International Journal of Algebra, Vol. 4, 2010, no. 3, 119 - 142

Gabidulin Codes that are Generalized Reed Solomon Codes

R. F. Babindamana and C. T. Gueye

Departement de Mathematiques et Informatique
Faculte des Sciences et Techniques
Universite Cheikh Anta Diop de Dakar
regisbab@ucad.sn, cgueye@ucad.sn

Abstract

In this paper, we describe permutations that allow to move from a Gabidulin code to a GENERALIZED REED SOLOMON code.

Mathematics Subject Classification: 11T71

Keywords: Gabidulin codes, Generalized Reed Solomon codes, cauchy matrix

1 Introduction

In [7] Pierre LOIDREAU raised a problem on the existence of a link between Gabidulin codes and generalized Reed Solomon codes. There are *GRS* codes having the same parameters than Gabidulin codes which are in the same ambient space. One may ask the following question: are there simple transformations like permutations or semi-linear isometries of Hamming metric that transform Gabidulin codes into *GRS* codes ?

When we consider a Gabidulin code of parameters *(n,k)*, where n is the length of the code and k the dimension of the code, algebraic transformations of generator matrix of the code allow to write generator matrix in the form $(I_k|P_i(g_j))$ with $i = 1, ..., k$ and $j = k+1, ..., n$. Saying that a Gabidulin code is transformed to a *GRS* code, implies that the following equation $P_i(g_j) = \frac{c_i d_j}{x_i + y_j}$, where P_i, c_i, d_j, x_i, y_j are unknown, has solutions up to affine permutations keeping the Hamming metric.

In this paper, we describe the existence of an application that conserves the Hamming distance and transforms a Gabidulin code into a *GRS* code. The idea of our method consists to transform the matrix of the GABIDULIN code to the form of systematic matrix by multiplying this matrix by the $k \times k$ invertible matrix extracted from the generator matrix of the

GABIDULIN code. So, using usual methods of the calculation of the invertible matrix,we obtain $(P_i(g_j))$ and by application of ψ_{ij} that are affine permutations that keep Hamming metric defined of $(GF(q^m))^n \longrightarrow (GF(q^m))^n$ such that $\psi_{ij}(P_i(g_j)) = a_{ij}P_i(g_j)$; we obtain, up to affine permutations, unknown elements of the equation $P_i(g_j) = \frac{c_i d_j}{x_i+y_j}$ i.e. P_i, c_i, d_j, x_i, y_j where $1 \leq i \leq k$ and $k+1 \leq j \leq n$, that ensure that a GABIDULIN code is transformed to a *GRS* code.

The paper is organized as follow: we recall basic facts about Cauchy matrices, *GRS* codes and GABIDULIN codes, we introduce our transformation that allow to transform a GABIDULIN code into *GRS* code, we generalize results for any parameters, and we present a example . At last we propose a natural algorithm to resolve the equation $P_i(g_j) = \frac{c_i d_j}{x_i+y_j}$ up to affine permutations.

2 Preliminary

In this section we recall basic definitions and properties over *GRS* codes and GABIDULIN codes that will be used in the sequel.

2.1 Matrix terminology

2.1.1 Diagonal matrix

Let be `K` a finite field. Given $v = (v_1, ..., v_n)$, where $(v_1, ..., v_n) \in$ `K`. we define `D(v)` to be the $n \times n$ diagonal matrix as

$$\texttt{D(v)} = \begin{pmatrix} v_1 & 0 & . & . & . & 0 \\ 0 & v_2 & . & . & . & 0 \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ 0 & 0 & . & . & . & v_n \end{pmatrix}$$

2.1.2 Cauchy Matrix:

Let K be a field, $x_i \in K$ for $1 \leq i \leq k$ and $y_j \in K$ for $1 \leq j \leq r$ such that $\{x_1, ..., x_k\}$ are pairwise distinct and $\{y_1, ..., y_r\}$ are pairwise distinct and $x_i + y_j \neq 0$ for $1 \leq i \leq k$ and $1 \leq j \leq r$.

The matrix

$$\begin{pmatrix} \frac{1}{x_1+y_1} & \frac{1}{x_1+y_2} & . & . & . & \frac{1}{x_1+y_r} \\ \frac{1}{x_2+y_1} & \frac{1}{x_2+y_2} & . & . & . & \frac{1}{x_2+y_r} \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ \frac{1}{x_k+y_1} & \frac{1}{x_k+y_2} & . & . & . & \frac{1}{x_k+y_r} \end{pmatrix}$$

is called a Cauchy matrix over K generated by $\{x_1, ..., x_k\}$ and $\{y_1, ..., y_r\}$.

2.1.3 Generalized Cauchy Matrix :

A $k \times r$ matrix $\mathbf{A}$ is a generalized Cauchy matrix if $A = D(c)CD(d)$
Where $\mathbf{C}$ is a $k \times r$ Cauchy matrix and $c = (c_1, ..., c_k)$, $c_i \neq 0$ for $1 \leq i \leq k$ and $d = (d_1, ..., d_r)$, $d_j \neq 0$ for $1 \leq j \leq r$.

2.2 Generalized Cauchy Codes

Let $k \in \mathbf{N}$ and $k < n$ for some $n \in \mathbf{N}$.
Let $\mathbf{C}$ be $k \times (n-k)$ Cauchy matrix over a field $\mathbf{K}$. Let $c = (c_1, ..., c_k)$ such that $c_i \in \mathbf{K}$ and $c_i \neq 0$, $\forall\ 1 \leq i \leq k$ and $d = (d_1, ..., d_{n-k})$ where $d_j \in \mathbf{K}$ and $d_j \neq 0\ \forall\ 1 \leq j \leq n-k$. Let $A = D(c)CD(d)$ ($\mathbf{A}$ is a generalized Cauchy matrix by definition). Then the code generated by the generator matrix $[I_k|A]$ is called the generalized cauchy code.

2.3 Definitions and Properties (CODES *GRS*)

Definition 1 *[10]:*
Let $GF(q^m)$ be a finite field with q^m elements. Let $n \in N$ with $1 \leq n \leq q^m$ and $\alpha = (\alpha_1, ..., \alpha_n)$ an n-tuple of distinct elements of $GF^(q^m)$ and let $v = (v_1, ..., v_n)$ be an n-tuple of non -zero elements of $GF^(q^m)$. Let $k \in N$ with $1 \leq k \leq n$. Then the Generalized Reed Solomon codes, denoted by : $GRS_{n,k}(\alpha, v)$ is
$GRS_{n,k}(\alpha, v) = \{v_1 f(\alpha_1), ..., v_n f(\alpha_n)/f \in GF(q^m)[x], deg(f) \leq k-1\}$.
We can thus write the generator matrix of Generalized Reed Solomon code as

$$G = \begin{pmatrix} v_1 & v_2 & . & . & . & v_n \\ v_1\alpha_1 & v_2\alpha_2 & & & & v_n\alpha_n \\ . & & & & & \\ . & & & & & \\ . & . & . & . & . & . \\ v_1{\alpha_1}^{k-1} & v_2{\alpha_2}^{k-1} & . & . & . & v_n{\alpha_n}^{k-1} \end{pmatrix}$$

is noted $GRS_k(\alpha, v)$.

The set of $GRS_k(\alpha, v)$ *codes is called by Generalized REED SOLOMON codes family.*

Proposition 1 *[8]:*
Let be $GRS_k(\alpha, v)$ *a Generalized REED SOLOMON code of length* n*, with* k *and* d *respectively the dimension and thye minimal distance of the code. We have the following conditions:*

1. $GRS_k(\alpha, v)$ *is a* MDS *code, i.e.* $d = n - k + 1$
2. *The* $GRS_k(\alpha, v)$ *dual code is the* $GRS_{n-k}(\alpha, v')$ *code for a certain vector* v' *determined in fonction of* v

Theorem 1 *:[10]*
Every generalized Cauchy code is a Generalized Reed Solomon code. Furthermore, given $v = (v_1, ..., v_n)$ *and* $\alpha = (\alpha_1, ..., \alpha_n)$ *for* $GRS_{n,k}(\alpha, v)$ *code, taking*
$x_i = -\alpha_i$, $y_j = \alpha_{k+j}$
$c_i = \frac{v_i^{-1}}{\prod_{t=1, t\neq i}^{k}(\alpha_i - \alpha_t)}$, $d_j = v_{j+k} \prod_{t=1}^{k}(\alpha_{j+k} - \alpha_t)$.
for $1 \leq i \leq k$, $1 \leq j \leq n-k$ *and* $c = (c_1, ..., c_k)$, $d = (d_1, ..., d_{n-k})$ *and* $\boldsymbol{C}$ *is the Cauchy matrix generated from* $\{x_1, ..., x_k\}$ *and* $\{y_{k+1}, ..., y_n\}$.
Then $A = [I_k | D(c)CD(d)]$ *is a generator matrix for* $GRS_{n,k}(\alpha, v)$.

2.4 Rank metric and GABIDULIN codes

Consider any finite field *GF(q)*. Given a vector $a = (a_1, ..., a_n) \in GF(q^m)^n$, the rank weight of a is by definition the rank of the $m \times n - matrix$ over *GF(q)* formed by extending every coordinate a_i on a basis of $GF(q^m)/GF(q)$. The construction is independent of the chosen basis.

The rank weight being a norm, it also defines a metric. With the distance related to the metric, we define minimum rank distance of a linear code, in the same way as the classical minimum distance for a code in the Hamming metric.

Definition 2 *:*
Let C *be a linear code over* $GF(q^m)$*. the minimum rank distance of* C *is* $d = Min_{c \in C^*}(Rk(c))$*. Given any matrix over* $GF(q^m)$ *we also define the rank of a matrix over* $GF(q)$*, and the minimum rank distance of a code.*

Definition 3 *:*
Let X *be a* $k \times n$ *matrix with coefficients in* $GF(q^m)$.
The column rank of X *over* GF(q) *is equal to the maximum number of columns of* X *that are linearly independent over* GF(q).

In 1985 Gabidulin, [4] published a family of codes which are optimal for the rank metric. Namely, they reach the "Singleton bound" for the rank metric.

Let $g_1, ..., g_n \in GF(q^m)^n$ *be* n *elements, which are linearly independent over* GF(q). *The matrix*

$$\boldsymbol{G} = \begin{pmatrix} g_1^{[0]} & . & . & . & g_n^{[0]} \\ . & . & . & & . \\ . & & . & & . \\ . & & . & & . \\ {g_1}^{[k-1]} & . & . & . & {g_n}^{[k-1]} \end{pmatrix}$$

where $[i] = q^i$ *is of rank* k, *is a generator matrix of Gabidulin code.*

Proposition 2 *([3]):*

- *The linear code* C *with generator matrix* G *reaches the Singleton bound for the rank metric. That is, let* d *be the minimum rank distance of* C, *we have* $d - 1 = n - k$.
- *The dual of the Gabidulin code is a Gabidulin code.*

Proposition 3 *([7]):*
A generator matrix of Gabidulin code $Gab_k(g)$ *can be in the form*

$$\begin{pmatrix} 1 & 0 & . & . & . & 0 & P_1(g_k+1) & . & . & . & P_1(g_n) \\ 0 & 1 & & & & 0 & P_2(g_k+1) & . & . & . & P_2(g_n) \\ . & & & . & & & & & & & . \\ . & & & . & & & & & & & . \\ . & & & . & & & & & & & . \\ 0 & 0 & . & . & . & 1 & P_k(g_k+1) & . & . & . & P_k(g_n) \end{pmatrix}$$

where for $i = 1...k$, P_i *is the unique q-polynomial of degree* k *checking for all* $j \leq k$, $P_i(g_j) = \delta_{i,j}$, $j = k+1, ..., n$.

2.5 Equivalence of linear codes

Two linear codes over $GF(q^m)$ are called equivalent if one can be obtained from the other by a combination of operations of the following types:
(A) permutation of the positions of code;
(B) multiplication of the symbols appearing in a fixed position by a non-zero scalar.

Remark 1 *If a code is displayed as an* $M \times n$ *matrix whose rows are the codewords, where* M *is the cardinal of the code and* n *the length of codewords, then an operation of type (A) corresponds to a permutation, or rearrangement of the columns of the matrix, while an operation of type (B) corresponds to a re-labelling of the symbols appearing in a given column.*
Clearly the distances between codewords are unchanged.

Theorem 2 *Two $k \times n$ matrices generate equivalent linear $[n,k]$-codes over $GF(q^m)$ if one matrix can be obtained from the other by a sequence of operations of the following types:*
(R_1) Permutation of the rows.
(R_2) Multiplication of a row by a non-zero scalar.
(R_3) Addition of a scalar multiple of one row to another.
(C_1) Permutation of the columns.
(C_2) Multiplication of any column by a non-zero scalar.

Proof:

The row operations (R_1), (R_2) and (R_3) preserve the linear independence of the rows of a generator matrix and simply replace one basic by another of the same code. Operations of type (C_1) and (C_2) convert a generator matrix to one for an equivalent code.

3 Results

Gabidulin operator ∇_{Gab}

Given a vector $(g_1, ..., g_n)$ over $GF(q^m)^n$ and $Gab_k(g)$ a generator matrix of a Gabidulin code $\mathbf{C}$ of length n and dimension k.
Let put $Gab_k(g) = (L, M)$.
We may suppose , to the rank of $rg(Gab_k(g)) = k$, that L is an invertible matrix. If not we can permute columns of $Gab_k(g)$ in order to obtain an invertible matrix L. This is possible because $rg(Gab_k(g)) = k$. So, we obtain a matrix $Gab_k^{\sigma}(g) = Gab_k P_{\sigma}(g)$, where P_{σ} is the permutated matrix corresponding to the permutation σ.
In the sequel we suppose $Gab_k(g) = (L, M)$ where L is $k \times k$ invertible matrix and $GF(q^m) = K$
.
Let $E_{Gab_k}(K)$ be the set of generator matrices of Gabidulin code of length n and dimension k and $S_{Gab_k}(K)$ the set of corresponding systematic matrices of Gabidulin code of length n and dimension k.
Let us consider $\nabla_{Gab} : E_{Gab_k}(K) \longrightarrow S_{Gab_k}(K)$
$(L, M) \longmapsto L^{-1}(L, M) = (I_k | L^{-1}M)$.
∇_{Gab} acts on $Gab_k(g)$ by transforming the $Gab_k(g)$ matrix of the code $\mathbf{C}$ by another matrix of the code $\mathbf{C}$. It replaces the base $(L_i)_{1 \leq i \leq k}$ by the base $(\delta_{ij}, P_i(g_j))$, where $(L^{-1}M) = (P_i(g_j))$. This transformation allows to obtain $(P_i(g_j))$.
∇_{Gab} is called Gabidulin operator.
`Conclusion:` We have two cases.

1. If L is invertible then the equivalent matrix $Gab_k(g)$ of Gabidulin code generate the same code as $Gab_k(g)$.

2. But if L is not invertible, the operations of (C_1) and (C_2) of theorem 2 are also used, the equivalent matrix of $Gab_k(g)$ generate code which is equivalent to (not necessarily the same) that generate by $Gab_k(g)$.
   Therefore by the Gabidulin operator, the Hamming distance between codewords is unchanged.

Proposition 4 *:*
Every Gabidulin code of length n and of dimension $k = 2$ over $GF(q^m)$, with generator vector $g = (g_1, ..., g_n)$ is a GRS code with parameters $v = a = g = (g_1, ..., g_n)$

Proof

Let be $Gab_2(g) = \begin{pmatrix} g_1 & . & . & . & g_n \\ g_1^2 & . & . & . & g_n^2 \end{pmatrix}$ a generator matrix of Gabidulin code of length n and of dimension 2.
Let us set $L_1 = (g_1, ..., g_n)$ the first line of the matrix and $L_2 = (g_1^2, ..., g_n^2)$ the second one. Since $v = a = g$ we have $L_1 = (v_1, ..., v_n)$ and $L_2 = (g_1.g_1, ..., g_n.g_n) = (v_1.a_1, ..., v_n.a_n)$
Thus $GRS_2(v, a) = \begin{pmatrix} L_1 \\ L_2 \end{pmatrix}$

Corrolary 1 *Every Gabidulin code of length n and dimension $n - 2$ over $GF(q^m)$, with generator vector $g = (g_1, ..., g_n)$ is a GRS code.*

Proof:

In fact the dual code of a Gabidulin code is a Gabidulin code, and the dual of a GRS *code is also a* GRS *code.*

Theorem 3 *:*
If $Gab_k(g)$ is a generator matrix of the Gabidulin code of the length n and of the dimension $k = 3$, $(I, P_i(g_j))$ the systematic generator matrix corresponding, then unknown elements of the following equation $P_i(g_j) = \frac{c_i d_j}{x_i + y_j}$ are obtained up to affine permutations keeping the hamming distance, with $1 \leq i \leq k$ and $k + 1 \leq j \leq n$.

Proof:

Let be

$$Gab_k(g) = \begin{pmatrix} g_1^{[0]} & . & . & . & g_n^{[0]} \\ . & . & . & & . \\ . & & . & & . \\ . & & . & & . \\ {g_1}^{[k-1]} & . & . & . & {g_n}^{[k-1]} \end{pmatrix}$$

a generator matrix of Gabidulin code of the length n and of the dimension k
For $k = 3$ we have

$$Gab_3(g) = \begin{pmatrix} g_1^{[0]} & g_2^{[0]} & g_3^{[0]} & . & . & g_n^{[0]} \\ g_1^{[1]} & g_2^{[1]} & g_3^{[1]} & . & . & g_n^{[1]} \\ g_1{}^{[2]} & g_2^{[2]} & g_3^{[2]} & . & . & g_n{}^{[2]} \end{pmatrix}$$

Without losing generality, we may suppose that

$$L = \begin{pmatrix} g_1^{[0]} & g_2^{[0]} & g_3^{[0]} \\ g_1^{[1]} & g_2^{[1]} & g_3^{[1]} \\ g_1{}^{[2]} & g_2^{[2]} & g_3^{[2]} \end{pmatrix}$$

is invertible

$$and \;\; M = \begin{pmatrix} g_4^{[0]} & . & . & . & g_n^{[0]} \\ g_4^{[1]} & & . & & g_n^{[1]} \\ g_4^{[2]} & . & . & . & g_n{}^{[2]} \end{pmatrix}$$

Then the form $(I_3|L^{-1}M)$ is a systematic matrix of the Gabidulin code with generator matrix $Gab_3(g)$. By identification $(I_3|L^{-1}M)$ and $(I_3|P_i(g_j)$, we obtain:

$$(L^{-1}M) = (P_i(g_j)) \tag{1}$$

where $1 \leq i \leq 3$ and $4 \leq j \leq n$. By calculating $(L^{-1}M)$,
where $L^{-1} = \frac{1}{detL} tCof(L)$, we can determine P_i.
We have

$$tCof(L) = t(Cof(g_i^{[l-1]})), \;\; 1 \leq i \leq 3 \;\; , 1 \leq l \leq 3 \;\; and \;\; [l] = q^l.$$

$det(L) \neq 0$ *because* L *is invertible*. By equality (1), we have:

$$P_i(z) = \frac{\sum_{l=1}^{3} p_i^l z^{[l-1]}}{det(L)} \tag{2}$$

where $p_i^l = Cof(g_i^{[l-1]})$ with $1 \leq i \leq 3$ and $1 \leq l \leq 3$.

$$det(L)P_i(z) = \sum_{l=1}^{3} p_i^l z^{[l-1]} = p_i^1 z^{[0]} + p_i^2 z^{[1]} + p_i^3 z^{[2]}. \tag{3}$$

The L matrix is a Vandermonde generalized matrix, then the determinant of L is :

$$det(L) = g_1 \prod_{i=1}^{2} \prod_{\lambda_1,\ldots,\lambda_i \in GF(q)} (g_{i+1} - \sum_{l=1}^{i} \lambda_l g_l)$$

Let us consider the polynomial $D_t(z)$ obtained by the determinant of L matrix by substituting the column t by elements z, z^q, z^{q^2}, where $1 \leq t \leq 3$. For instance

$$D_1(z) = \begin{vmatrix} z & g_2 & g_3 \\ z^q & g_2^q & g_3^q \\ z^{q^2} & g_2^{q^2} & g_3^{q^2} \end{vmatrix} = (g_2 g_3^q - g_3 g_2^q) z^{q^2} + \sum_{i=0}^{2} \lambda_i z^{q^i}$$

Where $\lambda_i \in GF(q^m)$ for $0 \preceq i \preceq 2$
First, since g_1, g_2, g_3 are linearly independent over $GF(q)$, g_2, g_3 are linearly independent over $GF(q)$. We have $D_1(g_2) = D_1(g_3) = 0$, and since $D_1(z)$ is a $q-$polynomial over $GF(q^m)$, all combination $\lambda_2 g_2 + \lambda_3 g_3$ with $\lambda_2,\ \lambda_3\ \in\ GF(q)$ are roots of $D_1(z)$. Thus $D_1(z)$ has q^2 distinct roots, so that we obtain a factorization

$$D_1(z) = (g_2 g_3^q - g_2^q g_3) \prod_{\lambda_2,\ \lambda_3 \in\ GF(q)} (z - \sum_{l=2}^{3} \lambda_l g_l)$$

Thus

$$D_i(z) = (g_t g_m^q - g_t^q g_m)_{1 \leq t < m \leq 3,\ t,m \neq i} \prod_{\lambda_1, \lambda_2,\ \lambda_3 \in\ GF(q)} (z - \sum_{l=1,\ l \neq i}^{3} \lambda_l g_l) \tag{4}$$

where the product (4) is did over the set of linear combinations of $(g_t)_{1 \leq t \leq 3,\ t \neq i}$
In the other way, by expansion along the first column we get :

$$D_1(z) = p_1^1 z^{[0]} + p_1^2 z^{[1]} + p_1^3 z^{[2]}, \quad [t] = q^t, \quad 0 \leq t \leq 2$$

i.e.

$$D_i(z) = p_i^1 z^{[0]} + p_i^2 z^{[1]} + p_i^3 z^{[2]} \tag{5}$$

So we get by (2), (3), (4), and (5) :

$$P_i(z) = \frac{\sum_{l=1}^{3} p_i^l z^{[l-1]}}{det(L)} = \frac{(g_t g_m^q - g_t^q g_m)_{1 \leq t < m \leq 3,\ t,m \neq i} \prod_{\lambda_1, \lambda_2, \lambda_3 \in GF(q)} (z - \sum_{l=1, l \neq i}^{3} \lambda_l g_l)}{g_1 \prod_{i=1}^{2} \prod_{\lambda_1, \ldots, \lambda_i \in GF(q)} (g_{i+1} - \sum_{l=1}^{i} \lambda_l g_l)}$$

We have for $z = g_j$

$$P_i(g_j) = \frac{(g_t g_m^q - g_t^q g_m)_{1 \leq t < m \leq 3,\ t,m \neq i} \prod_{\lambda_1, \lambda_2, \lambda_3 \in GF(q)} (g_j - \sum_{l=1, l \neq i}^{3} \lambda_l g_l)}{g_1 \prod_{i=1}^{2} \prod_{\lambda_1, \ldots, \lambda_i \in GF(q)} (g_{i+1} - \sum_{l=1}^{i} \lambda_l g_l)}$$

Let us consider the affine permutations ψ_{ij} defined by

$$\psi_{ij} : GF(q^m) \longrightarrow GF(q^m)$$

$$x \longmapsto a_{ij}x$$

where

$$a_{ij} = \frac{\prod\limits_{\lambda_1,\lambda_2,\lambda_3 \in GF(q)} (g_j - \lambda_i g_i - \sum\limits_{l=1, l\neq i}^{3} \lambda_l g_l)}{g_j - g_i}$$

with $\lambda_i \neq 0$ and $(\lambda_1, \lambda_2, \lambda_3) \neq (0, 0, 0)$.
Thus a_{ij} is all products of linear combinations containing g_i except the term $(g_j - g_i)$ Those affine permutations keep the hamming distance.
We are going to extend the action of ψ_{ij} to $GF(q^m)^n$ by the following form :

$$\psi : (GF(q^m))^n \longrightarrow (GF(q^m))^n$$

$$(x_1, x_2, x_3, x_j, ..., x_n) \longmapsto (x_1, x_2, x_3, \psi_{ij}(x_j), ..., \psi_{in}(x_n))$$

i.e the ψ act over the elements of $GF(q^m)^n$ by leaving invariant the k first components and in transforming the $n - k$ components.
Let be $Gab(g)$ a generator matrix of a Gabidulin code put under the systematic form. We set in the application ψ such as $\psi(Gab(g)) = (\psi(L_i))$ where $L_i = (I_i, P_i(g_j))$ with $I_i = \delta_{ij}$ $1 \leq i \leq k,\ k+1 \leq j \leq n$

$$\psi(Gab(g)) = (I_i, \psi_{ij}(P_i(g_j)))$$

By applying $\psi_{ij}(x) = a_{ij}x$ we get :

$$\psi_{ij}(P_i(g_j)) = \frac{(g_t g_m^q - g_t^q g_m)_{1\leq t<m\leq 3,\ t,m\neq i} \prod\limits_{\lambda_1,\lambda_2,\lambda_3 \in GF(q)} (g_j - \sum\limits_{l=1}^{3} \lambda_l g_l)}{g_1 \prod\limits_{i=1}^{2} \prod\limits_{\lambda_1,...,\lambda_i \in GF(q)} (g_{i+1} - \sum\limits_{l=1}^{i} \lambda_l g_l)(g_j - g_i)} \tag{6}$$

Where $\prod\limits_{\lambda_1,\lambda_2,\lambda_3 \in GF(q)} (g_j - \sum\limits_{l=1}^{3} \lambda_l g_l)$ is all products of linear combinations of g_1, g_2, g_3, $\lambda_l \in GF(q)$
Let us put

$$\psi_{ij}(P_i(g_j)) = \frac{c_1 d_j}{x_1 + y_j} \tag{7}$$

Then by identification of (6) and (7), we have :

$$\begin{cases} c_i = & \frac{(g_t g_m^q - g_t^q g_m)_{1\leq t<m\leq 3,\ t,m\neq i}}{g_1 \prod_{i=1}^{2} \prod_{\lambda_1,\lambda_2,\lambda_3 \in GF(q)} (g_{i+1} - \sum_{l=1}^{i} \lambda_l g_l)} \\ d_j = & \prod_{\lambda_1,\lambda_2,\lambda_3 \in GF(q)} (g_j - \sum_{l=1}^{3} \lambda_l g_l) \\ x_i = & -g_i \\ y_j = & g_j, \qquad\qquad with\ 4 \leq j \leq n \end{cases}$$

Thus we get :

1. For $i = 1 \quad \psi_{1j}(P_1(g_j)) = a_{1j}(P_1(g_j)) = \frac{c_1 d_j}{x_1 + y_j}$, we have :

$$\begin{cases} c_1 = & \frac{g_2 g_3^q - g_3 g_2^q}{g_1 \prod_{i=1}^{2} \prod_{\lambda_1,\ldots,\lambda_i \in GF(q)} (g_{i+1} - \sum_{l=1}^{i} \lambda_l g_l)} \\ d_j = & \prod_{\lambda_1,\lambda_2,\lambda_3 \in GF(q)} (g_j - \sum_{l=1}^{3} \lambda_l g_l) \\ x_1 = & -g_1 \\ y_j = & g_j,\ with\ 4 \leq j \leq \end{cases}$$

2. For $i = 2$

$$\psi_{2j}(P_2(g_j)) = \frac{c_2 d_j}{x_2 + y_j}$$

with

$$\begin{cases} c_2 = & \frac{g_1 g_3^q - g_3 g_1^q}{g_1 \prod_{i=1}^{2} \prod_{\lambda_1,\ldots,\lambda_i \in GF(q)} (g_{i+1} - \sum_{l=1}^{i} \lambda_l g_l)} \\ x_2 = & -g_2 \end{cases}$$

3. For $i = 3$

$$\psi_{3j}(P_3(g_j)) = \frac{c_3 d_j}{x_3 + y_j}$$

with

$$\begin{cases} c_3 = & \frac{g_1 g_2^q - g_2 g_1^q}{g_1 \prod_{i=1}^{2} \prod_{\lambda_1,\ldots,\lambda_i \in GF(q)} (g_{i+1} - \sum_{l=1}^{i} \lambda_l g_l)} \\ x_3 = & -g_3 \end{cases}$$

We have $x_1 \neq x_2 \neq x_3$, because $g_1,\ g_2,\ and\ g_3$ are distinct and y_j *are pairwise distinct with* $4 \leq j \leq n$

3.1 Example

- Let be the Gabidulin code of the length $n = 4$ and of the dimension $k = 3$ over $GF(2^4)$

- Let be α a primitive element of $GF(2^4)$, we have $\alpha^{15} = 1$ and $\alpha^4 = \alpha + 1$
  $GF(16)= \{0, 1, \alpha, \alpha^2, \alpha^3, \alpha+1, \alpha^2+\alpha, \alpha^3+\alpha^2, \alpha^2+1, \alpha^3+\alpha, \alpha^3+\alpha+1, \alpha^3+\alpha^2+\alpha, \alpha^3+\alpha^2+\alpha+1, \alpha^3+\alpha^2+1, \alpha^3+1\}$
  Let be $(1, \alpha, \alpha^2, \alpha^3)$ a base of $GF(16)$.
  Let us take $g = (g_1, g_2, g_3, g_4) \in (F_{2^4})^4$ such as :
  $g_1 = \alpha \longrightarrow 0100$
  $g_2 = \alpha^2 \longrightarrow 0010$
  $g_3 = \alpha^3 \longrightarrow 0001$
  $g_4 = \alpha^4 = \alpha + 1 \longrightarrow 1100$
  $g = (g_1, g_2, g_3, g_4)$ is linearly independent.

$$Gab_3(g)= \begin{pmatrix} {g_1}^{[0]} & {g_2}^{[0]} & {g_3}^{[0]} & {g_4}^{[0]} \\ {g_1}^{[1]} & {g_2}^{[1]} & {g_3}^{[1]} & {g_4}^{[1]} \\ {g_1}^{[2]} & {g_2}^{[2]} & {g_3}^{[2]} & {g_4}^{[2]} \end{pmatrix}$$

$$Gab_3(g)= \begin{pmatrix} \alpha & \alpha^2 & \alpha^3 & \alpha+1 \\ \alpha^2 & \alpha+1 & \alpha^3+\alpha^2 & \alpha^2+1 \\ \alpha+1 & \alpha^2+1 & \alpha^3+\alpha^2+\alpha+1 & \alpha \end{pmatrix}$$

Let us put

$$L = \begin{pmatrix} \alpha & \alpha^2 & \alpha^3 \\ \alpha^2 & \alpha+1 & \alpha^3+\alpha^2 \\ \alpha+1 & \alpha^2+1 & \alpha^3+\alpha^2+\alpha+1 \end{pmatrix} \quad and \quad M = \begin{pmatrix} \alpha+1 \\ \alpha^2+1 \\ \alpha \end{pmatrix}$$

We have

$$\begin{aligned} detL = & \; g_1 \prod_{i=1}^{2} \prod_{\lambda_1,\ldots,\lambda_i \in GF(2)} (g_{i+1} - \sum_{l=1}^{i} \lambda_l g_l) \\ detL = & \; g_1(g_2 - g_1)g_2 \prod_{\lambda_1,\lambda_2 \in GF(2)} (g_3 - (\lambda_1 g_1 + \lambda_2 g_2)) \end{aligned}$$

$$detL = \; g_1 g_2 (g_2 - g_1)[g_3(g_3 - g_2)(g_3 - g_1)(g_3 - g_1 - g_2)]$$

$$detL = \; \alpha.\alpha^2(\alpha^2 - \alpha)[\alpha^3(\alpha^3 - \alpha^2)(\alpha^3 - \alpha)(\alpha^3 - \alpha^2 - \alpha)]$$

$$detL = \; \alpha^3 + \alpha + 1$$

$$\psi_{ij}(P_i(g_j)) = \frac{(g_t g_m^q - g_t^q g_m)_{1 \le t < m \le 3,\ t,m \neq i} \prod_{\lambda_1,\lambda_2,\lambda_3 \in GF(q)} (g_j - \sum_{l=1}^{3} \lambda_l g_l)}{det(L)}$$

$1 \le i \le 3,\ j = 4 \quad where \quad \psi_{ij}(x) = a_{ij}x$ and $a_{ij} = \frac{\prod_{\lambda_1,\lambda_2,\lambda_3 \in GF(q)} (g_j - \lambda_i g_i - \sum_{l=1,l\neq i}^{3} \lambda_l g_l)}{g_j - g_i}$

1. for $i = 1$

$$P_1(g_4) = \; \frac{(g_2 g_3^2 - g_3 g_2^2) \prod_{\lambda_2,\lambda_3 \in GF(2)} (g_4 - \lambda_2 g_2 - \lambda_3 g_3)}{detL}$$

$$P_1(g_4) = \; \frac{(g_2 g_3^2 - g_3 g_2^2) g_4 (g_4 - g_2)(g_4 - g_3)(g_4 - g_2 - g_3)}{detL}$$

$a_{14} = (g_4 - g_1 - g_2)(g_4 - g_1 - g_3)(g_4 - g_1 - g_2 - g_3)$

$\psi_{14}(P_1(g_4)) = \frac{(g_2 g_3^2 - g_3 g_2^2) g_4 (g_4 - g_2)(g_4 - g_3)(g_4 - g_2 - g_3)(g_4 - g_1 - g_2)(g_4 - g_1 - g_3)(g_4 - g_1 - g_2 - g_3)}{detL}$

we multiply and we divide by $(g_4 - g_1)$. We get

$\psi_{14}(P_1(g_4)) = \frac{(g_2 g_3^2 - g_3 g_2^2) g_4 (g_4 - g_1)(g_4 - g_2)(g_4 - g_3)(g_4 - g_2 - g_3)(g_4 - g_1 - g_2)(g_4 - g_1 - g_3)(g_4 - g_1 - g_2 - g_3)}{detL(g_4 - g_1)}$

$$\begin{cases} c_1 = \frac{g_2 g_3^2 - g_3 g_2^2}{detL} \\ \\ d_4 = g_4(g_4 - g_1)(g_4 - g_2)(g_4 - g_3)(g_4 - g_2 - g_3)(g_4 - g_1 - g_2)(g_4 - g_1 - g_3) \\ (g_4 - g_1 - g_2 - g_3) \\ x_1 = -g_1 \\ y_4 = g_4 \end{cases}$$

$$\begin{cases} c_1 = \frac{\alpha^2(\alpha^3)^2-\alpha^3(\alpha^2)}{\alpha^3+\alpha+1} \\ \\ d_4 = \alpha^4(\alpha^4-\alpha)(\alpha^4-\alpha^2)(\alpha^4-\alpha^3)(\alpha^4-\alpha-\alpha^2)(\alpha^4-\alpha-\alpha^3)(\alpha^4-\alpha^2-\alpha^3) \\ (\alpha^4-\alpha-\alpha^2-\alpha^3) \\ x_1 = -\alpha \\ y_4 = \alpha^4 \end{cases}$$

$$\Longrightarrow \begin{cases} c_1 = \alpha^{-1} \\ \\ d_4 = \alpha^8 \\ x_1 = \alpha \\ y_4 = \alpha^4 \end{cases}$$

2. for $i = 2$

$$P_2(g_4) = \frac{(g_3g_1^2-g_1g_3^2)\prod_{\lambda_1,\lambda_3\in GF(2)}(g_4-\lambda_1g_1-\lambda_3g_3)}{detL}$$

$$P_2(g_4) = \frac{(g_3g_1^2-g_1g_3^2)g_4(g_4-g_1)(g_4-g_3)(g_4-g_1-g_3)}{detL}$$

$a_{24} = (g_4-g_1-g_2)(g_4-g_2-g_3)(g_4-g_1-g_2-g_3)$

$\psi_{24}(P_2(g_4)) = \frac{(g_3g_1^2-g_1g_3^2)g_4(g_4-g_1)(g_4-g_3)(g_4-g_1-g_3)(g_4-g_1-g_2)(g_4-g_2-g_3)(g_4-g_1-g_2-g_3)}{detL}$

we multiply and we divide by (g_4-g_2). We get
$\psi_{24}(P_2(g_4)) = \frac{(g_3g_1^2-g_1g_3^2)g_4(g_4-g_1)(g_4-g_2)(g_4-g_3)(g_4-g_2-g_3)(g_4-g_1-g_2)(g_4-g_1-g_3)(g_4-g_1-g_2-g_3)}{detL(g_4-g_2)}$

$$\begin{cases} c_2 = \frac{g_3g_1^2-g_1g_3^2}{detL} \\ \\ d_4 = g_4(g_4-g_1)(g_4-g_2)(g_4-g_3)(g_4-g_2-g_3)(g_4-g_1-g_2)(g_4-g_1-g_3) \\ (g_4-g_1-g_2-g_3) \\ x_2 = -g_2 \\ y_4 = g_4 \end{cases}$$

By replacing $g_1,\ g_2,\ g_3,\ g_4$, we get

$$\Longrightarrow \begin{cases} c_2 = \alpha^{13} \\ \\ d_4 = \alpha^8 \\ x_2 = \alpha \\ y_4 = \alpha^4 \end{cases}$$

3. for $i = 3$

$$P_3(g_4) = \frac{(g_1 g_2^2 - g_2 g_1^2)\prod_{\lambda_1,\lambda_2 \in GF(2)}(g_4 - \lambda_1 g_1 - \lambda_2 g_2)}{detL}$$

$$P_3(g_4) = \frac{(g_1 g_2^2 - g_2 g_1^2) g_4 (g_4 - g_1)(g_4 - g_2)(g_4 - g_2 - g_1)}{detL}$$

$a_{34} = (g_4 - g_3 - g_1)(g_4 - g_3 - g_2)(g_4 - g_1 - g_2 - g_3)$

$$\psi_{34}(P_3(g_4)) = \frac{(g_1 g_2^2 - g_2 g_1^2) g_4 (g_4 - g_1)(g_4 - g_2)(g_4 - g_1 - g_3)(g_4 - g_1 - g_2)(g_4 - g_2 - g_3)(g_4 - g_1 - g_2 - g_3)}{detL}$$

we multiply and we divide by $(g_4 - g_3)$. We get

$$\psi_{34}(P_3(g_4)) = \frac{(g_1 g_2^2 - g_2 g_1^2) g_4 (g_4 - g_1)(g_4 - g_2)(g_4 - g_3)(g_4 - g_2 - g_3)(g_4 - g_1 - g_2)(g_4 - g_1 - g_3)(g_4 - g_1 - g_2 - g_3)}{detL(g_4 - g_3)}$$

$$\begin{cases} c_3 = \frac{g_1 g_2^2 - g_2 g_1^2}{detL} \\ \\ d_4 = g_4(g_4 - g_1)(g_4 - g_2)(g_4 - g_3)(g_4 - g_2 - g_3)(g_4 - g_1 - g_2)(g_4 - g_1 - g_3) \\ (g_4 - g_1 - g_2 - g_3) \\ x_3 = -g_3 \\ y_4 = g_4 \end{cases}$$

By replacing $g_1,\ g_2,\ g_3,\ g_4$, we get

$$\Longrightarrow \begin{cases} c_3 = \alpha^8 \\ \\ d_4 = \alpha^8 \\ x_3 = \alpha^3 \\ y_4 = \alpha^4 \end{cases}$$

Example 2

Let us consider the Gabidulin code with following parameters $q = 2$, $k = 3$, $n = 6$ and $(\alpha_0, \alpha_1, \alpha_2, \beta_0, \beta_1, \beta_2)$ are linearly independent over $GF(2)$.

A generator matrix of the code is given par :

$$G = \begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \beta_0 & \beta_1 & \beta_2 \\ \alpha_0^2 & \alpha_1^2 & \alpha_2^2 & \beta_0^2 & \beta_1^2 & \beta_2^2 \\ \alpha_0^4 & \alpha_1^4 & \alpha_2^4 & \beta_0^4 & \beta_1^4 & \beta_2^4 \end{pmatrix}$$

Let us put

$$L = \begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 \\ \alpha_0^2 & \alpha_1^2 & \alpha_2^2 \\ \alpha_0^4 & \alpha_1^4 & \alpha_2^4 \end{pmatrix} \quad and \quad M = \begin{pmatrix} \beta_0 & \beta_1 & \beta_2 \\ \beta_0^2 & \beta_1^2 & \beta_2^2 \\ \beta_0^4 & \beta_1^4 & \beta_2^4 \end{pmatrix}$$

$$det(L) = \alpha_0 \prod_{i=0}^{1} \prod_{\lambda_0,\ldots,\lambda_i \in GF(2)} (\alpha_{i+1} - \sum_{l=0}^{i} \lambda_l \alpha_l)$$

$$det(L) = \alpha_0(\alpha_1 - \alpha_0)\alpha_1 \prod_{\lambda_0,\lambda_1 \in GF(2)} (\alpha_2 - (\lambda_0\alpha_0 + \lambda_1\alpha_1))$$

$$det(L) = \alpha_0\alpha_1(\alpha_1 - \alpha_0)[\alpha_2(\alpha_2 - \alpha_1)(\alpha_2 - \alpha_0)(\alpha_2 - \alpha_1 - \alpha_0)]$$

$$det(L) = \alpha_0\alpha_1\alpha_2(\alpha_1 - \alpha_0)(\alpha_2 - \alpha_1)(\alpha_2 - \alpha_0)(\alpha_2 - \alpha_1 - \alpha_0)$$

$det(L) \neq 0$ because $(\alpha_0, \alpha_1, \alpha_2)$ is a sub family of free family.
L is so invertible. The generator matrix G of the code is equivalent to $(I_3|L^{-1}M)$.
Let us put $(L^{-1}M) = (P_i(\beta_j))$, $0 \leq i \leq 2$, $0 \leq j \leq 2$.
where $L^{-1} = \frac{1}{detL} tCof(L)$, we can determine P_i.
We have

$$tCof(L) = t(Cof(\alpha_i^{[l]})), \quad 0 \leq i \leq 2 \ \ and \ \ 0 \leq l \leq 2.$$

$P_i(z) = \frac{\sum_{l=0}^{2} p_i^l z^{[l]}}{det(L)}$; $P_i^l = cof(\alpha_i^{[l]})$; $0 \leq i \leq 2$, $0 \leq j \leq 2$.

$$det(L)P_i(z) = \sum_{l=0}^{2} p_i^l z^{[l]} = p_i^0 z + p_i^1 z^2 + p_i^2 z^4$$

$$D_0(z) = p_i^0 z + p_i^1 z^2 + p_i^2 z^4 = (\alpha_1\alpha_2^2 - \alpha_2\alpha_1^2)z^4 + \sum_{i=0}^{2} \lambda_i z^{2^i}$$

$$D_0(z) = (\alpha_1\alpha_2^2 - \alpha_2\alpha_1^2) \prod_{\lambda_1,\ \lambda_2 \in\ GF(2)} (z - \sum_{l=1}^{2} \lambda_l \alpha_l)$$

$$P_i(z) = \frac{\sum_{l=0}^{2} p_i^l z^{[l]}}{det(L)} = \frac{(\alpha_t\alpha_m^2 - \alpha_t^2\alpha_m)_{0\le t<m\le 2,\ t,m\neq i} \prod_{\lambda_0,\lambda_1,\lambda_2\in GF(2)} (z - \sum_{l=0,l\neq i}^{2} \lambda_l\alpha_l)}{det(L)}$$

We have for $z = \beta_j$

$$P_i(\beta_j) = \frac{(\alpha_t\alpha_m^2 - \alpha_t^2\alpha_m)_{0\le t<m\le 2,\ t,m\neq i} \prod_{\lambda_0,\lambda_1,\lambda_2\in GF(2)} (\beta_j - \sum_{l=0,l\neq i}^{2} \lambda_l\alpha_l)}{det(L)}$$

Let us consider the affine permutations ψ_{ij} defined by

$$\psi_{ij} : GF(2^m) \longrightarrow GF(2^m)$$

$$x \longmapsto a_{ij}x$$

$\psi_{ij}(P_i(\beta_j)) = \frac{c_1 d_j}{x_1+y_j}$; $0 \le i \le 2$ *and* $0 \le j \le 2$

1.
For $i = 0$ $j = 0$ $\psi_{00}(P_0(\beta_0)) = a_{00}(P_0(\beta_0)) = \frac{c_0 d_0}{x_0+y_0}$

$P_0(\beta_0) = \frac{(\alpha_1\alpha_2^2-\alpha_2\alpha_1^2)\beta_0(\beta_0-\alpha_1)(\beta_0-\alpha_2)(\beta_0-\alpha_1-\alpha_2)}{detL}$

$a_{00} = (\beta_0 - \alpha_0 - \alpha_1)(\beta_0 - \alpha_0 - \alpha_2)(\beta_0 - \alpha_0 - \alpha_1 - \alpha_2)$

$\psi_{00}(P_0(\beta_0)) = \frac{(\alpha_1\alpha_2^2-\alpha_2\alpha_1^2)\beta_0(\beta_0-\alpha_1)(\beta_0-\alpha_2)(\beta_0-\alpha_1-\alpha_2)(\beta_0-\alpha_0-\alpha_1)(\beta_0-\alpha_0-\alpha_2)(\beta_0-\alpha_0-\alpha_1-\alpha_2)}{detL}$
We multiply and we divide by $(\beta_0 - \alpha_0)$, we get:
$\psi_{00}(P_0(\beta_0)) = \frac{(\alpha_1\alpha_2^2-\alpha_2\alpha_1^2)\beta_0(\beta_0-\alpha_0)(\beta_0-\alpha_1)(\beta_0-\alpha_2)(\beta_0-\alpha_1-\alpha_2)(\beta_0-\alpha_0-\alpha_1)(\beta_0-\alpha_0-\alpha_2)(\beta_0-\alpha_0-\alpha_1-\alpha_2)}{detL(\beta_0-\alpha_0)}$

$\psi_{00}(P_0(\beta_0)) = \frac{\alpha_1\alpha_2(\alpha_2-\alpha_1)\beta_0(\beta_0-\alpha_0)(\beta_0-\alpha_1)(\beta_0-\alpha_2)(\beta_0-\alpha_1-\alpha_2)(\beta_0-\alpha_0-\alpha_1)(\beta_0-\alpha_0-\alpha_2)(\beta_0-\alpha_0-\alpha_1-\alpha_2)}{\alpha_0\alpha_1\alpha_2(\alpha_1-\alpha_0)(\alpha_2-\alpha_1)(\alpha_2-\alpha_0)(\alpha_2-\alpha_1-\alpha_0)(\beta_0-\alpha_0)}$

After simplification, we get :
$\psi_{00}(P_0(\beta_0)) = \frac{\beta_0(\beta_0-\alpha_0)(\beta_0-\alpha_1)(\beta_0-\alpha_2)(\beta_0-\alpha_1-\alpha_2)(\beta_0-\alpha_0-\alpha_1)(\beta_0-\alpha_0-\alpha_2)(\beta_0-\alpha_0-\alpha_1-\alpha_2)}{\alpha_0(\alpha_1-\alpha_0)(\alpha_2-\alpha_0)(\alpha_2-\alpha_1-\alpha_0)(\beta_0-\alpha_0)}$
Thus we obtain :

$$\begin{cases} c_0 = \frac{1}{\alpha_0(\alpha_1-\alpha_0)(\alpha_2-\alpha_0)(\alpha_2-\alpha_1-\alpha_0)} \neq 0 \\ \\ d_0 = \beta_0(\beta_0 - \alpha_0)(\beta_0 - \alpha_1)(\beta_0 - \alpha_2)(\beta_0 - \alpha_1 - \alpha_2)(\beta_0 - \alpha_0 - \alpha_1)(\beta_0 - \alpha_0 - \alpha_2)(\beta_0 - \alpha_0 - \alpha_1 - \alpha_2) \\ d_0 \neq 0 \\ x_0 = -\alpha_0 \\ y_0 = \beta_0 \end{cases}$$

2
For $i = 1$ $j = 0$

$$\psi_{10}(P_1(\beta_0)) = a_{10}(P_1(\beta_0)) = \frac{c_1 d_0}{x_1 + y_0}$$

$$P_1(\beta_0) = \frac{(\alpha_2\alpha_0^2 - \alpha_0\alpha_2^2)\beta_0(\beta_0-\alpha_0)(\beta_0-\alpha_2)(\beta_0-\alpha_0-\alpha_2)}{detL}$$

$$a_{10} = (\beta_0 - \alpha_0 - \alpha_1)(\beta_0 - \alpha_1 - \alpha_2)(\beta_0 - \alpha_0 - \alpha_1 - \alpha_2)$$

$$\psi_{10}(P_1(\beta_0)) = \frac{(\alpha_2\alpha_0^2 - \alpha_0\alpha_2^2)\beta_0(\beta_0-\alpha_0)(\beta_0-\alpha_2)(\beta_0-\alpha_1-\alpha_2)(\beta_0-\alpha_0-\alpha_1)(\beta_0-\alpha_0-\alpha_2)(\beta_0-\alpha_0-\alpha_1-\alpha_2)}{detL}$$

We multiply and we divide by $(\beta_0 - \alpha_1)$, we get:

$$\psi_{10}(P_1(\beta_0)) = \frac{(\alpha_2\alpha_0^2 - \alpha_0\alpha_2^2)\beta_0(\beta_0-\alpha_0)(\beta_0-\alpha_1)(\beta_0-\alpha_2)(\beta_0-\alpha_1-\alpha_2)(\beta_0-\alpha_0-\alpha_1)(\beta_0-\alpha_0-\alpha_2)(\beta_0-\alpha_0-\alpha_1-\alpha_2)}{det(L)(\beta_0-\alpha_1)}$$

$$\psi_{10}(P_1(\beta_0)) = \frac{\alpha_0\alpha_2(\alpha_0-\alpha_2)\beta_0(\beta_0-\alpha_0)(\beta_0-\alpha_1)(\beta_0-\alpha_2)(\beta_0-\alpha_1-\alpha_2)(\beta_0-\alpha_0-\alpha_1)(\beta_0-\alpha_0-\alpha_2)(\beta_0-\alpha_0-\alpha_1-\alpha_2)}{\alpha_0\alpha_1\alpha_2(\alpha_1-\alpha_0)(\alpha_2-\alpha_1)(\alpha_2-\alpha_0)(\alpha_2-\alpha_1-\alpha_0)(\beta_0-\alpha_1)}$$

After simplification, we get :

$$\psi_{10}(P_1(\beta_0)) = \frac{-\beta_0(\beta_0-\alpha_0)(\beta_0-\alpha_1)(\beta_0-\alpha_2)(\beta_0-\alpha_1-\alpha_2)(\beta_0-\alpha_0-\alpha_1)(\beta_0-\alpha_0-\alpha_2)(\beta_0-\alpha_0-\alpha_1-\alpha_2)}{\alpha_1(\alpha_1-\alpha_0)(\alpha_2-\alpha_1)(\alpha_2-\alpha_1-\alpha_0)(\beta_0-\alpha_1)}$$

Thus we obtain :

$$\begin{cases} c_1 = \frac{-1}{\alpha_1(\alpha_1-\alpha_0)(\alpha_2-\alpha_1)(\alpha_2-\alpha_1-\alpha_0)} \neq 0 \\ \\ d_0 = \beta_0(\beta_0 - \alpha_0)(\beta_0 - \alpha_1)(\beta_0 - \alpha_2)(\beta_0 - \alpha_1 - \alpha_2)(\beta_0 - \alpha_0 - \alpha_1)(\beta_0 - \alpha_0 - \alpha_2)(\beta_0 - \alpha_0 - \alpha_1 - \alpha_2) \\ d_0 \neq 0 \\ x_1 = -\alpha_1 \\ y_0 = \beta_0 \end{cases}$$

3. In the same way, for $i = 2$, $j = 0$

$$\psi_{20}(P_2(\beta_0)) = \frac{\beta_0(\beta_0-\alpha_0)(\beta_0-\alpha_1)(\beta_0-\alpha_2)(\beta_0-\alpha_1-\alpha_2)(\beta_0-\alpha_0-\alpha_1)(\beta_0-\alpha_0-\alpha_2)(\beta_0-\alpha_0-\alpha_1-\alpha_2)}{\alpha_2(\alpha_2-\alpha_0)(\alpha_2-\alpha_1)(\alpha_2-\alpha_1-\alpha_0)(\beta_0-\alpha_2)}$$

Thus we obtain :

$$\begin{cases} c_2 = \frac{1}{\alpha_2(\alpha_2-\alpha_0)(\alpha_2-\alpha_1)(\alpha_2-\alpha_1-\alpha_0)} \neq 0 \\ \\ d_0 = \beta_0(\beta_0-\alpha_0)(\beta_0-\alpha_1)(\beta_0-\alpha_2)(\beta_0-\alpha_1-\alpha_2)(\beta_0-\alpha_0-\alpha_1)(\beta_0-\alpha_0-\alpha_2)(\beta_0-\alpha_0-\alpha_1-\alpha_2) \\ d_0 \neq 0 \\ x_2 = -\alpha_2 \\ y_0 = \beta_0 \end{cases}$$

4.
For $i = 0$ $j = 1$ $\psi_{01}(P_0(\beta_1)) = a_{01}(P_0(\beta_1)) = \frac{c_0 d_1}{x_0+y_1}$

$P_0(\beta_1) = \frac{(\alpha_1\alpha_2^2-\alpha_2\alpha_1^2)\beta_1(\beta_1-\alpha_1)(\beta_1-\alpha_2)(\beta_1-\alpha_1-\alpha_2)}{detL}$

$a_{01} = (\beta_1-\alpha_0-\alpha_1)(\beta_1-\alpha_0-\alpha_2)(\beta_1-\alpha_0-\alpha_1-\alpha_2)$

$\psi_{01}(P_1(\beta_1)) = \frac{\beta_1(\beta_1-\alpha_0)(\beta_1-\alpha_1)(\beta_1-\alpha_2)(\beta_1-\alpha_1-\alpha_2)(\beta_1-\alpha_0-\alpha_1)(\beta_1-\alpha_0-\alpha_2)(\beta_1-\alpha_0-\alpha_1-\alpha_2)}{\alpha_0(\alpha_1-\alpha_0)(\alpha_2-\alpha_0)(\alpha_2-\alpha_1-\alpha_0)(\beta_1-\alpha_0)}$

Thus we obtain : $c_0, c_1, c_2; x_0, x_1, x_2$ are the same

$$\begin{cases} \\ d_1 = \beta_1(\beta_1-\alpha_0)(\beta_1-\alpha_1)(\beta_1-\alpha_2)(\beta_1-\alpha_1-\alpha_2)(\beta_1-\alpha_0-\alpha_1)(\beta_1-\alpha_0-\alpha_2)(\beta_1-\alpha_0-\alpha_1-\alpha_2) \\ d_1 \neq 0 \\ y_1 = \beta_1 \end{cases}$$

For $i = 1$ $j = 1$ and for For $i = 2$ $j = 1$

$$\begin{cases} \\ d_1 = \beta_1(\beta_1-\alpha_0)(\beta_1-\alpha_1)(\beta_1-\alpha_2)(\beta_1-\alpha_1-\alpha_2)(\beta_1-\alpha_0-\alpha_1)(\beta_1-\alpha_0-\alpha_2)(\beta_1-\alpha_0-\alpha_1-\alpha_2) \\ d_1 \neq 0 \\ y_1 = \beta_1 \end{cases}$$

5.
for $i = 0$ $j = 2$, $i = 1$ $j = 2$ and for For $i = 2$ $j = 2$

$$\begin{cases} \\ d_2 = \beta_2(\beta_2-\alpha_0)(\beta_2-\alpha_1)(\beta_2-\alpha_2)(\beta_2-\alpha_1-\alpha_2)(\beta_2-\alpha_0-\alpha_1)(\beta_2-\alpha_0-\alpha_2)(\beta_2-\alpha_0-\alpha_1-\alpha_2) \\ d_2 \neq 0 \\ y_2 = \beta_2 \end{cases}$$

The generator matrix G of the Gabidulin code is equivalent to the systematic matrix $(I|L^{-1}M)$ that is transformed to a Generalized Cauchy matrix by affine permutations ψ_{ij} keeping the Hamming metric.

4 Generalization of main result

Theorem 4 *If $Gab_k(g)$ is a generator matrix of the Gabidulin code of the length n and of the dimension k, over $GF(q^m)$, $(I, P_i(g_j))$ the systematic generator matrix corresponding, then unknown elements of the following equation $P_i(g_j) = \frac{c_i d_j}{x_i + y_j}$ are obtained, up to affine permutations keeping Hamming metric, with $1 \leq i \leq k$ and $k+1 \leq j \leq n$.*

Proof

The proof is the same as for the case $k = 3$.
Let be

$$Gab_k(g) = \begin{pmatrix} g_1^{[0]} & . & . & . & g_n^{[0]} \\ . & . & . & & . \\ . & & . & & . \\ . & & . & & . \\ {g_1}^{[k-1]} & . & . & . & {g_n}^{[k-1]} \end{pmatrix}$$

a generator matrix of Gabidulin code of the length n and of the dimension k
and

$$L = \begin{pmatrix} g_1^{[0]} & . & . & . & g_k^{[0]} \\ . & . & . & & . \\ . & & . & & . \\ . & & . & & . \\ {g_1}^{[k-1]} & . & . & . & {g_k}^{[k-1]} \end{pmatrix}$$

We know that $Gab_k(g)$=$(I_k|P_i(g_j)$ and $P_i(g_j) = \frac{\sum_{l=1}^{k} p_i^l g_j^{[l-1]}}{det(L)}$

$$detL = g_1 \prod_{i=1}^{k-1} \prod_{\lambda_1,\ldots,\lambda_i \in GF(q)} (g_{i+1} - \sum_{l=1}^{i} \lambda_l g_l)$$

$$D_i(z) = p_i^1 z + p_i^2 z^q + \ldots + p_i^k z^{q^{k-1}}$$

Furthermore :

$$D_i(g_j) = Cof(g_i^{[k-1]}) \prod_{\lambda_1,\ldots,\lambda_k \in GF(q)} (g_j - \sum_{l=1, l\neq i}^{k} \lambda_l g_l)$$

Where $Cof(g_i^{[k-1]})$ is the cofactor obtained by eliminating the row and the column corresponding to element $g_i^{[k-1]}$.
Thus

$$P_i(g_j) = \frac{Cof(g_i^{[k-1]}) \prod\limits_{\lambda_1,...,\lambda_k \in GF(q)} (g_j - \sum\limits_{l=1, l\neq i}^{k} \lambda_l g_l)}{g_1 \prod\limits_{i=1}^{k-1} \prod\limits_{\lambda_1,...,\lambda_i \in GF(q)} (g_{i+1} - \sum\limits_{l=1}^{i} \lambda_l g_l)}$$

Let us take $\psi_{ij}(x) = a_{ij}x$

$$a_{ij} = \frac{\prod\limits_{\lambda_1,...,\lambda_k \in GF(q), \lambda_l \neq \lambda_i} (g_j - \lambda_i g_i - \sum\limits_{l=1, l\neq i}^{k} \lambda_l g_l)}{g_j - g_i}$$

with $\lambda_i \neq 0$ and $(\lambda_1, ..., \lambda_k) \neq (0, ..., 0)$.
Thus by applying ψ_{ij}, we have

$$\psi_{ij}(P_i(g_j)) = \frac{Cof(g_i^{[k-1]}) \prod\limits_{\lambda_1,...,\lambda_k \in GF(q)} (g_j - \sum\limits_{l=1}^{k} \lambda_l g_l)}{g_1 \prod\limits_{i=1}^{k-1} \prod\limits_{\lambda_1,...,\lambda_i \in GF(q)} (g_{i+1} - \sum\limits_{l=1}^{i} \lambda_l g_l)(g_j - g_i)} = \frac{c_i d_j}{x_i + y_j}$$

We get :

$$\begin{cases} c_i = \dfrac{Cof(g_i^{[k-1]})}{g_1 \prod\limits_{i=1}^{k-1} \prod\limits_{\lambda_1,...,\lambda_i \in GF(q)} (g_{i+1} - \sum\limits_{l=1}^{i} \lambda_l g_l)} \\ d_j = \prod\limits_{\lambda_1,...,\lambda_k \in GF(q)} (g_j - \sum\limits_{l=1}^{k} \lambda_l g_l) \\ x_i = -g_i \\ y_j = g_j \end{cases}$$

Proposition 5 *If $Gab_k(g)$ is a generator matrix of a Gabidulin code of the length n and of the dimension k over $GF(q^m)$ then there exist affine permutations keeping hamming metric that transform a generator matrix of the Gabidulin code to a Generalized Cauchy matrix.*

Proof

Let be $Gab_k(g) = (I_k|P_i(g_j))$ a systematic matrix of Gabidulin code.
We know that

$$\psi_{ij}(P_i(g_j)) = \begin{pmatrix} \frac{c_1 d_{k+1}}{x_1+y_{k+1}} & . & . & . & \frac{c_1 d_n}{x_1+y_n} \\ . & & . & & . \\ . & & & . & . \\ . & & & & . \\ \frac{c_k d_{k+1}}{x_k+y_{k+1}} & . & . & . & \frac{c_k d_n}{x_k+y_n} \end{pmatrix} \quad 1 \leq k \leq n \text{ and } k+1 \leq j \leq n$$

is a $k \times (n-k)$ matrix.

$$\psi_{ij}(P_i(g_j)) = \begin{pmatrix} c_1 & . & . & . & 0 \\ 0 & c_2 & 0 & . & 0 \\ . & 0 & . & & . \\ . & . & . & . & . \\ 0 & 0 & . & 0 & c_k \end{pmatrix} \begin{pmatrix} \frac{1}{x_1+y_{k+1}} & . & . & . & \frac{1}{x_k+y_n} \\ . & & . & & . \\ . & & & . & . \\ . & & & & . \\ \frac{1}{x_k+y_{k+1}} & . & . & . & \frac{1}{x_k+y_n} \end{pmatrix} \begin{pmatrix} d_{k+1} & . & . & . & 0 \\ 0 & d_{k+2} & 0 & . & 0 \\ . & 0 & . & & . \\ . & . & . & . & . \\ 0 & 0 & . & 0 & d_{k+n} \end{pmatrix}$$

i.e. $\psi_{ij}(P_i(g_j))= (D(c)CD(d))$
$\psi_{ij}(Gab_k(g)) = (I_k|D(c)CD(d))$ where $\mathbf{C} = (\frac{1}{x_i+y_j})$ is a Cauchy matrix with $1 \leq i \leq k$ and $k+1 \leq j \leq n$.
$(x_1, ..., x_k)$ are pairwise distinct and $(y_{k+1}, ..., y_n)$ are pairwise distinct because $x_i = -g_i$ and $y_j = g_j$ are elements of a free family.
$c_i = \frac{Cof(g_i^{[k-1]})}{detL} \neq 0$. In fact $Cof(g_i^{[k-1]})$ is a factor of $D_i(z) \neq 0$ and $detL \neq 0$
$d_j \neq 0$ because $D_i(z) \neq 0$

Theorem 5 *any Gabidulin code of the length n and of the dimension k is a GRS code up to affine permutations keeping Hamming metric.*

Proof

By the theorem 1 a Generalized cauchy code is a *GRS* code. Thus, by the proposition 5, all Gabidulin code is transformed to *GRS* code by affine permutations keeping Hamming metric.

5 Natural algorithm

1. Input : a generator matrix $Gab_k(g)$ of Gabidulin code of length n and dimension k.
2. Output: $P_i(g_j)$, ψ_{ij}, c_i, x_i, d_j, y_j.

Procedure:

3. First step: To put the matrix $Gab_k(g)$ in the systematic form:
   To extract a $k \times k$ square matrix L
   To put $Gab_k(g)$ in the form $Gab_k(g) = (I_k|L^{-1}M)$

4. Second step:
   calculating of $P_i(g_j) = (L^{-1}M)$

   calculating of : a_{ij}
   calculating of : $\psi_{ij}(P_i(g_j)) = a_{ij}P_i(g_j)$
   calculating of: c_i, x_i, d_j, y_j by $\psi_{ij}(P_i(g_j)) = \frac{c_i d_j}{x_i + y_j}$.

6 Conclusions and Future Work

We have shown that for any k and for any n, any Gabidulin code of the length n and of the dimension k is equivalent up to permutions that keep the Hamming distance to a GRS code. In our future work we will present an efficient method for decoding of Gabidulin code by using the structure of the inverse of the Vandermonde matrix.

7 References

[1] D. Augot Travaux de Madhu Sudan sur les codes correcteurs d'erreurs

[2] T.B. Berger, "Isometries for rank distance and permutation group of Gabidulin codes", in Proc. 8^th Int. Workshop on Agebraic and Combinatorial Coding Theory ($ACCT8$), St Petersbourg, Russia, Sept. 2002, pp. $30 - 33$.

[3] T.P. Berger, P. Loidreau " How to mask the structure of codes for a cryptographic use", August 29, 2006.

[4] E. Gabidulin "Theory of codes with maximum rank distance", Problemy Peredachi Informatsii, $vol.21, n.1, p1 - 12, 1985$.

[5] P. Gaborit "Shorter keys for code based cryptography" September $8, 2004$.

[6] LIDL and NIEDERREITER "Introduction to finite fields and their applications" CAMBRIDGE University, Presso

[7] P. Loidreau "An Algebraic Attack against Augot−Finiasz cryptosystem", INRIA $n°5662$, 2005.

[8] P. Loidreau "Metrique rang et cryptographie", memoire d'Habilitation à diriger des Recherches Universite Pierre et Marie Curie, Paris 6. 25 janvier 2007.

[9] P. Loidreau "Etude et Optimisation des Cryptosystemes à Cle Publique fondes sur la Theorie des Codes Correcteurs d'Erreurs", 4 mai 2001, these presentee à l'Ecole Nationale Superieure de Techniques Avancees (ENSTA), Universite Paris 6.

[10] R.M.Roth and G. Seroussi, On generator matrices of MDS codes, IEEE Trans. Inform. Theory, vol. 31, N°. 6 (1985) pp. 826-830.

[11] M. Shretha and L. Xu , Efficient Erasure Decoding for Generalized Reed Solomon Codes .

Received: June, 2009