation of block lines. There are $l!$ such block column permutations in total. Hence the group of automorphisms of the code $\mathbb{F}_q^l \otimes C$ has a power of at least $l!$. It yields the following lemma.

Lemma 3. A group of automorphisms $\mathrm{PAut}(\mathbb{F}_q^l \otimes C)$ of the code $\mathbb{F}_q^l \otimes C$ contains a subgroup $\mathcal{G}(\mathbb{F}_q^l \otimes C)$, being isomorphic to a group S_l.

Notice that each element of the group $\mathcal{G}(\mathbb{F}_q^l \otimes C)$ has the form

$$\begin{pmatrix} 1 & \dots & n & \dots & (l-1)n+1 & \dots & ln \\ (\sigma(1)-1)n+1 & \dots & \sigma(1)n & \dots & (\sigma(l)-1)n+1 & \dots & \sigma(l)n \end{pmatrix}, \ \sigma \in S_l. \tag{6}$$

Let $I_{i,n} = \{i, i+n, \dots, i+(l-1)\cdot n\}$ be, where $i \in \{1, \dots, n\}$, $S(I_{i,n})$ is a subgroup of the group S_{ln}, where the permutations involve only the elements of a set $I_{i,\,n}$, and the elements of a set $\{1,\dots,\mathrm{ln}\} \setminus I_{i,n}$ are fixed. Consider a group $Q = S(I_{1,n}) \times S(I_{2,n}) \times \ldots \times S(I_{n,n}) \subset S_{ln}$, $|Q| = (l!)^n$. It is easy to see that

$$\mathcal{G}(\mathbb{F}_q^l \otimes C) \subseteq \mathrm{PAut}(\mathbb{F}_q^l \otimes C) \cap Q. \tag{7}$$

We remind that an orbit of an element $i \in \{1, \dots, ln\}$ under the action of a subgroup $\mathcal{G} \subseteq S_{ln}$ is a set $\{g(i) : g \in \mathcal{G}\}$. Then $I_{i,n}$, $i = 1, \dots, n$, are the orbits that form by a subgroup $\mathcal{G}(\mathbb{F}_q^l \otimes C)$ on the elements of a set $\{1, \dots, ln\}$. Expression (7) yields the following auxiliary lemma.

Lemma 4. A length of each orbit, forming under the action of a group $\mathrm{PAut}(\mathbb{F}_q^l \otimes C)$to the elements of a set $\{1, \dots, ln\}$ is a multiple of l.

2.3. Support Splitting Algorithm

As aforementioned, two permutation-equivalent codes with a complete discriminant $\mathcal{S}$ for finding the most suitable permutation require not more than one iteration of the internal cycle of a SSA algorithm. It follows from **Lemma 3**, there is no complete discriminant for the code $\mathbb{F}_q^l \otimes C$, because the group of automorphisms of this code is nontrivial. Consider an algorithm plotting problem for the codes $\mathbb{F}_q^l \otimes C$ and $D = \sigma(\mathbb{F}_q^l \otimes C)$, which allows a suitable permutation σ' to be determined so that $\sigma'(\mathbb{F}_q^l \otimes C) = D$.

Lemma 5. Let C be a $[n, k, d]$-code. Then for $i = 1, \dots, n$ and any signature $\mathcal{S}$, found from the rule (1), the following equality is valid:

$$\mathcal{S}(\mathbb{F}_q^2 \otimes C, i) = \mathcal{S}(\mathbb{F}_q^2 \otimes C, i+n). \tag{8}$$

Proof. According to definition (1), $\mathcal{S}(\mathbb{F}_q^2 \otimes C, i) = \mathcal{V}((\mathbb{F}_q^2 \otimes C)_i)$. Let $\mathcal{V}((\mathbb{F}_q^2 \otimes C)_i) \neq \mathcal{V}((F_q^2 \otimes C)_{i+n})$ be. For any permutation π from the invariant definition: $\mathcal{V}((\mathbb{F}_q^2 \otimes C)_i) = \mathcal{V}(\pi((\mathbb{F}_q^2 \otimes C)_i))$. Consider an arbitrary nontrivial permutation $\pi \in \mathcal{G}(\mathbb{F}_q^2 \otimes C)$. Hence,

$$\mathcal{V}((\mathbb{F}_q^2 \otimes C)_i) = \mathcal{V}(\pi((\mathbb{F}_q^2 \otimes C)_i)) = \mathcal{V}(\pi(\mathbb{F}_q^2 \otimes C)_{\pi(i)}) = \mathcal{V}((\mathbb{F}_q^2 \otimes C)_{\pi(i)}).$$

Since π is the nontrivial permutation, then the elements (6) of a group $\mathcal{G}(\mathbb{F}_q^2 \otimes C)$ for $l = 2$ result in: $\pi(i) = i + n$. This is a contradistinction.

Taking the representation (6) into account, **Lemma 5** yields a corollary below.

Corollary 2. For the code $\mathbb{F}_q^l \otimes C$ and $i \in \{1, \dots, n\}$, the following equality is valid: $\mathcal{S}(\mathbb{F}_q^l \otimes C, i) = \mathcal{S}(\mathbb{F}_q^l \otimes C, i + n \cdot k)$, for all $k = \{0, \dots, l-1\}$.

Thus, any signature for the code $\mathbb{F}_q^l \otimes C$, determined using the rule (1), has not more than n various values. The more values the signature has for the code, the fewer cycles of the SSA algorithm are required for finding a suitable permutation.

Lemma 6. If $\mathcal{G}(\mathbb{F}_q^l \otimes C) \subset \mathrm{PAut}(\mathbb{F}_q^l \otimes C)$, then any signature for the code C, found from the rule (1), has less than n values.

Proof. It follows from **Corollary 2** that any signature for the code $\mathbb{F}_q^l \otimes C$, established from the rule (1), has not more than n various values. Hence, in according to statement 8 [8], the group $\mathrm{PAut}(\mathbb{F}_q^l \otimes C)$ in a set $\{1,\dots,ln\}$ leads to the formation of not more than n different orbits. Let $\sigma \in \mathrm{PAut}(\mathbb{F}_q^l \otimes C) \setminus \mathcal{G}(\mathbb{F}_q^2 \otimes C)$ be, then there is an element $i \in \{1,\dots,n\}$, so that $\sigma(i) \neq i + n \cdot k$ for some $k \in \{0,\dots,l-1\}$. Therefore, it obtains from **Lemma 4** that at least one orbit has a length not less than $2l$. Hence, the group $\mathrm{PAut}(\mathbb{F}_q^l \otimes C)$ favors the formation of not more than $n-1$ orbits. Based on statement 8 [8], a signature has not more than $n-1$ various values.

Corollary 3. Let a signature $\mathcal{S}$ be defined in accordance with the rule (1). If $\mathcal{S}$ for the code $\mathbb{F}_q^l \otimes C$ has n various values, then $\mathrm{PAut}(\mathbb{F}_q^l \otimes C) = \mathcal{G}(\mathbb{F}_q^l \otimes C)$.

Example 1. Consider a code $C^1 \times C^2$ and find a weight numerator of the code $(C^1 \times C^2)_i$, then $i \in \{1,\dots,n_1+n_2\}$. If $i \le n_1$, then $\mathcal{W}((C^1 \times C^2)_i) = \mathcal{W}(C_i^1) \cdot \mathcal{W}(C^2)$; if $n_1 < i \le n_1 + n_2$, then $\mathcal{W}((C^1 \times C^2)_i) = \mathcal{W}(C^1) \cdot \mathcal{W}(C_{i-n_1}^2)$. Hence

$$\mathcal{S}^W(C^1 \times C^2, i) = \mathcal{W}((C^1 \times C^2)_i) = \mathcal{W}(C^a_{i-(a-1)\cdot n_1}) \cdot \mathcal{W}(C^b), \tag{9}$$

where

$$a = \begin{cases} 1, & \text{at } 1 \le i \le n_1 \\ 2, & \text{at } n_1 + 1 \le i \le n_1 + n_2, \end{cases}$$

$b = \{1,2\} \setminus \{a\}$. For $C^1 = C^2 = C$ one obtains: $C^1 \times C^2 = \mathbb{F}_q^2 \otimes C$, thus

$$\mathcal{S}^W(\mathbb{F}_q^2 \otimes C, i) = \mathcal{W}(C) \cdot \mathcal{W}(C_j), \tag{10}$$

where

$$j = \begin{cases} i, & \text{at } 1 \le i \le n \\ i - n, & \text{at } n < i \le 2n. \end{cases}$$

If C is a $[n,k]$-code, so that $\mathcal{S}^W$ is its complete discriminant, then, according to Corollary 1 and generalizations of formula (10) to the case $l \ge 2$, a signature $\mathcal{S}^W$ for the code $\mathbb{F}_q^l \otimes C$ has n various values. Then it follows from **Corollary 3** that a group of automorphisms of the code $\mathbb{F}_q^l \otimes C$ is described in a simple manner.

Lemma 7. Let $\mathcal{S}$ be a signature found from the rule (1). Then for any $\pi \in Q$ there are the following equalities:

(1) $\mathcal{S}(\mathbb{F}_q^l \otimes C, i) = \mathcal{S}(\mathbb{F}_q^l \otimes C, \pi(i))$, $i = 1,\dots,ln$;

(2) $(\mathcal{S}(\mathbb{F}_q^l \otimes C, i))_{i=1}^{ln} = (\mathcal{S}(\mathbb{F}_q^l \otimes C, \pi(i)))_{i=1}^{ln}$;

(3) $(\mathcal{S}(\mathbb{F}_q^l \otimes C, i))_{i=1}^{ln} = \pi((\mathcal{S}(\mathbb{F}_q^l \otimes C, i))_{i=1}^{ln})$.

Proof. The proof of the equality (1) follows from **Corollary 2**; the equality (2) follows from the equality (1). Prove Statement (3). It appears from Statement (2) that

$$(\mathcal{S}(\mathbb{F}_q^l \otimes C, i))_{i=1}^{ln} = (\mathcal{S}(\mathbb{F}_q^l \otimes C, \pi(i)))_{i=1}^{ln} = (\mathcal{S}(\mathbb{F}_q^l \otimes C, j))_{\pi^{-1}(j)=1}^{ln} = \pi((\mathcal{S}(\mathbb{F}_q^l \otimes C, j))_{j=1}^{ln}).$$

Since π is the arbitrary permutation from the group Q, then the statement is proved.

A symbol σQ means a factor-class $\{\sigma\pi : \pi \in Q\}$ of a factor-set $-\; S_{nl}/Q$.

Lemma 8. If a signature $\mathcal{S}$ for the code $\mathbb{F}_q^l \otimes C$ is determined using the rule (1) and has n various values, then $D = \sigma(\mathbb{F}_q^l \otimes C)$ and

$$(\mathcal{S}(D,i))_{i=1}^{ln} = \gamma((\mathcal{S}(\mathbb{F}_q^l \otimes C,i))_{i=1}^{ln}), \tag{11}$$

hence $\sigma \in \gamma Q$.

Proof. It follows from Statement (3) of **Lemma 7** and condition (11) that for any$\pi \in Q$ the following equalities are valid:

$$(\mathcal{S}(D,i))_{i=1}^{ln} = \gamma((\mathcal{S}(\mathbb{F}_q^l \otimes C,i))_{i=1}^{ln}) = \gamma\pi((\mathcal{S}(\mathbb{F}_q^l \otimes C,i))_{i=1}^{ln}).$$

Since in accordance with the condition, a signature has a maximum amount of various values n, then $\sigma \in \gamma Q$ with respect to the construction of a group Q (Q is a maximum subgroup that does not change the order of elements in a set $(\mathcal{S}(\mathbb{F}_q^l \otimes C,i))_{i=1}^{ln}$).

Let $Q/\mathcal{G}(\mathbb{F}_q^l \otimes C) = \{G_i\}_{i=0}^{x}$ be a factor-set of a group Q with respect to a group $\mathcal{G}(\mathbb{F}_q^l \otimes C)$, $x+1 = |Q|/|\mathcal{G}(\mathbb{F}_q^l \otimes C)| = (l!)^n/l! = (l!)^{n-1}$. Let also $\Omega = \{\omega_0,\dots,\omega_x\}$ be a transversal of a factor-set $Q/\mathcal{G}(\mathbb{F}_q^l \otimes C)$, or a set of representatives of the adjacency classes $G_i = \omega_i\mathcal{G}(\mathbb{F}_q^l \otimes C)$, $i = 0,\dots,x$. Among the possible plotting schemes of a set Q, there is a MakeRepresentatives algorithm.

```
Initial parameters: $Q, \mathcal{G}(\mathbb{F}_q^l \otimes C)$
Result: $\Omega$ — is a set of representative classes of a factor-set
            $Q/\mathcal{G}(\mathbb{F}_q^l \otimes C)$
1. $\Omega = \varnothing$
2. Run until $|\Omega| < (l!)^{n-1}$
    Randomly generate permutation $\pi' \in Q$
    if $\pi' \notin \mathcal{G}(\mathbb{F}_q^l \otimes C)$ and $\pi'^{-1}\sigma \notin \mathcal{G}(\mathbb{F}_q^l \otimes C)\ \forall\sigma \in \Omega$ then
        $\Omega = \Omega \cup \{\pi'\}$
    end
end
return $\Omega$
```

Algorithm 2: MakeRepresentatives

Theorem 1. Let C be a $[n,k]$-code, $D = \sigma(\mathbb{F}_q^l \otimes C)$, $\mathcal{S}$ is a signature defined from the rule (1) and having n various values for the code $\mathbb{F}_q^l \otimes C$, Ω is a transversal of a factor-set $Q/\mathcal{G}(\mathbb{F}_q^l \otimes C)$. Then there is an algorithm with a computation complexity $\mathbb{O}(|\Omega|)$, which finds a suitable permutation σ', so that $D = \sigma'(\mathbb{F}_q^l \otimes C)$.

Proof. Let $(\mathcal{S}(D,i))_{i=1}^{ln} = \gamma((\mathcal{S}(\mathbb{F}_q^l \otimes C,i))_{i=1}^{ln})$. This permutation γ can be found by a simple calculation of signatures of codes D and C. Then **Lemma 8** gives $\sigma \in \gamma Q$. As seen from **Lemma 1**, a suitable permutation that converts the code $\mathbb{F}_q^l \otimes C$ into a code D, is a permutation $\sigma\pi$, where $\pi \in \mathrm{PAut}(\mathbb{F}_q^l \otimes C)$. Since the signature has n various values, then it obtains from **Corollary 3** that $\mathrm{PAut}(\mathbb{F}_q^l \otimes C) = \mathcal{G}(\mathbb{F}_q^l \otimes C)$. Hence a suitable permutation can be established by sorting out the elements in a transversalΩ. Thus, a suitable permutation can be found via the SSAForTensor algorithm, whose complexity is$\mathbb{O}(|\Omega|)$, due to sorting out the elements with respect to a transversal Ω.

```
**Initial parameters:** $\mathbb{F}_q^l \otimes C \in \mathcal{L}_{ln}$, $D = \sigma(\mathbb{F}_q^l \otimes C)$, $\mathcal{S}$, $\Omega$
**Result:** $\sigma' : \sigma'(\mathbb{F}_q^l \otimes C) = D$
1. Calculate $\mathfrak{D} = (\mathcal{S}(D, i))_{i=1}^{\mathrm{ln}}$.
2. Calculate $\mathcal{C} = (\mathcal{S}(\mathbb{F}_q^l \otimes C, i))_{i=1}^{\mathrm{ln}}$
3. Find a suitable permutation $\gamma$, so that $\gamma(\mathcal{C}) = \mathfrak{D}$.
4. **for each** $\omega \in \Omega$ **do**
    **if** $(\gamma\omega)^{-1}(D) = \mathbb{F}_q^l \otimes C$ **then**
        $\sigma' = \gamma\omega$
        exit
    **end**
**end**
**return** $\sigma'$
```

Algorithm 3: SSAForTensor

Mention that **Theorem 1** in the estimation of complexity of the SSAForTensor algorithm takes only the power of the transversal into account, but neglects the computation complexity of signatures (steps 1 and 2), as well as the complexity of checking the coincidence of two codes (steps 4) and the complexity of constructing a transversal Ω used as an input parameter. The coincidence of two codes can be verified by multiplying the generator matrix $(\gamma\omega)^{-1}(D)$ by a check matrix $H(\mathbb{F}_q^l \otimes C)$ of the code $\mathbb{F}_q^l \otimes C$. Thus, the complexity of this test depends polynomially on ln, i.e., the verification can be implemented by the effective way. On the other hand, plotting the effectively computable signatures is an individual task [8]. In particular [8], the effective signatures can be constructed using a numerator of a code hull, which is likely to have a small dimension. As follows from Eq. (5), the hull dimensionality for the induced code increases by l times in comparison with a hull of the base code. In turn, this may substantially slow down the calculation of numerators of its projection in the general case and complicate the computation of signatures, because it requires that the vectors of the hull projection are sorted out for all coordinates. Plotting a transversal Ω is also a complex problem at high enough values ln. The above proposed MakeRepresentatives algorithm has a ln-nonpolynomial complexity, although it can be done in advance.

Meanwhile, while the code $\mathbb{F}_q^l \otimes C$ possess the effectively computational signature, determined from the rule (1) and offering n various values along with a transversal Ω, a SSAForTensor algorithm is assumed to be more powerful than SSA at establishing a suitable permutation. This is due to the fact that SSA searches a suitable permutation over the whole adjacency class σQ and SSAForTensor algorithm makes a search over a set $\sigma\Omega$ only, whose power is lower by $l!$ times than $|\sigma Q|$ because $|Q|/|\Omega| = |\mathcal{G}(\mathbb{F}_q^l \otimes C)| = l!$.

3. APPLICATION OF INDUCED CODES IN CRYPTOGRAPHY

3.1. A McEliece Cryptosystem Based on Induced Codes

Consider a McEliece cryptosystem based on a $[ln, lk, d]$-code $\mathbb{F}_q^l \otimes C$, where C is a $[n, k, d]$-code with a generator matrix $G(C)$. In this cryptosystem, an open key $\mathbf{k}_{\mathrm{pub}}$ is a pair $(\widetilde{G}, t = \lfloor r(d-1)/2 \rfloor)$, and a secret key $\mathbf{k}_{\mathrm{sec}}$ is a matrix pair (S, P), where S is a random nondegenerate $(lk \times lk)$-matrxi, P is a random permutation $(ln \times ln)$-matrix, where $\widetilde{G} = S \cdot (E_l \otimes G(C)) \cdot P$ with E_l as a unit matrix of dimensions $l \times l$. The coding rule of an arbitrary message $\mathbf{s}(\in \mathbb{F}_q^{lk})$ has the form

$$\mathbf{z} = \mathbf{s}\widetilde{G} + \mathbf{e}, \tag{12}$$

where $\mathbf{e} \in \mathbb{F}_q^{ln}$ and $\mathrm{wt}(\mathbf{e}) \le t$.

The decoding uses a rule $\mathbf{s} = \mathrm{Dec}_C(\mathbf{z}P^{-1})S^{-1}$, where $Dec_{\mathbb{F}_q^l \otimes C} : \mathbb{F}_q^{ln} \to \mathbb{F}_q^{lk}$ is the decoder of the code $\mathbb{F}_q^l \otimes C$, which guarantees the correction of t and less errors and recovers a vector $\mathbf{s}$. A McEliece cryptosystem based on the code $\mathbb{F}_q^l \otimes C$ will be designated by $\mathrm{McE}(\mathbb{F}_q^l \otimes C)$.

Since the code with a generator matrix $\widetilde{G}$ and the code $\mathbb{F}_q^l \otimes C$ are permutation-equivalent, the $\mathrm{McE}(\mathbb{F}_q^l \otimes C)$ cryptosystem can be hacked by finding a matrix pair (S',P'), so that $S'(E_l \otimes G(C))P' = \widetilde{G}$ [4], and the permutation referred to a permutation matrix $P^{'-1}P$ belongs to $\mathrm{PAut}(\mathbb{F}_q^l \otimes C)$.

As shown in work [7], if the $\mathrm{McE}(C)$ cryptosystem on the base code C is unstable to attacks, there is an algorithm for establishing a suitable permutation matrix for the $\mathrm{McE}(\mathbb{F}_q^l \otimes C)$ cryptosystem, whose complexity is estimated by $\mathbb{O}\left(\frac{(nl)!}{(n!)^l l!}\right)$. Using the Stirling formula, it obtains that

$$\frac{(nl)!}{(n!)^l l!} \approx \frac{\sqrt{2\pi nl}\left(\frac{nl}{e}\right)^{nl}}{\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right)^l \sqrt{2\pi l}\left(\frac{l}{e}\right)^l} = \sqrt{\frac{n}{(2\pi n)^l}} \cdot e^l \cdot l^{l\cdot(n-1)}. \tag{13}$$

Furthermore, a suitable permutation in case of a discriminant existing for the code $\mathbb{F}_q^l \otimes C$ can be found using the SSA. Consider the most favorable condition in the viewpoint of the attacker, when the effectively calculated signature $\mathcal{S}$ with n various values is known for the code $\mathbb{F}_q^l \otimes C$ and a transversal Ω is constructed for a factor-set $Q/\mathcal{G}(\mathbb{F}_q^l \otimes C)$. Thus, the conditions of **Theorem 1** are fulfilled, and SSAForTensor can be substituted for SSA. It follows from **Theorem 1** that the persistence of a cryptosystem $\mathrm{McE}(\mathbb{F}_q^l \otimes C)$ is evaluated by $\mathbb{O}(|\Omega|) = \mathbb{O}((l!)^{n-1})$. Using the Stirling formula gives

$$(l)!^{n-1} \approx \left(\sqrt{2\pi l}\left(\frac{l}{e}\right)^l\right)^{n-1} = \left(\frac{\sqrt{2\pi l}}{e^l}\right)^{n-1} \cdot l^{l\cdot(n-1)}. \tag{14}$$

Notice that expressions (13) and (14) are the estimated powers of sets of keys, where suitable permutations are being searched by sorting out via the algorithm [7] and SSAForTensor. According to monograph [10], sorting out with respect to a key set with a power of 2^{128} and higher is considered computationally impracticable. In order to compare estimations (13) and (14), take as an example the construction of the induced code $\mathbb{F}_q^l \otimes C$ using a double Ride—Maller $[n,k,d]$-code C, where $n \in \{8,16,32,64,128,256\}$ and $q = 2$.

Tables 1 and 2 show the values $\log_2 K$ calculated for a hacked cryptosystem $\mathrm{McE}(\mathbb{F}_2^l \otimes C)$, $l \in \{2,3,4,5,6,7,8,9\}$, where the parameter K in Table 2 is evaluated from expression (13), and that in Table 1 is obtained from formula (14). The cells highlighted in both tables correspond to the parameters of the induced code $\mathbb{F}_2^l \otimes C$, for which the sort complexity is not less than 2^{128}. A comparative analysis of the corresponding values in tables reveals that hacking based on a SSAForTensor splitting algorithm is much more efficient than that described in work [7]. However, this hacking in selecting parameters n and l can also be impracticable.

As shown in monograph [11], the use of induced codes in McEliece cryptosystems causes the weakening of the resilience of a system to attacks on cipher based on the dataset decoding method. The acceptable resilience to these attacks is achieved at large code lengths, which is due to the fact that dimensionality and length of induced codes increase by l times at a fixed code distance. Meanwhile, a $\mathrm{McE}(\mathbb{F}_2^l \otimes C)$ cryptosystem can be applied when coding involves the error vectors with weights beyond the capacity of a decoder $\mathrm{Dec}_{\mathbb{F}_2^l \otimes C}$. So, a shared secret key generation protocol was obtained based on the above cryptosystem [7]. The next subsection is dedicated to another application of induced codes, i.e., their use in cryptographic identification protocols.

Table 1. Values of $\log_2\left(\frac{(nl)!}{(n!)^l l!}\right)$

l/n	8	16	32	64	128	256
2	12.73	28.23	59.73	123.23	**series 250.73**	**series 506.23**
3	30.63	67.67	**series 142.75**	**series 293.90**	**series 597.22**	**series >1024**
4	51.96	114.46	**series 240.96**	**series 495.46**	**series 1006**	**series >1024**
5	75.85	**series 166.72**	**series 350.48**	**series 719.99**	**series >1024**	**series >1024**
6	101.77	**series 223.34**	**series 469**	**series 962.81**	**series >1024**	**series >1024**
7	**series 129.37**	**series 283.59**	**series 595.01**	**series >1024**	**series >1024**	**series >1024**
8	**series 158.43**	**series 346.93**	**series 727.43**	**series >1024**	**series >1024**	**series >1024**
9	**series 188.75**	**series 412.99**	**series 865.46**	**series >1024**	**series >1024**	**series >1024**

Table 2. Values of $\log_2(((l)!)^{n-1})$

l/n	8	16	32	64	128	256
2	7	15	31	63	127	**series 255**
3	18.09	38.77	80.13	**series 162.85**	**series 328.29**	**series 659.16**
4	32.09	68.77	**series 142.13**	**series 288.85**	**series 582.29**	**series >1024**
5	48.34	103.60	**series 214.11**	**series 435.13**	**series 877.17**	**series >1024**
6	66.44	**series 142.37**	**series 294.24**	**series 597.98**	**series >1024**	**series >1024**
7	86.09	**series 184.48**	**series 381.27**	**series 774.85**	**series >1024**	**series >1024**
8	107.09	**series 229.48**	**series 474.27**	**series 963.84**	**series >1024**	**series >1024**
9	**series 129.28**	**series 277.03**	**series 572.54**	**series >1024**	**series >1024**	**series >1024**

3.2. Identification Protocol Based on Induced Codes

An identification protocol based on the complexity in finding the permutation for two permutation-equivalent codes over a binary field was constructed by Girault [12]. Consider this protocol for a case $\mathbb{F}_q$. Let H be a $(N-K\times N)$-matrix over a field $\mathbb{F}_q$, shared by all protocol users. Each user $\mathcal{P}$ randomly chooses a vector $\mathbf{e}$ with a small weight w and calculates $H\mathbf{e}^\top = \mathbf{s}^\top$. The vector $\mathbf{s}$ is a public identifier of a user $\mathcal{P}$. If the relying party $\mathcal{V}$ intends to authenticate a user $\mathcal{P}$, i.e., to check that the authenticated user knows a vector $\mathbf{e}$, a 3-step protocol is being implemented.

Step 1: $\mathcal{P}$ randomly and equally likely choses a permutation $(N\times N)$-matrix P and an undegenerated $(N-K\times N-K)$-matrix S, calculates $\widetilde{H} = SHP$ and $\mathbf{s}' = S\mathbf{s}$ and sends a matrix eH and a vector $\mathbf{s}'$to $\mathcal{V}$.

Step 2: $\mathcal{V}$ randomly and equally likely choses a bit $c \in \{0,1\}$ and sends it to $\mathcal{P}$.

Step 3a: If $c = 0$, then $\mathcal{P}$ transfers the matrices S and P to $\mathcal{V}$ that verifies that $SHP = \widetilde{H}$ and $\mathbf{s}' = S\mathbf{s}$.

Step 3b: If $c = 1$, then $\mathcal{P}$ transfers $\mathbf{e}' = P^{-1}\mathbf{e}$ to $\mathcal{V}$ that verifies that $\mathrm{wt}(\mathbf{e}') = w$ and $\widetilde{H}\mathbf{e}' = \mathbf{s}$.

This protocol is running m times, where a safety parameter m is chosen so that a proving party fraud probability $1/2^m$ is less than a predetermined threshold. Let the communication complexity of this protocol to be estimated. In Step 1, the proving party passes $(N-K)(N+1)\log_2 q$ bit of data. In Step 2, the relying party transfers one bit. The amount of data transferred at Step 3 depends on the bit value c: at $c = 0$ there are $(N-K)^2\log_2 q + N\log_2 N$ bit, and at $c = 1$ there are $N\log_2 q$ bit transferred. Taking into account the fact that the bit value c is chosen randomly and equally likely, then m iterations result in the following communication complexity of a protocol:

$$\left(1+\frac{3N-K}{2}\right)m(N-K)\log_2 q + N(\log_2 N + m/2\log_2 q) + m \text{ bit.} \qquad (15)$$

As mentioned in work [13], if a matrix H is selected in a random manner, then the Giraut protocol is unsteady. At the same time, it is possible to calculate the codes with high-dimensionality hulls, because the complexity of the calculation of a signature $\mathcal{S}^W$ proposed in monograph [8] is a linear function of hull dimensionality. These codes can be induced codes $\mathbb{F}_q^l \otimes C$, whose hull dimensionality is l times higher than that of the base code C (see Eq. (5)). If the complexity of the calculation of a signature is neglected in assuming the effective computation of the latter, then the Giraut protocol can be implemented via the Reed–Maller based code $\mathbb{F}_q^l \otimes C$ (see Table 1), for which a cell value exceeds 128. Remembering that the communicative complexity (15) increases with rising $N = ln$, it is essential to select the parameters of the induced code that provide the smallest ln among the permissible values. In the example considered in the previous subsection, a minimally permissible value ln is $72 = 9 \cdot 8$.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. McEliece, R.J., A public-key cryptosystem based on algebraic coding theory, *JPL Deep Space Network Prog. Rep.,* 1978, nos. 42–44, pp. 114–116.
2. Sendrier, N. and Tillich, J.P., Code-Based Cryptography: New Security Solutions against a Quantum Adversary, ERCIM News, ERCIM, 2016. https://hal.archives-ouvertes.fr/hal-01410068/document.
3. Morelos-Zaragoza, R.H., *The Art of Error Correcting Coding,* John Wiley & Sons, Inc., 2006, 2nd ed.
4. Sidel'nikov, V.M. and Shestakov, S.O., On an encoding system constructed on the basis of generalized Reed-Solomon codes, *Discrete Math. Appl.,* 1992, vol. 2, no. 4, pp. 439–444.
5. Borodin, M.A. and Chizhov, I.V., Effective attack on the McEliece cryptosystem based on Reed-Muller codes, *Discrete Math. Appl.,* 2014, vol. 24, no. 5, pp. 273–280.
6. Deundyak, V.M. and Kosolapov, Yu.V., Algorithms for majority decoding of group codes, *Model. Anal. Inf. Sist.,* 2015, vol. 22, no. 4, pp. 464–482.
7. Deundyak, V.M. and Kosolapov, Yu.V., Cryptosystem based on induced group codes, *Model. Anal. Inf. Sist.,* 2016, vol. 23, no. 2, pp. 137–152.
8. Sendrier, N., Finding the permutation between equivalent linear codes: The support splitting algorithm, *IEEE Trans. IT,* 2000, vol. 46, no. 4, pp. 1193–1203.
9. Haily, A. and Harzalla, D., On binary linear codes whose automorphism group is trivial, *J. Discrete Math. Sci. Cryptogr.,* 2015, vol. 18, no. 5, pp. 495–512.
10. Lenstra, A.K. and Verheul, E.R., Selecting cryptographic key sizes, *J. Cryptol.,* 2001, vol. 14, no. 4, pp. 255–293.
11. Deundyak, V.M. and Kosolapov, Yu.V., The use of the tensor product of Reed-Muller codes in asymmetric McEliece type cryptosystem and analysis of its resistance to attacks on the cryptogram, *Vychisl. Tekhnol.,* 2017, vol. 22, no. 4, pp. 43–60.
12. Girault, M., A (non-practical) three-pass identification protocol using coding theory, *Advances in Cryptology AUSCRYPT'90*; *Lect. Notes Comput. Sci.,* 1990, vol. 453, pp. 265–272.
13. Sendrier, N. and Simos, D.E., The Hardness of Code Equivalence over $\mathbb{F}_q$ and its application to code-based cryptography, *Post-Quantum Cryptography. PQCrypto 2013*; *Lect. Notes Comput. Sci.,* 2013, vol. 7932, pp. 203–216.

Translated by O. Maslova