République Algérienne Démocratique et Populaire
Ministère de L'enseignement Supérieure et de la recherche Scientifique
Université El Hadj Lakhdar Batna

Faculté des Sciences
Département de Mathématiques
Laboratoire des Techniques Mathématiques

MEMOIRE

Présenté pour obtenir le diplôme de

MAGISTER

Thème

Les codes Cortex et les codes auto- duaux de type I et de type II

Option : Mathématiques Appliquée

Par
Chatouh karima

Soutenue le : 03/06/2007

Devant le jury

Mr. S. E.REBIAI	Prof	Université de Batna	*Président*
Mr. M. BENLAHCENE	M.C	Université de Batna	*Rapporteur*
Mr. D.MIHOUBI	M.C	Université de Msila	*Examinateur*
Mr. A. AMROUN	M.C	Université de Msila	*Examinateur*

Table des matières

REMERCIMENTS

Je tiens à remercier chaleureusement Monsieur
BENLAHCENE MOUSSA pour l'aide inestimable qu'il m'a apporté dans la documentation nécessaire à la rédaction de cette thèse.

Je remercie également les nombreux professeurs du département de mathématiques, respectivement : **SALAH EDDINE REBIAI** (professeur à l'université de BATNA), Monsieur**D. MIHOUBI**. Monsieur **A. AMROUN** (Maîtres de conférence Université de M'SILA) qui ont accepté d'être des membres du jury de cette thèse.

Mes remerciements aussi à tous mes professeurs qui ont contribué à ma formation.

Je n'en tiens pas moins à leur signifier ma reconnaissance et ma gratitude.

Je dédie ce modeste travail à mes parents, à mon mari **LAZHAR** qui m'a beaucoup soutenu dans la réalisation de ce projet, sans oublier à remercier **KHADRAOUI FAIROUZE**, qui m'a aidé avec ses conseils judicieux, ainsi qu'à **GHALIA MERZOUGUI**.

Mes remerciements aussi à toute ma famille, surtout ma soeur **MALIKA**, mon frère **FARES** et **YACINE** ; à la famille de mon mari.

Mes respects à tous les membres du jury pour avoir accepter d'évaluer mon travail.

Introduction :

Le codage correcteur d'erreurs, appelé aussi codage du canal, consiste à rajouter à l'information numérique, des symboles binaires, appelés symboles redondance, suivant une loi mathématique particulière.

Le décodeur vérifie que la loi de codage n'a pas été modifiée lors des divers traitements réalisés sur l'information numérique codée. Si c'est le cas, le décodeur conclut à l'absence d'erreur ; dans le cas contraire, par un traitement approprié, il repaire les symboles erronés puis les corrige par simple inversion binaire. Malheureusement, le codage correcteur d'erreurs a des capacités de correction limitées et ainsi, certaines erreurs peuvent lui échapper.

Les chercheurs dans la théorie de codage s'intéressent à trouver les meilleurs codes qui serviront une fonction particulière. Parfois ils s'intéressent à trouver un " meilleur " code, tant que d'autre fois ils veulent tous les "meilleurs " codes. Par exemple si on désire utiliser des codes pour la transmission de l'information ou le stockage de donnés, l'objectif est d'avoir des codes avec petites longueurs pour la transmission rapide, un grand nombre de mots de code pour envoyer un grand nombre de message, et un grand " poids minimum " pour corriger plusieurs erreurs.

Ces deux objectifs sont en conflit, de telles recherches peuvent limiter un ou plus de ces paramètres et trouver alors un ou tous les " meilleurs " codes en termes des autres paramètres.

Un peu d'histoire

En 1948 Claude Shannon, ingénieur des Bell Labs, détermine une limite appelée la limite théorique de Shannon, au pouvoir de correction du codage correcteur d'erreurs. Ce résultat majeur de théorie de l'information ne donne toute fois pas de solution pratique pour réaliser le code permutant d'atteindre cette limite.

Dans les décennies qui suivirent ce résultat, la communauté scientifique se mobilisa et nombreux chercheurs, probablement des meilleurs à travers le monde, se mirent à la recherche d'un code susceptible d'atteindre la limite de Shannon. Des résultats significatifs furent obtenus mais tous les codes imaginés possédaient des performances un peu décevantes ; la limite de Shannon semblait inaccessible. En 1988, Claude Berroux et Alain Glavieux, professeurs à L'ENST Bretagne, tentent à leur toute l'aventure et suivent une démarche mêlant intuition et non-conformisme.

À peine trois ans plus tard, les premiers résultats sont là ! Les turbos codes sont nés et la limite théorique de Shannon est pratiquement atteinte. Les premiers brevets sont déposés dés 1991, suivis en 1993 d'une présentation des turbo codes à Genève dans le cadre de l'international conférence en

communication (ICS).

Les turbos codes ont ouvert une nouvelle voie de recherche qui dépasse très longuement le domaine strict du codage. En effet, le principe turbo code est maintenant appliqué à la plupart des fonctions de la chaîne de communication et notamment à la démodulation, la détection, l'égalisation ou encore à la détection multi-utilisateur.

Les turbos codes ont été adoptés par le CCSDS (consultative commite fore space Data systems), comité de normalisation pour les agences spéciales mondiales (ESA, NASA, NASDA,.....).

La NASA a déclaré vouloir remplacer l'ancien standard de codage par un turbo code, pour toutes ses missions en espace lointain à partir de 2003. La troisième génération de téléphonie mobile, L'UMTS en Europe, a également retenu les turbo codes pour la protection des transmissions de données dont le débit excède 64 k bits/s. Des liaisons par satellite pour des applications de télévision numérique ou de diffusion de l'Internet ont d'ores et déjà fait entrer les turbo codes dans leurs normes.

Les turbo codes devraient également être utilisés dans les réseaux locaux sans fil (WLAN) ainsi que dans la nouvelle génération d'ADSL. En fin, des études sont en cours pour utiliser les turbo codes dans la transmission sur fibres optiques ou pour la protection des données stockées sur disque dur ou CD-ROM.

Ce mémoire s'inscrit dans cette optique et prend comme idée les travaux de Carlach et Vervoux en 1998 **[15]**. Cette construction utilise tout comme les turbo codes des permutations, et des codes très courts que l'on assemble entre eux de différentes façons pour obtenir des codes de rendement $\frac{1}{2}$ et de longueurs importantes. Des travaux ont été menés sur ces codes notamment par Olocco, Tillich, et Otmani.

Le but de cette thèse est l'étude de ces nouveaux codes, dits codes Cortex. Ces codes sont construits à partir des codes de petites longueurs (le code de Hamming étendu entre autre pour obtenir des codes de type I et de type II), en général, concaténés et entrelacés. Philippe Gaborit a étudié la nouvelle méthode de construction de la famille des codes Cortex et a construit des codes avec des paramètres non connus surprenant, un code [92,46,16] sur $\mathbb{F}_2$, une code [52,26,15] sur $\mathbb{F}_3$ ou encore [46,23,14] sur $\mathbb{F}_4$ (Projet code INRIA). Les codes Cortex obtenus sont des codes auto-duaux.

Ce mémoire de thèse est constitué de quatre chapitres organisés comme suit :

- Le premier chapitre résume les notions de base : codes linéaires et leurs paramètres, code de Hamming, code de Golay, dualité. Nous donnons les définitions et les propriétés de base de classe des codes auxquelles nous nous sommes intéressés.

- Dans le deuxième chapitre, nous présentons les codes auto-duaux est leurs propriétés. On s'intéresse aussi à la classification des codes auto-duaux binaires, ternaire et quaternaire.

- Le troisième chapitre traite une famille de turbo codes appelés codes Cortex, nous donnons quelque rappels sur leurs construction et propriétés, dans la présentation des travaux de **[15]** et **[7].** Nous donnons une représentation Cortex des codes auto-duaux de longueur inférieur ou égal à 20.

- Le dernier chapitre est consacré à la construction des codes Cortex a partir de code de bases non auto-duaux de matrice redondante d'ordre 2 et 3 avec des suites de permutations de cardinal 1 et 2 ainsi que l'établissement des équivalences des codes construits.

Chapitre 1

Généralités sur les codes correcteurs d'erreurs

1.1 Introduction

Le principe de construction d'un code correcteur d'erreurs systématique consiste à ajouter aux mots constitué de m éléments d'information $a_1a_2...a_m$ où les a_i parcourent un corps fini $\mathbb{F}_q$, k éléments de contrôle (ou de redondance) $a_{m+1}a_{m+2}...a_{m+k}$ déterminés par le biais d'une fonction Ψ des m éléments d'information ; définie au préalable. La longueur d'un mot code est alors $n = m + k$, pour vérifier qu'un mot reçu $a_1a_2...a_ma_{m+1}...a_{m+k}$ appartient au code, on applique la fonction $f : \mathbb{F}_q^m \longrightarrow \mathbb{F}_q^{m+k}$à $a_1a_2...a_m$ on obtient le bits de redondance $b_{m+1}b_{m+2}...b_{m+k}$. Ensuite on compare cette grandeur aux éléments effectivement reçus $a_{m+1}a_{m+2}...a_{m+k}$. S'il y a coïncidence entre ces deux grandeurs, le mot reçu est un mot code, sinon on détecte une erreur.

Si q est de la forme $q = p^r$ avec p un nombre premier et r entier strictement positif, on sait qu'il existe un seule corps fini noté $G\mathbb{F}(q)$ ($G\mathbb{F}$ signifie le corps de Galois), constitué de q éléments. Une représentation de ce corps est obtenue en considérant les polynômes de degré inférieur ou égal à $r - 1$ et à coefficients dans le corps fini $G\mathbb{F}(q)$.

L'ensemble $\mathbb{F}_q^{m+k}$ de tous les mots possibles constitués de n éléments est un espace vectoriel sur $G\mathbb{F}(q)$. Son cardinal est q^n sa dimension est n. Considérons alors l'ensemble C des mots code (inclus dans $\mathbb{F}_q^{m+k}$) : Son cardinal est q^m(les q^mmots possibles résultant de la concaténation de m éléments).

Le but de ce chapitre est de définir les notions et les propriétés de bases permettant d'étudier la construction des codes Cortex et des codes Cortex auto-duaux de type I et de type II.

1.1.1 Généralités sur les codes linéaires

Définissons dans un premier temps les codes linéaires de deux grandes familles de codes :

a- Les codes en blocs : le codage et le décodage d'un bloc dépend uniquement des informations de ce bloc, par exemple :

* Code de Hamming
* Code cyclique
* Code de Golay

b- Les codes convolutionnels : le codage et le décodage d'un bloc dépend des informations d'autres blocs (généralement du bloc précédemment transmis).

Dans ce mémoire nous proposons de présenter seulement les codes en bloc.

Dans ce chapitre nous rappelons les définitions et les propriétés élémentaires de base sur les codes linéaires que nous utiliserons par la suite.

1.2 Codes en blocs

Un code est un ensemble de mots qui permet de représenter des messages dans le bits de leur protection. Les mots du code sont en générale plus longs que les messages qu'ils représentent pour ajouter une certaine redondance à ceux-ci (pour controler a fin de réstatier). Les symboles composant les mots du code sont tirés d'un alphabet précis .Un code est formellement défini comme suit.

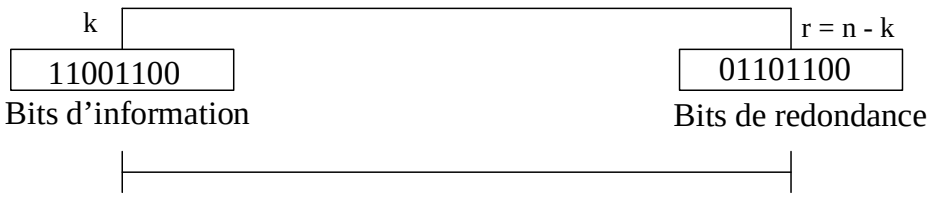
Définition 1.1 *Soient $\mathbb{F}$ un ensemble de cardinal q, dit alphabet, M et n deux entiers strictement positifs. On appelle code C sur l'alphabet $\mathbb{F}$ de longueur n,(bloc par bloc), toute partie $C \subset \mathbb{F}^n$ de cardinal*
card $(C) = M$.

Définition 1.2 *Si $\mathbb{F}_q$ est un corps fini et C est un sous-espace vectoriel de dimension k de $\mathbb{F}_q^n$, alors C est dit un code linéaire de longueur n et de dimension k qu'on note $C(n,k)$ et chaque élément de C est dit un mot de code.*

1.2.1 Code en bloc et redondance

Etant donné un code en bloc $C = C(n, k)$.

Un message est constitué d'un bloc de k symboles, appelés bit d'information. Pour cela, on adjoint un bloc de $r = n - k$ symboles supplémentaires dit de redondance. Ces symboles sont calculés à partir des bits d'information par l'intermédiaire d'une fonction f fixée à l'avance dite fonction du codage. Ils forment un bloc $v = f(u)$.



En concaténant information et redondance, on obtient les mot codes du code C. Dans un code en bloc C de cardinal M sur un corps $\mathbb{F}_q$, le taux d'information est $R = \frac{\log_q^M}{n} = \frac{\dim C}{\dim(\mathbb{F}_q^n)}$

1.2.2 Matrice génératrice d'un code linéaire

Une classe très importante est celle des codes linéaires, en raison des outils dont nous disposons pour manipuler les applications linéaires.

Nous considérons, dans ce qui suit l'alphabet $\mathbb{F}_q$ un corps fini de cardinal q ou q est une puissance d'un nombre premier p ($q = p^r$).

Par définition un codage linéaire est une application linéaire f de la forme :

$f : \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^n$

$m \longmapsto c = f(m)$

où :

$m = (m_1, m_2, ..., m_k)$ est l'information et $c = (c_1, c_2, ...c_n)$ le mot code associé.

Si on pose $\{e_1, ..., e_k\}$ et $\{\alpha_1, ...\alpha_n\}$ des bases respectivement pour $\mathbb{F}_q^k$ et $\mathbb{F}_q^n$ on a :

$$m = \sum_{i=1}^{k} m_i e_i$$

L'application f est linéaire, on a donc :

$$c = f(m) = \sum_{i=1}^{k} m_i f(e_i)$$

tels que les vecteurs $f(e_i)$ peuvent être exprimés dans la base $\{\alpha_1, ..., \alpha_n\}$ comme suit :

$$f(e_i) = \sum_{j=1}^{n} f_{ij}\alpha_j$$

c-à-d : $f(e_i) = (f_{i,1}, ..., f_{i,n-1})$ où $(1 \leq i \leq k)$.
Soit G la matrice dont les lignes sont les coordonnées de $f(e_i)$ dans la base $\{\alpha_1, ..., \alpha_n\}$.

$$G = \begin{bmatrix} f_{11} & f_{12} & . & . & . & f_{1n} \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & . \\ f_{k1} & f_{l2} & . & . & . & f_{kn} \end{bmatrix}$$

Cette matrice G à k lignes et n colonnes est appelée matrice génératrice du code.

Définition 1.4 *Une matrice génératrice d'un code linéaire $C(n, k)$ est une matrice d'ordre k dont les lignes forment une base de C.*

Matrice de contrôle

Etant donné un code linéaire $C = C(n, k)$ sur le corps $\mathbb{F}_q$. Supposons qu'on a un produit scalaire $\langle\ ,\ \rangle_{\mathbb{F}_q}$ sur $\mathbb{F}_q$. Ceci induit un produit scalaire $\langle\ ,\ \rangle$ sur $\mathbb{F}_q^n$ définis par : $\forall x, y \in \mathbb{F}_q^n$ tels que ; $x = (x_1, x_2, ..., x_n)$ et $y = (y_1, y_2, ..., y_n)$:

$$\langle x, y\rangle = \sum_{i=1}^{n} \langle x_i, y_i\rangle_{\mathbb{F}_q} = \sum_{i=1}^{n} x_i y_i$$

Considérons le sous espace orthogonal à C :

$$C^{\perp} = \{v \in \mathbb{F}_q^n : \langle v, c\rangle = 0, \forall c \in C\}$$

on obtient alors la définition suivante :

Définition 1.5 *$C^{\perp}$ est un code linéaire de dimension $n-k$ dont toute matrice génératrice H, est appelée matrice de contrôle de C .*

Théorème 1.1 [17] *Soit $C \subseteq \mathbb{F}_q^n$ un code linéaire $C(n,k)$ de matrice génératrice G et de matrice de contrôle H, alors on a :*

1/ $C = \{aG, a = (a_1, ...a_k) \in \mathbb{F}_q^k\}$.

2/ $H^t G = 0$.

3/ $C^{\perp} = \{x \in \mathbb{F}_q^n : xG^t = 0\}$.

4/ card $(C) = q^{\dim(C)}$.

5/ $C^{\perp}$ *est un code linéaire et* $\dim(C) + \dim(C^{\perp}) = n$.

6/ $\ker f = \{c \in C : cG^t = 0\}$.

Exemple 1.1 *a) Si on veut construire un code linéaire binaire $C(6,3)$ il faut choisir trois vecteurs linéairement indépendants de $\mathbb{F}_2^6$, ce qui donne une base de C supposons par exemple*

$$G_1 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

on a alors :

Message a :	$C_1 = \{v \in \mathbb{F}_2^6 : v = aG_1\}$
000	000000
001	110110
010	011101
011	101011
100	100101
101	010011
110	111000
111	001110

On cherche maintenant une matrice de contrôle de C. Soit $v = (v_1, ..., v_6) \in C^{\perp}$, alors $vG^t = (0,0,0)$

la résolution du système :

$$\begin{cases} v_1 + v_4 + v_6 = 0 \\ v_2 + v_3 + v_4 + v_6 = 0 \\ v_1 + v_2 + v_4 + v_5 = 0 \end{cases}$$

détermine :

$$H_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

b)

$$G_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Message a :	$C_2 = C(7,3)$
000	0000000
001	0011101
010	0111010
011	0100111
100	1110100
101	1101001
110	1001110
111	1010011

On a :

$$H_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

1.2.3 Code systématique

Etant donné un code linéaire $C = C(n,k)$.

Définition 1.6 *Si C possède une matrice génératrice sous la forme $G = [I_k \parallel P]$, on dit que C est un code systématique.*

Propsition 1.1 *Soit C un code systématique de matrice génératrice $G = [I_k \parallel P]$, alors $H = [-P^t \parallel I_{n-k}]$ est une matrice de contrôle.*
Preuve. Soit $G = [I_k \parallel P]$ avec :
I_k = matrice identité de taille $k \times k$
P = matrice de taille $k \times (n-k)$
$G^t = [I_k \parallel P^t]$
En effet si $u \in C^\perp$ on a $u = bH$ avec $b \in \mathbb{F}_q^{n-k}$ et de même façon :

$$\forall c \in C; c = aG \quad tel\ que;\ a \in \mathbb{F}_q^k$$

De plus, C et $C^{\perp}$ étant orthogonaux ; c-à-d :

$$\forall u \in C^{\perp} \; et \; \forall c \in C \;\; on \; a : \; \langle u, c \rangle = 0$$

alors :

$$(aG).(bH)^{\intercal} = 0 \Rightarrow aG.H^{\intercal}b^{t} = 0$$

Il faut donc satisfaire $GH^{\intercal} = 0$ ou encore $HG^{\intercal} = 0$

si :

$$G^{\intercal} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ 0 & 0 & 0 & 1 \\ P_{11} & P_{21} & . & P_{k1} \\ P_{12} & P_{22} & . & P_{k2} \\ . & & & . \\ . & & & . \\ . & & & . \\ P_{1n-k} & P_{2n-k} & . & P_{kn-k} \end{bmatrix}$$

Calculons le produit $HG^{\intercal}$:

$$HG^{\intercal} = \begin{bmatrix} -P_{11} & -P_{12} & . & . & . & -P_{1n-k} & 1 & 0 & 0 & 0 \\ -P_{21} & -P_{22} & . & . & . & -P_{2n-k} & 0 & 1 & 0 & 0 \\ . & & & & & & 0 & 0 & 1 & 0 \\ -P_{k1} & -P_{k2} & . & . & . & -P_{kn-k} & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ 0 & 0 & 0 & 1 \\ P_{11} & P_{21} & . & P_{k1} \\ P_{12} & P_{22} & . & P_{k2} \\ . & & & . \\ . & & & . \\ . & & & . \\ P_{1n-k} & P_{2n-k} & . & P_{kn-k} \end{bmatrix}$$

$$= \begin{bmatrix} -P_{11} + P_{11} & & & & \\ & -P_{22} + P_{22} & & & \\ & & . & & \\ & & & . & \\ & & & & . \\ & & & & -P_{kn-k} + P_{kn-k} \end{bmatrix}$$

$$= [0]$$

On a donc vérifié que :

$$H = \left[\begin{array}{cc} -P^t & I_{n-k} \end{array} \right]$$

■

1.3 Codes équivalents

Il existe des codes linéaires qui ne possèdent pas une forme systématique.

Exemple 1.2 *Le code* $C = \{000, 010, 001, 011\}$ *possède comme matrice génératrice possible :*

$$G_1 = \left[\begin{array}{ccc} 0 & 0 & 1 \\ 0 & 1 & 0 \end{array} \right], G_2 = \left[\begin{array}{ccc} 0 & 0 & 1 \\ 0 & 1 & 1 \end{array} \right], G_3 = \left[\begin{array}{ccc} 0 & 1 & 0 \\ 0 & 1 & 1 \end{array} \right]$$

Ce sont des matrices non systématiques. Par contre le code $C' = \{000, 100, 010, 110\}$, *obtenu du code* C *en permutant la première et la troisième composante, possède une forme systématique* $G' = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \end{array} \right]$

Exemple 1.3 $C_1 = \{000, 011, 022\} \subset C(3, 1)$
de matrice génératrice possible $G_1 = \left[\begin{array}{ccc} 0 & 2 & 2 \end{array} \right]$ *qui n'est pas systématique. Par contre le code* $C_1' = \{000, 102\}$, *obtenu du* C_1 *par permutation des deux premières composantes par 2, est de forme systématique.*
Ceci motive la définition des codes équivalents.

Définition 1.7 *Soient* C *et* C'*de codes linéaires de longueur* n *et de dimension* k *sur un corps premier* $\mathbb{F}_q$.*On dit que* C et C' *sont équivalents s'ils existent.*
1/ $\delta \in S_n$.
2/ l'application $\pi_i : \mathbb{F}_q \longrightarrow \mathbb{F}_q$
$\alpha \longmapsto \pi_i(\alpha) = \alpha_i \alpha \quad , \qquad \alpha \in \mathbb{F}_q^*$
telles que :
$C = (C_1, ..., C_n) \in C \Longrightarrow (\pi_1(C_{\delta(1)}), \pi_2(C_{\delta(2)}), ..., \pi_n(C_{\delta(n)})\ ) \in C'$

Remarque 1.1 *Il est clair que pour les codes binaires,* $\forall i \ \ \pi_i = 1$

Proposition 1.2 *Tout code linéaire* $C = C(n, k)$ *est équivalent à un code systématique.*

1.3.1 Distance minimale d'un code linéaire

Nous considérons dans cette partie uniquement les codes linéaires et nous introduisons dans un premier temps la notion de distance de Hamming. Etant donné un code $C = C(n, k)$.

Définition 1.8 *On appelle distance de Hamming $d(x, y)$ entre deux mots de code x et y de $\mathbb{F}_q^n$ le nombre de composantes de même position qui sont différents.*

Définition 1.9 *Le poids de Hamming d'un mot x est sa distance au mot code nul : $\omega(x) = d(x, 0)$*

Remarque 1.2 *L'ensemble $\mathbb{F}_q^n$ muni de la distance de Hamming est un espace métrique.*

Définition 1.10 *On appelle distance minimale d'un code linéaire $C = C(n, k)$, la plus petite distance entre deux mots codes distincts du code $d = \min\{d(x, y) \ , \forall \ x, y \in C \ / \ x \neq y\}$.* Un code linéaire est donc caractréser des trois paramétres la longueur, la dimension et la distance minimale *on note $C = C(n, k, d)$*

Théorème 1.2 [13] (Borne de Singleton)
Soit C un code linéaire de paramètres (n, k, d), alors : $d \leq n - k + 1$.

1.4 Les codes de Hamming

La famille des codes de Hamming est une famille importante des codes linéaires, leur décodage est simple et ils sont très utilisés en pratique. Il en existe sur tout corps fini. Ils ont été introduits par Golay en 1949 et par Hamming en 1950.

Définition 1.11 *Un code binaire dont les colonnes de la matrice de contrôle sont tous les $m-$uplets non nuls possibles est dit code de Hamming.*

Proposition 1.3 *Soit H_m le code de Hamming, alors H_m est de type $C(2^m - 1, 2^m - 1 - m, 3)$.*

1.4.1 Construction du code de Hamming systématique

Pour déterminer m on se propose de construire un code linéaire systématique. On sait que ses paramètres doivent vérifier la relation $2^m = n+1$. Pour chaque valeur de n supérieure ou égale à 3, on va chercher s'il existe ou non un entier m vérifiant $2^m = n + 1$, on obtient ainsi les solutions suivantes :

n	m	$k = n - m$
3	2	1
4		
5		
6		
7	3	4

1.5 Code de Golay

Les codes binaires de Golay forment une famille importante du code en bloc.

Définition 1.12 *Le code de Golay est un* $(23, 12, 7)$ *code linéaire engendré par :*

$$G = \begin{bmatrix} & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ I_{12} & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

1.6 Les codes étendus

Il est intéressant de trouver des nouveaux codes à partir de codes déjà connus. C'est le cas des codes étendus.

Définition 1.13 *Soit un code linéaire* $C(n, k, d)$*. On considère le code linéaire étendu* $C'(n + 1, k)$*de distance minimal* d *ou* $d + 1$*, où chaque mot*

du code $v' = (v_1, ..., v_{n+1})$*est tel que* $v = (v_1, ..., v_n) \in C$ *et* $v_{n+1} = f(v)$ *pour une certaine fonction* f*.*

Remarque 1.3 [19] *Le sous-espace formé par les mots du code étendu est isomorphe au sous-espace du code d'origine, mais la distance minimale augmente, ou reste inchangée.*

Exemple 1.4 *On construit une matrice de contrôle du code étendu sans modifier les caractéristiques du code initial, et en assurant un contrôle de parité sur les données pour le code de Hamming* $C(7, 4)$ *on a :*

$$\hat{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Les codes de Hamming étendus ont une longueur $n = 2^m$ *bits.*

Pour le code de Golay étendu le C (23,12) code de Golay peut étre prolongé en ajoutant un contrôle de parité à chaque mot de code pour former le C (24,12) code de Golay étendu engendré par $G = [I_{12} \| B]$ *telle que :*

$$B = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Le code de Golay étendu est de distance minimale $d = 8$ et de rendement $\frac{1}{2}$.

Chapitre 2

Les codes auto-duaux et leurs paramètres

2.1 Les codes auto-duaux et leurs propriétés

En préambule à cette partie, nous rappelons quelques définitions et propriétés les plus importantes que vérifient les codes auto-duaux.

Définition 2.1 *Soit $C = C(n,k)$ un code sur $\mathbb{F}_q$ où $\mathbb{F}_q$ est un corps fini muni d'un produit scalaire $\langle\ ,\ \rangle_{\mathbb{F}_q}$.*
1/ On dit que C est auto-orthogonal si $C \subset C^{\perp}$.
2/ On dit que C est auto-dual si $C = C^{\perp}$.

Proposition 2.1 [16] *On à les relations suivantes pour tout code C linéaire sur $\mathbb{F}_q$:*
1/ $|C|\ |C|^{\perp} = |\mathbb{F}_q|^n$.
2/ $(C^{\perp})^{\perp} = C$.
3/ Si C est auto-dual, alors $n = 2k$.

Proposition 2.2 [19] *Un code linéaire $C = C(n,k)$* systématique *de matrice génératrice $G = [I_k \| P]$ est auto-dual ssi $PP^t = -I_k$.*
Preuve. *$G = [I_k \| P]$, comme $G = H$ alors :*

$$\begin{aligned} GG^t = 0_n &\iff [I_k \| P] \begin{bmatrix} I_k \\ P^t \end{bmatrix} = 0_n \\ &\iff I_k + PP^t = 0_k \\ &\iff PP^t = -I_k \ \blacksquare \end{aligned}$$

Définition 2.2 Soit *$C = C(n,k)$ un code auto-dual sur $\mathbb{F}_2$ dont tous les poids sont congrus à 0 mod 2 et pour lesquels il existe au moins un mot de*

poids ne congruant pas à 0 mod 4 est de type I (i.e. simplement pair). Si tous les mots d'un code binaire C sont de poids congrus à 0 mod 4, C est dit de type II (i.e .doublement pair).

Définition 2.3 *Un code auto-dual sur $\mathbb{F}_3$ (i.e. ternaire) est dit de type III.*

Définition 2.4 *Un code auto-dual hermitien sur $\mathbb{F}_4$ est dit de type IV.*

2.2 Classification des codes auto-duaux

Le problème de la classification des codes à été le sujet d'étude de nombreux auteurs **[1],[2],[4],[5],[10],[14] ,[18]** et **[22].**

Théorème 2.1 **[4]** *Soit C un code binaire auto-dual $(n, \frac{n}{2}, d)$. Alors :*
i) $d \leq 2[\frac{n}{8}] + 2$.
ii) si C est de type II, Alors $d \leq 4[\frac{n}{24}] + 4$.

Un code atteint une borne si l'égalité se maintient en cette borne. Ward **[4]** a prouvé que la borne dans le théorème 2.1 (i) est atteinte seulement pour $n = 2, 4, ..., 22, 24$. Cette borne est renforcée pour les codes de type I en 1990 **[4]**.

Théorème 2.2 **[4]** *Soit $C = C(n, \frac{n}{2}, d)$ un code de type I avec $n \neq 2, 12, 22$ ou 32. Alors : $d \leq 2 \left\lfloor \frac{n+6}{10} \right\rfloor$.*

Théorème 2.3 **[4]** *Soit $C = C(n, \frac{n}{2}, d)$ un code auto-dual. Alors :*
1/ Si $n \ncong 22[\mathrm{mod}\, 24] \Rightarrow d \leq 4 \left\lfloor \frac{n}{24} \right\rfloor + 4$.
2/ Si $n \equiv 22[\mathrm{mod}\, 24] \Rightarrow d \leq 4 \left\lfloor \frac{4}{24} \right\rfloor + 6$.
3/ Si $n \equiv 0[\mathrm{mod}\, 24]$ et $d = 4 \left\lfloor \frac{n}{24} \right\rfloor + 4$ alors le code et de type II.

Définition 2.5 *Un code auto-dual qui, selon son type, atteint une des bornes de ci-dessus, est appelé extrémal. Un code est optimal si son poids minimum est le plus grand connu pour sa longueur et sa dimension.*

2.3 Les codes de type I et de type II de longueur 2 à 36

Dans le tableau suivant, nous présentons les codes auto-duaux de type I et de type II de longueur n avec $2 \leq n \leq 32$. Le nombre de code de type

I et de type II non équivalents cité sous $\#_1$ et $\#_2$, respectivement. Dans le tableau "$d_{\max,I}$ " respectivement "$d_{\max,II}$" le plus grand poids minimum pour lequel un code de type I, respectivement de type II, existe. L'exposant "E" indique que le code est extrémal, l'exposant "O" indique que le code n'est pas extrémal mais optimal Ainsi le nombre de code de type I et de type II non équivalents des ces poids minimal élevés sont cités sous " $\#_{\max,I}$" et " $\#_{\max,II}$", respectivement. Une classification complète de tous les codes de type I de longueur 34 et 36 n'est pas encore connue [**4**].

n	$\#_I$	$\#_{II}$	$d_{\max,I}$	$\#_{\max,I}$	$d_{\max,II}$	$\#_{\max,II}$
2	1		2^O	1		
4	1		2^O	1		
6	1		2^O	1		
8	1	1	2^O	1	4^E	1
10	2		2^O	2		
12	3		4^E	1		
14	4		4^E	1		
16	5	2	4^E	1	4^E	2
18	6		4^E	2		
20	16		4^E	7		
22	25		6^E	1		
24	46	9	6^E	1	8^E	1
26	103		6^O	1		
28	261		6^O	3		
30	731		6^O	13		
32	3210	85	8^E	3	8^E	5
34	?		6^O	938		
36	?		8^E	41		

Classification des codes de type I et de type II de longueur $2 \leq n \leq 32$

*Tableau*1

Pour les longueurs $n \geq 34$, nous suggérons les références suivantes : **[1],[2], [4],[5],[6],[8],[9],[11]** et **[14]**.

2.3.1 L'énumurateur de poids

L'énumurateur de poids W du code C est le polynôme $W_c(X,Y)$ défini par :

$$W_c(X,Y) = \sum A_i X^{n-i} Y^i$$

où A_i est le nombre des mots codes dans C de poids de Hamming i,.
$A_0, ..., A_n$ sont des distributions de poids.

Théorème 2.4 [22]
1. L'énumérateur des poids de tout code auto-dual binaire de type I est une combinaison linéaire de W_1 et W_2 où :

$$W_1(x,y) = x^2 + y^2 \text{ et } W_2(x,y) = x^8 + 14x^4y^4 + y^8$$

2. L'énumérateur des poids de tout code auto-dual binaire de type II est une combinaison linéaire de W_3 et W_4 où :

$$W_3(x,y) = x^{24}+759x^{16}y^8+2576x^{12}y^{12}+759x^8y^{16}+y^{24} \text{ et } W_4(x,y) = x^4+8xy^3$$

3. L'énumérateur des poids de tout code auto-dual ternaire de type III est une combinaison linéaire de W_4 et W_5 où :

$$W_4(x,y) = x^4 + 8xy^3 \text{ et } W_5(x,y) = x^{12} + 264x^6y^6 + 440x^3y^9 + 24y^{12}$$

Remarque 2.1 *Si le code est extrémal, son polynôme énumérateur est unique.*

Plusieurs théorèmes sont etablis donnant la construction des codes auto-duaux **[22].**

2.4 Les codes ternaires

Les codes de type III existent seulement pour des longueurs multiples de 4 et ont seulement des mots codes de Hamming de poids multiple de 3. De plus, si C est un code ternaire avec tous les mots codes de poids multiple de 3, alors C est auto-orthogonal. Le poids minimum de Hamming est donné comme suit.

Théorème 2.5 [10] *Si C est un code de type III de paramètres $[n, \frac{n}{2}, d]$, alors : $d \leq 3\left\lfloor \frac{n}{12} \right\rfloor + 3$.*

Remarque 2.2 *Les codes auto-duaux extrémaux de type III n'existent pas pour les longueurs*

$n = 72, 96, 120,$ *et tous* $n \geq 144$**[4]**.

Les codes de type III de longueur n ont été classifiés complètement seulement pour $n \leq 24$. Le tableau 2 suivant donne l'état actuel pour le nombre $4 \leq n \leq 72$. Les codes auto-duaux de type III inéquivalents sont énumérés sous "#".

"$d_{\max}$" est le plus grand poids minimal pour lequel un code de type III existe, et "$\#_{\max}$" est le nombre de tels codes excepté la longueur 68,

$d = 3\left\lfloor \frac{n}{12} \right\rfloor + 3$.Il y a deux codes de type III extrémaux de paramètres (24, 12).

On donnent maintenant le tableau de classification des codes auto-duaux ternaire.

n	#	$d_{\max}$	$\#_{\max}$
4	1	3	1
8	1	3	1
12	3	6	1
16	7	6	1
20	24	6	6
24	$\geq$140	9	2
28	?	9	≥ 32
32	?	9	≥ 239
36	?	12	≥ 1
40	?	12	≥ 20
44	?	12	≥ 8
48	?	15	≥ 2
52	?	15	≥ 1
56	?	15	≥ 1
60	?	18	≥ 2
64	?	18	≥ 1
68	?	15 ou bien 18	≥ 1 ou bien ?
72	?	18	≥ 1

Classification des codes de type III de longueur $4 \leq n \leq 72$

Tableau 2

2.5 Les codes auto-duaux sur $\mathbb{F}_4$

IL y a plusieurs familles des codes auto-duaux sur $\mathbb{F}_4$. Ces familles utilisant différents produits scalaires et ont des différentes notions sur l'équivalence. Nous présentons :

2.5.1 Les codes auto-duaux Hermitiens sur $\mathbb{F}_4$

Le produit scalaire Hermitien sur $\mathbb{F}_4^n$ est donné par :

$$\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i^2$$

où $x, y \in \mathbb{F}_4^n$ avec $x = (x_1, x_2, ..., x_n)$ et $y = (y_1, y_2, ..., y_n)$.

Le code Hermitien auto-dual est appelé aussi un code de type IV. Tous les mots codes dans le code de type IV ont un poids de Hamming pair. Les codes de type IV existent pour toutes les longueurs pairs. Il y a une borne sur le poids minimum de Hamming qui est donné dans le théorème de ci-dessous.

Théorème 2.6 [3] *Si C est un code de paramètres $[n, \frac{n}{2}, d]$ de type IV, alors : $d \leq 2\left\lfloor \frac{n}{6} \right\rfloor + 2$.*

Aucun code extrémal de type IV existent pour les longueurs $n = 102, 108, 114, 120, 122, 126, 128$, et pour $n \geq 132$ **[4]**.

Les codes auto-duaux de type IV ont été complètement classifiés pour des longueurs $2 \leq n \leq 16$; tous les codes extrémaux, ont été déterminés pour les longueurs 18 et 20. Le tableau 3 de ci- dessous donne ce qui est connu pour la longueur 40. Le nombre des codes de type IV inéquivalents est énuméré sous "#". Et aussi "$d_{\max}$" et "$\#_{\max}$" indiquent le plus grand poids minimal pour lequel un code de type IV existe et le nombre de tels codes.

L'existence des codes extrémaux de longueur 32 à 40 est inconnue.

n	#	$d_{\max}$	$\#_{\max}$
2	1	2^E	1
4	1	2^E	1
6	2	4^E	1
8	3	4^E	1
10	5	4^E	2
12	10	4^O	5
14	21	6^E	1
16	55	6^E	4
18	?	8^E	1
20	?	8^E	2
22	?	8^E	≥ 46
24	?	8^O	≥ 217
26	?	8^O	≥ 49
28	?	10^E	≥ 3
30	?	12^E	≥ 1
32	?	10 ou bien 12^E	≥ 19 ou bien ?
34	?	10 ou bien 2^E	≥ 105 ou bien ?
36	?	12 ou bien 2^E	≥ 1 ou bien ?
38	?	12 ou bien 2^E	≥ 1 ou bien ?
40	?	12 ou bien 2^E	≥ 1 ou bien ?

Classification des codes de type IV de longueur $2 \leq n \leq 40$

Tableau 3

Chapitre 3

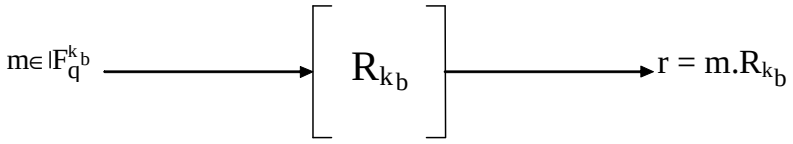
Les codes Cortex et la construction de ces codes à base auto-dual

3.1 Les codes Cortex

Cette famille des codes en blocs est issue des travaux conjoints de Carlach et Vervoux. Cette construction a compté beaucoup sur les travaux de Olocco, Tilich et Otmani. Leurs études concernaient plus particulièrement les aspects théoriques de la construction avec ses différentes propriétés**.** **[3],[13],[15]** et **[16]**.

Soit B un code linéaire de dimension k_b, de rendement $\frac{1}{2}$ sur le corps $\mathbb{F}_q$ et ayant une matrice génératrice sous forme systématique notée $[I_{k_b} \| \ R_{k_b}]$ avec R_{k_b} une matrice carrée d'ordre k_b.

Pour tout $m \in \mathbb{F}_q^{k_b}$ on a :



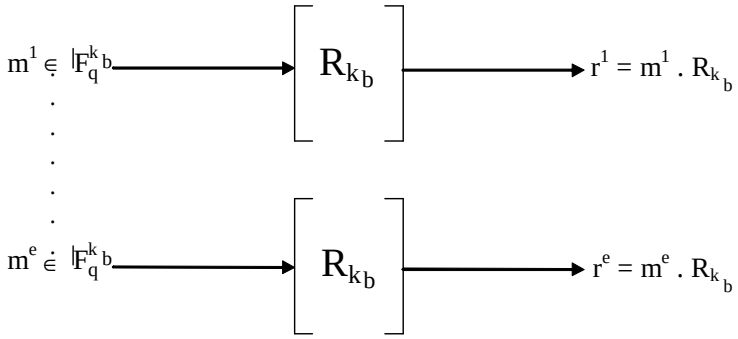
Le vecteur (m . r) ∈ B

3.1.1 Construction des codes de plus grandes longueurs

Soit k un entier non nul multiple de k_b et posons $k = ek_b$ pour tout vecteur m de $\mathbb{F}_q^k$ on forme les vecteurs

$$m^{i+1} = (m_{ik_b+1}, ..., m_{(i+1)k_b}) \text{ où } i \in \{0, ...e-1\}$$

L'encodage parallèle tel que : $e \geq 1$ est donné par :



On construit alors le vecteur (m^1, …,m^e ,r^1,…,r^e)

Si on pose r de $\mathbb{F}^k$ vérifiant $r^{i+1} = m^{i+1}R_{k_b}$

On considère la concaténation parallèle des vecteurs m et r pour obtenir l'ensemble C_k.

3.1.2 Encodage parallèle du point de vue matriciel

Pour tout $k = ek_b$ et $m \in \mathbb{F}_q^k : m = (m^1, ..., m^e)$ avec $m^i \in \mathbb{F}_q^{k_b}$:

On définit la matrice :

$$R = \begin{bmatrix} R_{k_b} & & & \\ & . & 0 & \\ & & . & \\ & 0 & & . \\ & & & R_{k_b} \end{bmatrix}$$

On a le vecteur $r = m.R$

Définition 3.1 *L'ensemble* $C_k = \{(m : r) \ / \ m \in F_q^k \text{ et } r = m.R\}$.

est un code linéaire équivalent au code B^e, *de dimension* $k = e.k_b$, *de rendement* $\frac{1}{2}$, *de matrice génératrice* $G = [I_k \parallel R]$ *où* R *est dite partie redondante du code* C_k *donnée par le produit de Kronecker de* I_e *par* R_{k_b}(*i.e* $R = I_e \otimes R_{k_b}$).

Définition 3.2 *Soient* $A = (a_{ij})_{\substack{1\leq i\leq n \\ 1\leq j\leq n}}$ *et* $B = (b_{ij})_{\substack{1\leq i\leq q \\ 1\leq j\leq p}}$ *deux matrices, la matrice résultante du produit de Kronecker de* A *et* B*, notée* $A \otimes B$*, est la matrice de dimension* $mp \times nq$ *donnée par la relation* $(a_{ij}B)$:

$$A \otimes B = \begin{bmatrix} a_{11}B & . & . & . & a_{1n}B \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ a_{m1}B & . & . & . & a_{mn}B \end{bmatrix}$$

Remarque 3.1 *L'encodage parallèle fait augmenter la longueur et la dimension mais n'améliore en rien la distance du code obtenu.*

Avant de définir la construction Cortex, on introduit quelques notations

Définition 3.3 *Soit* $\pi \in S_k$*, On définit la matrice carrée d'ordre* k*, notée* Π*, par :*

$\Pi = (h_{ij})$ *où* $h_{ij} = \delta_{\pi(i)j}$ *telque :* $\delta_{ij} = \begin{cases} 1 & si \ \ i = j \\ 0 & si \ \ non \end{cases}$

Exemple 3.1 *Si* $\pi \in S_5$ *définie par :* $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix}$ *alors :*

$$\Pi = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Remarque 3.2 Π *est obtenue de* I_k *en appliquant la permutation* π *sur les colonnes de* I_k*.*

3.1.3 L'encodage parallèle des codes Cortex

Le processus d'encodage parallèle systématique se fait de la façon suivante :

Dans un premier temps en transforme le vecteur de bits d'information à encoder

$$m = (m_1^{(1)}, ..., m_k^{(1)}) = (m^1, ...m^e) \in \mathbb{F}_q^k$$

où :

$$m^{i+1} = (m_{ik_b+1}, ..., m_{(i+1)k_b}) \ pour \ i \in \{0, ..., e-1\}$$

en un vecteur de bits de redondance

$$r = (r_1^{(1)}, ..., r_k^{(1)}) = (r^1, ..., r^e) \in \mathbb{F}_q^k$$

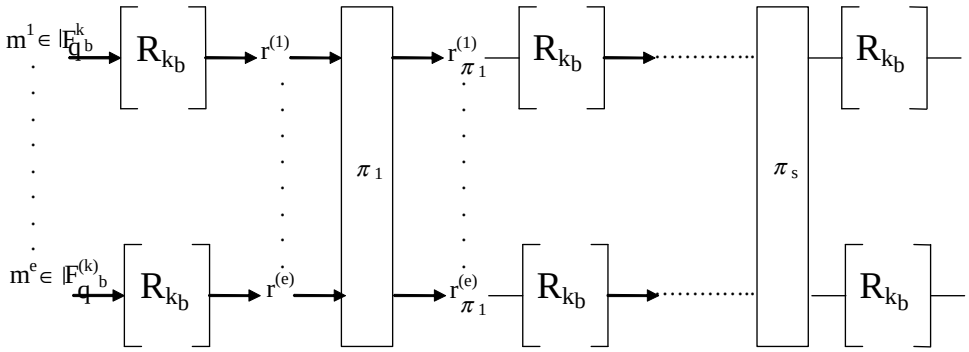
où :

$$r^{i+1} = m^{i+1} R_{k_b} \ pour \ i \in \{0, ..., e-1\}$$

Les éléments de ce vecteur redondance modifient alors une première permutation π_1 pour obtenir :

$$m^{(2)} = (r^{(1)}_{\pi_1(1)}, ..., r^{(1)}_{\pi_1(k)})$$

En pratique ainsi pour $(s+1)$ étages et les s permutations qui composent la structure. Le mot code correspond à la concaténation du vecteur m de bits d'information avec le vecteur r de bits de redondance.

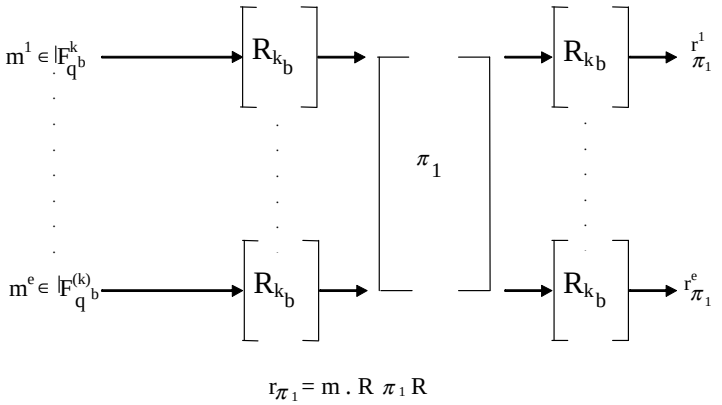


Codeur « cortex »

pour tout $m \in \mathbb{F}_q^k$:

$$R^{(s)}(m) = m \cdot R\Pi_1 ... \Pi_s R$$

Si on pose $s = 1$:
Soit π_1 une permutation quelconque de S_k si $m \in \mathbb{F}_q^k$ on a :



L'ensemble des vecteurs (m, r_{π_1}) forme un code de dimension k.

Ces codes sont des codes linéaires correcteurs d'erreurs en blocs systématiques de longueur $n = 2k$ ayant k bits d'information. Ces codes se présentent sous la forme de concaténation série et parallèle de code de base B. Les codes de base sont regroupés en étages séparés par des permutations non nécessairement identiques.

Remarque 3.3 *Le code ainsi construit n'est pas forcément equivalent à* B^e.

Définition 3.4 Des codes Cortex (Carlach-Vervoux 1998)
Soient $B = C(2k_b, k_b) \subset \mathbb{F}^{2k_b}$ *un code de rendement* $\frac{1}{2}$, *ayant une matrice génératrice*

$$G_{k_b} = [I_{k_b} \quad \| \quad R_{k_b}]$$

$\forall e \in \mathbb{N}^*$ *posons* $k = ek_b$. *Pour tout* $\Pi = \{\pi_{1,}..., \pi_s\}$ *de* S_k, *l'ensemble*

$$C_k(B, \Pi) = \left\{(m : R^{(s)}(m))/m \in \mathbb{F}^k \text{ et } R^{(s)}(m) = m \cdot R\Pi_1...\Pi_s R\right\}$$

est dit code Cortex de base B *suivant la suites de permutations* $\Pi = \{\pi_1, ..., \pi_s\}$, *où* $R = I_e \otimes R_{k_b}$.

Remarque 3.4 *Le code $C_k(B, \Pi)$ est un code linéaire de dimension k, de rendement $\frac{1}{2}$ et de matrice génératrice*

$$G = [\ I_k \parallel\ R\Pi_1 ... \Pi_s R]$$

Remarque 3.5 *Le code C_k est un code Cortex obtenu suivant la suite vide de permutation.*

Exemple 3.2 [15] *Soit $B = C(4, 2)$ de matrice génératrice* $G_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$ *avec $k_b = 2$ et la partie redondante de G_2 est* $R_2 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$. *Pour construire un code Cortex, on prend une suite quelconque de permutation $\Pi = \{\pi_1, \pi_1\}$ et $e = 2$ on a : $\pi_1 \in S_4$ avec* $\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix}$

La matrice génératrice de ce code est

$$G_4 = [\ I_4 \parallel\ R\Pi_1 R\Pi_2 R]$$

où

$$R = I_2 \otimes R_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

et $k = 4$
donc :

$$R\Pi_1 R\Pi_2 R = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

est la partie redondante de la matrice génératrice du code Cortex $C_4(C(4, 2), \{\pi_1, \pi_1\})$.
C'est le code de Hamming étendu de matrice génératrice :

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

C'est un code auto-dual de type II de distance minimal $d = 4$.

Exemple 3.3 *Si on prend le code de base $B = C(6,3)$ sur $\mathbb{F}_5$ de matrice génératrice :*

$$G_3 = \begin{bmatrix} 1 & 0 & 0 & 2 & 2 & 4 \\ 0 & 1 & 0 & 3 & 1 & 3 \\ 0 & 0 & 1 & 4 & 2 & 2 \end{bmatrix}$$

Pour la suite de permutation $\Pi = \{\pi_1, \pi_2\}$ avec $e = 2$ et
$\pi_1(z) = (z+1)(\mathrm{mod}\, 6)$ pour tout $z \in \mathbb{Z}_6 = \{1,2,3,4,5,6\}$.
si

$$\pi_1 = \pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 4 & 5 & 6 & 1 \end{pmatrix}$$

$$\Pi_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad et \quad R = \begin{bmatrix} 2 & 2 & 4 & 0 & 0 & 0 \\ 3 & 1 & 3 & 0 & 0 & 0 \\ 4 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 4 \\ 0 & 0 & 0 & 3 & 1 & 3 \\ 0 & 0 & 0 & 4 & 2 & 2 \end{bmatrix}$$

de sorte :

$$R\Pi_1 R\Pi_1 R = \begin{bmatrix} 3 & 3 & 3 & 1 & 4 & 0 \\ 3 & 2 & 2 & 4 & 0 & 4 \\ 3 & 2 & 3 & 0 & 4 & 4 \\ 1 & 4 & 0 & 3 & 3 & 3 \\ 4 & 0 & 4 & 3 & 2 & 2 \\ 0 & 4 & 4 & 3 & 2 & 3 \end{bmatrix}$$

Le code $C_6(C(6,3), \{\pi_1, \pi_2\}) = C(12,6,4)$ est un code Cortex ayant comme matrice génératrice

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 1 & 4 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 3 & 2 & 2 & 4 & 0 & 4 \\ 0 & 0 & 1 & 0 & 0 & 0 & 3 & 2 & 3 & 0 & 4 & 4 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 4 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 4 & 0 & 4 & 3 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 4 & 4 & 3 & 2 & 3 \end{bmatrix}$$

$C_6(C(6,3), \{\pi_1, \pi_1\})$ est un code auto-dual sur $\mathbb{F}_5$.

Remarque 3.6 *On peut généraliser la construction des codes Cortex sur un anneau.*

Exemple 3.4 *On choisit comme code de base l'octacode O_8 défini par la matrice génératrice :*

$$G_{4^4} = \begin{bmatrix} 1 & 0 & 0 & 0 & 2 & 1 & 3 & 1 \\ 0 & 1 & 0 & 0 & 3 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 3 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 \end{bmatrix}$$

Le code construit à partir de O_8 sur $\mathbb{Z}_4$ suivant la suite de permutation $\Pi = \{\pi_1, \pi_2\}$ tel que :

$$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 1 & 5 & 2 & 6 & 3 & 9 & 4 & 10 & 7 & 11 & 8 & 12 \end{pmatrix}$$

$$\pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 2 & 6 & 1 & 5 & 4 & 10 & 3 & 9 & 8 & 12 & 7 & 11 \end{pmatrix}$$

est un code auto-dual Cortex de paramètres $[24, 4^{12}, 12]$ sur $\mathbb{Z}_4$ ayant comme matrice génératrice

$$G_{4^{12}} = \begin{bmatrix} & 2 & 0 & 1 & 0 & 2 & 1 & 3 & 1 & 1 & 0 & 1 & 1 \\ & 1 & 0 & 0 & 0 & 3 & 2 & 1 & 1 & 2 & 3 & 1 & 3 \\ & 1 & 0 & 0 & 1 & 1 & 1 & 2 & 0 & 3 & 2 & 1 & 1 \\ & 0 & 1 & 3 & 2 & 1 & 0 & 1 & 1 & 1 & 1 & 2 & 0 \\ & 1 & 1 & 2 & 0 & 0 & 3 & 2 & 3 & 1 & 1 & 3 & 0 \\ & 2 & 3 & 1 & 3 & 2 & 1 & 2 & 0 & 3 & 1 & 2 & 3 \\ I_{12} & 1 & 2 & 3 & 3 & 0 & 0 & 1 & 2 & 1 & 1 & 1 & 2 \\ & 0 & 3 & 3 & 1 & 3 & 2 & 3 & 0 & 0 & 2 & 3 & 1 \\ & 0 & 2 & 3 & 1 & 1 & 3 & 0 & 3 & 2 & 3 & 3 & 0 \\ & 3 & 1 & 2 & 3 & 0 & 2 & 1 & 3 & 3 & 0 & 0 & 1 \\ & 3 & 3 & 3 & 2 & 3 & 1 & 2 & 3 & 0 & 0 & 0 & 3 \\ & 1 & 3 & 0 & 3 & 3 & 3 & 3 & 2 & 0 & 3 & 0 & 2 \end{bmatrix}$$

3.2 Code auto-dual Cortex à base auto-dual

Dans cette partie nous nous intéressons à l'étude des codes Cortex auto-dual à base d'un code auto-dual et à partir d'une suite de permutations quelconque Π de S_k, ainsi qu' à l'équivalence des codes Cortex et la construction des codes Cortex à base du code de Hamming étendu.

3.2.1 Auto-dualité

Théorème 3.1[15] *Soit B un code auto-dual sur $\mathbb{F}_q$. Alors :*

1/ Si $\mathbb{F}_q$ est de caractéristique 2, alors quelque soit Π de S_k, le code $C_k(B,\Pi)$ est auto-dual.

2/ Si $\mathbb{F}_q$ est de caractéristique différente de 2, le code $C_k(B,\Pi)$ pour Π une suite de permutations de S_k est auto-dual ssi card $(\Pi) \equiv 0[2]$.

Preuve. Etant donné le code C_k de matrice génératrice

$$G_{k_b} = [I_{k_b} \parallel R_{k_b}]$$

on a alors :

$$R_{k_b}.R_{k_b}^t = -I_{k_b}$$

La matrice génératrice de $C_k(B,\Pi)$ est :

$$G = [\ I_k \parallel R\Pi_1 R\Pi_2 ... \Pi_s R]$$

où

$$R = I_e \otimes R_{k_b}$$

comme :

$$R^t = (I_e \otimes R_{k_b})^t = I_e^t \otimes R_{k_b}^t$$

alors :

$$\begin{aligned} RR^t &= (I_e \otimes R_{k_b}).(I_e \otimes R_{k_b})^t \\ &= I_e \otimes R_{k_b} \\ &= I_e \otimes (-I_{k_b}) \\ &= -I_k \end{aligned}$$

d'où :

$$\begin{aligned} &(R\Pi_1 R\Pi_2 ... \Pi_s R)(R\Pi_1 R .... \Pi_s R)^t \\ &\qquad = R\Pi_1 R\Pi_2 ... \Pi_s R R^t \Pi_s^t ... \Pi_1^t R^t \\ &\qquad = (-1)^{s+1} I_k. \ \blacksquare \end{aligned}$$

Remarque 3.7 *Il se peut qu'il existe des codes Cortex auto-duaux à base non auto-dual, le code de Hamming étendu est un exemple.*

3.2.2 Groupe de permutations

Etant donné :
Un code Cortex C, i.e. $C_k(B,\Pi)$, notons par $\mathbb{N}_k$ l'ensemble $\{1,...,k\}$.
Pour tout $\pi \in S_k$, On définit :

$$\begin{aligned} \pi : \mathbb{F}^k &\longrightarrow \mathbb{F}^k \\ f = (f_1,...,f_k) &\longmapsto \pi(f) = (f_{\pi^{-1}(1)}\ ,...,f_{\pi^{-1}(k)}). \end{aligned}$$

Notation 3.1 *Pour toute suite de permutations* Π *de* S_k*, on note par* $G_k(B,\Pi)$ *le groupe de permutations de* $C_k(B,\Pi)$ *et* G_k *le groupe de permutations de* C_k*.*

Notation 3.2 *Pour tout sous groupe* H *de* S_{2k}*, Notons par :*
$H^* = \{(\delta_1,\delta_2) \in H \ / \ \delta_1$ *agit sur les* k *premiers éléments et* δ_2 *sur les* k *derniers*$\}$

Remarque 3.8 *On peut définir* H^* *par :*
$H^* = \{\delta \in H \ / \ \delta = \delta_1 \circ \delta_2 = \delta_2 \circ \delta_1$ *tel que* $\forall i \in \mathbb{N}_k \ , \delta_1(i+k) = i+k$ *et* $\delta_2(i) = i\}$.

Comment construire un mot code de $C_k(B,\Pi)$?
Prenons $f^0 \in \mathbb{F}_q^k$
Rappelons que :

$$C_k = \{(m : mR)/m \in \mathbb{F}_q^k \ et \ R = I_e \otimes R_{k_b}\}$$

Posons
$f^1 \in \mathbb{F}_q^k$ tell que : $(f^0, f^1) \in C_k$
$f^2 \in \mathbb{F}_q^k$ tell que : $(\pi_1 f^1, f^2) \in C_k$
$\vdots$
$\vdots$
$\vdots$
$f^{s+1} \in \mathbb{F}_q^k$ tell que : $(\pi_s f^s, f^{s+1}) \in C_k$
le vecteur $(f^0, f^{s+1}) \in C_k(B,\Pi)$ est un mot code de $C_k(B,\Pi)$.

Proposition 3.1 *Soient* $\sigma_1, ..., \sigma_{s+1} \in G_k^*$ *telles que :*

$$\forall j \in \{1, ..., s\} \Rightarrow \sigma_{j+1,1} = \pi_j \circ \sigma_{j,2} \circ \pi_j^{-1}$$

Alors : $(\sigma_{1,1}, \sigma_{s+1,2})$ *appartient au groupe de permutations de* $C_k(B,\Pi)$.
Preuve. On considère l'ensemble de vecteurs $f^0, ..., f^{s+1}$ de $\mathbb{F}_q^k$ qui vérifie la relation suivante :

$$\left.\begin{array}{c} (f^0, f^1) \in C_k \\ (\pi_1 f^1, f^2) \in C_k \\ (\pi_2 f^2, f^3) \in C_k \\ . \\ . \\ . \\ (\pi_s f^s, f^{s+1}) \in C_k \end{array}\right\} \Rightarrow (f^0, f^{s+1}) \in C_k(B,\Pi) \qquad (*)$$

On a :

$(\sigma_{1,1}f^0, \sigma_{1,2}f^1) \in C_k$
de plus :

$$(\pi_1 f^1, f^2) \in C_k \Rightarrow (\sigma_{2,1}\pi_1 f^1, \sigma_{2,2} f^2) \in C_k$$

d'aprés les hypothèses (*) on a :
$(\pi_1\sigma_{1,2}f^2, f^3) \in C_k$ et :

$$\begin{aligned}(\pi_2 f^2, f^3) \in C_k &\Rightarrow (\sigma_{3,1}\pi_2 f^2, \sigma_{3,2} f^3) \in C_k \\ &\Rightarrow (\pi_2\sigma_{2,2} f^2, \sigma_{3,2} f^3) \in C_k \\ (\pi_s f^s, f^{s+1}) \in C_k &\Rightarrow (\sigma_{s+1,1}\pi_s f^s, \sigma_{s+1,2} f^{s+1}) \in C_k \\ &\Rightarrow (\pi_s\sigma_{s,2} f^s, \sigma_{s+1,2} f^{s+1}) \in C_k\end{aligned}$$

de sorte que :
$(\sigma_{1,1}f^0, \sigma_{s+1,2}f^{s+1}) \in C_k(B,\Pi) \Rightarrow (\sigma_{1,1}, \sigma_{s+1,2}) \in G_k(B,\Pi)$ ■

3.2.3 L'équivalence des codes Cortex

Cas des codes binaires obtenus suivant une seule permutation

Proposition 3.2 *Soient σ une permutation de S_k, $\varepsilon = (\varepsilon_1, \varepsilon_2)$ et $\eta = (\eta_1, \eta_2)$ deux élements de G_k^*.*
Posons : $\sigma' = \eta_1\sigma\varepsilon_2^{-1}$
Alors :
1/ $C_k(B,\sigma') = (\varepsilon_1, \eta_2)(C_k(B,\sigma))$
2/ $C_k(B,\sigma')$ et $C_k(B,\sigma)$ sont des codes équivalents.
Preuve. 1/ Soit $(f^0, f^2) \in C_k(B,\sigma)$ c-à-d il existe f^1 de $\mathbb{F}_q^k$ tel que :

$$\begin{cases} (f^0, f^1) \in C_k \\ (\sigma f^1, f^2) \in C_k \end{cases} \Rightarrow \begin{cases} \varepsilon(f^0, f^1) \in C_k \\ \eta(\sigma f^1, f^2) \in C_k \end{cases}$$
$$\Rightarrow \begin{cases} (\varepsilon_1 f^0, \varepsilon_2 f^1) \in C_k \\ (\eta_1\sigma f^1, \eta_2 f^2) \in C_k \end{cases}$$
$$\Rightarrow \begin{cases} (\varepsilon_1 f^0, \varepsilon_2 f^1) \in C_k \\ (\eta_1\sigma\varepsilon_2^{-1}\varepsilon_2 f^2, \eta_2 f^2) \in C_k \end{cases}$$
$$\Rightarrow \begin{cases} (\varepsilon_1 f^0, \varepsilon_2 f^1) \in C_k \\ (\sigma'\varepsilon_2 f^1, \eta_2 f^2) \in C_k \end{cases}$$
$$\Rightarrow (\varepsilon_1 f^0, \eta_2 f^2) \in C_k(B,\sigma')$$

2/ $(\varepsilon_1, \eta_2) \in G_k(B,\sigma)$ donc :
$(\varepsilon_1, \eta_2)(C_k(B,\sigma))$ et $C_k(B,\sigma')$ sont des codes équivalents. ■

Corollaire 3.1 L'ensemble $\{(\varepsilon,\eta) \ / \ \exists\ \sigma \in S_k : \sigma = \eta_1\sigma\varepsilon_2^{-1}\}$ est un sous groupe de $Aut(C_k(B,\sigma))$.

Définition 3.5 *Soient $\sigma, \theta \in S_k$*

On dit que σ et θ sont G_k^*-équivalents, s'ils existent $\varepsilon = (\varepsilon_1, \varepsilon_2)$ et $\eta = (\eta_1, \eta_2)$ de G_k^* telles que :

$$\eta_1 \circ \sigma = \theta \circ \varepsilon_2$$

Cas des codes obtenus suivant une suite de permutations

Définition 3.6 [15] *Soient Σ et Π deux suites de permutations de S_k. On dit que θ_1* et *θ_2 sont :*

$(G_k^*(B,\Sigma), G_k^*(B,\Pi))-$équivalentes si et selement si ils existent $h = (h_1, h_2)$ et
$g = (g_1, g_2)$ dans $G_k^*(B,\Sigma)$ et $G_k^*(B,\Pi)$ telles que : $\theta_2 = g_1\theta_1 h_1^{-1}$.

Au moyen de la G_k^*- équivalence et la $(G_k^*(B,\Sigma), G_k^*(B,\Pi))-$équivalence on étudie l'équivalence des codes Cortex.

Proposition 3.3 *Soit θ_1 et θ_2 deux permutations*
$(G_k^*(B,\Sigma), G_k^*(B,\Pi))-$équivalentes

Posons : $\varepsilon_1 = (\Sigma, \theta_1, \Pi)$ et $\varepsilon_2 = (\Sigma, \theta_2, \Pi)$ deux suite de permutations
on a :
1/$(\eta_1, \rho_2)(C_k(B,\varepsilon_1)) = C_k(B,\varepsilon_2)$
2/$C_k(B,\varepsilon_1)$ et $C_k(B,\varepsilon_2)$ sont des codes équivalents.

Preuve. $(\eta_1, \rho_1)(C_k(B,\varepsilon_1) = C_k(B,\varepsilon_2)$.
Posons $\Sigma = (\pi_1, ...\pi_s)$ et $\Pi = (\acute{\pi_1}, ..., \acute{\pi_s})$.
Soit $(f^0, f^{s+1}) \in C_k(B,\Sigma)$ c-à-d : il existe $f^1, ..., f^s$ de $\mathbb{F}_q^k$ telles que :

$$\begin{cases} f^0 f^1 \in C_k \\ \pi_1 f^1 f^2 \in C_k \\ \quad . \\ \quad . \\ \pi_s f^s f^{s+1} \in C_k \end{cases}$$

Posons le vecteur $(\theta_1 f^{s+1}, g^{s+1}) \in C_k(B,\Pi)$ c-à-d : il existe $g^1, ..., g^s$ telles que :

$$\begin{cases} (\theta_1 f^{s+1}, g^1) \in C_k \\ (\acute{\pi_1} g^2, g^2) \in C_k \\ \quad . \\ \quad . \\ (\acute{\pi_s} g^s, g^{s+1}) \in C_k \end{cases}$$

de sorte que $(f^0, g^{s+1}) \in C(B,\varepsilon_1)$
on a :

$$\eta(f^0, f^{s+1}) \in C_k(B,\Sigma) \quad \Longrightarrow (\eta_1 f^0, \eta_2 f^{s+1}) \in C_k(B,\Sigma)$$
$$\rho(\theta_1 f^{s+1}, g^{s+1}) \in C_k(B,\Pi) \Longrightarrow (\rho_1\theta_1 f^{s+1}, \rho_2 g^{s+1}) \in C_k(B,\Pi)$$

$$\Longrightarrow (\rho_2\eta_2 f^{s+1}, \rho_2 g^{s+1}) \in C_k(B, \Pi)$$
$$\Longrightarrow (\eta_1 f^0, \rho_2 g^{s+1}) \in C_k(B, \varepsilon_2). \ \blacksquare$$

Si les permutations sont équivalentes alors les codes Cortex sont équivalents mais l'inverse n'est pas toujours valable c-à-d pour déterminer l'équivalence des codes Cortex il faut caractériser le groupe de permutations qui est, en général difficile à déterminer.

Nous traitons donc quelques représentations Cortex des codes auto-duaux.

3.3 Construction des codes Cortex

Remarque 3.9 Pour n = 2 *Il est triviale que $C(2,1)$ est un code auto-dual si la matrice génératrice $G' = [1 \| 1]$, ne peut pas construire $C(2,1)$ sous forme de code Cortex. $C(2,1)$est un code auto-dual de type I.*

- **Pour n = 4**

On prend comme code de base $C(2,1)$ de matrice génératrice $G' = [1 \parallel 1]$ et la suite de permutations $\Pi = \{\pi_1, \pi_2\}$ telle que $\pi_1 = \pi_2 = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}$, on obtient le code $C_2(C(2,1), \Pi)$ de matrice génératrice $G_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$ et pour $\Pi' = \{\pi\}$ telle que $\pi = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$ on obtient le code $C_2(c(2,1), \Pi')$ de matrice génératrice $G_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$.

Proposition 3.4 *G_1 et G_2 sont des matrices génératrices de codes équivalents.*

Preuve. Quelque soit la suite de permutations, et quelque soit $s \geq 1$ on obtient le code Cortex $C_2(C(2,1), \Pi)$ où bien $C_2(C(2,1), \Pi')$ qui sont des codes auto-duaux de type I tels que, $C_2(C(2,1), \Pi) = \sigma C_2(C(2,1), \Pi')$ avec

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix}. \ \blacksquare$$

- **Pour n = 6**

On prend le code de base $C(2,1)$ de matrice génératrice $G = [1 \parallel 1]$pour une suite de permutations quelconque on obtient un code Cortex $C_2(C(2,1),\Pi_i)$ où $i = \overline{0,5}$ de matrice génératrice :

$$\begin{cases} G_0 = [I_3 \parallel I_3] \quad \text{si } \Pi_0 = \{\pi_1,\pi_2\} \text{ avec} & \pi_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \\ & \pi_2 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \\ G_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \text{ si } \Pi_1 = \{\pi_1,\pi_2\} \text{ avec} & \pi_1 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \\ & \pi_2 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \\ G_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \text{ si } \Pi_2 = \{\pi_1,\pi_2\} \text{ avec} & \pi_1 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \\ & \pi_2 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \\ G_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \text{ si } \Pi_3 = \{\pi_1,\pi_2\} \text{ avec} & \pi_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \\ & \pi_2 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \\ G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \text{ si } \Pi_4 = \{\pi_1,\pi_2\} \text{ avec} & \pi_1 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \\ & \pi_2 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \\ G_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \text{ si } \Pi_5 = \{\pi_1,\pi_2\} \text{ avec} & \pi_1 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \\ & \pi_2 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \end{cases}$$

La partie redondante de G_i telle que $i = \overline{0,5}$ est la matrice associée à la permutation $\pi_1 \circ ... \circ \pi_s$ telles que $s \geq 1$.

Le code construit est un code auto-dual de type I .

- **Pour n=8**

le code Cortex est de base $\begin{cases} C(2,1) \\ \text{où bien} \\ C(4,2 \end{cases}$

Dans les deux cas on obtient un code Cortex $C_4(C(2,1),\Pi)$ où bien $C_4(C(4,2),\Pi)$ de matrice génératrice $G = \begin{bmatrix} I_4 \parallel R \end{bmatrix}$ où R est la matrice associéeà la permutation $\pi_1 \circ ... \circ \pi_s$ telles que $s \geq 1$.

Pour cette longueur on a un code auto-dual qui est une base non auto-dual voir l'exemple de construction de code de Hamming.

- **Pour n=10**

Le seul code de base dans ce cas est $C(2,1)$.

Si $C(2,1)$ est un code auto-dual on obtient le code Cortex $C_5(C(2,1),\Pi)$ de matrice génératrice

$G = \left[\begin{array}{c||c} I_5 & R \end{array} \right]$ avec R est la matrice associée à la permutation $\pi_1 \circ ... \circ \pi_s$.$C_5(C(2,1),\Pi)$ est un code auto-dual de type I, pour tout $s \geq 1$.

- **Pour n=12**

Les codes de base possible pour cette longueur sont : $C(2,1), C(4,2)$ et $C(6,3)$, pour les codes auto-duaux suivant on obtient le $C(12,6)$ de matrice génératrice $G = \left[\begin{array}{c||c} I_6 & R \end{array} \right]$ telles que :

R est la matrice associée à la permutation $\pi = \pi_1 \circ ... \circ \pi_s$ pour tout $s \geq 1$.

si on prend $\Pi = \{id\}$ on obtient $G = \left[\begin{array}{c||c} I_6 & I_6 \end{array} \right]$

- **Pour n=14**

La seule possibilité de code de base est $C(2,1)$ qui donne un code Cortex auto-dual de type I de matrice génératrice $G = \left[\begin{array}{c||c} I_7 & R \end{array} \right]$ telle que R est la partie redondante de G qu'est associée à la permutation

$\pi = \pi_1 \circ ... \circ \pi_s$, pour tout $s \geq 1$.

Remarque 3.10 *Avec les codes $C(2,1), C(4,2), C(6,3)$ toute suite de permutations Π donne un code cortex de matrice génératrice sous la forme $G = \left[\begin{array}{c||c} I_k & \pi(I_k) \end{array} \right]$, où $\pi(I_k)$ et la matrice obtenue de I_k en permutant les colonnes de I_k suivant π.*

Quelque représentation sous forme Cortex pour la longueur n=20

Si on prend comme code de base de matrice redondante

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

On construit certains codes Cortex auto-duaux de type I inéquivalents avec un certain choix de permutations.

Nous laissons le soin des autres codes auto-duaux de même longueur pour un traitement numérique de la question.

la suite de permutation $\Pi = \{\pi_1, \pi_2\}$	La partie redondante R de la matrice génératrice du code $C_{10}(C(10,5), \Pi)$
$\pi_1 = (10,9,8,7,6,5,4,3,2,1)$ $\pi_2 = (10,9,8,7,6,5,4,3,2,1)$	$R = \begin{bmatrix} 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&1&1 \\ 0&0&0&0&0&1&0&0&1&1 \\ 0&0&0&0&0&1&0&1&1&0 \\ 0&0&0&0&0&1&0&1&0&1 \\ 1&0&1&1&0&0&0&0&0&0 \\ 0&0&1&1&1&0&0&0&0&0 \\ 1&0&0&1&1&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 1&0&1&0&1&0&0&0&0&0 \end{bmatrix}$
$\pi_1 = (5,2,3,4,1,10,7,8,9,6)$ $\pi_2 = (1,2,9,8,5,6,7,4,3,10)$	$R = \begin{bmatrix} 0&0&0&1&1&0&0&1&1&1 \\ 0&0&0&0&0&0&1&0&1&1 \\ 1&0&0&1&1&0&0&0&0&0 \\ 1&1&1&1&0&0&0&1&1&1 \\ 1&1&1&0&1&0&0&1&1&1 \\ 0&0&1&1&1&0&0&0&1&1 \\ 0&1&0&1&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&1&1 \\ 0&0&1&1&1&1&1&1&1&0 \\ 0&0&1&1&1&1&1&1&0&1 \end{bmatrix}$
$\pi_1 = (3,2,1,6,5,4,9,8,7,10)$ $\pi_2 = (5,6,3,7,8,1,2,9,10,4)$	$R = \begin{bmatrix} 0&0&0&0&0&1&0&1&0&1 \\ 1&0&1&1&0&1&1&1&1&0 \\ 1&0&0&0&0&0&0&1&0&1 \\ 1&0&1&1&0&1&1&0&1&1 \\ 0&1&1&0&1&0&0&1&0&1 \\ 0&1&0&1&1&0&0&1&0&1 \\ 0&0&0&0&0&0&1&1&0&1 \\ 0&0&0&0&0&0&0&1&1&1 \\ 0&1&1&1&0&0&0&0&0&0 \\ 0&0&1&1&1&0&0&0&0&0 \end{bmatrix}$

$\pi_1 = (1,9,8,7,6,5,4,3,2,10)$ $\pi_2 = (3,2,1,4,5,6,7,8,9,10)$	$R = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$

$\pi_1 = (1,9,8,7,6,5,4,3,2,10)$ $\pi_2 = (5,6,3,7,8,1,2,9,10,4)$	$R = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

$\pi_1 = (8,7,6,5,4,3,2,1,9,10)$ $\pi_2 = (5,6,3,7,8,1,2,9,10,4)$	$R = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

$\pi_1 = (3,2,1,4,5,6,7,8,9,10)$ $\pi_2 = (7,6,5,4,3,2,1,8,9,10)$	$R = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$

$\pi_1 = (1,9,8,7,6,5,4,3,2,10)$ $\pi_2 = (7,6,5,4,3,2,1,8,9,10)$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$

$\pi_1 = (3,2,1,6,5,4,9,8,7,10)$ $\pi_2 = (3,2,1,6,5,4,9,8,7,10)$	$R = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$

$\pi_1 = (3,2,1,6,5,49,8,7,10)$ $\pi_2 = (8,7,6,5,4,3,2,1,9,10)$	$R = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$

- **Pour n≥16**

La méthode de construction des codes Cortex prend plusieurs axes à savoir à :

* le code de base.

* le nombre de codes de base non équivalents.

* la suite de permutations et leur nombre.

Pour n≥16 nous nous intéressons aux codes Cortex binaires dont le code de base est le code de Hamming étendu noté H_8 de matrice génératrice

$$G = [I_4 \parallel h_4] \text{ où } h_4 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Proposition 3.5 [15] *Si le code de base B est un code auto-dual de type II alors tout code Cortex construit à partir de B avec un nombre pair (resp. impair) de permutation de type II (resp. type I)*

la démenstration est basée sur le lemme suivant

Lemme 3.1[15] *Soit $e \geq 1$ et posons $k = 4e$ et $R = I_e \otimes h_4$, pour tout υ de $\mathbb{F}_q^k$ on a la relation suivante :*

Preuve. $\omega(\upsilon) + \omega(\upsilon p) \equiv 0[4]$. ■

Preuve. Si B un code de dimension k_b et $k = ek_b$ la dimension d'un code Cortex.

Pour tout m de $\mathbb{F}_q^k$ et pour tout suite de permutations $\pi_1, ..., \pi_s$

telles que $s \geq 1$ on a d'après le lemme précédent la relation suivante :

$\omega(m) + \omega(R^s(ms)) \equiv 0[4] \Longrightarrow \omega(m) + (-1)^s \omega(R^{(s)}(m)) \equiv 0[4]$

$$\Longrightarrow \omega(m) \equiv (-1)^{s+1}\omega(R^s(m))[4].$$

$*$ Si s est pair, comme le poids d'un mot du code Cortex est $[\omega(m)+\omega(R^s(m)]$, on en déduit qu'ils ont tous un poids divisible par 4 c-à-d que le code est de type II.

$*$ Si s est impair, on peut alors écrire :

$$\omega(m) + \omega(R^{(s)}(m)) \equiv 2\omega(m)[4]$$

■

Cette relation signifie que certains mots du code Cortex ont un poids multiple de 2. Cela entraîne donc que le code Cortex est de type I.

Remarque 3.11

1/ *Il existe des codes de type I de longueur multiple de 8 qui ne peuvent être obtenus sous forme de codes Cortex à partir de code de Hamming étendu* H_8 .

2/ *Tous les codes auto-duaux de type II peuvent être mis sous forme de code Cortex à partir du code de base le code de Hamming étendu.*

3/ *Il existe des codes auto-duaux Cortex dont la base peut être non auto-dual (Voir le quatrième chapitre).*

3.3.1 Les codes extrémaux de type II sous forme de code Cortex

Vu l'importance des codes Cortex auto-duaux extrémaux, nous présentons les travaux de **[3],[6],[7],[8],[15]** et **[16].**

Ces codes sont plus intéressants à étudier puisqu'ils ont une distance minimale maximale.

Tableau de liens

Définition 3.7 *Soit σ une permutation de S_k. On définit le tableau de liens de la permutation σ à partir de matrice $T(\sigma) = [t_{ij}]$ de taille $e \times e$ vérifiant :*

$$t_{ij} = |\sigma(B_i) \cap B_J|$$

où t_{ij} sont des éléments dont $\mathbb{N}_4$ vérifiant :

$$\sum_{s=1}^{e} t_{si} = 4 \quad et \quad \sum_{s=1}^{e} t_{is} = 4$$

et pour tout $i \leq e$ *,* B_i *est l'ensemble* $4i + \mathbb{N}_4$.

Exemple 3.5 *Si* $e = 3$ *alors* $k = 12$
En calculant les tableaux de liens des permutations suivantes :

$$\sigma_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 4 & 2 & 3 & 1 & 8 & 6 & 7 & 5 & 12 & 10 & 11 & 9 \end{pmatrix}$$

$$\sigma_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 1 & 2 & 5 & 4 & 3 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{pmatrix}$$

on obtient :

$$T(\sigma_1) = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix} \quad et \quad T(\sigma_2) = \begin{bmatrix} 3 & 0 & 1 \\ 0 & 4 & 0 \\ 1 & 0 & 3 \end{bmatrix}$$

Borne sur la distance minimale

On peut majorer la distance minimale du code Cortex construit suivant une seule permutation, deux permutations et trois permutations.

Proposition 3.6 [15] *Soient* σ *une permutation de* S_k *et* d *la distance minimale de* $C_k(B, \sigma)$*, on a :*

- *Si* $|T(\sigma)| \geq 3$ *alors* $d = 2$.
- *Si* $|T(\sigma)| = 2$ *alors* $d = 4$.
- *Si* $|T(\sigma)| = 1$ *alors (pour* $k \geq 16$ *),* $d \leq 16$.

Proposition 3.7 [15] *pour* $s = 2$, $\Pi = \{\pi_1, \pi_2\}$ *on a :*

- *S'il existe* i *dans* $\{1, 2\}$ *telle que* $|T(\pi_i)| \geq 3$ *alors* $d = 4$.
- *S'il existe* i *dans* $\{1, 2\}$ *telle que* $|T(\pi_i)| = 2$ *alors* $d \leq 8$.
- *Si* $|T(\pi_1)| = |T(\pi_2)| = 1$*(pour* $k \geq 16$*), alors* $d \leq 12$.

Proposition 3.8 [15] *pour* $s = 3$*, on a :*

- *Si* $|T(\pi_2)| \geq 3$ *alors* $d \leq 6$.
- *Si* $|T(\pi_2)| = 2$ *alors* $d \leq 12$.
- *Si* $|T(\pi_2)| = 1$ *alors* $d \leq 18$.

Les codes obtenus

Les codes Cortex sont obtenus à l'aide des permutations suivantes (dites transformations affines) :
pour $a, b \in \mathbb{Z}_k$:

$$\delta_{a,b} : \mathbb{Z}_k \longrightarrow \mathbb{Z}_k$$
$$i \longmapsto \delta_{a,b}(i) = (ai + b)[k]$$

Pour $n \leq 88$ les codes de paramétres $[n, \frac{n}{2}, d]$ sont obtenus par Otmani et les codes de paramétres $[96, 48, 16]$ $[128, 64, 25]$ et $[256, 128, d \geq 22]$ sont construits dans **[3].**

- **Pour n = 16**

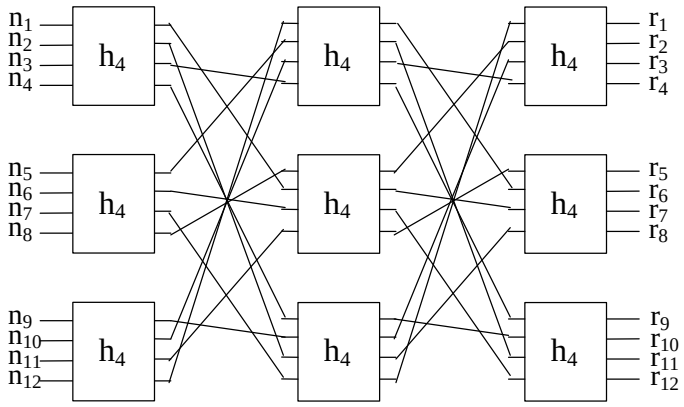
On obtient un code extrémal de type II de paramètres $[16, 8, 4]$ avec les permutations $\pi_1 = \pi_2 = \delta_{1,0}$ où :

$$\pi_1 : \{1, ..., 8\} \longrightarrow \{1, ..., 8\}$$
$$i \longmapsto \pi_1(i) = i\,[8] = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}$$

- **Pour n = 24**

Pour le code Cortex de paramètres $[24, 12, 8]$ qui est le code de Golay obtenus suivant la suite de transformation $\delta = \{\delta_{5,1}^{(1)}\ ,\ \delta_{5,1}^{(2)}\}$ où $\delta_{5,1}^{(1)} = \delta_{5,1}^{(2)}$ avec $\delta_{5,1}^{(1)}(i) = (5i + 1)[12]$ en utilisant le codeur Cortex pour lequel les boites (h_4) est la partie redondante de la matrice génératrice des codes de base le Hamming étendu $[8, 4, 4]$ les bits $n_i$correspondent aux bits d'information alors que les bits r_i sont ceux de redondance.

On obtient la description graphique suivante :



Le code de Golay étendu de paramètres [24,12,8]

- **Pour n = 32**

Il y a cinq codes extrémaux obtenus à partir des permutations suivantes :

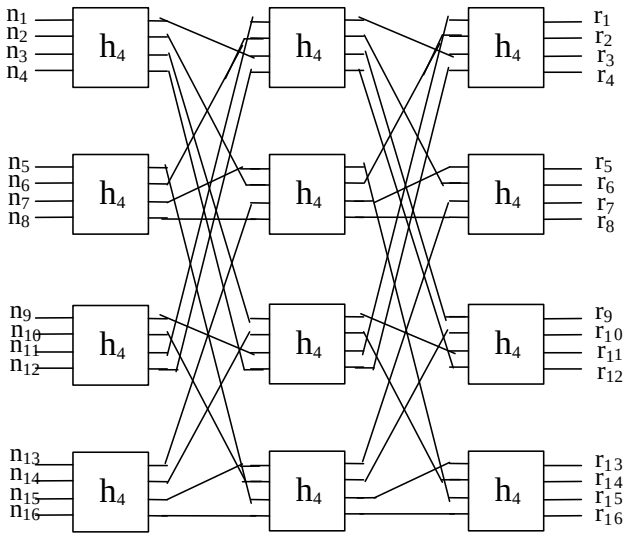
Le code	permutations
CP81	$\pi_1 = 1\ ,\ 5\ ,\ 9\ ,\ 13\ ,\ 2\ ,\ 6\ ,\ 10\ ,\ 14\ ,\ 3\ ,\ 7\ ,\ 11\ ,\ 15\ ,\ 4\ ,\ 8\ ,\ 12\ ,\ 16$ $\pi_2 = 1\ ,\ 14\ ,\ 7\ ,\ 8\ ,\ 5\ ,\ 2\ ,\ 11\ ,\ 16\ ,\ 9\ ,\ 6\ ,\ 15\ ,\ 4\ ,\ 13\ ,\ 10\ ,\ 3\ ,\ 12$
CP82	$\pi_1 = 1\ ,\ 5\ ,\ 9\ ,\ 13\ ,\ 2\ ,\ 3\ ,\ 10\ ,\ 14\ ,\ 4\ ,\ 6\ ,\ 7\ ,\ 15\ ,\ 8\ ,\ 11\ ,\ 12\ ,\ 16$ $\pi_2 = 1\ ,\ 6\ ,\ 15\ ,\ 4\ ,\ 5\ ,\ 14\ ,\ 11\ ,\ 8\ ,\ 9\ ,\ 10\ ,\ 3\ ,\ 16\ ,\ 13\ ,\ 2\ ,\ 7\ ,\ 12$
CP83	$\pi_1 = 1\ ,\ 5\ ,\ 6\ ,\ 9\ ,\ 2\ ,\ 3\ ,\ 7\ ,\ 13\ ,\ 4\ ,\ 8\ ,\ 10\ ,\ 14\ ,\ 11\ ,\ 12\ ,\ 15\ ,\ 16$ $\pi_2 = 1\ ,\ 2\ ,\ 11\ ,\ 12\ ,\ 5\ ,\ 14\ ,\ 7\ ,\ 16\ ,\ 9\ ,\ 10\ ,\ 3\ ,\ 8\ ,\ 13\ ,\ 6\ ,\ 15\ ,\ 4$
CP84	$\pi_1 = 1\ ,\ 5\ ,\ 9\ ,\ 13\ ,\ 2\ ,\ 6\ ,\ 10\ ,\ 14\ ,\ 3\ ,\ 7\ ,\ 11\ ,\ 15\ ,\ 4\ ,\ 8\ ,\ 12\ ,\ 16$ $\pi_2 = 1\ ,\ 14\ ,\ 7\ ,\ 4\ ,\ 5\ ,\ 2\ ,\ 11\ ,\ 12\ ,\ 9\ ,\ 6\ ,\ 15\ ,\ 8\ ,\ 13\ ,\ 10\ ,\ 3\ ,\ 16$
CP85	$\pi_1 = 1\ ,\ 5\ ,\ 9\ ,\ 13\ ,\ 2\ ,\ 6\ ,\ 10\ ,\ 14\ ,\ 3\ ,\ 7\ ,\ 11\ ,\ 15\ ,\ 4\ ,\ 8\ ,\ 12\ ,\ 16$ $\pi_2 = 1\ ,\ 14\ ,\ 7\ ,\ 12\ ,\ 5\ ,\ 2\ ,\ 11\ ,\ 16\ ,\ 9\ ,\ 6\ ,\ 15\ ,\ 4\ ,\ 13\ ,\ 10\ ,\ 3\ ,\ 8$

• De plus, on obtient un code extrémal de type II, de paramètres [32,16,8] à partir de la suite de transformation $\delta = \{\delta_{3,0}^{(1)}, \delta_{3,0}^{(2)}\}$ avec $\delta_{3,0}^{(1)} = \delta_{3,0}^{(2)}$ et :

$$\delta_{3,0}^{(1)} : \{1, ..., 16\} \longrightarrow \{1, ..., 16\}$$
$$i \longmapsto \delta_{3,0}^{(1)}(i) = 3i[16]$$

telle que :

$$\delta_{3,0}^{(1)}(i) = (3, 6, 9, 12, 14, 2, 5, 8, 11, 12, 1, 4, 7, 10, 13, 16)$$



Un code Cortex de paramètres auto-dual [32,16,8]

le code de Golay de paramètres [32,16,8] avec la suite de permutations $\Pi = \{\pi_1, \pi_2\}$ où $\pi_1 = \pi_2$ et :

$$\pi_1(1, ..., 16) = (1, 5, 2, 6, 3, 9, 4, 10, 7, 13, 8, 14, 11, 15, 12, 16)$$

pour la transformation affine $\delta_{7,1}$ définie par

$$\delta_{7,1} : \{1, ..., 16\} \longrightarrow \{1, ..., 16\}$$
$$i \longmapsto \delta_{7,1}(i) = (7i + 1)[16]$$

telle que :

$$\delta_{7,1}^{(1)}(i) = (3, 6, 9, 12, 15, 18, 1, 4, 7, 10, 13, 16, 19, 2, 5, 8, 11, 14, 17, 20)$$

On obtient le code de paramètres [32,16,6] sous forme de code Cortex dont le polynôme enumurateur des poids est :

$$\begin{aligned} W(X) &= 1 + 32X^6 + 300X^8 + 1952X^{10} + 6976X^{12} + 14400X^{14} + 18214X^{16} \\ &\quad +14400X^{18} + 6976X^{20} + 1952X^{22} + 300X^{24} + 32X^{26} + X^{32}. \end{aligned}$$

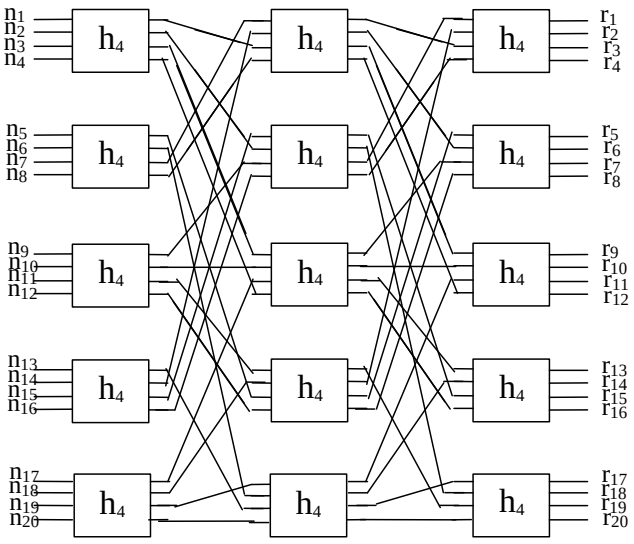
- **Pour n = 40**

Le code extrémal de type II de paramètres [40,20,8] est obtenu suivant la suite de permutations

$\Pi = \{\pi_1, \pi_2\}$ où $\pi_1 = \pi_2$ et :

$$\pi_1 = (1, 5, 2, 6, 3, 9, 4, 10, 7, 13, 8, 14, 11, 17, 12, 18, 15, 19, 16, 20)$$

Pour la suite de transformation affine $\delta = \{\delta_{3,0}^{(1)}, \delta_{3,0}^{(2)}\}$ avec $\delta_{3,0}^{(1)} = \delta_{3,0}^{(2)}$ où $\delta_{3,0}^{(1)}(i) = 3i[20]$ on obtient le code extrémal de type II de paramètres [40,20,8].



le code Cortex auto-duaux de paramétres [40,20,8]

- **Pour n = 56**

Pour la suite de transformation $\delta = \{\delta_{5,1}^{(1)}, \delta_{5,1}^{(2)}\}$ avec $\delta_{5,1}^{(1)} = \delta_{5,1}^{(2)}$ où $\delta_{5,1}^{(1)}(i) = (5i+1)[28]$ on obtient le code extrémal de type II de paramètres [56,28,12].

- **Pour n = 64**

Pour la suite de transformation $\delta = \{\delta_{19,0}^{(1)}, \delta_{19,0}^{(2)}\}$ avec $\delta_{19,0}^{(1)} = \delta_{19,0}^{(2)}$ où $\delta_{19,0}^{(1)}(i) = 19i[32]$ on obtient le code extrémal de type II de paramètres [64,32,12] et suivant le suite de permutation :

$$\Pi = \begin{cases} \pi_1, \pi_5 : i \mapsto i[4] \\ \pi_2, \pi_6 : i \mapsto (3i+1)[8] \\ \pi_3, \pi_7 : i \mapsto (5i+1)[16] \\ \pi_4 : i \mapsto 19i[32] \end{cases}$$

On obtint un code de paramètres [64,32,10].

- **Pour n = 72**

Suivant la transformation affine $\delta = \{\delta_{5,0}^{(1)}, \delta_{5,0}^{(2)}\}$ avec $\delta_{5,0}^{(1)} = \delta_{5,0}^{(2)}$ où $\delta_{5,0}^{(1)}(i) = 5i[36]$ on obtient le code Cortex de type II de paramètres [72,36,12]. Ce code n'est pas un code extrémal mais possède une meilleur distance minimale connue pour cette longueur .

- **Pour n = 88**

La transformation affine $\delta = \{\delta_{35,0}^{(1)}, ..., \delta_{35,0}^{(6)}\}$ où $\delta_{35,0}^{(1)} = ... = \delta_{35,0}^{(6)}$ avec $\delta_{35,0}^{(1)}(i) = 35i[44]$ donne un nouveau code Cortex de type II extrémal C_{88} de paramètres [88,44,16]. Il existe trente-trois codes extrémaux de type II pour cette longueur , et il y a deux codes extrémaux de type II inéquivalent de paramétres [88,44,16] de polynôme enumurateur :

$$\begin{aligned} W(X) &= 1 + 32164X^{16} + 6992832X^{20} + 535731625X^{24} + 16623384448X^{28} \\ &\quad +225426781470^{32} + 1405590745152X^{36} + 4163803131796X^{40} \\ &\quad +5968212445440X^{44} + 4163803131796X^{48} + 1405590745152X^{52} \\ &\quad +225426781470X^{56} + 16623384448X^{60} + 535731625X^{64} \\ &\quad +6992832X^{68} + 32164X^{84} + X^{88}. \end{aligned}$$

- **Pour n = 96**

On obtient un code Cortex de type II de paramètres [96,48,16] suivant la transformation affine $\delta = \{\delta_{37,1}^{(1)}, ...\delta_{37,1}^{(20)}\}$où $\delta_{37,1}^{(1)} = ... = \delta_{37,1}^{(20)}$ avec $\delta_{37,1}^{(1)}(i) = (37i + 1)[48]$.

Pour le code de type II de paramètres [96,48,20] on a le polynôme enumurateur suivant

$$\begin{aligned} W(X) &= 1 + 3217056X^{20} + 369844880X^{24} + 18642839520X^{28} \\ &\quad +422069980215X^{32} + 4552866656416X^{36} + 24292689565680X^{40} \\ &\quad +65727011639520X^{44} + 91447669224080X^{48} + 65727011639520X^{52} \\ &\quad +24292689565680X^{56} + 4552866656416X^{60} + 422069980215X^{64} \\ &\quad +18642839520X^{68} + 369844880X^{72} + 3217056X^{76} + X^{96}. \end{aligned}$$

• **Pour n = 128**

Pour la transformation affine $\delta = \{\delta_{19,0}^{(1)}, ..., \delta_{19,0}^{(20)}\}$ où
$\delta_{19,0}^{(1)} = ... = \delta_{19,0}^{(20)}$ avec $\delta_{19,0}^{(1)}(i) = 19i[64]$ on obtient le code Cortex de type II de paramètres [128,64,20]

Pour le code Cortex de type II de paramètres [128,64,14] obtenu suivant la suite de permutations :

$$\Pi = \begin{cases} \pi_1, \pi_6 : i \mapsto i[4] \\ \pi_2, \pi_7 : i \mapsto i[8] \\ \pi_3, \pi_8 : i \mapsto (3i+1)[16] \\ \pi_4, \pi_9 : i \mapsto (19i+1)[32] \\ \pi_5 : i \longmapsto (37i+1)[64] \end{cases}$$

est un code obtenu en 10 couches. Dans ce cas les deux codes de base utilisés sont le code de Hamming étendu [8,4,4] et le code auto-dual [32,16,8] de matrice génératrice $G = \left[\begin{array}{c|c} I_{16} & R \end{array} \right]$ avec :

$$R = \left[\begin{array}{cccccccccccccccc} 1&1&0&0&0&1&1&1&1&0&1&1&0&1&1&1 \\ 0&1&1&1&1&1&0&0&0&1&1&1&1&0&1&1 \\ 1&0&1&1&0&1&1&1&1&1&0&0&0&1&1&1 \\ 0&1&1&1&1&0&1&1&0&1&1&1&1&1&0&0 \\ 1&1&0&0&1&0&1&1&0&1&1&1&1&0&1&1 \\ 1&0&1&1&1&1&0&0&1&0&1&1&0&1&1&1 \\ 0&1&1&1&1&0&1&1&1&1&0&0&1&0&1&1 \\ 1&0&1&1&0&1&1&1&1&0&1&1&1&1&0&0 \\ 0&0&1&1&1&1&0&1&1&1&1&0&1&1&0&1 \\ 1&1&0&1&0&0&1&1&1&1&0&1&1&1&1&0 \\ 1&1&1&0&1&1&0&1&0&0&1&1&1&1&0&1 \\ 1&1&0&1&1&1&1&0&1&1&0&1&0&0&1&1 \\ 0&0&1&1&1&1&1&0&1&1&0&1&1&1&1&0 \\ 1&1&1&0&0&0&1&1&1&1&1&0&1&1&0&1 \\ 1&1&0&1&1&1&1&0&0&0&1&1&1&1&1&0 \\ 1&1&1&0&1&1&0&1&1&1&1&0&0&0&1&1 \end{array}\right]$$

où il est obtenu par la méthode Cortex dont la base est le code de Hamming étendu suivant la suite de transformations affines $\delta = \{\delta_{3,0}^{(1)}, \delta_{3,0}^{(2)}\}$ telles que : $\delta_{3,0}^{(1)}(i) = 3i[16]$.

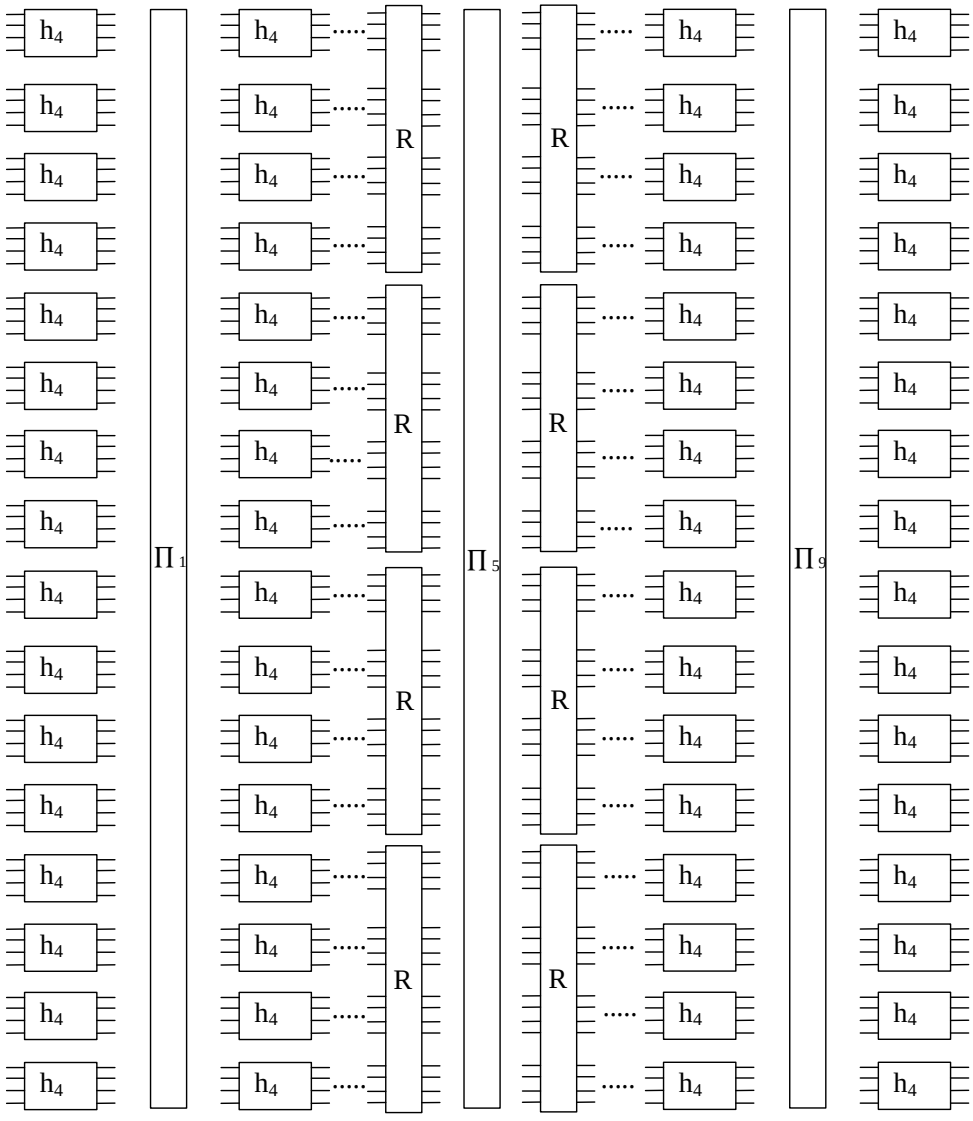
• Le principe de construction, consiste à remplacer les deux couches de code de Hamming du centre par

deux couches de codes [32,16,8] mais la structure générale reste inchangé.

• En utilisant cette variante pour construire un code de paramètres

[128,64,16] suivant la suite de permutations :

$$\Pi = \begin{cases} \pi_1, \pi_6 : i \mapsto i[4] \\ \pi_2, \pi_7 : i \mapsto i[8] \\ \pi_3, \pi_8 : i \mapsto 3i[16] \\ \pi_4, \pi_9 : i \mapsto 19i[32] \\ \pi_5 : i \longmapsto 37i[64] \end{cases}$$



Codeur cortex des codes [128,64,16] à base de code de Hamming étendu [8,4,4] et le code de paramètres [32,16,8]

Remarque3.12 *R définit la partie redondante de la matrice génératrice du code [32,16,8]et h_4 est la partie redondante de la matrice génératrice du code de Hamming [8,4,4].*

- **Pour n = 256**

On obtient un code de paramètres [256,128, $d \geq 22$] suivant la suite de permutations :

$$\Pi = \begin{cases} \pi_1, \pi_7 : i \mapsto i[4] \\ \pi_2, \pi_8 : i \mapsto i[8] \\ \pi_3, \pi_9 : i \mapsto 3i[16] \\ \pi_4, \pi_{10} : i \mapsto 19i[32] \\ \pi_5, \pi_{11} : i \longmapsto 37i[64] \\ \pi_6 : i \longmapsto 73i[128] \end{cases}$$

3.4 Quelques codes de base pour obtenir des codes auto-duaux Cortex extrémaux sur un corps premier

Dans cette section nous proposons les paramètres des codes de bases pour construire des codes auto-duaux Cortex extrémaux sur un alphabet A **[15]**.

- **Sur $\mathbb{F}_2$**

$16 \leq n \leq 88$ et $n = 96, 128, 256$.

Dans le cas binaire le seul code de base considéré est le code de Hamming étendu H_8.

- **Sur $\mathbb{F}_3$**

On considère comme code de base le tetracode de paramètres $[4, 2, 3]$ et le code de Golay ternaire de paramètres $[12, 6, 6]$ ayant respectivement comme matrice génératrices :

$$G_1 = \begin{bmatrix} 1 & 0 & 2 & 2 \\ 0 & 1 & 2 & 1 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 2 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 2 & 2 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 2 & 1 & 2 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

• **Sur** $\mathbb{Z}_4$

On considère comme code de base l'octacode de paramètres [8,4^4,6] défini par la matrice génératrice suivante :

$$G_{4^4} = \begin{bmatrix} 1 & 0 & 0 & 0 & 2 & 1 & 3 & 1 \\ 0 & 1 & 0 & 0 & 3 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 3 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 \end{bmatrix}$$

• **Sur** $\mathbb{F}_4$

Dans le cas hermitien le produit scalaire définis sur $\mathbb{F}_4$ est $x \cdot y = \sum_{i=1}^{n} x_i y_i^2$ pour tout $x, y \in \mathbb{F}_4^n$.

On considère comme code de base l'hexacode et le code de paramètres [8,4,4] ayant respectivement comme matrice génératrices :

$$G_1 = \begin{bmatrix} 1 & 0 & 0 & 1 & \omega & \omega \\ 0 & 1 & 0 & \omega & 1 & \omega \\ 0 & 0 & 1 & \omega & \omega & 1 \end{bmatrix}$$

et

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & \omega & \omega & \omega \\ 0 & 1 & 0 & 0 & \omega & 1 & \omega & \omega \\ 0 & 0 & 1 & 0 & \omega & \omega & 1 & \omega \\ 0 & 0 & 0 & 1 & \omega & \omega & \omega & 1 \end{bmatrix}$$

Dans le cas euclidien le produit scalaire défini sur $\mathbb{F}_4$ est $x \cdot y = \sum_{i=1}^{n} x_i y_i$ pour tout $x, y \in \mathbb{F}_4^n$. Le code de base défini par la matrice génératrice suivante :

$$G = \begin{bmatrix} 1 & 0 & \omega^2 & \omega \\ 0 & 1 & \omega & \omega^2 \end{bmatrix}$$

• **Sur $\mathbb{F}_5$**

Les codes de bases choisis sont définis par les matrices génératrices suivantes :

$$G_1 = \begin{bmatrix} 1 & 0 & 0 & 2 & 2 & 4 \\ 0 & 1 & 0 & 3 & 1 & 3 \\ 0 & 0 & 1 & 4 & 2 & 2 \end{bmatrix}$$

et

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 2 & 2 & 4 \\ 0 & 1 & 0 & 0 & 3 & 0 & 4 & 3 \\ 0 & 0 & 1 & 0 & 3 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 1 & 2 & 3 & 0 \end{bmatrix}$$

3.5 Les codes auto-duaux optimaux sur un corps premier

Nous présentons dans cette partie la méthode de construction des codes auto-duaux généralisant la méthode Cortex. Cette généralisation est donnée par[**15**].

Cette méthode est expérimentale pour construire rapidement et facilement de bons codes auto-duaux pour n'importe longueur et alphabet. Ces bons codes ont une distance minimale qui vaut ou qui est très proche de la plus grande distance minimale connue pour un code auto-dual à une longueur donnée. Nous renforçons la représentation avec des exemples. Des meilleurs code auto-duaux connus jusqu'aujourd'hui. En particulier le code auto-dual quaternaire et beaucoup de code sur $\mathbb{F}_5$ et $\mathbb{F}_7$, [**7**],[**9**],[**10**],[**15**] et [**16**].

3.5.1 Principe de construction de la nouvelle méthode

Etant données des matrices carrées M_r d'ordre n, d'éléments dans un anneau R. Satisfaisant :

$$M_r \cdot M_r^t = \lambda_r I_n.(*)$$

où $\lambda_r \in R$ inversible

Pour toute permutations $\pi_1, ..., \pi_r$ de S_n, on lui associée des matrices $\Pi_i = \delta_{\pi_i(k)j}$ où

$$\delta_{\pi_i(k)j} = \begin{cases} 1 & \text{si } \pi_i(k) = i \\ 0 & \text{si non} \end{cases}$$

Pour un code initial C de matrice génératrice G, on construit le code C_r comme suit :

1/ On choisit une suite finie de permutations $\pi_1, ..., \pi_r$ dans S_n.

2/ Pour chaque permutation π_i, on lui associe la matrice Π_i.

3/ On définit C_r comme eton le code de matrice génératrice G_r définie par : $G_r = GM_1\Pi_1...M_r\Pi_r$.

Proposition 3.9 [15] *Si le code C est auto-dual alors C_r est auto-dual.*

Preuve. Le code C est auto-dual c-à-d $GG^t = 0$. Ainsi, on à :

$$\begin{aligned} G_rG_r^t &= (GM_1\Pi_1...M_r\Pi_r)(GM_1...M_r\Pi_r)^t \\ &= GM_1\Pi_1...M_r\Pi_r\Pi_r^tM_r^t...\Pi_1^tM_1G^t \\ &= \lambda_r^rGG^t \\ &= 0 \ \blacksquare \end{aligned}$$

Remarque 3.12 *Pour obtenir une matrice vérifiant (*), ou choisit une matrice B d'ordre $b \prec n$ telle que : $BB^t = \lambda I_b$ pour $\lambda = \alpha^2$ inversible dans R.*

On a M consistant en k_1 fois la matrice B sur la diagonale et k_2 fois α avec : $bk_1 + k_2 = n$.

$$M = \begin{bmatrix} B & & & & \\ & . & & 0 & \\ & & . & & \\ & & & . & \\ & 0 & & B & \\ & & & & \alpha I_{k_2} \end{bmatrix}$$

Définition 3.8 *Pour toutes suite de permutations affine :*

$$\Pi = \left\{\pi_i \ : \ i = \overline{1,r}\right\} \text{ où } \pi_i(z) = (az + b)[n]$$

1/ Π et dit de type $(f_1; a, r)$ si :$\forall i = 1, ..., r : a_i = b_i = a$ avec $a \in \mathbb{Z}_n$.

2/ Π et dit de type$(f_2; a, r)$ si : $\forall i = 1, ..., r : a_i = b_i = a^i$ avec $a \in \mathbb{Z}_n$.

3/ Π et dit de type$(f_3; a, r)$ si : $\forall i = 1, ..., r : a_i = a^i, b_i = ia^i$ avec $a \in \mathbb{Z}_n$.

Remarque 3.13 *La matrice $G = I_e \otimes G_b$ où G_b est la matrice génératrice d'un code auto-dual B_b. Le code C est égale au produit cartésien B_b^e.*

Exemple 3.6 *Si on prend comme code initial C_{ini} de matrice génératrice*

$$G_{ini} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

C'est *la matrice génératrice du code* $C_{ini} = i_2 \oplus i_2 \oplus i_2$. *où* $i_2 = \{00, 11\}$ *sur* $\mathbb{F}_2$.

$$B = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \qquad \textit{on obtient} \qquad M = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Si en prend π *de* S_6 *telle que :*

$$\forall z \in \mathbb{Z}_6 : \pi(z) = (z+1)[6]$$

Alors la matrice associe a π *est :*

$$\Pi = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Le code C_1 *obtenu à partir d'une seule permutation de type* $(f_1; 1, 1)$ *et suivant la matrice génératrice* G_{ini} *de* C_{ini} *est de matrice génératrice :*

$$G_1 = GM\Pi = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

C_1 est équivalent à C_{ini} ce qui évident que pour cette longueur il y a un seul code auto-dual de type I.

Exemple 3.7 *Soit* C_{ini} *un code auto-dual sur* $\mathbb{F}_4$ *ayant comme matrice génératrice :*

$$G_{ini} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

pour

$$M = B = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & \omega & \omega^2 & 0 \\ 1 & \omega^2 & \omega & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

si en prend la permutation de type $(f_1; 1, 1)$

On obtient le code de matrice génératrice :

$$G_1 = \begin{bmatrix} 0 & 0 & \omega^2 & \omega \\ 1 & 1 & \omega^2 & \omega \end{bmatrix}$$

Dans la partie suivante, nous donnons les matrices B, et les matrices génératrices des codes initiaux, sur plusieurs alphabets. **[7]** et **[15],** qui donnent des codes optimaux.

- **Codes auto-duaux sur $\mathbb{F}_2$**

La construction que nous considérons suppose que l'on prend :

$$B_{ini} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \qquad et \qquad G_{ini} = \begin{bmatrix} 1 & 1 & & & & \\ & & . & & 0 & \\ & & & . & & \\ & 0 & & & . & \\ & & & & 1 & 1 \end{bmatrix}$$

où bien

$$B_{ini}{}' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

et avec un choix adéquat de permutation de type $(f_i; a, r)$, on obtient le tableau suivant :

n	d	f_i	a	r
2	2	f_1	1	1
4	2	f_1	1	1
6	2	f_1	1	1
8	2	f_1	1	1
10	2	f_1	1	1
12	4	f_1	1	4
14	4	f_1	1	4
16	4	f_1	1	4
18	4	f_1	1	4
20	4	f_1	1	4
22	6	f_1	1	20
24	6	f_1	1	3
26	6	f_1	1	76
28	6	f_1	1	4
30	6	f_1	1	17
32	6	f_1	3	3
34	6	f_1	1	10
36	8	f_1	1	8
38	8	f_1	1	116
40	8	f_1	1	8
42	8	f_1	1	94
44	8	f_1	1	8
46	8	f_1	1	10
48	8	f_1	1	12
50	8	f_1	1	12
52	10	f_1	1	19
54	10	$'f_1$	5	9
56	10	f_1	1	10
58	10	f_1	1	112
60	12	f_1	1	78
62	10	f_1	1	21
64	12	f_1	3	11
66	12	$'f_1$	7	23
68	12	f_1	1	11
70	10	f_1	1	18
72	12	f_1	1	11
74	12	f_1	1	234
76	14	f_1	65	276
78	14	$'f_2$	61	3820
80	14	f_1	43	38
82	12	f_1	1	24
84	14	f_1	1	14
86	12	f_1	1	18
88	14	f_1	1	20
90	14	$'f_1$	7	15
92	14	f_1	1	21
94	14	f_1	1	85
96	16	f_1	13	114
98	14	f_1	1	36
100	16	f_1	1	97
102	16	$'f_1$	5	10
104	16	f_1	1	25
106	14	f_1	1	21
108	16	f_1	1	21
110	16	f_1	1	37
112	12	f_1	1	20
114	18	$'f_1$	25	75
116	18	f_1	1	26
118	16	f_1	1	25
120	18	f_1	1	112
122	16	f_1	1	21
124	18	f_1	1	23
126	18	$'f_1$	5	19
128	18	f_1	3	27
130	18	f_1	1	44

Les codes auto – duaux de type I construits sur $\mathbb{F}_2$

Tableau 4

- **Sur** $\mathbb{F}_3$

La construction suppose que $k_2 = 0$ et :

$$B_{ini} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \quad et \quad G_{ini} = \begin{bmatrix} 1 & 0 & 1 & 1 & & & & & & \\ 0 & 1 & 1 & 2 & & & & & & \\ & & & & . & & & 0 & & \\ & & & & & . & & & & \\ & & & 0 & & & . & & & \\ & & & & & & & 1 & 0 & 1 & 1 \\ & & & & & & & 0 & 1 & 1 & 2 \end{bmatrix}$$

n	d	f_i	a	r
24	9	f_1	1	8
28	9	f_1	1	8
32	9	f_1	1	9
36	12	f_2	29	22
40	12	f_1	1	59
44	12	f_1	1	12
48	12	f_1	1	14
52	15	f_3	7	187

56	15	f_1	43	49
60	15	f_1	1	16
64	15	f_1	1	16
68	15	f_2	3	4
72	15	f_2	5	4
76	18	f_1	1	35
80	18	f_1	1	19
84	18	f_1	1	17
88	18	f_1	1	17

Les codes auto – duaux construits sur $\mathbb{F}_3$

Tableau 5

- **Sur $\mathbb{F}_4$**

Le cas hermitien
On prend

$$B_{ini} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & \omega & \omega^2 & 0 \\ 1 & \omega^2 & \omega & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad et \qquad G_{ini} = \begin{bmatrix} 1 & 1 & & & & \\ & & . & & 0 & \\ & & & . & & \\ & 0 & & & . & \\ & & & & 1 & 1 \end{bmatrix}$$

n	d	f_i	a	r
2	2	f_1	1	1
4	2	f_1	1	1
6	4	f_1	1	3
8	4	f_1	1	3
10	4	f_1	1	3
12	4	f_1	1	3
14	6	f_1	1	6
16	6	f_1	1	4
18	6	f_1	1	6
20	8	f_1	1	5
22	8	f_1	1	125
24	8	f_1	1	5
26	8	f_1	1	11
28	10	f_1	5	45
30	8	f_1	1	80
32	10	f_1	1	7
34	10	f_1	1	81
36	10	f_1	1	7
38	10	f_1	1	9
40	12	f_1	1	14
42	10	f_1	1	9
44	12	f_1	1	10
46	12	f_1	1	16
48	14	f_2	17	57
50	12	f_1	1	11
52	14	f_1	1	15
54	14	f_1	1	694
56	14	f_1	1	11
58	14	f_1	1	25
60	16	f_1	1	51
62	14	f_1	1	14
64	16	f_1	1	14
66	16	f_1	1	26
68	18	f_1	1	1282
70	16	f_1	1	16
72	18	f_1	1	15

Les codes auto – duaux hermitien sur $\mathbb{F}_4$

Tableau 6

Le cas euclidien

Pour construire M on prend ω un élément de $\mathbb{F}_4$ vérifions : $\omega^2+\omega+1=0$ et $\omega^3=1$.

$$B_{ini}=\begin{bmatrix}1 & \omega\\ \omega & 1\end{bmatrix} \qquad et \qquad G_{ini}=\begin{bmatrix}1 & 1 & & & & \\ & & . & & 0 & \\ & & & . & & \\ & 0 & & & . & \\ & & & & 1 & 1\end{bmatrix}$$

n	d	f_i	a	r
2	2	f_1	1	1
4	2	f_1	1	1
6	3	f_1	1	2
8	4	f_1	1	2
10	4	f_1	1	2
12	4	f_1	1	3
14	6	f_1	1	3
16	6	f_1	1	3
18	6	f_1	1	3
20	6	f_1	1	8
22	8	f_1	1	6
24	8	f_2	5	5
26	8	f_1	1	5
28	8	f_1	1	5
30	10	f_2	11	15
32	10	f_2	13	6
34	10	f_2	3	8
36	10	f_2	5	21
38	11	f_2	5	74
40	12	f_1	1	7
42	12	f_2	11	17
44	12	f_2	13	11
46	13	f_2	35	83
48	12	f_2	5	5
50	14	f_2	11	14
52	14	f_2	3	22
54	14	f_2	7	7
56	14	f_1	1	15
58	15	f_1	1	244
60	15	f_2	7	54
62	16	f_2	3	14
64	16	f_1	5	12
66	16	f_1	1	11
68	17	f_1	1	145
70	18	f_1	43	448
72	18	f_2	7	728

Les codes auto – duaux euclidiens sur $\mathbb{F}_4$

Tableau 7

- **Sur $\mathbb{F}_5$**

Sur le corps $\mathbb{F}_5$ on suppose que :

$$B_{ini} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \qquad et \qquad G_{ini} = \begin{bmatrix} 1 & 2 & & & \\ & & . & & 0 \\ & & & . & \\ & 0 & & & \\ & & & & 1 \quad 2 \end{bmatrix}$$

Exemple 3.8 *On obtient les codes de paramètres [18,9,7] et [22,11,8] à partir de B_{ini}, G_{ini} et $(f_1; 1, 9)$ et $(f_1; 1, 4)$ respectivement, de matrices*

génératrices :

$$G_9 = \begin{bmatrix} 1 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 3 & 1 & 2 & 4 & 2 & 3 & 4 & 3 & 4 & 4 \\ 4 & 4 & 1 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 3 & 1 & 2 & 4 & 2 & 3 & 4 & 3 \\ 4 & 3 & 4 & 4 & 1 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 3 & 1 & 2 & 4 & 2 & 3 \\ 2 & 3 & 4 & 3 & 4 & 4 & 1 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 3 & 1 & 2 & 4 \\ 2 & 4 & 2 & 3 & 4 & 3 & 4 & 4 & 1 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 3 & 1 \\ 3 & 1 & 2 & 4 & 2 & 3 & 4 & 3 & 4 & 4 & 1 & 3 & 3 & 2 & 1 & 2 & 1 & 1 \\ 1 & 1 & 3 & 1 & 2 & 4 & 2 & 3 & 4 & 3 & 4 & 4 & 1 & 3 & 3 & 2 & 1 & 2 \\ 1 & 2 & 1 & 1 & 3 & 1 & 2 & 4 & 2 & 3 & 4 & 3 & 4 & 4 & 1 & 3 & 3 & 2 \\ 3 & 2 & 1 & 2 & 1 & 1 & 3 & 1 & 2 & 4 & 2 & 3 & 4 & 3 & 4 & 4 & 1 & 3 \end{bmatrix}$$

et

$$G_4 = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 2 & 1 & 2 & 1 & 1 \\ 1 & 1 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 2 & 1 & 2 \\ 1 & 2 & 1 & 1 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 2 \\ 3 & 2 & 1 & 2 & 1 & 1 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 2 & 1 & 2 & 1 & 1 & 4 & 0 \end{bmatrix}$$

respectivement.

- **Sur $\mathbb{F}_7$**

Pour

$$B_{ini} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \qquad et \qquad G_{ini} = \begin{bmatrix} 1 & 0 & 2 & 3 & & & & & & \\ 0 & 1 & 4 & 2 & & & & 0 & & \\ & & & & . & & & & & \\ & & & & & . & & & & \\ & & & & & & . & & & \\ & & 0 & & & & & 1 & 0 & 2 & 3 \\ & & & & & & & 0 & 1 & 4 & 2 \end{bmatrix}$$

Nous considérons les matrices B_{ini} et G_{ini} pour obtenir les codes de paramètres [12,6,6], [16,8,6] et [28,14,10] suivant les permutations : $(f_1; 1, 3), (f_1; 1, 3)$

et $(f_1; 1, 11)$ respectivement, de matrices génératrices :

$$G_3 = \begin{bmatrix} 2 & 6 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 5 & 3 & 0 \\ 0 & 2 & 5 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 6 & 3 \\ 1 & 5 & 3 & 0 & 2 & 6 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 3 & 0 & 2 & 5 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 5 & 3 & 0 & 2 & 6 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 6 & 3 & 0 & 2 & 5 & 0 \end{bmatrix}$$

et

$$G_3 = \begin{bmatrix} 2 & 6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 5 & 3 & 0 \\ 0 & 2 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 6 & 3 \\ 1 & 5 & 3 & 0 & 2 & 6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 6 & 3 & 0 & 2 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 5 & 3 & 0 & 2 & 6 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 6 & 3 & 0 & 2 & 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 5 & 3 & 0 & 2 & 6 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 6 & 3 & 0 & 2 & 5 & 0 \end{bmatrix}$$

et : $G_{11} =$

$$\begin{bmatrix} 3 & 6 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 6 & 4 & 3 & 4 & 1 & 2 & 3 & 5 & 4 & 0 & 3 & 4 & 1 & 5 & 0 & 3 & 4 & 3 & 6 \\ 5 & 2 & 5 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 2 & 4 & 2 & 2 & 5 & 3 & 6 & 2 & 0 & 3 & 3 & 3 & 2 & 0 & 5 & 3 & 5 \\ 3 & 4 & 3 & 6 & 3 & 6 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 4 & 6 & 3 & 4 & 1 & 2 & 3 & 5 & 4 & 0 & 3 & 4 & 1 & 5 & 0 \\ 0 & 5 & 3 & 5 & 5 & 2 & 5 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 2 & 4 & 2 & 2 & 5 & 3 & 6 & 2 & 0 & 3 & 3 & 3 & 2 \\ 4 & 1 & 5 & 0 & 3 & 4 & 3 & 6 & 3 & 6 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 6 & 4 & 3 & 4 & 1 & 2 & 3 & 5 & 4 & 0 & 3 \\ 3 & 3 & 3 & 2 & 0 & 5 & 3 & 5 & 5 & 2 & 5 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 2 & 4 & 2 & 2 & 5 & 3 & 6 & 2 & 0 \\ 5 & 4 & 0 & 3 & 4 & 0 & 5 & 0 & 3 & 4 & 3 & 6 & 3 & 6 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 6 & 4 & 3 & 4 & 1 & 2 & 3 \\ 3 & 6 & 2 & 0 & 3 & 3 & 3 & 2 & 0 & 5 & 3 & 5 & 5 & 2 & 5 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 2 & 4 & 2 & 2 & 5 \\ 4 & 1 & 2 & 3 & 5 & 4 & 0 & 3 & 4 & 1 & 5 & 0 & 3 & 4 & 3 & 6 & 3 & 6 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 6 & 4 & 3 \\ 4 & 2 & 2 & 5 & 3 & 6 & 2 & 6 & 3 & 3 & 3 & 2 & 0 & 5 & 3 & 5 & 5 & 2 & 5 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 2 \\ 1 & 6 & 4 & 3 & 4 & 1 & 2 & 3 & 5 & 4 & 0 & 3 & 4 & 1 & 5 & 0 & 3 & 4 & 3 & 6 & 3 & 6 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 2 & 0 & 2 & 4 & 2 & 2 & 5 & 3 & 6 & 2 & 0 & 3 & 3 & 3 & 2 & 0 & 5 & 3 & 5 & 5 & 2 & 5 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 6 & 4 & 3 & 4 & 1 & 2 & 3 & 5 & 4 & 0 & 3 & 4 & 1 & 5 & 0 & 3 & 4 & 3 & 6 & 3 & 6 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 2 & 0 & 2 & 4 & 2 & 2 & 5 & 3 & 6 & 2 & 0 & 3 & 3 & 3 & 2 & 0 & 5 & 3 & 5 & 5 & 2 & 5 & 0 \end{bmatrix}$$

respectivement.

n	d	f_i	a	r
4	3	f_1	1	1
8	4	f_1	1	2
12	6	f_1	1	3
16	6	f_1	1	3
20	6	f_1	1	10
24	9	f_1	1	25
28	10	f_1	1	11
32	11	f_1	1	54
36	12	f_1	1	32
40	13	f_1	1	74
44	14	f_1	1	26
48	15	f_1	1	94
52	15	f_1	1	11
56	16	f_1	1	10

Les codes auto – duaux sur $\mathbb{F}_7$

Tableau 8

Chapitre 4

La construction Cortex à base non auto-dual

L'informatique et la mécanique quantique sont deux des plus importantes théories du 20^e siècle. Elles se combinent élégamment pour former la théorie de l'information, le découverte qu'un ordinateur peut factoriser les grands nombres en temps polynômial est un problème considéré difficile classiquement.

Depuis, plusieurs résultats surprenants montrent que l'ordinateur peut faire des choses qu'on ne pourrait jamais faire classiquement, autant du point de vue algorithmique (comme l'algorithme à proposé).

Comme on vient de voir dans le chapitre précédent, la théorie nous assure l'obtention de certains codes Cortex auto-duaux comme code Cortex à base auto-dual. Dans ce chapitre nous donnons un algorithme qui est concerné à la construction des codes Cortex auto-duaux à base non auto-dual.

C'est un algorithme qui cherche les différents codes de base pour construire un code Cortex auto-dual à partir d'une suite de permutations composée d'une ou de deux permutations.

Nous utilisons la construction que nous présentons dans le chapitre précédent, nous présentons par la suite quelques résultats de ce programme qui traite les codes de longueur supérieur ou égal à 8 et divisible par 4.

Description de l'algorithme

1. Préliminaires

• Soit P_k la partie redondante de la matrice génératrice du code de base.

• Si en prend la suite de permutations $\Pi = \{\pi_1\}$ ou bien $\Pi = \{\pi_1, \pi_2\}$.

• On obtient le code Cortex auto-dual où R est la partie redondante de la matrice génératrice de ce code.

2. Les étapes de la construction de l'algorithme

•**Premier pas :** Détermination de toutes les matrices carrées P_k satisfaisant $P_k \cdot P_k^t \neq -I_k$ pour $k = \overline{2,3}$.

•**Deuxièmes pas :** Détermination des matrices associées à toutes les permutations π de S_{2k}.

•**Troisièmes pas :** Calculer la partie redondante P de la matrice génératrice du code C_k.

•**Quatrième pas :** Calculer la matrice $R = P(\prod_{i=1}^{s} \Pi_i P)$ telle que :

$s = \overline{1,2}$et Vérifier si la relation $RR^t = -I_{4k}$ est satisfaite.

4.1 Résultat numérique

On donne la partie redondante de la matrice génératrice du code de base, la suite de permutations et la partie redondante de la matrice génératrice du code Cortex obtenu.

- **Pour** $k = 2$

Dans le cas ou la suite est composée d'une seule permutation $\Pi = \{\pi\}$.

$P = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 2 & 1 & 4 & 3 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 4 & 3 & 2 & 1 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 4 & 1 & 2 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 4 & 1 & 2 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	$\pi = \begin{pmatrix} 2 & 1 & 4 & 3 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	$\pi = \begin{pmatrix} 4 & 3 & 2 & 1 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

Remarque 4.1 *Nous remarquons que :*

R est la matrice associée à la permutation Π ($\Pi = R$).

Dans le cas où la suite de permutations est composée de deux permutations $\Pi = \{\pi_1, \pi_2\}$.

$P = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 3 & 4 & 1 & 2 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 4 & 3 & 2 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 4 & 3 & 2 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 4 & 3 & 2 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 3 & 2 & 1 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 3 & 2 & 1 & 4 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 3 & 2 & 1 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 3 & 4 & 1 & 2 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 3 & 4 & 1 & 2 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 2 & 1 & 4 & 3 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 2 & 1 & 4 & 3 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 2 & 1 & 4 & 3 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 4 & 3 & 2 & 1 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 2 & 3 & 4 & 1 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 2 & 3 & 4 & 1 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 2 & 3 & 4 & 1 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 4 & 1 & 2 & 3 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 4 & 1 & 2 & 3 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 4 & 1 & 2 & 3 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 4 & 3 & 2 & 1 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 4 & 3 & 2 & 1 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 2 & 1 & 4 & 3 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 2 & 1 & 4 & 3 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 2 & 1 & 4 & 3 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 4 & 3 & 2 & 1 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 2 & 3 & 4 & 1 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 2 & 3 & 4 & 1 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 2 & 3 & 4 & 1 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 4 & 1 & 2 & 3 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 4 & 1 & 2 & 3 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 4 & 1 & 2 & 3 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 4 & 3 & 2 & 1 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 2 & 1 & 4 & 3 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 4 & 3 & 2 & 1 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 4 & 3 & 2 & 1 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 3 & 4 & 1 & 2 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 4 & 3 & 2 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 4 & 3 & 2 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 4 & 3 & 2 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 3 & 2 & 1 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 3 & 2 & 1 & 4 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 3 & 2 & 1 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 3 & 4 & 1 & 2 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 3 & 4 & 1 & 2 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

4.2 Etude de l'équivalence des codes cortex obtenus :

Soit P_k la partie redondante de la matrice génératrice du code de base. Pour une suite de permutations $\Pi = \{\pi_1, \pi_2, ..., \pi_s\}$ avec $(s = \overline{1,2})$ on construit la partie redondante R du code cortex auto-dual éventuel

$$R = P(\prod_{i=1}^{s} \Pi_i P)$$

4.2.1 Pour k=2, dans le cas où $|\Pi| = 1$

Proposition 4.1 *Si $C_{2k}(P_2, \{\pi\})$ est un code Cortex auto-dual alors :*
$C_{2k}(P_2, \{\pi \circ \varepsilon_0\})$ est le seul code Cortex *auto-dual à base P_2 équivalent à $C_{2k}(P_2, \{\pi\})$ et on a :*

$$C_{2k}(P_2, \{\pi \circ \varepsilon_0\}) = C_{2k}(P_2, \pi)^{(\varepsilon_0, Id_{2k})}$$

où :

$$\varepsilon_0(l) = \begin{cases} l+2 & 1 \leq l \leq 2 \\ l-2 & 3 \leq l \leq 4 \end{cases}$$

Preuve. Soient $\varepsilon = (\varepsilon_0, \varepsilon_0)$ et $\eta = (Id_{2k}, Id_{2k})$ (deux éléments du groupe G^*_{2k}), $(f^0, f^2) \in C_{2k}(P_2, \{\pi\})$ alors il existe $f^1 \in \mathbb{F}_2^{2k}$
tel que :

$$\begin{cases} (f^0, f^1) \in C_{2k} \\ (\pi f^1, f^2) \in C_{2k} \end{cases} \Longrightarrow \begin{cases} (\varepsilon_0 f^0, \varepsilon_0 f^1) \in \mathbb{C}_{2k} \\ (\pi \circ \varepsilon_0 \circ \varepsilon_0 f^1, f^2) \in C_{2k} \end{cases}$$
$$\Longrightarrow (\varepsilon_0 \circ f^0, f^2) \in C_{2k}(p_2, \{\pi \circ \varepsilon_0\})$$

L'unicité est obtenue numériquement. ■

Prposition 4.2 *Soit J la matrice associée à la permutation $\sigma = \begin{pmatrix} 2 & 1 \end{pmatrix}$.Les codes Cortex $C_{2k}(P_2, \{\pi\})$ et $C_{2k}(J^k P_2 J^{k'}, \{\varepsilon^k \circ \pi \circ \varepsilon^{k'}\})$ pour :*

$$(k, k' \geq 0) \text{ et } \Pi_\varepsilon = \begin{bmatrix} J & 0 \\ 0 & J \end{bmatrix}$$

sont équivalents.

Corollaire 4.1 *Soit J la matrice associée à la permutation $\sigma = \begin{pmatrix} 2 & 1 \end{pmatrix}$. Alors les codes Cortex $C_{2k}(P_2, \{\pi\})$ et $C_{2k}(J^k P_2 J^{k'}, \{\varepsilon^k \circ \pi \circ \varepsilon^{k'} \circ \varepsilon_0^l\})$ pour :*

$$(k, k', l \geq 0) \text{ et } \varepsilon_0(t) = \begin{cases} t+2 & 1 \leq t \leq 2 \\ t-2 & 3 \leq t \leq 4 \end{cases}$$

sont équivalents.

Preuve. D'après la proposition 4.2, les codes $C_{2k}(P_2, \{\pi\})$ et $C_{2k}(J^k P_2 J^{k'}, \{\varepsilon^k \circ \pi \circ \varepsilon^{k'}\})$ sont équivalents. Or d'après la proposition 4.1, es codes $C_{2k}(P_2, \{\pi\})$ et $C_{2k}(J^k P_2 J^{k'}, \{\varepsilon^k \circ \pi \circ \varepsilon^{k'} \circ \varepsilon_0^l\})$ sont équivalents. ■

4.2.2 Pour k=2, dans le cas où $|\Pi| = 2$

Proposition 4.3 *Soient π_1, π_2 deux permutations de S_{2k}, si $C_{2k}(P_2, \{\pi_1, \pi_2\})$ est un code Cortex auto-dual alors :*

1/ $C_{2k}(P_2, \{\pi_1, \pi_2\}) = C_{2k}(P_2, \{\pi_2, \pi_1\})$

2/ Les codes $C_{2k}(P_2, \{\pi_1, \pi_2 \circ \varepsilon_0\})$ et $C_{2k}(P_2, \{\pi_1, \pi_2\})$ sont équivalent et on a :

$$C_{2k}(P_2, \{\pi_1, \pi_2 \circ \varepsilon_0\}) = C_{2k}(P_2, \{\pi_1, \pi_2\})^{(Id_{2k}, \varepsilon_0)}$$

3/ Les codes $C_{2k}(P_2, \{\pi_1 \circ \varepsilon_0, \pi_2\})$ et $C_{2k}(P_2, \{\pi_1, \pi_2\})$ sont équivalent et on a :

$$C_{2k}(P_2, \{\pi_1 \circ \varepsilon_0, \pi_2\}) = C_{2k}(P_2, \{\pi_1, \pi_2\})^{(\varepsilon_0, Id_{2k})}$$

4/Les codes $C_{2k}(P_2, \{\pi_1 \circ \varepsilon_0, \pi_2 \circ \varepsilon_0\})$ *et* $C_{2k}(P_2, \{\pi_1, \pi_2\})$ *sont équivalents.*

Preuve. : Soient $\varepsilon = (\varepsilon_0, \varepsilon_0)$ et $\eta = (Id_{2k}, Id_{2k})$ (deux permutations du groupe G^*_{2k}).

1/ Soit (a^0, a^3) un mot code de $C_{2k}(P_2, \{\pi_1, \pi_2\})$ alors ils existent a^1, a^2deux vecteurs de $\mathbb{F}_2^{2k}$ tels que :

$$\left\{\begin{array}{c}(a^0,a^1)\in C_{2k}\\(\pi_1 a^1,a^2)\in C_{2k}\\(\pi_2 a^2,a^3)\in C_{2k}\end{array}\right. \Longrightarrow \left\{\begin{array}{c}(a^0,a^1)\in C_{2k}\\(\pi_1 a^1,a^2)\in C_{2k}\\(\pi_2\circ\varepsilon_0 a^2,\varepsilon_0 a^3)\in C_{2k}\end{array}\right. \begin{cases}\text{car} & \varepsilon_0\circ\pi_2=\pi_2\circ\varepsilon_0\\ \text{et} & \varepsilon_0^2=Id_{2k}\end{cases}$$

$$\Longrightarrow \left\{\ (a^0,\varepsilon_0 a^1)\in C_{2k}(P_2,\{\pi_1,\pi_2\circ\varepsilon_0\})\right.$$

3/Soient (f^0, f^3) un mot code de $C_{2k}(P_2, \{\pi_1, \pi_2\})$ alors ils existent f^1, f^2deux vecteurs de $\mathbb{F}_2^{2k}$ tels que :

$$\left\{\begin{array}{c}(f^0,f^1)\in C_{2k}\\(\pi_1 f^1,f^2)\in C_{2k}\\(\pi_2 f^2,f^3)\in C_{2k}\end{array}\right. \Longrightarrow \left\{\begin{array}{c}(f^0,f^1)\in C_{2k}\\(\varepsilon_0\circ\pi_1\circ\varepsilon_0\circ\varepsilon_0 f^1,\varepsilon_0 f^2)\in C_{2k}\\(\ \pi_2\circ\varepsilon_0\circ\varepsilon_0 f^2,f^3)\in C_{2k}\end{array}\right.$$

$$\Longrightarrow \left\{\begin{array}{c}(f^0,f^1)\in C_{2k}\\(\pi_1\circ\varepsilon_0 f^1,\varepsilon_0 f^2)\in C_{2k}\\(\ \pi_2\circ\varepsilon_0\circ\varepsilon_0 f^2,f^3)\in C_{2k}\end{array}\right.$$

$$\Longrightarrow \left\{\ (f^0,f^3)\in C_{2k}(P_2,\{\pi_1\circ\varepsilon_0,\pi_2\circ\varepsilon_0\})\right.$$

ce qui achève la preuve. ■

Proposition 4.4 *Les codes* $C_{2k}(J^k P_2 J^{k'}, \{\varepsilon^k \circ \pi_1 \circ \varepsilon^{k'}, \varepsilon^k \circ \pi_2 \circ \varepsilon^{k'}\})$ *et* $C_{2k}(P_2, \{\pi_1, \pi_2\})$*pour :*

$$k, k' \geq 0 \text{ et } \Pi_\varepsilon = \begin{bmatrix} J & 0 \\ 0 & J \end{bmatrix}$$

et J *la matrice associée à la permutation* $\sigma = \begin{pmatrix} 2 & 1 \end{pmatrix}$ *sont équivalents.*

Théorème 4.1 *Les codes* $C_{2k}(P_2, \{\varepsilon^k \circ \pi_1 \circ \varepsilon^{k'} \circ \varepsilon_0^l, \varepsilon^k \circ \pi_2 \circ \varepsilon^{k'} \circ \varepsilon_0^l\})$ *et* $C_{2k}(P_2, \{\pi_1, \pi_2\})$ *pour :* $k, k', l \geq 0$ *sont équivalents.*

Preuve. En appliquant les proposition 4.3 et 4.4. ■

- **pour** $k = 3$

Dans le cas ou la suite de permutations est composée d'une seule permutation $\Pi = \{\pi\}$.

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 2 & 1 & 6 & 5 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 5 & 1 & 6 & 2 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 2 & 1 & 6 & 5 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 2 & 4 & 6 & 5 & 1 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 1 & 2 & 6 & 4 & 5 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 1 & 2 & 6 & 5 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 2 & 1 & 6 & 4 & 5 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 2 & 1 & 6 & 5 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 2 & 1 & 6 & 5 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 5 & 4 & 6 & 2 & 1 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 2 & 3 & 1 & 5 & 6 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 2 & 3 & 1 & 6 & 5 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 3 & 2 & 1 & 5 & 6 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$

$P = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	$\pi = \begin{pmatrix} 4 & 6 & 5 & 1 & 3 & 2 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

Remarque 4.2 1/ *Il y a 492 résultat dans ce cas.*

Dans le cas où la suite de permutations est composée de deux permutations $\Pi = \{\pi_1, \pi_2\}$.

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 2 & 6 & 4 & 5 & 3 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 2 & 6 & 4 & 5 & 3 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 2 & 6 & 4 & 5 & 3 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 5 & 6 & 4 & 2 & 3 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 4 & 5 & 3 & 2 & 6 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 4 & 6 & 3 & 2 & 5 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 4 & 6 & 3 & 2 & 5 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 4 & 3 & 6 & 5 & 2 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 3 & 2 & 5 & 6 & 4 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 6 & 5 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 3 & 2 & 5 & 6 & 4 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 2 & 3 & 5 & 6 & 4 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 5 & 4 & 2 & 6 & 3 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 5 & 4 & 2 & 3 & 6 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 5 & 4 & 2 & 6 & 3 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 5 & 4 & 2 & 6 & 3 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 3 & 2 & 4 & 6 & 5 \end{pmatrix}$	$R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 6 & 5 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 2 & 3 & 6 & 4 & 5 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 3 & 2 & 6 & 4 & 5 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 3 & 2 & 6 & 4 & 5 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$

$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\pi_1 = \begin{pmatrix} 1 & 6 & 2 & 4 & 3 & 5 \end{pmatrix}$ $\pi_2 = \begin{pmatrix} 1 & 3 & 5 & 4 & 6 & 2 \end{pmatrix}$	$R = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$

Remarque 4.3 *Il y a 74 résultats dans ce cas.*

4.2.3 Pour k=3, dans le cas où $|\Pi| = 1$

Proposition 4.5 *Soit π une permutation de S_{2k}.Les codes $C_{2k}(J^k P_3 J^{k'}, \{\varepsilon^k \circ \pi \circ \varepsilon^{k'}\})$ et*
$C_{2k}(P_3, \{\pi\})$ pour :

$$k, k' \geq 0 \text{ et } \Pi_\varepsilon = \begin{bmatrix} J & 0 \\ 0 & J \end{bmatrix}$$

et J la matrice associée à la permutation $\sigma = \begin{pmatrix} 2 & 1 \end{pmatrix}$ sont équivalents.

Proposition 4.6 *Soient $C_{2k}(P_3, \{\pi\})$ est un code Cortex auto-dual et $\pi' = \sigma \circ \pi$ où σ est un élément de ces groupes $G(P_3)[\sigma_1], G(P_3)[\sigma'_1, \sigma'_2]$ et $G(P_3)[\sigma'_1, \sigma'_2, \sigma'_3]$ alors :*
les codes $C_{2k}(P_3, \{\pi\})$ et $C_{2k}(P_3, \{\pi'\})$ sont équivalents.

Remarque 4.4 On a un code auto-dual
1/ code Cortex à base auto-dual
2/ code Cortex à base non auto-dual

Annexe :

Algorithme (La construction Cortex auto- duaux à base non auto-dual)

Constantes

```
 e=2
 max=14
```

Types

```
 Vect=tableau(max)de entier
 Mat=matrice(max,max) de entier
```

Variables

```
 K :entier
 I,J,S,M :entier //long entier
 TAB,TabApp :vect
 IK,B,P,PI,A : Mat //matrices principales
 M1,M2 : Mat  //matrices utilitaires
```

Début

```
 Ecrire ('entrer la valeur de K :')
 Lire (K)      //K est la dimension de la matrice B
 genereI(K*e)  //générer la matrice unité IK de dim k*e
 InitTAB(k) // initialiser le vecteur unité TAB avec (1,2,3,…k)
 i←1
 TantQue(i ≤ 2^k*k) faire
  GenereB(B,i,k) //générer la ième matrice B de dim K
  GenereP(P,e,B) //générer la matrice P à partir de B et de e
  Transpose(B,k,M1) //calculer la transposé de B dans M1
  ProdMat(B,M1,k,M2) //calculer M2=B*M1
  Transforme(M2,K) //substituer les valeurs paires de M2 par 0 et les
                   //impaires par 1
  Si non EgaleI(M2,K) alors
   J←1
   TantQue j≤(k*e)! faire
    CPT←0 //compteur du Rang du vecteur TabApp à générer
    App(j,cpt,TAB,TabApp,k*e)//generation du vecteur application
                             //TabApp numero j
    Permut(TabApp,PI,K*e) //generation de la permutation PI à partir
                          //TabApp numero j
    ProdMat(P,PI,k*e,M1) //calculer M1=P*PI
    ProdMat(M1,P,k*e,A) //calculer A=M1*P
    Transpose(A,k*e,M1) //calculer la transposé de A dans M1
    ProdMat(A,M1,k*e,M2) //calculer M2=A*M1
    Transforme(M2,K*e)
    Si EgaleI(M2,K*e) alors
     Afficher1(B,PI)
    fsi
    J←j+1
   FTQ
```

```
    S←1
    TantQue S≤(k*e)! faire
     CPT←0 //compteur du Rang du vecteur TabApp à générer
     App(S,cpt,TAB,TabApp,k*e)//generation du vecteur application
                              //TabApp numero S
     Permut(TabApp,PS,K*e) //generation de la permutation PS à partir
                           //TabApp numero S
     M←1
     TantQue M≤(k*e)! faire
      CPT←0 //compteur du Rang du vecteur TabApp à générer
      App(M,cpt,TAB,TabApp,k*e)//generation du vecteur application
                               //TabApp numero M
      Permut(TabApp,PI,K*e) //generation de la permutation PI à partir
                            //TabApp numero M

      ProdMat(P,PS,k*e,M1) //calculer M1=P*PS
      ProdMat(M1,P,k*e,M2) //calculer M2=M1*P
      ProdMat(M2,PI,k*e,M1) //calculer M1=M2*PI
      ProdMat(M1,P,k*e,A) //calculer A=M1*P⇒ A=P*PS*P*PI*P
      Si non EgaleI(A,K*e) alors
       Transpose(A,k*e,M1) //calculer la transposé de A dans M1
       ProdMat(A,M1,k*e,M2) //calculer M2=A*M1
       Transforme(M2,K*e)
       Si EgaleI(M2,K*e) alors
        Afficher2(B,PS,PI)
       Fsi
      Fsi
      M←M+1
     FTQ
     S←S+1
    FTQ

   FSI
   i←i+1
  FTQ
FIN
```

Conclusion :

A partir d'un code de base donné on a des conditions suffisantes sur les suites de permutations pour l'obtention des codes Cortex équivalents ceci motive la représentation des codes auto-duaux comme code Cortex pour ainsi déterminer leurs équivalence. Nous donnons cette représentation pour les codes auto-duaux de longueurs inférieur ou égal à 20. Aussi, un code de base non nécessairement auto-dual peut générer un code Cortex auto-dual. Nous avons déterminé les matrices d'ordre 2 et 3 comme matrice redondante de certains codes non auto-duaux, ainsi que l'équivalence de ces codes générés.Peut on représenter des codes auto-duaux de longueurs plus grandes pour ainsi déterminer leurs équivalences ? Peut on augmenter l'ordre de matrice de redondance au code de base ?

Bibliographie

[1] **I. BOUYUKLIEV** and **PATRIC R. J. Östergård**–"*Classification of Self-Orthogonal Codes over* $\mathbb{F}_3$ *and* $\mathbb{F}_4$*",* SIAM Journal on Discrete Mathematics, vol**.** 19, p**.** 363-370 **(2005)**.

[2] **S. BUYUKLIEVA**–"*On the Binary Self-Dual Codes with an Automorphism of Order 2",* Designs, Codes and Cryptography, vol**.** 12, p**.** 39–48 **(1997)**.

[3] **E. CADIC**–"Construction de Turbo Codes Courts possédant de bonnes propriétés de distance minimale", Thèse de doctorat, Université De Limoges **(2003)**.

[4] **W. CARY HUFFMAN**–"*On the classification and enumeration of self-dual codes",* Finite Fields and Their Application,vol**.** 11, p**.** 451-490 **(2005)**.

[5] **N. J. H CONWAY** and **N. J. A. SLOANE**–"*A New Upper Bound on the Minimal Distance of Self-Dual Codes",* IEEE Transactions on Information Theory, vol**.** 36. p**.** 1319–1333 **(1990)**.

[6] **D. B. DALAN**–"*Type I neighbors of extremal type II codes of length 40 derived from Hadamard matrices",* Discrete Mathematics, vol**.** 259, p**.** 285-291 **(2002)**.

[7] **P. GABORIT** et **A. OTMANI**–*"Experimental Constructions of Self-Dual Codes",* Finite Fields and their Applications **(2002)**.

[8] **M. HARADA** and **T. A. GUKKIVER** –*"Extremal Binary Self-Dual Codes",* IEEE Transactions on Information Theory , p**.** 2036–2047 **(1997)**.

[9] —,*"Optimal Formally Self-Dual Codes over*$\mathbb{F}_5$ *and* $\mathbb{F}_7$*",* AAECC, 10, p**.** 227–236 **(2000)**.

[10] —,*"Optimal Ternary Formally Self-Dual Codes",* Discrete Mathematics,vol**.** 196, p**.** 117–135 **(1999)**.

[11] **JON-LARK KIM**–*"Euclidean Type I Codes over* $GF(4)$*"*, à paraitre dans Ars Combinatoria **(2007).**

[12] **R. LIDL et G. PILZ**–"Applied Abstract Algebra",Springer- Verlag , p**.** 188–195 **(1991)**.

[13] **G.NAANAA**–"Etude comparaive des codes classiques ", Thèse de Magister, Université de Batna, **(2001)**

[14] **G. NEBE, H.-G. QUEBBEMANN, E. M. RAINS** and **N. J. A. SLOANE**–"*Complete Weight Enumerators of Generalized Doubly-Even Self-Dual Codes"*, arXiv :math.NT/0311289, vol**.** 1, **(2003)**.

[15] **A. OTMANI**–"Codes Cortex et construction de codes auto-duaux optimaux", Thèse de doctorat, Université De Limoges,**(2002)**.

[16] **A. OTMANI** et **G. KAHN**–*"Caractérisation des codes auto-duaux binaires de type II à partir du code de Hamming étendu [8,4,4]",* C R Acad Sci, vol**.** 336, p**.** 971–974 **(2003)**.

[17] **A. A. PANTCHICHKINE**–"Mathématique des codes correcteurs d'erreurs", http ://www-fourier.ujf-grenoble.fr/~panchish/cv11**(2006)**.

[18] **C. PAQUIN**–"Les codes correcteurs quantiques et leurs applications cryptographiques", Mémoire de (M. Sc.),Université de Montréal **(2000)**.

[19] **A. POLI, L. I. HUGUET**–"Codes correcteurs", Théorie et Applications, Masson, Paris, **(1988)**.

[20] **E. M. RAINS** and **N. J. A. SLOANE**–*"Self dual codes",* arXiv :math.CO/0208001, vol**.** 1 **(2002)**.

[21] **S. ROMAN** –"Coding and Information Theory",Springer-Verlag , **(1991)**.

[22] **CARMAN-SIMONA NEDELOAIA**–"Étude des énumérateurs des poids des codes linéaires utilisant des formes décomposées des matrices génératrices", Thèse de doctorat, Université de Limoges **(2005)**.

Résumé

Cette thèse est consacrée à l'étude de la construction des codes Cortex auto-duaux à base auto-dual ou bien non. Les codes Cortex introduis fournissant un moyen simple de construire des codes auto-duaux binaires lorsque le code de base l'est [2.1]. Ils offrent de plus une méthode efficace pour construire des codes extrémaux lorsque le code de Hamming étendu H_8 de longueur 8 est le code de base.

Cette construction permet en effet d'obtenir des codes auto-duaux extrémaux de type II.

Par la suite, En utilisant un algorithme qui calcule la suite de permutation, le code de base non auto-dual et le code Cortex auto-dual obtenu.

Mot clé : Code linéaire, code auto-dual, polynôme énumérateur des poids, code Cortex, permutation, matrice génératrice.

Abstract

This thesis is devoted to the study of the construction of the self-dual codes Cortex the self-dual at base or not. The Cortex codes introduce providing a means simple to build binary self-dual codes when the basic code is [2.1]. They offer moreover one effective method to build extremal codes when the Hamming code wide H_8 of length 8 is the basic code

This construction indeed makes it possible to obtain extremal self-dual codes of type II.

Thereafter, by using an algorithm which calculation the continuation of permutation, the no self-dual basic code and the self-dual codes Cortex obtained.

Key words : linear code, self-dual code, weight enumurateur, Cortex Codes, permutation, generator matrix.