Permutation decoding: Towards an approach using algebraic properties of the -subcode

Matthieu Legeay

Permutation decoding: Towards an approach using algebraic properties of the σ-subcode

Matthieu Legeay

IRMAR - University of Rennes 1 - FRANCE
matthieu.legeay@univ-rennes1.fr

Abstract. In this paper, we show a manner to use properties of the permutation group on some binary linear codes to improve the decoding algorithms. We search especially for particular permutations and we prove bounds on dimension of a special subcode of the idempotent subcode. This σ-subcode can have very lower dimension in practice than the original code. We give several examples at the end and explain what can be the gain with this way of decoding.

Keywords: permutation group, σ-subcode, idempotent subcode, decoding, Goppa code, quasi-cyclic code

1 Introduction

Binary-linear codes with non-trivial automorphism groups appear in various domains in the field of coding theory (Reed-Solomon codes, Reed-Muller codes, quasi-cyclic codes and BCH codes, see [4]). Usually they come with a polynomial time decoding algorithm.

In code-based public key cryptography equally, it happens that one can find codes with non-trivial permutation group :

- ◇ Goppa codes are used in the original McEliece cryptosystem [1]. When these codes have a generator polynomial whose coefficients are in a subfield of the support, the corresponding codes have a non trivial permutation group [11] and it can be used to strengthen the cryptosystem against the decoding attacks [12].
- ◇ More recently, quasi-cyclic codes have been used in McEliece type cryptosystem and these codes have a non trivial permutation group too [13].
- ◇ Some compression functions in hash function involve quasi-cyclic codes too [14].

These systems can be attacked by generic decoding algorithms [5], [6], [7], [8], [9]. For binary linear codes, the NP-hardness of the Maximum-Likelihood Decoding (i.e. decoding up to w errors, with w fixed) was proved in [2]. However, none of these generic decoding algorithms take into account the fact that the permutation group is non-trivial, whereas it is possible to recover information on the permutation group by using the *support splitting algorithm* [15], [16].

A first attempt to improve generic decoding algorithm using the permutation group was by MacWilliams in 1964, on cyclic codes [3]. She presented an information set type decoding algorithm for cyclic codes by using the cyclic permutation on the codewords and the multiplication by 2 modulo the length of the code. For the cyclic permutation, a theoretical evaluation was made more recently in [10].

In this paper, however, we investigate another manner to use the permutation group of codes.

The idea is the following one : Suppose that someone receives the vector y (in data transmission for example) such as $y = c + e$, where c is a word of a binary $[n, k, d]$ linear code C and e is an error vector of length n and weight t. Decoding consists in removing errors (due to noise for instance) and recovering c (or equivalently e) from the only knowledge of y and C.

Moreover, suppose that the receiver knows the permutation group $Perm(C)$ of C (or part of the permutation group ; we will define it more rigorously in the next section). Then the receiver can compute the vector :

$$\sum_i \lambda_i \sigma_i(y) = \sum_i \lambda_i \sigma_i(c) + \sum_i \lambda_i \sigma_i(e)$$

where $(\sigma_i)_i \in Perm(C)$ and $(\lambda_i)_i \in \mathbb{F}_2$. Thus, $c' = \sum_i \lambda_i \sigma_i(c)$ lives in a particular subcode, the σ-subcode (it will be defined more rigorously in the next section), of C with probably lower dimension than k and $e' = \sum_i \lambda_i \sigma_i(e)$ is an error vector of weight $\leq \lambda t$, where λ is the number of non-zero λ_i.

Therefore, if the dimension of the σ-subcode is small enough, by doing a decoding exhaustive search, one can expect to find the possible values of e' (there is no longer a unique solution) which will give information on the positions of error in e.

This particular use of permutations is strongly based on the trade-off : moderate increase of the number of errors (by choosing adequate λ_i) and significant decrease of the dimension of the σ-subcode compared to k.

However, in general case, it is difficult to estimate the properties and the dimension of the σ-subcode. We are interested here in the paper by the simplest linear combination c', and we suppose that their exists an element of the permutation group of order 2.

In the first part we recall some basic background, then we prove bounds on the dimension of the σ- subcode and we explain how to use it for decoding, finally we give some examples.

2 Background

2.1 Permutation group of a code

We will only focus on binary linear codes of length n, that is to say $\mathbb{F}_2$-vector subspaces of $\mathbb{F}_2^n$. Let C be such a code with dimension k (as a vector subspace). Since we talk about it from beginning, define precisely now what we call the permutation group of C.

Definition 1. *Let $\sigma \in \mathfrak{S}_n$ be a permutation of the set $\{1, \ldots, n\}$. These permutations act on words of $\mathbb{F}_2^n$ as follows:*

$$\text{if } c = (c_i)_{i\in\{1,\ldots,n\}} \text{ is a word of } \mathbb{F}_2^n,\ \sigma(c) = (c_{\sigma^{-1}(i)})_{i\in\{1,\ldots,n\}}$$

We define the permutation group of C as the set

$$Perm(C) = \{\sigma \in \mathfrak{S}_n \mid \sigma(C) = C\} \text{ where } \sigma(C) = \{\sigma(c) \mid c \in C\}.$$

For instance, since we introduce them :

- the permutation group of an l-quasi-cyclic code contains the cyclic shift of l positions and its powers.
- the permutation group of a Goppa code whose generating polynomial has coefficients in the subfield $\mathbb{F}_{2^s}$ of $\mathbb{F}_{2^m}$ contains the group generated by the Frobenius automorphism of $\mathbb{F}_{2^m}/\mathbb{F}_{2^s}$.

2.2 Definition of the σ-subcode

In the introduction, we supposed that the receiver knew a part of $Perm(C)$ and we took linear combinations of $\sigma_i(y)$. To simplify and to be able to show properties of the σ-subcode, we have to restrict to special permutations in $Perm(C)$. Suppose now that we can find in $Perm(C)$ a permutation σ of order 2 ($\sigma^2 = Id$), which is a transposition, or a product of transpositions with disjoint cycles. Thus, we can define the sets C_{id} and C_{ad} :

$$C_{id} = \{c \in C \mid \sigma(c) = c\} \text{ and } C_{ad} = \{c + \sigma(c) \mid c \in C\},$$

with respective dimension k_{id} and k_{ad}. C_{id} is by definition the idempotent subcode of C. C_{ad} is what we call the σ-subcode of C. It is the one which interests us.

3 Dimension of the σ-subcode

First, we prove their inclusion and that they are really subcodes.

Proposition 1. *C_{id} and C_{ad} are subcodes of C and we have the chain of inclusion $C_{ad} \subseteq C_{id} \subseteq C$.*

Proof. It is easy to show that :

- Let $c_1, c_2 \in C_{id}$. Then, by linearity, $\sigma(c_1 + c_2) = \sigma(c_1) + \sigma(c_2) = c_1 + c_2$ and C_{id} is a subcode of C.
- Let $c_1, c_2 \in C_{ad}$. Then $c_1 = c + \sigma(c)$ and $c_2 = c' + \sigma(c')$ for $c, c' \in C$. So $c_1 + c_2 = c + c' + \sigma(c) + \sigma(c') = (c + c') + \sigma(c + c') \in C_{ad}$ and C_{ad} is also a subcode of C.
- Let $c_1 \in C_{ad}$. Then $c_1 = c + \sigma(c)$ for some $c \in C$. So $\sigma(c_1) = \sigma(c) + \sigma^2(c) = \sigma(c) + c = c_1$, because $\sigma^2 = Id$, and $c_1 \in C_{id}$, which completes the proof.

□

There are two possibilities :

- either all the codewords verify $\sigma(c) = c$ and then $C_{id} = C$ with $k_{id} = k$.
- either there is at least one codeword (and so many others) which do not verify this relation and then $C_{id} \subset C$ with $k_{id} < k$.

Our interest is on the other inclusion $C_{ad} \subseteq C_{id}$.

What is the relation between k, k_{id} and k_{ad} ?
Can we have $C_{ad} = C_{id}$, that is $C_{id} \subseteq C_{ad}$?

The following proposition answers the first question :

Proposition 2. *Let k_{id} and k_{ad} be the respective dimension of the subcodes defined above. We have the relation :$k_{ad} = k - k_{id}$*

Proof. Let G be a generator matrix of C ,

$$G = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{pmatrix}$$

with $g_i \in (\mathbb{F}_2)^n$. Let $\sigma(G)$ be the matrix

$$\sigma(G) = \begin{pmatrix} \sigma(g_1) \\ \sigma(g_2) \\ \vdots \\ \sigma(g_k) \end{pmatrix}.$$

We rearrange G by making linear combinations on its rows and we obtain another generator matrix :

$$G' = \begin{pmatrix} c_1 \\ \vdots \\ c_{k_{id}} \\ c'_1 \\ \vdots \\ c'_{k-k_{id}} \end{pmatrix} \text{ with } (c_i)_{1 \leq i \leq k_{id}} \in C_{id} \text{ and } (c'_i)_{1 \leq i \leq k-k_{id}} \notin C_{id}.$$

In fact to do this, just take a basis of C_{id} (for example by computing the kernel of $G - \sigma(G)$) and complete it to have a basis for C.

The first k_{id} rows is a generator matrix $Idem$ for C_{id}. The other $k - k_{id}$ rows form an other matrix which we call X. So $G' = \begin{pmatrix} Idem \\ X \end{pmatrix}$.

We have

$$G' + \sigma(G') = \begin{pmatrix} 0 \\ X + \sigma(X) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ c'_1 + \sigma(c'_1) \\ \vdots \\ c'_{k-k_{id}} + \sigma(c'_{k-k_{id}}) \end{pmatrix}.$$

By definition, $G' + \sigma(G')$ is a generator matrix for C_{ad}. Hence, to obtain information on k_{ad}, we need information on the rank of $G' + \sigma(G')$, that is the rank of $X + \sigma(X)$. Yet, we only know that $k_{ad} = rank(X + \sigma(X)) \leq k - k_{id}$ Suppose that $rank(X + \sigma(X)) < k - k_{id}$.

Then there exists $(b_i)_{2 \leq i \leq k-k_{id}} \in (\mathbb{F}_2)^{k-k_{id}-1}$, non constant zero sequence, such that $c'_1 + \sigma(c'_1) = \sum_{i=2}^{k-k_{id}} b_i(c'_i + \sigma(c'_i))$.

So, because of characteristic 2 and by linearity of the action of σ on the vector coordinates, $c'_1 + \sum_{i=2}^{k-k_{id}} b_i * c'_i = \sigma\left(c'_1 + \sum_{i=2}^{k-k_{id}} b_i * c'_i\right)$.

This implies that $c'_1 + \sum_{i=2}^{k-k_{id}} b_i * c'_i \in C_{id}$ and again there exists $(d_j)_{1 \leq j \leq k_{id}} \in (\mathbb{F}_2)^{k_{id}}$, non constant zero sequence, such that $c'_1 + \sum_{i=2}^{k-k_{id}} b_i * c'_i = \sum_{j=1}^{k_{id}} d_j * c_j$.

But, in this case, $(c_1, \ldots, c_{k_{id}}, c'_1, \ldots, c'_{k-k_{id}})$ is a linearly dependent set of vectors, which contradicts the fact that C is of dimension k. So, we must have $k_{ad} = rank(X + \sigma(X)) = k - k_{id}$.

□

Remark : The construction of G' can be done differently which permits to understand this proposition in another way. Take again the generator matrices

$$G = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{pmatrix} \text{ and } G' = \begin{pmatrix} c_1 \\ \vdots \\ c_{k_{id}} \\ c'_1 \\ \vdots \\ c'_{k_{ad}} \end{pmatrix}$$

of C. Without loss of generality, we can begin by g_1. There are two possibilities:

- either $g_1 \in C_{id}$, equivalently $\sigma(g_1) = g_1$: in this case $g_1 = c_i$ (for some $i \in [1, \ldots, k_{id}]$) and this increases k_{id} by 1
- either $g_1 \notin C_{id}$, equivalently $\sigma(g_1) \neq g_1$: in this other case, $g_1 = c'_i$ and $\sigma(g_1) = c'_j(\text{possibly} + \sum c_l)$ (for some $i, j \in [1, \ldots, k_{ad}]$). Then $g_1 + \sigma(g_1) \in C_{ad}$ and $g_1 + \sigma(g_1) = c_a$ (for some $a \in [1, \ldots, k_{id}]$), which increases k_{ad} and k_{id} by 1.

And so on with $g_2, \ldots, g_k$, taking care to drop the g_s which have already been used before.

Corollary 1. *Let k_{id} and k_{ad} be the respective dimension of the subcodes defined above. We have $k_{id} \geq \frac{k}{2}$ and $k_{ad} \leq \frac{k}{2}$.*

Proof. The inclusion $C_{ad} \subseteq C_{id}$ implies $k_{ad} = k - k_{id} \leq k_{id}$. □

4 Idea to use the permutation group for decoding

A way to use, as in introduction, linear combinations of type $\sum_i \lambda_i \sigma^i(y) = \sum_i \lambda_i \sigma^i(c) + \sum_i \lambda_i \sigma^i(e)$ has already been studied in [12] to improve the McEliece cryptosystem. The aim was to choose good combinations so that $\sum_i \lambda_i \sigma^i(e)$ is still of weight $\leq t$, where σ is the Frobenius automorphism of an extension field.

Here is another way to use it for decoding. Suppose again that $\sigma \in Perm(C)$ is of order 2 and restart from $y = c + e$. We obtain $\sigma(y) = \sigma(c) + \sigma(e)$, where $\sigma(c)$ is still a word of C and $\sigma(e)$ is still an error vector of length n and weight t.

Then we can compute $y' = y + \sigma(y) = c + \sigma(c) + e + \sigma(e)$, where $c' = c + \sigma(c)$ is a codeword living in the σ-subcode of C and $e' = e + \sigma(e)$ is still an error vector of length n but of weight $\leq 2 * t$ (mostly $2 * t << n$).

As seen in previous section, the dimension k_{ad} of the σ-subcode C_{ad} is such that $0 \leq k_{ad} \leq \frac{k}{2}$. Although the number of error bits in e' can have been doubled, if this dimension k_{ad} is low enough, we can do a decoding exhaustive search in C_{ad}. It will probably enable to find the possible values for e' and will give informations on the position of the error bits of e.

5 Examples

Here are several examples on some types of codes with different permutations σ. All computations were performed with magma.

5.1 BCH code

Let C be a binary primitive narrow-sense $[15, 5, 7]$ BCH code with designed distance 7, generated by

$$G = \begin{pmatrix} 1\,0\,0\,0\,0\,1\,1\,1\,0\,1\,1\,0\,0\,1\,0 \\ 0\,1\,0\,0\,0\,0\,1\,1\,1\,0\,1\,1\,0\,0\,1 \\ 0\,0\,1\,0\,0\,1\,1\,0\,1\,0\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,0\,0\,1\,1\,0\,1\,0\,1\,1\,1\,1 \\ 0\,0\,0\,0\,1\,1\,1\,0\,1\,1\,0\,0\,1\,0\,1 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix}.$$

We obtain that $\#Perm(C) = 20160$ and we have 315 permutations of order 2 (product of transpositions with disjoint cycles). There are two cases :

* If we take $\sigma = (1,2)(6,9)(10,15)(12,14) \in Perm(C)$, then

$$\{0, c_1 + c_2\} = C_{ad} \subset C_{id} = \langle c_3, c_4, c_5, c_1 + c_2 \rangle$$

$$k_{ad} = 1 \text{ and } k_{id} = 4$$

105 permutations of order 2 in $Perm(C)$ give this result. These permutations are all a product of 4 transpositions with disjoint cycles. In this case, the σ-subcode is a $[15, 1, 8]$ code.

* If we take $\sigma = (1,2)(3,12)(4,8)(6,13)(7,9)(11,14) \in Perm(C)$, then

$$< c_5, c_1 + c_2 + c_4 >= C_{ad} \subset C_{id} = \langle c_3, c_5, c_1 + c_2 + c_4 \rangle$$

$$k_{ad} = 2 \text{ and } k_{id} = 3$$

210 permutations in $Perm(C)$ give this result. These permutations are all a product of 6 transpositions with disjoint cycles. In this case, the σ-subcode is a $[15, 2, 7]$ code.

5.2 Quasi-cyclic code

Let C be a binary $[15, 5, 6]$ quasi-cyclic code, generated by

$$G = \begin{pmatrix} 1\,0\,0\,0\,0\,1\,1\,1\,0\,1\,0\,0\,0\,0\,1 \\ 0\,1\,0\,0\,0\,1\,1\,1\,1\,0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,1\,1\,1\,1\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,0\,1\,0\,1\,1\,1\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1\,1\,1\,0\,1\,1\,0\,0\,0\,1\,0 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix}.$$

We obtain that $\#Perm(C) = 3840$ and we have 311 permutations of order 2 (product of transpositions with disjoint cycles). There are three cases :

* If we take $\sigma = (1,15) \in Perm(C)$, then

$$\{0\} = C_{ad} \subset C_{id} = C$$

$$k_{ad} = 0 \text{ and } k_{id} = 5$$

31 permutations of order 2 in $Perm(C)$ give this result. These permutations are a product from 1 to 5 transpositions with disjoint cycles.

* If we take $\sigma = (1,2)(4,13)(9,10)(11,15) \in Perm(C)$, then

$$\{0, c_1 + c_2\} = C_{ad} \subset C_{id} = \langle c_3, c_4, c_5, c_1 + c_2 \rangle$$

$$k_{ad} = 1 \text{ and } k_{id} = 4$$

160 permutations in $Perm(C)$ give this result. These permutations are a product from 3 to 6 transpositions with disjoint cycles. In this case, the σ-subcode is a $[15, 1, 6]$ code.

* If we take $\sigma = (1,2)(3,4)(5,14)(6,7)(9,10)(11,15)(12,13)$, then

$$\langle c_1 + c_2, c_4 + c_5 \rangle = C_{ad} \subset C_{id} = \langle c_3, c_1 + c_2, c_4 + c_5 \rangle$$

$$k_{ad} = 2 \text{ and } k_{id} = 3$$

120 permutations in $Perm(C)$ give this result. These permutations are a product of 6 or 7 transpositions with disjoint cycles. In this case, the σ-subcode is a $[15, 2, 6]$ code.

5.3 Goppa code

Let now C be a binary $[16, 4, 7]$ Goppa code with $\mathbb{F}_{2^4}$ as support and whose generator polynomial is $g(x) = x^3 + x + 1$.

$$G = \begin{pmatrix} 1\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,1\,1\,0\,1\,1 \\ 0\,1\,0\,0\,1\,0\,0\,1\,1\,0\,1\,0\,1\,1\,1\,1 \\ 0\,0\,1\,0\,0\,1\,1\,1\,0\,0\,1\,0\,1\,0\,0\,1 \\ 0\,0\,0\,1\,1\,0\,0\,0\,1\,1\,0\,1\,0\,1\,1\,0 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}.$$

We obtain that $\#Perm(C) = 256$ and we have 95 permutations of order 2 (product of transpositions with disjoint cycles). There are three cases :

* If we take $\sigma = (3,6) \in Perm(C)$, then

$$\{0\} = C_{ad} \subset C_{id} = C$$

$$k_{ad} = 0 \text{ and } k_{id} = 4$$

63 permutations in $Perm(C)$ give this result. These permutations are a product from 1 to 5 transpositions with disjoint cycles.

* If we take $\sigma = (1,2)(3,16)(4,9)(6,13)(8,11)(10,15) \in Perm(C)$, then

$$\{0, c_1 + c_2\} = C_{ad} \subset C_{id} = \langle c_3, c_4, c_1 + c_2 \rangle$$

$$k_{ad} = 1 \text{ and } k_{id} = 3$$

24 permutations in $Perm(C)$ give this result. These permutations are a product from 5 to 7 transpositions with disjoint cycles. In this case, the σ-subcode is a $[16, 1, 8]$ code.

* If we take $\sigma = (1,2)(3,9)(4,16)(5,8)(6,15)(7,12)(10,13)(11,14)$, then

$$\langle c_1 + c_2, c_3 + c_4 \rangle = C_{ad} = C_{id} = \langle c_1 + c_2, c_3 + c_4 \rangle$$

$$k_{ad} = 2 \text{ and } k_{id} = 2$$

8 permutations in $Perm(C)$ give this result. These permutations are a product of 8 transpositions with disjoint cycles. In this case, the σ-subcode is a $[15, 2, 8]$ code.

5.4 Remarks

1. The last case $C_{id} = C_{ad}$ answers the second question of section 3. This case can obviously happen only if k is even, but the reciprocal is not true.
2. We can notice that the number of transpositions with disjoint cycles involved in the permutation σ chosen may have an impact on the dimensions. Indeed, this is understandable : the more you have transpositions in the permutation, the more you have constraints on the structure, and the more it is difficult for a codeword to be idempotent. That's why k_{id} decreases, and on the contrary k_{ad} increases, when we choose σ with many transpositions.
3. Which interests us is C_{ad} and particularly its dimension k_{ad}. We see in examples than k_{ad} can be reduced small in practice and this seems to happen when σ is composed of not many transpositions. Moreover, k_{ad} can even be 0. That's the best information we can have. Indeed, if we take again $y' = c' + e'$, this means that c' is directly the zero codeword and that $e' = y'$.
   For instance, resume the quasi-cyclic case with $\sigma = (1, 15)$. Two cases :
   - The word y' has a 1-bit on positions 1 and 15, then e has an error position on 1 **or** 15 (exclusive or) ;
   - The word y' has a 0-bit on positions 1 and 15, then either e has no error positions on 1 and 15, either e has an error position on 1 **and** 15.

6 Conclusion

In this paper, we prove bounds on the dimension of a particular subcode of the idempotent subcode. The results are attractive in practice because the obtained dimensions can be very low. This gives a manner to improve decoding by collecting information on the error positions of the original error vector.
The next work is to improve the suggested way of decoding in section 4 and to apply it with the already existing algorithms to see if we can have a real gain. Try to understand the correlation between the dimensions k_{id}, k_{ad} and the number of transpositions with disjoint cycles involved in σ is also an interesting way of research. Finally, the ultimate aim is also to see what happens if we take $\sigma \in Perm(C)$ of order 3 and possibly to generalize.

References

1. R.J. McEliece, "A public-key cryptosystem based on algebraic coding theory", *JPL DSN Progress Report*, pages 114-116, 1978.
2. E.R. Berlekamp, R.J. McEliece and H.C.A. Van Tilborg, "On the inherent intractability of certain coding problems", *IEEE Transactions on Information Theory*, 24(3):384-386, 1978.
3. F.J. MacWilliams, "Permutation decoding of systematic codes", in *Bell System Technical Journal*, Vol 43, pp 485-505, 1964.
4. F.J. MacWilliams and N.J.A.Sloane, "The theory of Error-Correcting Codes", 3nd edition Amsterdam, The Netherlands : North-Holland, 2009.

5. P.J. Lee and E.F. Brickell, "An observation on the security of McEliece's public-key cryptosystem", in *Advances in Cryptology - EUROCRYPT'88*, Vol 330 of *Lecture Notes in Computer Science*, pp 275-280, Springer-Verlag, 1988.
6. J.S. Leon, "A probabilistic algorithm for computing minimum weights of large error-correcting codes", *IEEE Transactions on Information Theory*, 34(5):1354-1359, 1988.
7. J. Stern, "A method for finding codewords of small weight", in *Coding Theory and Applications*, Vol 388 of *Lecture Notes in computer Science*, pp 106-113, Springer-Verlag, 1989.
8. A. Canteaut and F. Chabaud, "A new algorithm for finding minimum-weight words in a linear code : application to a primitive narrow-sense BCH codes of length 511", in *IEEE Transactions on Information Theory*, 44(1):367-378 1998.
9. D.J. Bernstein, T. Lange and C. Peters, "Attacking and defending the McEliece cryptosystem",in *Post-quantum cryptography: second international workshop, PQCrypto 2008*, Springer, 2008.
10. C. Chabot and M. Legeay, "Using permutation group for decoding", in *Proceedings of the Twelfth International Workshop on Algebraic and Combinatorial Coding Theory, ACCT-12*, pp 86-92, 2010.
11. P. Loidreau, "Codes derived from binary Goppa codes", in *Problems of Information Transmission 37*,n2 , pp 91-99, 2001.
12. P. Loidreau, "Strengthening McEliece cryptosystem", in *Advances in Cryptology (ASIACRYPT 200)*, pp 585-598, 2000.
13. T. P. Berger, P.L. Cayrel, P. Gaborit, and A. Otmani, "Reducing key length of the McEliece cryptosystem", in Bart Preneel, editor, *Progress in Cryptology - Second International Conference on Cryptology in Africa (AFRICACRYPT 2009)*, Vol 5580 of *Lecture Notes in Computer Science* , pp 77-97, Gammarth, Tunisia, 2009.
14. D. Augot, M. Finiasz, P. Gaborit, S. Manuel, and N. Sendrier, "Sha-3 proposal : FSB.", Submission to NIST, 2008.
15. N. Sendrier, "Finding the permutation between equivalent codes: the support splitting algorithm", *IEEE Transactions on Information Theory*, Vol 46, pp. 1193-1203, 2000.
16. P. Loidreau and N. Sendrier, "Weak keys in McEliece public-key cryptosystem", *IEEE Transactions on Information Theory*, 47(3):1207-1212, 2001.