Table des matières

Introduction générale

La cryptographie et la théorie des codes sont des domaines distincts mais il existe des liens puissants entre ces deux sciences.

Le but de la cryptographie est de sécuriser l'information contre ceux qui ne sont pas habilités à en prendre connaissance. La théorie des codes correcteurs d'erreurs quand à elle a pour but de sécuriser l'information contre d'eventuels erreurs inevitables qu'elle subit lors de son transfert ou son stockage.

La cryptographie existait avant notre ère, elle fut autrefois l'art de secret et elle est aujourd'hui la science du secret. Elle s'est longtemps limitée au domaine militaire, elle possède aujourd'hui de nombreuses applications dans le civil. Elle fait partie des deux composantes de la cryptologie, la seconde étant la cryptanalyse.

La cryptanalyse est une discipline dont le but est d'analyser la sécurité des systèmes cryptographiques, et permet donc de connaitre leurs faiblesses et résistance contre toute attaque.

Avant les années 70, la sécurité d'un cryptosytème se basait sur la connaissance de sa clé, dite clé secrete utilisée pour le chiffrement et le déchiffrement.

Ces cyptosystèmes avaient le problème de la distribution de la clé secrete au utilistateurs et ils etaient moins sécurisés. C'est qu'en 1976 que Diffie et Hellman ont inventés le premier cryptosysteme à clé publique où le probleme de la distribution des clés est résolu, et sa sécurité repose seulement sur la difficulté de trouver l'inverse d'une fonction à sens unique.

Les codes correcteurs sont encore utilisés dans diverses technologies de communication

(ADSL, fibre optique et USB entre autres).

Leur histoire remonte à la fin des années 50 avec l'apparition de la théorie de l'information par C.Shannon et la théorie des codes avec R. Hamming, M. Golay ... etc.

Dès 1978, McEliece a imaginé le premier et le plus célèbre des crypto systèmes à clef publique utilisant des codes correcteurs d'erreurs.

Ce mémoire est composé d'une introduction générale, conclusion et trois chapitre.

Le premier chapitre est consacré à quelques notions algébriques : groupes, anneaux, le corps fini.

Dans le deuxième chapitre on traite en details les codes lineaires et les codes cycliques, et leur propriétés.

Dans le dernier chapitre on commence par donner quelques notions de base de la cryptographié . Ensuite on présente l'un des plus célèbre cryptosystème basés sur les codes correcteurs du à McEliece en utilisant les matrices génératrices. On termine ce chapitre par présenter un autre cryptosystème basés sur les codes correcteurs du à Neidereitter mais en se basant sur les codes cycliques au lieu des codes de Goppa et en utilisant les matrices de contrôle.

Chapitre 1

Notions d'algèbre.

Dans ce chapitre on va présenter quelques notions et structures algébriques : les groupes, les anneaux, les corps finis et en particulier le corps de Galois. On termine par la définition d'une matrice de permutation et quelques propriétés de ce type. Pour vérifier les calculs voir [2], [3], [5], [4], [10].

1.1 Groupes.

Définition 1.1.1 *Soit G un ensemble muni d'une loi interne notée $(.)$. $(G,.)$ est un groupe si et seulement si :*

1. *La loi $(.)$ est associative.*
2. *La loi $(.)$ admet un élément neutre noté 1.*
3. *Tout élément x de G admet un symétrique unique par la loi $(.)$, noté x^{-1}.*

Remarque 1.1.2

1. *Si la loi est notée additivement c-à-d $(+)$ l'élément neutre est noté 0, et le symétrique de x est noté $(-x)$.*
2. *Si la loi $(.)$ est commutative le groupe $(G,.)$ est dit commutatif.*

Exemple 1.1.3 *$(\mathbb{Z},+)$, $(\mathbb{R},+)$, $(\mathbb{C},+)$ sont des groupes abéliens.*

Proposition 1.1.4 *Soit $(G,.)$ un groupe et $H \subset G$, H est dit sous-groupe de G et on note $H \leqslant G$ si et seulement si :*

1. $H \neq \phi$.
2. $\forall\ x,\ y \in H \Rightarrow x.y \in H$.
3. $\forall\ x \in H \Rightarrow x^{-1} \in H$.

Définition 1.1.5 *Un sous-groupe H de G est dit normal dans G (on note $H \lhd G$) si et seulement si : $\forall\ x \in G,\ h \in H \Rightarrow x^{-1}.h.x \in H$.*

Exemple 1.1.6

1. $\mathbb{R}_+^* \lhd (\mathbb{R}^*,.)$.
2. $\mathbb{R}_n[X] \lhd (\mathbb{R}[X],+)$.
3. $n\mathbb{Z} \lhd (\mathbb{Z},+)$.

Remarque 1.1.7 *Tout sous-groupe H d'un groupe abélien G est un sous-groupe normal.*

Définition 1.1.8 *Soit $(G,.)$ un groupe et H un sous-groupe normal de G, la relation $\Re$ définie par : $x,\ y \in G$; $x\ \Re\ y\ \Leftrightarrow x.y^{-1} \in H$ est une relation d'équivalence compatible avec la loi $(.)$ et l'opération définie dans G/H par : $\overline{x}.\overline{y} = \overline{x.y}$ est une loi intèrne, et $(G/H,.)$ est un groupe dit groupe quotient de G par H d'élement neutre H.*

Exemple 1.1.9

1. *Soit $G = (\mathbb{Z},+)$ un groupe abélien et $H = n\mathbb{Z}$ un sous-groupe normal de $\mathbb{Z}$, le groupe quotient $\mathbb{Z}/n\mathbb{Z}$ est définie par : $\mathbb{Z}/n\mathbb{Z} = \{k + n\mathbb{Z},\ k \in \mathbb{Z}\} = \{\overline{r},\ 0 \leqslant r \leqslant n-1\}$ dit ensemble des entiers modulo n.*
2. *Soit $G = (\mathbb{R}[X],+)$ un groupe abélien et $H =< X^2+1 >$, le groupe quotient $\mathbb{R}[X]/ < X^2+1 >$ est un groupe abélien isomorphe au groupe des polynômes de degré inférieur où égal à 1.*

Définition 1.1.10 *Un groupe $(G,.)$ est dit monogène si G admet un générateur $a \in G$. C-à-d : $G =< a >= \{a^k,\quad k \in \mathbb{Z}\}$.*

Si $\mid G \mid = n$ *alors* G *est dit groupe cyclique et tout générateur* a *du groupe* G *est appelé élément primitif du groupe.*
Dans ce cas : $G = \{a^k,\ 0 \leq k \leq n-1\}$.

Remarque 1.1.11 *Si la loi est notée* $(+)$ *on écrit :* $G = \langle a \rangle = \{k.a,\ k \in \mathbb{Z}\}$.

Exemple 1.1.12

1. $(\mathbb{Z}, +)$ *est un groupe monogène engendré par* $a = 1$ *ou* $a = -1$.

2. $(\mathbb{Z}/n\mathbb{Z})$ *est un groupe cyclique. Il est l'unique groupe cyclique d'ordre* n *à isomorphisme près.*

Théorème 1.1.13 *Soit* G *un groupe cyclique d'ordre* n *engendré par* a *et* $H = \{x \in G : x^k = 1,\ k \in \mathbb{N}\}$. *Alors* H *est un sous-groupe cyclique de* G *d'ordre* $m = PGCD(n,\ k)$ *engendré par* $a^{\frac{n}{m}}$.

Définition 1.1.14 *Soit* $(G, .)$, $(G', .)$ *deux groupes, et soit* f *une application de* G *dans* G'. f *est dit morphisme de groupes si et seulement si :* $\forall x,\ y \in G :\ f(x.y) = f(x).f(y)$.

Définition 1.1.15 *Soit* $f : G \to G'$ *un morphisme de groupes.*

1. *L'image de* G *par* f *, notée* imf*, tel que :*
$imf = \{y \in G' \setminus \exists\ x \in G \setminus y = f(x)\} = \{f(x) \setminus x \in G\}$, *est un sous groupe de* G'.

2. *Le noyau de* f *, noté* $Kerf$ *, tel que :* $Kerf = \{x \in G \setminus f(x) = 1_{G'}\}$, *est un sous groupe normal de* G.

Théorème 1.1.16 *Si* $f : G \to G'$ *un morphisme de groupes alors :*
$G/Kerf \simeq imf$ (1^{er}*théorème d'isomophisme*).

1.2 Anneaux.

Définition 1.2.1 *Soit A un ensemble muni de deux lois $(+)$ et $(.)$ alors $(A, +, .)$ est dit anneau si et seulement si :*

1. $(A, +)$ *groupe abélien d'élément neutre* 0.
2. $(.)$ *associative et distributive sur* $(+)$.
3. $(.)$ *admet un élément neutre noté* 1.

Remarque 1.2.2 *Si la loi $(.)$ est commutative alors A est dit anneau commutatif.*

Définition 1.2.3 *Soit $a \in A$ tel que $a \neq 0$, a est dit diviseur de zéro s'il existe $b \in A$ tel que $b \neq 0$ et $ab = 0$.*

Définition 1.2.4 *Un anneau A est dit intègre s'il n'admet pas des diviseur de zéro i.e ;* $\forall\ a,\ b \in A :\ ab = 0 \Rightarrow a = 0$ *ou* $b = 0$.

Exemple 1.2.5

1. $(\mathbb{Z}, +, .)$ *est un anneau commutatif intègre.*
2. $(\mathbb{R}[X], +, .)$ *est un anneau commutatif intègre.*

Définition 1.2.6 *Soit $(A, +, .)$ un anneau. Un élément $a \in A - \{0\}$ est dit inversible dans A s'il existe un élément $b \in A - \{0\}$ tel que : $a.b = b.a = 1$, b est noté a^{-1} dit inverse de a (ou unité). L'ensemble des éléments inversibles est noté $U(A)$.*

Remarque 1.2.7 *$(U(A), .)$ est un groupe dit groupe des unités de A.*

Exemple 1.2.8

1. $U(\mathbb{Z}) = \{1,\ -1\}$.
2. $U(\mathbb{R}[X]) = \mathbb{R}^*$.

Définition 1.2.9 *Un corps $\mathbb{k}$ est un anneau dans lequel tout élément non nul est inversible.*

Définition 1.2.10 *Soit $(A,+,.)$ un anneau et $I \subset A$, I est dit idéal de A si et seulement si :* 1. $I \leq (A,+)$.

2. $\forall x \in I,\ \forall a \in A : x.a \in I$ *et* $a.x \in I$.

Exemple 1.2.11

1. *Si A est commutatif et $a \in A$, l'ensemble $I = \{x.a \setminus x \in A\}$ est un idéal de A dit idéal principal de générateur a, et noté $\langle a \rangle$.*

2. $I = n\mathbb{Z}$ *est un idéal principal de* $\mathbb{Z}$.

Définition 1.2.12 *Un anneau A est dit principal s'il est commutatif, intègre et si tout idéal de A est un idéal principal.*

Exemple 1.2.13

1. *L'anneau* $\mathbb{Z}$ *est un anneau principal.*

2. *L'anneau* $\mathbb{R}[X]$ *est un anneau principal.*

Définition 1.2.14 *Soit A un anneau commutatif intègre, A est dit anneau Euclidienne s'il existe une application* $\Phi : A-\{0\} \to \mathbb{N}-\{0\}$ *vérifiant :*

$\forall\ a \in A,\ \forall\ b \in A-\{0\},\ \exists\ q,\ r \in A : a = bq + r$ *avec* $r = 0$ *où* $\Phi(r) < \Phi(b)$.

Exemple 1.2.15

1. $(\mathbb{Z},+,.)$ *est un anneau Euclidienne avec* $\Phi(k) = \mid k \mid$ *pour* $k \in \mathbb{Z}-\{0\}$.

2. $\mathbb{R}[X]$ *est un anneau Euclidienne avec* $\Phi(P) = d^{\circ}(P)$ *pour* $P \in \mathbb{R}[X] - \{0\}$.

Théorème 1.2.16 *Soient A, A' deux anneaux et f un morphisme d'anneaux de A dans A' alors :* $A/\ ker f\ \simeq im f$.

Définition 1.2.17 *Soit A un anneau commutatif et considérons l'application*

$$f : \mathbb{Z} \to A \text{ définie par : } f(k) = k.1_A = \begin{cases} 1 + ... + 1 & (k \text{ fois}),\ si\ k > 0, \\ 0 & k = 0, \\ -1 - ... - 1 & (-k \text{ fois}),\ si\ k < 0. \end{cases}$$

f est un morphisme d'anneaux de noyau $ker f$ de la forme $n\mathbb{Z}$ tel que $n \in \mathbb{N}$.

L'entier n est appelé la caractéristique de A noté $car(A)$.

Remarque 1.2.18 *$(A,+,.)$ un anneau et $car(A)=n$.*

1. *S'il existe n, c'est le plus petit entier différent de zéro tel que : $n.1_A=0$.*

Sinon $n=0$ dit caractéristique nulle.

2. *Si A est intègre alors $n=0$ si A est infini, ou bien $n=p$ entier premier si A est fini.*

Exemple 1.2.19

1. *$\mathbb{Z}$, $\mathbb{R}$, $\mathbb{C}$ sont des anneaux de caractéristique nulle.*

2. *$A=\mathbb{Z}/p\mathbb{Z}$ (p premier) est un anneau de caractéristique premier.*

3. *$A=\mathbb{Z}/n\mathbb{Z}$ (n non premier) est un anneau de caractéristique non premier.*

Théorème 1.2.20 *Soit $(A,+,.)$ un anneau commutatif et soit I idéal de A, comme $I \lhd (A,+)$ alors $(A/I,+)$ est un groupe abélien, tel que la loi $(+)$ est définie par : $\overline{x}+\overline{y}=\overline{x+y}$, $x,\ y\in A$. La relation d'équivalance définie par :*

$x,\ y\in A$; $x\ \Re\ y \Leftrightarrow x-y\in I$ est compatible avec la loi $(.)$, on définit, dans A/I la loi $(.)$ par : $\overline{x}.\overline{y}=\overline{x.y}$.

Alors $(A/I,+,.)$ est un anneau commutatif dit anneau quotient de A par I .

Exemple 1.2.21

1. *$(\mathbb{Z}/n\mathbb{Z},+,.)$ est un anneau commutatif tel que : $\forall x,\ y\in\mathbb{Z}$:*

$\overline{x}+\overline{y}=\overline{x+y} \Leftrightarrow (x+n\mathbb{Z})+(y+n\mathbb{Z}) = (x+y)+n\mathbb{Z}$.

$\overline{x}.\overline{y}=\overline{x.y} \Leftrightarrow (x+n\mathbb{Z}).(y+n\mathbb{Z}) = (x.y)+n\mathbb{Z}$.

2. *$\mathbb{R}[X]/ < P(X) >$ où $P(X)$ est un polynôme de degré n de $\mathbb{R}[X]$, est un anneau commutatif.*

Théorème 1.2.22 *Soit A un anneau commutatif et I un idéal de A on a alors :*

i) I maximal si et seulement si l'anneau A/I est un corps.

ii) I premier si et seulement si l'anneau A/I est intègre.

Définition 1.2.23 *Un élément a de A est premier (résp. irréductible) si et seulement si l'idéal $< a >$ est un idéal premier (résp. maximal).*

Théorème 1.2.24

i) a est premier si et seulement si $A/ < a >$ *est un anneau intègre.*

ii) a est irréductible si et seulement si l'anneau $A/ < a >$ *est un corps.*

Exemple 1.2.25

1. $\mathbb{Z}/n\mathbb{Z} = < n >$ *est un corps, si et seulement si n est un entier premier.*

2. $P = X^2 + 1 \in \mathbb{R}[X]$ *irréductible alors :* $\mathbb{R}[X] / < P >$ *est un corps isomorphe au corps des complexes* $\mathbb{C}$*.*

1.3 Corps finis.

Proposition 1.3.1 *Soit* $(\Bbbk, +, .)$ *un corps alors :*

i) Si $\Bbbk$ *est infini alors* $car(\Bbbk)$ *est nulle.*

ii) Si $\Bbbk$ *est fini alors* $car(\Bbbk)$ *un entier premier p .*

Théorème 1.3.2 *Si* $\Bbbk$ *est un corps fini de caractéristique un entier premier p, alors* $\Bbbk$ *admet un sous-corps isomorphe au corps premier* $F_p = \mathbb{Z}/p\mathbb{Z}$ *et le cardinal de* $\Bbbk$ *est de la forme* p^n / $n \in \mathbb{N}^*$.

Théorème 1.3.3 *Si* $\Bbbk$ *est un corps commutatif fini alors :*
$(\Bbbk^* = \Bbbk - \{0\}, .)$ *est un groupe cyclique d'ordre* $p^n - 1$ *c-à-d* $\forall x \in \Bbbk : \ x^{p^n} = x$.

Définition 1.3.4 *Soit* $\Bbbk$ *un corps commutatif fini de cardinal* p^n*, alors tout générateur* α *du groupe cyclique* $\Bbbk^*$ *est dit élément primitif ou racine primitive de* $\Bbbk$ *et*
$\Bbbk^* = < \alpha > = \{\alpha^i \ / \ 0 \leq i \leq p^n - 2\}$.

Définition 1.3.5 *Soit* $\Bbbk$ *un corps commutatif fini tel que* $car(\Bbbk) = p$ *(premier) et* $\alpha \in \Bbbk^*$ *une racine primitive de* $\Bbbk$.
Le polynôme minimal associé à cette racine est appelé polynôme primitif de $\Bbbk$*, qu'on le note* M_α.

Propriété.

1. Si $card(\Bbbk) = p^n$ alors $n = \deg(M_\alpha) = \dim_{F_p}(\Bbbk)$.
2. M_α est unitaire, irréductible sur F_p.
3. M_α est le polynôme de plus petit degré vérifiant $M_\alpha(\alpha) = 0$.

Théorème 1.3.6 *Si $\Bbbk$ est un corps commutatif fini de caractéristique p et α est une racine primitive de $\Bbbk$ alors : $\Bbbk \simeq F_p[X]/ < M_\alpha >$.*

Remarque 1.3.7 *Rappelons que le théorème de Wedderburn affirme que tout corps fini est un corps commutatif. Alors les résultats ci-dessous restent vrais si on ommet la proprieté "commutatif".*

Les résultats ci-dessous nous montrent l'existence de corps finis de cardinal p^n**,** pour n un entier non nul et p premier.

Définition 1.3.8 *Soit $\Bbbk$ un corps commutatif et H un sous-corps de $\Bbbk$.*
Une extension L du corps $\Bbbk$ est dit corps de décomposition d'un polynôme $P \in \Bbbk[X]$ si $\exists\ a \in \Bbbk$ et $\alpha_1, \alpha_2, ..., \alpha_n \in L$ tels que : $P = a \prod_{i=1}^{n} (X - \alpha_i)$.

Proposition 1.3.9 *Si $\Bbbk$ est un corps commutatif et $P \in \Bbbk[X]$, alors P admet un corps de décomposition L sur $\Bbbk$.*

Théorème 1.3.10 *Si p est un entier premier et $n \in \mathbb{N}^*$, alors il existe un corps fini $\Bbbk$ de cardinal p^n et un polynôme irréductible $P \in \Bbbk[X]$ tel que : $d^\circ(P) = n$.*

Preuve. Soit le polynôme $P_1(X) = X^{p^n} - X \in F_p[X]$ et $\Bbbk_1$ le corps de décomposition de P_1 sur F_p alors toutes les racines de $P_1(X)$ sont différentes et l'ensemble $\Bbbk$ de ces racines forme un corps fini de cardinal p^n.
Si α est une racine primitive du corps $\Bbbk$ alors le polynôme $P = M_\alpha$ est le polynôme minimal associé à α, c'est un polynôme irréductible unitaire de degré n. ■

Définition 1.3.11 *Le corps fini* $\mathbb{k}$ *de cardinal* $q = p^n$, *tel que* p *est un entier premier et* $n \in \mathbb{N}^*$, *est dit corps de Galois noté* F_q.

Remarque 1.3.12

1. *Tous les corps finis de cardinal* $q = p^n$ *sont isomorphes.*

2. *Pour décrire le corps de Galois* F_q, *il suffit de connaitre une racine primitive* α *de* F_q *et son polynôme minimal* M_α, *et* $F_q \simeq F_p[\alpha] = \{0, 1, \alpha, \alpha^2, ..., \alpha^{p^n-2}\}$ *ou de connaitre un polynôme irréductible de degré* n *sur* F_p.

Exemple 1.3.13 *Construction d'un corps de Galois de cardinal* 8, $\mathbb{k} = F_8$.

On a $card(F_8) = 8 = 2^3$ *donc* $p = car(F_8) = 2$ *et* $n = 3 = d^\circ(M_\alpha)$ *tel que* α *est une racine primitive de* $\mathbb{k}$.

Soit M_α *polynôme primitif de* $\mathbb{k}$ (*polynôme minimal de* α) *de degré* $n = 3$, *irréductible, unitaire sur* F_2. *On choisit :* $M_\alpha(X) = X^3+X+1$ (*ou* X^3+X^2+1).

$F_8 \simeq F_2[\alpha] = \{0,\ 1,\ \alpha,\ \alpha^2,\ \alpha^3,\ \alpha^4,\ \alpha^5,\ \alpha^6\}$.

$M_\alpha(\alpha) = 0 \Leftrightarrow \alpha^3 + \alpha + 1 = 0 \Leftrightarrow \alpha^3 = -\alpha - 1 = \alpha + 1,\ \alpha^4 = \alpha^2 + \alpha,\ \alpha^5 = \alpha^2 + \alpha + 1,$
$\alpha^6 = \alpha^2 + 1$. *Donc :* $F_8 = \{0, 1, \alpha, \alpha^2, \alpha + 1, \alpha^2 + \alpha, \alpha^2 + \alpha + 1, \alpha^2 + 1\}$.

Définition 1.3.14 *Soit* $\mathbb{k}$ *un corps commutatif et* $n \in \mathbb{N}^*$, $\alpha \in \mathbb{k}$ *est dit racine n-ième de l'unité si :* $\alpha^n - 1 = 0$ *c-à-d* α *racine du polynôme* $X^n - 1 \in \mathbb{k}[X]$.

On note $G_n(\mathbb{k})$ *l'ensemble des racines n-ième de l'unité alors :*

$G_n(\mathbb{k}) = \{x \in \mathbb{k} : x^n - 1 = 0\}$.

Théorème 1.3.15 *Soit* $\mathbb{k} = F_{q=p^r}$ *tel que* $r \in \mathbb{N}^*$ *un corps de Galois, alors* $G_n(\mathbb{k})$ *est un sous-groupe cyclique d'ordre* $card(G_n(\mathbb{k})) = PGCD(p^r - 1,\ n)$.

Définition 1.3.16 $G_n(\mathbb{k})$ *est appelé groupe des racines n-ièmes de l'unité et tout générateur du groupe cyclique* $G_n(\mathbb{k})$ *est dit racine n-ième primitive de l'unité.*

L'ensemble de ces racines n-ièmes primitives est noté $P_n(\mathbb{k})$ *i.e ;*

$P_n(\mathbb{k}) = \{\gamma \in \mathbb{k}\ /\ \gamma$ *générateur de* $G_n(\mathbb{k})\}$.

Proposition 1.3.17 *Soient $n \in \mathbb{N}^*$ et β une racine n-ième primitive de l'unité alors :* $P_n(\Bbbk) = \{\beta^j \ / \ \ 1 \leq j \leq n-1$ *et* $j \ \Lambda \ n = 1 \ \}$.

Le theorème suivant nous montre comment décomposer le polynôme $X^n - 1$ de $\Bbbk[X]$ en produit de polynômes de degré 1 sur une extension L du corps fini $\Bbbk$.

Théorème 1.3.18 *Soit p un entier premier et $n \in \mathbb{N}^*$ avec $n = N.p^m$ / $N\Lambda\ p = 1$. Alors il existe un unique (le plus petit) corps de décomposition du polynôme $X^n - 1$ sur F_p, c'est le corps de Galois $\Bbbk = F_{p^r}$ avec r est le plus petit entier non nul tel que N divise $p^r - 1$. Le corps $\Bbbk$ est dit corps des racines n-ièmes de l'unité sur F_p et on a :*
$X^n - 1 = \prod_{i=0}^{N-1} (X^N - \beta^i)^{p^m}$ *avec* $\beta = \alpha^{\frac{p^r-1}{N}}$, *et α est une racine primitive de $\Bbbk$ et β une racine n-ième primitive de l'unité.*

Exemple 1.3.19 *Décomposition de $X^{15}-1$ sur F_2.*
On a $n = 15$, $p = 2$ donc $n\Lambda\ p = 1$.
Le corps des racines $15^{ème}$ de l'unité est $\Bbbk = F_{2^r}$ où r est le plus petit entier non nul tel que 15 divise $2^r - 1$. On trouve $r = 4$ et donc $\Bbbk = F_{16}$. Si β est une racine 15-ième primitive de l'unité (β est un générateur de $G_{15}(\Bbbk)$) c-à-d : $G_{15}(\Bbbk) =< \beta >= \{\beta^i \ / \ 0 \leq i \leq 14\}$. Alors :

$$X^{15}-1 = (X-1)(X-\beta)...(X-\beta^{14}).$$

De plus si α est une racine primitive de $\Bbbk$ alors : $\beta = \alpha^{\frac{p^r-1}{n}} = \alpha$.

Définition 1.3.20 *Soit $\Bbbk = F_{p^r}$ le corps des racines n-ièmes de l'unité sur F_p.On appelle polynôme cyclotomique d'indice n et à coéfficients dans F_p, le polynôme noté $\Phi_n(X) \in F_p[X]$ dont les racines sont les racines n-ièmes primitives de l'unité c-à-d :*

$$\Phi_n(X) = \prod_{\substack{j=1 \\ j\Lambda\ n=1}}^{N-1} (X - \beta^j).$$

Remarque 1.3.21 *Par la définition de $\Phi_n(X)$ on a : $d^\circ(\Phi_n(X)) = \varphi(n)$ (φ est la fonction d'Euler).*

La proposition ci-dessous nous permet de décomposer le polynôme $X^n - 1$ en produit de polynômes cyclotomiques sur le sous-corps premier du corps $\Bbbk$ des racines n-ièmes de l'unité sur F_p .

Proposition 1.3.22 *Le polynôme* $X^n - 1$ *se décompose par :* $X^n - 1 = \prod_{d \setminus n} \Phi_d(X)$.

La proposition et le théorème suivants servent à calculer le polynôme cyclotomique $\Phi_n(X)$ pour n donné.

Proposition 1.3.23 *Si* p *est un entier premier et* $k \in \mathbb{N}^*$ *alors :*

1. $\Phi_p(X) = 1 + X + ... + X^{p-1}$.
2. $\Phi_{p^k}(X) = \frac{X^{p^k}-1}{X^{p^{k-1}}-1}$.

Théorème 1.3.24 *Soient* $k,\ n \in \mathbb{N}^*$ *et* p *premier.*

1. *Si* p *divise* n *alors :* $\Phi_{np}(X) = \Phi_n(X^p)$.
2. *Si* p *ne divise pas* n *alors :* $\Phi_{np^k}(X) = \frac{\Phi_n(X^{p^k})}{\Phi_n(X^{p^{k-1}})}$ *et en particulier* $\Phi_{np}(X) = \frac{\Phi_n(X^p)}{\Phi_n(X)}$.

La proposition ci-dessous nous permet de décomposer tout polynôme cyclotomique en produit de polynômes irréductibles sur le sous-corps premier du corps des racines n-ièmes de l'unité.

Proposition 1.3.25 *Soient* p *un entier premier et* $n \in \mathbb{N}^*$, $\Bbbk = F_{p^r}$ *le corps des racines n-ièmes de l'unité sur* F_p. *Alors* $\Phi_n(X)$ *se décompose en produit de* $\varphi(n)/r$ *polynômes irréductibles de degré* r *à coéfficients dans* F_p.

Le corollaire ci-dessous est une conséquence immédiate du proposition ci-dessus qui détermine quand un polynôme cyclotomique est irréductible sur F_p.

Corollaire 1.3.26 *Si* $\varphi(n) = r$ *alors* $\Phi_n(X)$ *est un polynôme irréductible sur* F_p.

Le corollaire ci-dessous nous montre qu'en choisissant un entier non nul n et en utilisant les propositions ci-dessus, on peut décomposer le polynôme $X^n - 1$ en produit de polynômes irréductibles sur le sous-corps premier du corps des racines n-ièmes de l'unité.

Corollaire 1.3.27 *Si $n \in \mathbb{N}^*$ alors le polynôme $X^n - 1$ peut être décomposer en produit de polynômes irréductibles sur F_p.*

Exemple 1.3.28

1. *Décomposer le polynôme $X^{15}-1$ en produit des polynômes irréductibles sur F_2.*

On a $n = 15,\ p = 2$ et $n\ \Lambda\ p = 1$ donc :

$$X^{15}-1 = \prod_{d\ \backslash 15} \Phi_d(X) = \Phi_1(X).\Phi_3(X).\Phi_5(X).\Phi_{15}(X).$$

Soit r le plus entier non nul tel que $n = 15$ divise $2^r - 1$, alors d'après les calculs on trouve $r = 4$. Donc $\Bbbk = F_{2^4} = F_{16}$ est le corps des racines $15^{ème}$ de l'unité sur F_2 .

$\Phi_1(X) = X - 1$, $\Phi_3(X)$ et $\Phi_5(X)$ sont des polynômes irréductibles et $\Phi_{15}(X)$ se décompose en produit de $\varphi(15)/r = 2$ polynômes irréductibles de degré 4 à coefficients dans F_2. Après calcul et indentification on trouve :

$$\Phi_{15}(X) = (X^4 + X + 1)(X^4 + X^3 + 1) \text{ et donc :}$$

$$X^{15} - 1 = (X-1)(X^2 + X + 1)(X^4 + X^3 + X^2 + X + 1)(X^4 + X + 1)(X^4 + X^3 + 1).$$

2. *Décomposition de $X^{30} - 1$ en produit de polynômes irréductibles sur F_2. Dans ce cas $n = 30$ qui n'est pas premier avec $p = 2$ et qui s'ecrit :*

$n = N.p^m = 15.2$ ($N = 15$ et $m = 1$). Il suffit de décomposer $X^{15} - 1$ et déduire $X^{30}-1 = (X^{15}-1)^2$. Donc d'après l'exemple ci-dessus :

$$X^{30} - 1 = (X-1)^2(X^2 + X + 1)^2(X^4 + X^3 + X^2 + X + 1)^2(X^4 + X + 1)^2(X^4 + X^3 + 1)^2.$$

1.4 Matrice de permutation.

Définition 1.4.1 *Une matrice de permutation d'ordre n est une matrice carré P d'ordre n dont les colonnes sont une permutation des colonnes de la matrice identité I_n, c-à-d : Si $P = (p_{ij})$, $\exists\ \sigma \in S_n$ (S_n le groupe symétrique d'indice n) tel que :*

$$p_{ij} = \delta_{i,\sigma(j)} = \begin{cases} 1 & si\ i = \sigma(j) \\ 0 & si\ non \end{cases}$$

tel que $\delta_{i,j}$ représente le symbole de Kronecker.

Si σ est la permutation associée à la matrice de permutation P on note P_σ au lieu de P.

Proposition 1.4.2 *Si P_σ est la matrice de permutation associée à la permutation σ. Alors :*

1. *L'ensemble des matrices de permutation d'ordre n noté P_n, forme un sous-groupe isomorphe au groupe symétrique S_n.*

2. *L'inverse de P_σ est $P_\sigma^{-1} = P_\sigma^t$, et $\det(P_\sigma) = 1$ si σ est une permutation paire, et $\det(P_\sigma) = -1$ si σ est une permutation impaire.*

3. *Multiplier une matrice A à droite par P_σ revient à permuter les colonnes de la matrice A, en suivant la permutation σ.*

4. *Multiplier une matrice A à gauche par P_σ revient à permuter les lignes de la matrice A, en suivant la permutation inverse.*

5. *Si P_σ est une matrice symétrique alors $P_\sigma^{-1} = P_\sigma$.*

Exemple 1.4.3 *Soit la matrice* $P_\sigma = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$ *tel que* $\sigma = \begin{pmatrix} 1\ 2\ 3\ 4 \\ 4\ 3\ 2\ 1 \end{pmatrix}$.

Chapitre 2

Les codes linéaires et les codes cycliques.

2.1 Les codes linéaires.

Introduction.

La classe des codes linéaires est une sous-classe des codes dits codes par blocs où le message à transmettre est découpé en blocs (mots) de longueur fixée k.

Nous commencerons par en donner une définition, puis nous donnerons une description des matrices génératrices et de contrôle, puis nous définirons plus précisément ce qu'est la distance minimale d'un code et nous expliquerons quel est son intérêt. Nous définirons ensuite quelques bornes sur le cardinal du code par rapport à sa distance minimale. Et pour finir, nous présentons un exemple de codes linéaires binaires très connu en pratique sous le nom de son inventeur Hamming.

Si l'alphabet A est un corps fini $\mathbb{k}$ de cardinal q, alors $\mathbb{k}^n$ est un $\mathbb{k}-$espace vectoriel pour les lois habituelles (l'addition et la multiplication par un scalaire) de dimension n, muni du produit scalaire usuel défini par : $< x, y >= \sum_{i=1}^{n} x_i y_i$ pour $x = (x_1, ..., x_n)$, $y = (y_1, ..., y_n)$ dans $\mathbb{k}^n$.

Définition 2.1.1 *Un code linéaire C de longueur n et de dimension k sur le corps fini $\Bbbk$ est un sous-espace vectoriel du $\Bbbk$-espace vectoriel $\Bbbk^n$ de dimension k et de cardinal $M = \mid C \mid = q^k$, noté $C(n,\ k)$.*

Remarque 2.1.2

1. *C est un code linéaire de longueur n sur $\Bbbk$ si et seulement si :*

$\forall\ x_1,\ x_2 \in C,\ \alpha_1,\ \alpha_2 \in \Bbbk\ :\ \alpha_1 x_1 +\ \alpha_2 x_2 \in C.$

2. *Si $\Bbbk = F_2 = \{0, 1\}$ le code $C(n,\ k)$ est dit code binaire et a comme cardinal 2^k.*

Définition 2.1.3 *Une application $\Phi : \Bbbk^k \to \Bbbk^n$ sur l'alphabet $\Bbbk$ est dite codage linéaire si et seulement si Φ est une application linéaire injective.*
Son image $C = im(\Phi)$ est un code linéaire sur $\Bbbk$, dit code linéaire associé à Φ.

Exemple 2.1.4

1. *$\{0\}$ et $\Bbbk^n$ les codes linéaires triviaux.*
2. *$\Phi : \Bbbk^k \to \Bbbk^{k+1}$ l'application codage par bit de parité.*
$x = (x_1, x_2, ..., x_k) \mapsto \Phi(x) = (x_1, x_2, ..., x_k, \sum_{i=1}^{k} x_i)$ est un codage linéaire.
3. *$\Phi : \Bbbk \to \Bbbk^n$ l'application codage à répétition.*
$x = x_1 \mapsto C = \Phi(x) = (x_1, x_1, ..., x_1)$ est un codage linéaire.

2.1.1 Distance de Hamming et distance minimale.

Définition 2.1.5 *On appelle dinstance de Hamming entre deux mots $x = (x_1, x_2, ..., x_n)$, $y = (y_1, y_2, ..., y_n) \in \Bbbk^n$ et on note $d(x, y)$, le nombre de position où x et y sont des composantes différentes :* $d(x, y) = card\{i \in \overline{1,\ n} : x_i \neq y_i\}$.

Exemple 2.1.6

1. $d(1111, 1010) = 2$.
2. $d(0101, 1010) = 4$.

Définition 2.1.7 *On appelle distance minimale (ou simplement distance) d'un code C et on note $d_{\min}$ (ou tout simplement d), la plus petite des distances entre deux mots distincts de ce code, c-à-d :* $d_{\min} = d = \min\{d(x, y)\ /\ (x, y) \in C \times C,\ x \neq y\}$.

Exemple 2.1.8

1. C= $\{000,\ 110,\ 111\},\ d = d_{\min} = \min\{1, 2, 3\} = 1.$

2. *Code à répitition* $C(n,\ k = 1,\ d)$ *est de distance* $d_{\min} = n.$

3. *Le code par bit de parité* $C(n = k + 1,\ k,\ d)$ *est de distance* $d = 2.$

Notation :

• Un code C de longueur n, de dimention k et de distance minimale d est dit "Code $C(n, k, d)$".

• Les entiers $n,\ k$ et d sont dits "paramètres du code".

Remarque 2.1.9 *La distance* d *joue un role important dans le décodage car elle est en relation directe avec le nombre d'erreurs susceptibles d'être détectées ou corrigées.*

Définition 2.1.10 *Le poids d'un élément* $x = (x_1, x_2, ..., x_n) \in \mathbb{k}^n$ *noté* $w(x)$ *est le nombre de ses composantes non nulles.*
$w(x) = card\{i \in \{1, 2, ..., n\} : x_i \neq 0\}.$

Exemple 2.1.11 *Soit l'alphabet* $A = F_2$ *et* $x = (1, 0, 1, 0) \in F_2^4,\ \ w(x) = 2.$

Propriétés du poids.

$\forall\ x,\ y \in \mathbb{k}^n, \forall\ \lambda \in \mathbb{k}$. On a :

$i)\quad d(x, y) = w(x - y).$

$ii)\quad w(x) = d(x, 0).$

$iii)\quad w(x) = 0 \Leftrightarrow x = 0.$

$iv)\quad w(\lambda x) = w(x),\ \forall\ \lambda \neq 0.$

$v)\quad w(x + y) \leq w(x) + w(y).$

Preuve. Les propriétés $i)$ à $iv)$ sont évidentes.

$v)\ \forall\ x,\ y \in \mathbb{k}^n$

$w(x + y) = d(x + y, 0) \leq d(x + y, x) + d(x, 0)$ (d est une dintance).

Donc : $w(x + y) \leq d(x + y - x, 0) + d(x, 0) = d(y, 0) + d(x, 0) = w(x) + w(y).$

■

Définition 2.1.12 *On appelle poids minimum d'un code linéaire $C(n,k)$ le plus petit poids des mots non nul du code C et on le not $P_{\min}$.*

Exemple 2.1.13 *Soit $C = \{00000, 10010, 01011, 00101, 11001, 10111, 01110, 11100\}$ un code linéaire de longueur $n = 5$ et de dimention $k = 3$ sur F_2.*
$P_{\min} = \min\{2,\ 3,\ 4\} = 2.$

Proposition 2.1.14 *La distance minimale d'un code linéaire C est égale au poids minimum de ses mots non nuls.*

Preuve.

$d_{\min} = min\{d(x,y)\ /\ x,y \in C\ ,\ \ x \neq y\} = min\{w(x-y)\ /\ x,y \in C\ ,\ x \neq y\}$
$= min\{w(z)\ /\ z = x - y \in C\ ,\ z \neq 0\} = P_{min}.$ ■

2.1.2 Capacité de correction et de détection d'un code.

Définition 2.1.15 *Soit C un code sur un corps fini $\mathbb{k}$. On appelle capacité de correction (résp. de détection) du code C le nombre d'erreurs e (résp. t) de transmission que ce code peut corriger (résp. détecter), on dit dans ce cas que C est un code $e-$correcteur (résp. $t-$détecteur).*

Théorème 2.1.16 *Soit C un code de distance d, soit y un mot reçu comportant au plus r erreurs de transmission par rapport au mot envoyé x.*
i) Si $e < d$ le code C permet de détecter si le mot reçu y est erroné.
ii) Si $e < \frac{d}{2}$ le code C permet de corriger le mot reçu y.

Preuve.

• **Cas** $e < d$:
On suppose qu'il y a au moins une erreur de transmission, le mot reçu est détecté comme si ce n'est pas un mot du code. Par définition pour tout mot z du code C différent de x on a $d(x,z) \geq d$, or comme $d(x,y) \leq e < d,\ y$ ne peut pas être un mot du code.

• **Cas** $e < \frac{d}{2}$:

Pour tout mot z du code C différent de x, on a par définition $d \leq d(x,z)$.

L'inégalité triangulaire donne : $d(x,z) \leq d(x,y)+d(y,z)$.

Maintenant par hypothèse $d(x,y) \leq e < \frac{d}{2}$ donc $d < \frac{d}{2} + d(y,z)$, soit $\frac{d}{2} < d(y,z)$.

Comme $d(x,y) < \frac{d}{2}$, on déduit que pour tout mot z du code C différent de x, $d(x,y) < d(y,z)$.

Le mot reçu y est bien corrigé, car x est le mot du code le plus proche de y. ■

Du théorème ci-dessus découle le théoreme suivant qui nous montre l'intérêt de la distance minimale dans la détection et la correction des erreurs comisent lors de transfert ou de stocage de messages.

Corollaire 2.1.17 *Soit C un code de distance minimale d, alors C permet de corriger jusqu'à : $e = [\frac{d-1}{2}]$ erreurs et permet de détecter jusqu'à : $t = d-1$ erreurs.*

Exemple 2.1.18 *Soit le code à répétition de longueur $n = 5$, $C = \{00000,\ 11111\}$ sa distance $d = 5$. Il peut détecter jusqu'à $t = 4$ erreurs et corriger jusqu'à $e = 2$ erreurs.*

2.1.3 Codes équivalents.

Définition 2.1.19 *Soit $\Bbbk$ un corps fini quelconque et n un entier $n \geq 1$, pour chaque permutation $\sigma \in S_n$ (où S_n est le groupe symétrique) on définit l'application $\overline{\sigma}$ de $\Bbbk^n$ dans $\Bbbk^n$ par : $\overline{\sigma} : (x_1, x_2, ..., x_n) \longmapsto (x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(n)})$.*

Deux codes C et C' de longueur n sur $\Bbbk$ sont dits équivalents s'il existe une permutation $\sigma \in S_n$ tel que $C' = \overline{\sigma}(C\ )$.

Proposition 2.1.20 *Deux codes équivalents sur un corps fini $\Bbbk$ ont la même longueur, la même dimension et la même distance.*

Exemple 2.1.21 *Soit le code $C = \{111,\ 000,\ 101\}$ sur $\Bbbk = \{0,\ 1\}$.*

Alors le code $C' = \{111,\ 000,\ 011\}$ est un code équivalent à C.

Car $C' = \overline{\sigma}(C\ )$ tel que : $\overline{\sigma} = \tau_{12} = (213)$.

2.1.4 Code linéaire systématique.

Définition 2.1.22 *Une martice génératrice d'un code linéaire $C(n,\ k)$ sur le corps fini $\Bbbk$ est une matrice de type $k \times n$ à coéficients dans $\Bbbk$ dont les lignes forment une base de C.*

Exemple 2.1.23 *Soit* $G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$ *une matrice génératrice du code linéaire $C(5,\ 3)$ sur F_2 dont les mots sont :*

$C = \{00000, 10010, 01011, 00101, 11001, 10111, 01110, 11100\}$.

Proposition 2.1.24 *Si G est une martice génératrice d'un code linéaire $C(n,\ k)$ sur $\Bbbk$, alors toute martice génératrice de C est de la forme $A.G$ où A une martice carrée inversible d'orde k sur $\Bbbk$.*

Preuve. Soit G'=$A.G$ qui est une matrice de type $k \times n$.

G' est une matrice génératrice de $C \Leftrightarrow rg(G') = k$?

$rg(G') = rg(A.G) = \min\{rg(A),\ rg(G)\} = \min\{k,\ k\} = k$.

Donc G'=$A.G$ est une matrice génératrice de C. ■

Remarque 2.1.25

1. *Le code C est le sous-espace de $\Bbbk^n$ des mots de la forme $y = x.G$ avec*

$x = (x_1, ..., x_k) \in \Bbbk^k$.

2. *Si $c_1, c_2, ..., c_n$ sont les vecteurs colonnes de G, les mots du code C sont tous de la forme : $y = \{< c_1, x >, < c_2, x >, ..., < c_n, x >\}$ avec $x \in \Bbbk^k$ et $< .,. >$ est le produit scalaire usuel de $\Bbbk^k$.*

Exemple 2.1.26 *Soit C un code linéaire de matrice génératrice* $G = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$ *de type $(5,\ 2)$, alors le code linéaire associé :*

$C = \{(x_1,\ x_2).G\ /\ x_1,\ x_2 \in \{0,\ 1\}\}$

$= \{(x_1{+}x_2,\ x_2, x_1{+}x_2,\ x_1{+}x_2,\ x_1)\ /\ x_1,\ x_2 \in \{0,\ 1\}\}$ *d'où :* $C = \{00000,\ 10111,\ 11110,\ 01001\}$.

Définition 2.1.27 *Une matrice génératrice G d'un code linéaire $C(n,\ k)$ est normalisée (canonique ou standard) si la matrice formée par les k premières colonnes de G est la matrice unité I_k. Donc G est de la forme $(I_k,\ M)$ tq : $M \in M_{k,n-k}(\mathbb{k})$ est la matrice dite de parité.*

Exemple 2.1.28 *Soit C un code linéaire $(5,\ 2)$ sur F_2 de matrice génératrice* $G = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix}$. *$G$ est une matrice génératrice normalisée du code C et* $C = \{00000,\ 10111,\ 01111,\ 11000\}$.

Définition 2.1.29 *Un code linéaire est dit systématique (ou standard) s'il possède une matrice génératrice normalisée G_N.*

Proposition 2.1.30 *La matrice génératrice standard d'un code linéaire C est unique, on l'obtient en appliquant l'algorithme de GAUSS sur les lignes d'une matrice génératrice quelconque de C.*

Preuve. Soit $G = (A,\ B)$ où A est une matrice carré de rang k (donc inversible), en appliquant l'algorithme de GAUSS sur les lignes de G (donc de A) pour avoir la matrice unité I_k, alors on obtient une matrice G' (équivalente à G) de la forme $G' = (I_k,\ B')$ qui est une matrice normalisée de C. ■

Remarque 2.1.31 *Certains codes linéaires n'admettent pas de matrices génératrices standard. En général si G est une matrice génératrice d'un code $C(n,\ k)$ de la forme $G = (A,\ B)$ tel que A matrice carrée d'ordre k, alors C est systématique si et seulement si la matrice A est inversible.*
La matrice génératrice normalisée dans ce cas est $G_N = (I_k,\ A^{-1}.B)$.

Exemple 2.1.32 *Soit C un code linéaire défini par sa matrice génératrice :* $G = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$ *de la forme $G = (A,\ B)$ où* $A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ *inversible donc*

C est systématique. On applique l'algorithme de GAUSS sur les lignes de G, pour passer de $G = (A,\ B)$ à $G_N = (I_3,\ B')$.

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} L_1 \leftarrow L_1 + L_2 \sim \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \ L_1 \leftarrow L_1 + L_3 \sim$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} = G_N \text{ qui est la matrice génératrice normalisée de } C.$$

Théorème 2.1.33 *Tout code linéaire est équivalent à un code linéaire systématique.*

Preuve. *Supposons que $C(n,\ k)$ est un code linéaire qui n'est pas un code systématique, soit G une matrice génératrice du code C comme le rang de G est égal à k, il existe un mineur dang de type $k \times k$ non nul.*

Par une permutation des colonnes de G, on amène ce mineur aux k premières colonnes et on obtient ainsi une matrice génératrice d'un code systématique C' équivalent à C. ■

Définition 2.1.34 *Un code binaire de Hamming de longueur $n = 2^m - 1$ tel que $m \in \mathbb{N} - \{0,\ 1\}$ est un code linéaire binaire qui admet comme matrice de contrôle toute matrice dont les colonnes sont tous les vecteurs non nuls de l'espace F_2^m.*

Ils ont été inventés par Richard Hamming et sont utilisés dans le domaine des "Digital Communications" et des systèmes de sauvegardes de données.

Remarque 2.1.35 *De la définition on déduit que la dimension d'un code de Hamming est $k = 2^m - m - 1$ et sa distance $d = 3$ car chaque matrice de contrôle H n'admet pas de colonne nulle et tous les colonnes sont distinctes et la somme de chaque deux colonnes est une autre colonne de H.*

Exemple 2.1.36 *Un code de Hamming binaire $C(7,\ 4,\ 3)$ admet comme matrice de contrôle H la matrice suivante : $H = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$.*

2.1.5 Quelques bornes caracterisant un code.

Soit $C(n,\ k,\ d)$ est un code linéaire sur un corps fini $\mathbb{k}$ de cardinal fixé q. Les grandeurs $M = q^k$ (et donc k) et la distance minimale d "jouent" l'une contre l'autre. En effet, si on a un très grand nombre de mots (i.e ; k très grand), alors on aura une distance d faible.

Alors que si on a k petit, alors la distance d sera grande, permettant ainsi de détecter et de corriger plus d'erreurs.

Il va donc être nécessaire de trouver un compromis entre ces deux valeurs, afin d'avoir un nombre de mots suffisants et une distance minimale suffisamment grande pour pouvoir détecter et corriger un certain nombre d'erreurs. Pour cela, il existe des bornes qui caractérisent les grandeurs k et d. Parmis ces bornes on trouve la borne de singleton et la borne de Hamming.

Borne de singleton.

Cette borne permet de trouver une borne maximale sur la distance minimale d par rapport aux valeurs n et k.

Théorème 2.1.37 *Soit un code $C(n,\ k,\ d)$. On a l'inégalité suivante : $d \leq n - k + 1$.*

Preuve. *On sait qu'un code linéaire C est équivalent à un code linéaire systématique C' est que les paramètres $(n,\ k,\ d)$ des codes C et C' sont les mêmes (même longueur, même dimension, même cardinal, même distance), on peut donc faire la démonstration pour le code C'.*
Soit G la matrice génératrice normalisée du code C' donc tout mot du code C' s'écrit comme combinaison linéaire des lignes de G, le poids minimum du code est donc nécessairement inférieur au poids minimum des vecteurs composant les lignes de G, les k premières colonnes de la matrice normalisée G formant la matrice identité, le poids maximum d'une ligne est donc majoré par $1 + (n - k)$, donc la distance minimale d'un code linéaire est inférieur à $n - k + 1$. ■

Définition 2.1.38 *Un code linéaire $C(n,\ k)$ est dit code M.D.S (Maximum Distance Séparable) si sa distance minimal d atteint la borne de singleton c-à-d : $d = n - k + 1$.*

Exemple 2.1.39

1. *Le code à répétition $C(n,\ k = 1,\ d = n)$ est un code M.D.S car $n - k + 1 = n = d$.*
2. *Le codage par bit de parité $C(n = k + 1,\ k,\ d = 2)$ est un codage M.D.S.*

Borne de Hamming.

Définition 2.1.40 *Soit $C(n,\ k,\ d)$ un code q-aire (c.à.d. sur le corps fini F_q) et $r \in \mathbb{N}^*$. Pour tout $x \in F_q^n$ on définit la boule $B(x,\ r)$ de centre x et de rayon r par : $B(x,\ r) = \{y \in F_q^n : d(x,\ y) \leq r\}$ et la sphère $S(x,\ i)$ de centre x et de rayon i par : $S(x,\ i) = \{y \in F_q^n : d(x,\ y) = i\}$.*

Remarque 2.1.41 $B(x,\ r) = \cup_{i\in\{0,\ldots,r\}} S(x,\ i)$ *et* $\cup_{x\in C} B(x,\ r) \subset F_2^n$.

Théorème 2.1.42 *Soit $C(n,\ k,\ d)$ un code q-aire de capacité de correction $e = [\frac{d-1}{2}]$. Alors $q^k \sum_{i=0}^{e} C_n^i (q-1)^i \leq q^n$. Cette borne est dite borne de Hamming.*

Preuve. Par définition de $S(x,\ i)$ on a pour tout i de 0 à e :
$card(S(x,\ i)) = C_n^i (q-1)^i$ avec $C_n^i = \frac{n!}{i!(n-i)!}$ et donc :
$card(B(x,\ e)) = \sum_{i=0}^{e} card(S(x,\ i)) = \sum_{i=0}^{e} C_n^i (q-1)^i$, comme $card(C) = q^k$ alors :
$card(\cup_{x\in C} B(x,\ e)) = \sum_{x\in C} card(B(x,\ e)) = q^k \sum_{i=0}^{e} C_n^i (q-1)^i \leq card(F_q^n) = q^n$. ■

Corollaire 2.1.43 *Soit $C(n,\ k,\ d)$ un code binaire de capacité de correction $e = [\frac{d-1}{2}]$. Alors $2^k \sum_{i=0}^{e} C_n^i \leq 2^n$.*

Codes parfaits.

Définition 2.1.44 *Un code $C(n,\ k,\ d)$ est dit parfait si la borne de Hamming est atteinte, c'est-à-dire si : $q^k \sum_{i=0}^{e} C_n^i (q-1)^i = q^n$ ($où \cup_{x\in C} B(x,\ e) = F_q^n$).*
Dans le cas binaire on a : $2^k \sum_{i=0}^{e} C_n^i = 2^n$.

Exemple 2.1.45

Les Codes de Hamming sont des codes parfaits.

2.1.6 Code orthogonal d'un code linéaire.

Définition 2.1.46 *Soit $C(n,\ k)$ un code linéaire sur un corps fini $\mathbb{k}$ le code orthogonal (où dual) de C, noté $C^{\perp}$ est le sous-espace vectoriel orthogonal de C pour le produit scalaire usuel de $\mathbb{k}^n$ c-à-d : $C^{\perp} = \{x \in \mathbb{k}^n,\ \forall y \in C : < x,\ y >= 0\}$.*
Pour le produit scalaire usuel de $\mathbb{k}^n$ i ;e $< x,\ y >= \sum_{i=1}^{n} x_i y_i$.

Le code dual de C est un code linéaire de longueur n et de dimension $k^{'} = n - k$.

Preuve. $C^{\perp}$ est un code linéaire car $C^{\perp}$ est un sous-espace vectoriel de $\mathbb{k}^n$.
$C^{\perp} \subset \mathbb{k}^n \Rightarrow$ la longueur de $C^{\perp}$ est n, $\dim C^{\perp} = \dim \mathbb{k}^n - \dim C = n - k$. ■

Définition 2.1.47 *On appelle matrice de contrôle d'un code linéaire $C(n,\ k)$ toute matrice génératrice de son code dual, qu'on note H.*

Exemple 2.1.48 *Soit $C(5,\ 3)$ un code linéaire sur F_2 de matrice génératrice normalisée $G_N = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$ et la matrice $H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix}$ est une matrice de contrôle de C.*

Remarque 2.1.49 *H est de type $(n \times n - k)$ et de rang $n - k$.*

Théorème 2.1.50 *Soit H une matrice de contrôle d'un code linéaire $C(n,\ k)$ sur $\mathbb{k}$ et $c = (c_1,\ ...,\ c_n) \in \mathbb{k}^n$ alors $c \in C \Leftrightarrow c.H^t = 0$.*

Preuve. Soit $c = (c_1,\ ...,\ c_n) \in \mathbb{k}^n$, $c \in C \Leftrightarrow c \in (C^{\perp})^{\perp} \Leftrightarrow \forall y \in C^{\perp} : < c,\ y >= 0$
$\Leftrightarrow \forall i \in \overline{1, n-k}\ : < c,\ L_i >= 0$, tel que (L_i) est la $i^{ème}$ ligne de H.
$\Leftrightarrow (< c,\ L_1 >,\ < c,\ L_2 >, ...,\ < c,\ L_{n-k} >) = (0,\ 0,\ ...,\ 0) \Leftrightarrow c.H^t = 0$. ■

Proposition 2.1.51 *Soit G une matrice génératrice d'un code linéaire $C(n,\ k)$.*
$H \in M_{n-k,n}(\Bbbk)$ une matrice de rang $n-k$ alors :
H est une matrice de contrôle de $C \Leftrightarrow H.G^t = 0$.

Preuve. Soit $h : \Bbbk^n \longrightarrow \Bbbk^{n-k}$ l'application liéaire associée à la matrice H dans les bases canoniques de $\Bbbk^n$ et $\Bbbk^{n-k}$ et soit $f : \Bbbk^k \longrightarrow \Bbbk^n$ l'application associée à la matrice G^t dans les bases canoniques de $\Bbbk^k$ et $\Bbbk^n$.
Alors H matrice de controle de $C \Longleftrightarrow$ pour tout $x \in \Bbbk^k : h[f(x)] = 0$
(car $f(x) \in C$).

$\Longleftrightarrow$ pour tout $x \in \Bbbk^k : (h \circ f)(x) = 0$.
$\Longleftrightarrow$ l'pplication $h \circ f$ est nulle $\Longleftrightarrow H.G^t = 0$. ■

Théorème 2.1.52 *Soit G_N la matrice génératrice normalisée d'un code linéaire systématique $C(n,\ k)$ tel que $G_N = (I_k,\ M)$ alors : $H = (-M^t,\ I_{n-k})$ est une matrice de contrôle de C. Réciproquement si $H = (A,\ I_{n-k})$ une matrice de control de C alors la la matrice génératrice normalisée de C est : $G_N = (I_k,\ -A^t)$.*

Preuve. Il suffit de faire un calcul du produit des matrice H et G^t par bloc on trouve $H.G^t = 0$. ■

Proposition 2.1.53 *Soit H une matrice de contrôle d'un code linéaire C et $C_1,\ C_2, ..., C_n$ les colonnes de H alors, il existe un mot c de C de poids r si et seulement si il existe r colonnes de H linéairement dépendants.*

Preuve. $\exists\ m = (m_0,\ m_1, ...,\ m_{n-1}) \in C$ tel que $w(m) = r \Leftrightarrow \exists\ \alpha_{i_1},\ \alpha_{i_2}, ...,\ \alpha_{i_r} \in \Bbbk$ tous non nuls tel que $m = (0,\ ...,\ \alpha_{i_1},\ 0,\ ..., \alpha_{i_2},\ 0, ...,\ \alpha_{i_r},\ ...0) \in C$.

$\Leftrightarrow m.H^t = 0 \Leftrightarrow \alpha_{i_1}C_{i_1} + \alpha_{i_2}C_{i_2} + ... + \alpha_{i_r}C_{i_r} = 0 \Leftrightarrow$ les r colonnes $C_{i_1},\ C_{i_2},\ ..., C_{i_r}$ sont linéairement dépendants. ■

2.2 Les codes cycliques.

2.2.1 Code cyclique et sa représentation polynomiale.

Définition 2.2.1 *Un code linéaire $C(n,\ k)$ sur le corps fini $\Bbbk$ est dit cyclique si et seulement si :* $\forall\ c = (c_0,\ c_1,\ ...,\ c_{n-1}) \in C \Rightarrow c^{'} = (c_{n-1},\ c_0,\ ...,\ c_{n-2}) \in C$.
Le mot $c^{'}$ est dit le shift de c.

Exemple 2.2.2

1. $\{0\}$, $\Bbbk^n$ *sont les codes cycliques triviaux.*
2. *Un code de Hamming H* $(n = 2^{m-1},\ k = 2^m - n - 1,\ d = 3)$ *est un code cyclique.*
3. *Le code à répétition et le code par bit de parité sont des codes cycliques.*

Définition 2.2.3 *Soit C un code linéaire sur le corps fini $\Bbbk$ et $c = (c_0,\ ...,\ c_{n-1}) \in C$. On appelle représentation polynomiale de c le polynôme*

$c(X) = c_0 + c_1X + ... + c_{n-1}X^{n-1}$ *de* $\Bbbk[X]$.

On associe au mot shift $c = (c_{n-1},\ c_0,\ ...,\ c_{n-2}) \in C$, la représentation polynomiale de $c^{'}$, $c^{'}(X) = c_{n-1} + c_0X + ... + c_{n-2}X^{n-1}$ de $\Bbbk[X]$, ce polynôme peut être obtenu en calculant le produit $X.c(X)$ et considérons que $X^n = 1$ c-à-d en calculant modulo $X^n - 1$ et précisément dans l'anneau quotient $\Bbbk[X] / < X^n - 1 >$.

Définition 2.2.4 *L'application* $\theta : \Bbbk^n \to \Bbbk[X] / < X^n - 1 >$ *définie par*
$c = (c_0, c_1, ..., c_{n-1}) \mapsto \theta(X) = \sum_{i=0}^{n-1} c_iX^i$ *est une application linéaire dite représentation polynômiale de $\Bbbk^n$, et $\theta(C)$ est dite représentation polynômiale du code C c-à-d :* $\theta(C) = \{\theta(c)\ /\ c \in C\}$.

Exemple 2.2.5 *Soit le code* $C = \{000,\ 110,\ 011,\ 101\}$, *sa représentation polynômiale est* $\theta(X) = \{0\ (le\ polynôme\ nul),\ 1 + X,\ X + X^2, 1 + X^2\}$.

Proposition 2.2.6 *Soit $C(n, k)$ un code linéaire sur $\Bbbk$. C est un code cyclique si et seulement si $\theta(X)$ est un idéal de l'anneau quotient* $\Bbbk[X] / < X^n - 1 >$.

Preuve. Supposons que C un code cyclique et soit $c(X) = \sum_{i=0}^{n-1} c_i X^i$ et $d(X) = \sum_{i=0}^{n-1} d_i X^i$ dans $\theta(C)$ donc : $c = (c_0, c_1, ..., c_{n-1})$, $d = (d_0, d_1, ..., d_{n-1}) \in C$ et comme C est un sous-espace vectoriel alors pour $\alpha,\ \beta \in \Bbbk$:

$g(X) = \alpha c(X) + \beta d(X) \in \theta(C)$. D'où $\theta(C)$ est un espace vectoriel sur $\Bbbk$. De plus soit $p(X) = \sum_{i\geq 0} \alpha_i X^i \in \Bbbk[X] \diagup < X^n - 1 >$ alors comme $c(X) \in \theta(C)$ et C cyclique alors : $c(X),\ Xc(X), X^2c(X), ..., X^ic(X), ... \in \theta(C)$, et donc :

$$a_0c(X)+a_1Xc(X)+a_2X^2c(X)+...+a_iX^ic(X) \in \theta(C).$$

Par suit $p(X)c(X) \in \theta(C)$ et donc $\theta(C)$ est un idéal de $\Bbbk[X] \diagup < X^n-1 >$.

Réciproquement : Soit $\theta(C)$ un idéal de $\Bbbk[X] \diagup < X^n-1 >$ alors si :

$c = (c_0, c_1, ..., c_{n-1}) \in C$ alors : $c(X) = \sum_{i=0}^{n-1} c_i X^i \in \theta(C)$.

Donc : $Xc(X) \in \theta(C) \Rightarrow c^{'} = (c_{n-1},\ c_0,\ ...,\ c_{n-2}) \in C$ donc C est cyclique. ■

Théorème 2.2.7 *Soit $\Bbbk$ un corps fini et n un entier non nul. Si C est un code linéaire non nul ($C \neq \{0\}$) alors $\theta(C)$ est un idéal principal de $\Bbbk[X] \diagup < X^n - 1 >$.*

Preuve. Soit $g(X)$ un polynôme non nul de $\theta(C)$ de degré minimum unitaire.

va montrer que $\theta(C) =< g(X) >$.

$g(X) \in \theta(C)$ alors $< g(X) > \subset \theta(C)$ ... (1).

part soit $c(X) \in \theta(C)$ donc $c(X) = g(X).q(X) + r(X)$ . . . $(*)$ avec $r(X) = 0$ où $d^\circ(r(X) < d^\circ(g(X))$.

Supposons que $d^\circ(r(X)) < d^\circ(g(X))$.

Pour $(*)$ on a $r(X) = c(X)-g(X).q(X) \in \theta(C)$ absurde.

$r(X) = 0$ alors $c(X) = g(X).q(X)$ et donc : $c(X) \in < g(X) >$ ... (2).

De (1) et (2) on trouve $\theta(C) = < g(X) > = \{q(X).g(X) \ / \ q(X) \in \Bbbk[X] \ \diagup < X^n - 1 >\}$

. ■

2.2.2 Polynôme générateur et matrice génératrice d'un code cyclique.

Définition 2.2.8 *Le polynôme $g(X)$ donné en théorème ci-dessus engendrant $\theta(C)$ est appelé le générateur du code cyclique C.*

Proposition 2.2.9

i) $g(X)$ *de degré minimum dans* $\theta(C)$.

ii) $g(X)$ *est unitaire unique.*

iii) Tout polynôme de $\theta(C)$ *est un multiple de* $g(X)$.

iv) $g(X)$ *divise* $X^n - 1$ *dans* $\Bbbk[X]$.

Exemple 2.2.10 *Soit* $C(n,\ k)$ *un code linéaire sur* F_2 *tel que :*
$\theta(C) = \{0,\ 1+X,\ X+X^2,\ 1+X^2\}$, $g(X) = 1+X$ *est un générateur de* C.

Remarque 2.2.11 *Comme* $g(X)$ *divise* $X^n - 1$ *dans* $\Bbbk[X]$ *alors pour trouver tous les codes cycliques de longueur* n*, il suffit de trouver tous les diviseurs du polynôme* $X^n - 1$ *dans* $\Bbbk[X]$ *pour cela il faut décomposer le polynôme* $X^n - 1$ *en produit de polynômes irréductibles sur* F_p.

Exemple 2.2.12 *Les codes cycliques non nuls de longueur* $n = 9$ *sur le corps des racines* $9^{ème}$ *de l'unité* $\Bbbk = F_{64}$. *La décomposition de* $X^9 - 1$ *en produit de polynômes irréductibles sur* F_2 *est donnée par :*

$$X^9 - 1 = (X-1)(X^2+X+1)(X^3+X+1)(X^3+X^2+1).$$

Le tableau ci-dessous nous donne les différentes valeurs possibles du polynôme générateur $g(X)$ *des codes cycliques de longueur* $n = 9$.

Le générateur $g(X)$	*Le Code* C
1	$<1>=\mathbb{k}^9(trivial)$
X^2+X+1	$<X^2+X+1>$
X^3+X^2+1	$<X^3+X^2+1>$
$(X-1)(X^3+X+1)$	$<X^4+X^3+X^2+1>$
$(X^2+X+1)(\ X^3+X+1)$	$<X^2+X+1(X-1)(\ X^3+X^2+1)>$
$(X^3+X+1)(X^3+X^2+1)$	$<(X^3+X+1)(X^3+X^2+1)>$
$(X-1)(\ X^2+X+1)(X^3+X^2+1)$	$<(X-1)(\ X^2+X+1)(X^3+X^2+1)>$
$(\ X^2+X+1)(X^3+X+1)(X^3+X^2+1)$	$<(\ X^2+X+1)(X^3+X+1)(X^3+X^2+1)>$
$X-1$	$<X-1>$

Le générateur $g(X)$	*Le Code* C
X^3+X+1	$<X^3+X+1>$
$(X-1)(X^2+X+1)$	$<X^3+1>$
$(X-1)(X^3+X^2+1)$	$<X^4+X^2+X+1>$
$(X^2+X+1)(X^3+X^2+1)$	$<(X^2+X+1)(X^3+X^2+1)>$
$(X-1)(\ X^2+X+1)(X^3+X+1)$	$<(X-1)(\ X^2+X+1)(X^3+X+1)>$
$(X-1)(X^3+X+1)(X^3+X^2+1)$	$<(X-1)(X^3+X+1)(X^3+X^2+1)>$
$X^9-1=0$	$\{0\}$

Théorème 2.2.13 *Soit* C *un code cyclique de longueur* n *sur un corps fini* $\mathbb{k}$ *et de générateur* $g(X)=g_0+g_1X+g_2X^2+...+X^t$ *avec* $d^\circ g=t$. *Alors* dim $C=n-t$. *et* C *admet comme matrice génératrice* $G\in M_{k,n}(\mathbb{k})$. *Tel que :*

$$G=\begin{pmatrix} g_0 & g_1 & ... & g_{t-1} & g_t & 0 & ... & 0 & 0 \\ 0 & g_0 & g_1 & ... & g_{t-1} & g_t & ... & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & g_0 & g_1 & \cdots & g_{t-1} & g_t & 0 \\ 0 & 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_{t-1} & g_t \end{pmatrix}=\begin{pmatrix} g(X) \\ Xg(X) \\ \vdots \\ \vdots \\ X^{n-t-1}g(X) \end{pmatrix}.$$

Preuve. Soit C un code cyclique de longueur n sur $\mathbb{k}$ et $g(X)$ le générateur de C tel que $d^\circ(g(X))=t$. Tout polynôme $c(X)$ de la représentation $\theta(X)$ est de la

forme :

$c(X) = a(X).g(X) = (a_0+a_1X+...+a_sX^s)g(X) = a_0g(X)+...+a_sX^sg(X)$,

avec $a_s \in \Bbbk$ et $0 \leq s \leq n-1$, les polynômes $g(X)$, $Xg(X)$, ..., $X^sg(X)$ forment donc une famille génératrice de $\theta(X)$. On va extraire de cette famille génératrice une base.

Soit $c(X) = a(X).g(X) \in \theta(C)$ et $h(X) = X^n - 1/\ g(X)$ dans $\Bbbk[X]$. En utilisant la division Euclidienne de $a(X)$ par $h(X)$ dans $\Bbbk[X]$, on obtient :

$a(X) = q(X)h(X)+r(X)$, $d^\circ(r(X)) < d^\circ(h(X)) = n-t$ donc :

$r(X) = r_0+r_1X+...+r_{n-t-1}X^{n-t-1}$, en conséquence :

$a(X).g(X) = q(X)h(X)g(X)+r(X)g(X) = (X^n-1)q(X)+r(X)g(X)$.

En calculant dans l'éspace quotient $\Bbbk[X]$ / $g(X)$ on déduit que : $a(X).g(X) = r(X).g(X)$ donc $c(X) = r_0g(X) + r_1Xg(X) \ + ... + r_{n-t-1}X^{n-t-1}g(X)$ d'où la famille des polynômes $g(X), Xg(X)$, $..., X^{n-t-1}g(X)$ est une famille génératrice de $\theta(C)$. Montrons que cette famille est libre : Dans $\Bbbk[X]/ < X^n - 1 >$ concidérons l'égalité :

$a_0g(X) + a_1Xg(X) \ + ... + a_{n-t-1}X^{n-t-1}g(X) = 0.....(*)$ avec $a_i \in \Bbbk$ et $i \in \{0, n-t-1\}$.

L'égalité $(*)$ implique que dans $\Bbbk[X]$: $(a_0 + a_1X \ + ... + a_{n-t-1}X^{n-t-1})g(X) \equiv 0 \pmod{X^n - 1}$ posons : $d(X) = (a_0 + a_1X \ + ... + a_{n-t-1}X^{n-t-1})g(X)$ alors $d(X)$ est de degré au plus $n-1$ et divisible par $X^n - 1$ alors $d(X) = 0$ (polynôme nul) comme $\Bbbk[X]$ est intègre et $g(X)$ n'est pas nul alors $a_0+a_1X \ +...+a_{n-t-1}X^{n-t-1} = 0$ (polynôme nul).

Donc : $a_0 = a_1 = ... = a_{n-t-1} = 0$, d'où la famille $g(X), Xg(X)$, $..., X^{n-t-1}g(X)$ est libre, donc la famille $\{g(X), Xg(X)$, $..., X^{n-t-1}g(X)\}$ est une base de $\theta(C)$.

La dimension de C est $n-t$, et les mots $L_0, L_1, ..., L_{n-t}$ correspond respectivement au polynôme $g(X), Xg(X)$, $..., X^{n-t-1}g(X)$ forme une base du code C et la matrice G dont les lignes $L_1 = g_0g_1g_2...g_t0...0$, , $L_2 = 0g_0g_1g_2...g_t0...0$, ..., $L_{n-t} = 0...0g_0g_1g_2...g_t$, est une matrice génératrice de C. ■

Exemple 2.2.14 *Le code de Hamming est un code cyclique de paramètre* $(7,\ 4,\ 3)$ *et de polynôme générateur* $g(X) = 1+X+X^3$ *admet pour matrice génératrice,*

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

2.2.3 Orthogonal et matrice de contrôle d'un code cyclique.

Définition 2.2.15 *Soit C un code cyclique de longueur n sur un corps fini $\mathbb{k}$ dont le générateur est $g(X)$ tel que $d^\circ(g(X)) = t$. Le code orthogonal (ou dual) du code C noté $C^\perp$, est le sous-espace orthogonal à C au moyen du produit scalaire usuel dans $\mathbb{k}^n$.*

Remarque 2.2.16

i) Si $t = 0$ alors $k = \dim(C) = n - t = n$, et $C = \mathbb{k}^n$ (code trivial) et $C^\perp = \{0\}$.

ii) Si $t \geq 1$ alors $k = \dim(C) = n - t = k$, donc $\dim(C^\perp) = (n - t) = k$ et $\dim(C^\perp) = n - (n - t) = t$.

Définition 2.2.17 *Soit C un code cyclique de longueur n sur un corps fini $\mathbb{k}$ et de polynôme générateur $g(X)$. Le polynôme $h(X) \in \mathbb{k}[X]$ tel que $h(X) = \frac{X^n-1}{g(X)}$ est dit polynôme de contrôle de C.*

Remarque 2.2.18

i) Le degré de $h(X)$ est donc $n-\deg(g(X)) = n-t = k$.

ii) Si c est un mot de C alors $c(X).h(X) = 0$ dans $\mathbb{k}[X]/ < X^n - 1 >$.

Théorème 2.2.19 *Soit C un code cyclique de longueur n sur un corps fini $\mathbb{k}$.*

1. *L'orthogonal $C^\perp$ d'un code cyclique est un code cyclique.*

2. *Si $h(X) = h_0 + h_1X + h_2X^2 + ... + h_kX^k$ est le polynôme de contrôle du code cyclique C non trivial alors le générateur de $C^\perp$ est $h_1(X) = h_0^{-1}\overline{h}(X)$ où*
$\overline{h(X)} = X^k h(X^{-1})$ est le polynôme réciproque de $h(X)$.

3. *La matrice H_1 suivante est une matrice de contrôle de C.*

$$H_1 = \begin{pmatrix} h_k & h_{k-1} & ... & h_1 & h_0 & 0 & ... & 0 & 0 \\ 0 & h_k & h_{k-1} & ... & h_1 & h_0 & ... & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & h_k & h_{k-1} & \cdots & h_1 & h_0 & 0 \\ 0 & 0 & \cdots & 0 & h_k & h_{k-1} & \cdots & h_1 & h_0 \end{pmatrix}.$$

Preuve. On a $X^n - 1 = h(X)g(X)$ alors $h(X)g(X) = 0$ dans $\Bbbk[X] / < X^n - 1 >$, soit $a(X) = \sum_{i=0}^{n-1} a_i X^i \in \theta(X)$ donc c'est un multiple de $g(X)$ dans $\Bbbk[X] / < X^n - 1 >$, si $h(X) = \sum_{j=0}^{n-1} h_j X^j$ alors $h(X)a(X) = \sum_{i=0}^{n-1} (\sum_{j=0}^{i} a_j h_{i-j}) X^{i+j} = 0$ (où les différences $i+j$ sont calculées modulo n), on déduit que $\forall i \in [0, n-1] : \sum_{j=0}^{i} a_j h_{i-j} = 0$ pour $i \in [k, n-1]$ on trouve les relations suivantes :

Si $(i = k)$ alors $a_0h_k+a_1h_{k-1}+a_2h_{k-2}+...+a_kh_0+a_{k+1}0+...+a_{n-1}0 = 0$ donc :

$(h_k, h_{k-1}, h_{k-2}, ..., h_0, 0, ..., 0) \perp (a_0, a_1, a_2, ..., a_k, a_{k+1}, ..., a_{n-1})$.

Si $(i = k+1)$ alors $a_0 0 + a_1h_k + a_2h_{k-1} + ... + a_kh_1 + a_{k+1}h_0 + ... + a_{n-1}0 = 0$ donc :

$(0, h_k, h_{k-1}, ..., h_0, 0, ..., 0) \perp (a_0, a_1, a_2, ..., a_k, a_{k+1}, ..., a_{n-1})$.

Si $(i = k+2)$ alors $a_0 0 + a_1 0 + a_2h_k + ... + a_kh_2 + a_{k+1}h_1 + ... + a_{n-1}0 = 0$ donc :

$(0, 0, h_k, ..., h_0, 0, ..., 0) \perp (a_0, a_1, a_2, ..., a_k, a_{k+1}, ..., a_{n-1})$.

Si $(i = n-1)$ alors $a_0 0 + a_1 0 + a_2 0 + ... + a_{n-k-2}0 + a_{n-k-1}h_k + ... + a_{n-1}h_0 = 0$ donc :

$(0, 0, 0, ..., 0, h_k, h_{k-1}, ..., h_1, h_0) \perp (a_0, a_1, a_2, ..., a_k, a_{k+1}, ..., a_{n-1})$. Les relations précédentes montrent que les shifts du mot $(h_k, h_{k-1}, h_{k-2}, ..., h_0, 0, ..., 0)$ sont orthogonaux au mot $a = (a_0, a_1, a_2, ..., a_k, a_{k+1}, ..., a_{n-1}) \in C$. En d'autres termes les mots suivants : $(h_k, h_{k-1}, h_{k-2}, ..., h_1, h_0, 0, 0, 0, ..., 0)$, $(0, h_k, h_{k-1}, h_{k-2}, ..., h_1, h_0, 0, 0, ..., 0)$, $(0, 0, 0, ..., 0, h_k, h_{k-1}, ..., h_1, h_0)$ sont orthogonaux au code C donc ils appartiennent à $C^\perp$. La matrice constituée des $t = n-k$ premières colonnes extraites du tableau ci-dessus est triangulaire inversible car $h_k \neq 0$. La matrice H_1 formée par ces mots est de rang t et ces lignes forment une base à $C^\perp$ donc c'est une matrice de controle du code C. $h(0) = h_0 \neq 0$ car $h(X)$ divise $X^n - 1$ alors la matrice $H = h_0^{-1} H_1$ est aussi

une matrice de controle de C, de plus le polynôme associé à la première ligne de H. $h_1(X) = h_0^{-1}h_k + h_0^{-1}h_{k-1}X + ... + X^k = h_0^{-1}\overline{h(X)}$ (où $\overline{h(X)}$ est le polynôme réciproque de $h(X)$), est un polynôme unitaire divisant $X^n - 1$. ■

Exemple 2.2.20 *Soit C un code cyclique sur F_2 de longueur 9 et de générateur $g(X) = X^3+X^2+1$. Alors le polynôme de contrôle de C est :*
$h(X) = X^9 - 1 \ / \ X^3 + X^2 + 1 \ = X^6 + X^4 + X + 1$. *L'othogonal de C est engendré par le polynome : $h_1(X) = h_0^{-1}\overline{h(X)}$ tel que $\overline{h(X)} = X^6h(X^{-1}) = 1 + X^2 + X^5 + X^6$, et sa matrice génératrice :* $H_1 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$.

2.2.4 Codes cycliques systématiques et codages systématique.

Définition 2.2.21 *Un code cyclique $C(n,\ k)$ sur un corps fini $\Bbbk$ est dit systématique (où normalisé) s'il admet une matrice génératrice G_N dite normalisée dont les k denières colonnes forment la matrice identité I_k et non pas les k premières colonnes dans le cas des codes linéaires, c-à-d : $G_N = (M,\ I_k)$ où $M \in M_{k,n-k}(\Bbbk)$.*

Exemple 2.2.22 *Le code cyclique C de longueur $n = 7$ et de générateur $g(X) = 1 + X + X^3$ sur F_2, admet comme matrice génératrice :*

$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$ *de la forme $G = (A, B)$ avec* $B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$

matrice carrée d'ordre 4 inversible donc c'est un code cyclique systématique de matrice génératrice

$G_N = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$.

Définition 2.2.23 *On appelle syndrome polynômiale d'un mot* $y(X) \in F_q[X] \ / < X^n - 1 >$, *le reste de la division Euclidiene de* $y(X)$ *par* $g(X)$ *dans* $F_q[X]$, *on le note* $S(\ y(X))$.

Proposition 2.2.24 *Soit* C *un code cyclique de longueur* n *et de générateur* $g(X)$ *tel que* $d^\circ(g(X)) = t$. *Le codage systématique d'un mot* $a = (a_0, a_1, ..., a_{k-1})$ *de* $\Bbbk^k$ *se fait en représentation polynômiale par :* $\quad a(X) \to c(X) = -S(X^t a(X)) + X^t a(X)$.

Preuve. Soit C un code cyclique de longueur n sur un corps fini $\Bbbk$ de générateur $g(X) = g_0 + g_1 X + g_2 X^2 + ... + X^t$ tel que $d^\circ(g(X)) = t$.
Le code C admet une matrice génératrice (pas nécessairement normalisée) G_1 de la forme :

$$G_1 = \begin{pmatrix} g_0 & g_1 & g_2 & \cdots & g_t & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & \cdots & g_t & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & g_0 & \cdots & g_t \end{pmatrix} = \begin{pmatrix} g(X) \\ Xg(X) \\ \vdots \\ X^{n-t-1}g(X) \end{pmatrix}.$$

$G_1 = (M, T)$ où $M \in M_{k,n-k}(\Bbbk)$ et T est constitué des $k = n - t$ dernières colonnes de G_1.
De plus T est inversible car $g_t \neq 0$. On considère la matrice $G_N = T^{-1}.G_1 = (N,\ I_k)$ tel que $N = T^{-1}.M$ donc G_N est la matrice génératrice normalisée de C qu'on utilise pour le codage comme suit :
Soit le mot $a = (a_0, a_1, ..., a_{k-1}) \in \Bbbk^k$, le mot code c est le produit de a par la
matrice G_N. On a $c = a.G_N = (a.N,\ a) = [(a_0, a_1, ..., a_{k-1}).N, a_0, a_1, ..., a_{k-1}]$ et en posant $(a_0, a_1, ..., a_{k-1}).N = (b_0, b_1, ..., b_{n-k-1})$ donc $c = (b_0, b_1, ..., b_{n-k-1}, a_0, a_1, ..., a_{k-1})$ alors en langage polynômiale on obtient :
$c(X) = b_0 + b_1 X + ... + b_{n-k-1}X^{n-k-1} + a_0 X^{n-k} + a_1 X^{n-k-1} + ... + a_{k-1}X^{n-1}$ d'où $c(X) = b(X) + X^{n-k}a(X)$ où $b(X) = b_0 + b_1 X + ... + b_{n-k-1}X^{n-k-1}$ qu'il faut déterminer. Comme $c(X) \in \theta(C)$ alors $\exists\ u(X) \in \Bbbk[X]$ tel que $c(X) = u(X)g(X)$ et donc $X^{n-k}a(X) = u(X)g(X) + (-b(X))$ avec $d^\circ(-b(X)) < d^\circ(g(X))$ cela veut dire que $(-b(X))$ n'est que $r(X)$ le reste de la division Euclidienne de $X^{n-k}a(X)$ par $g(X)$. Donc le codage systématique d'un mot $a = (a_0, a_1, ..., a_{k-1})$ de $\Bbbk^k$ se fait en représentation

polynômiale par : $a(X) \rightarrow c(X) = -S(X^t a(X)) + X^t a(X)$. ■

2.2.5 Quelques codes cycliques particulièrs.

On présente dans cette section quelques codes cycliques particulières utiliser en pratique tel que les codes B.C.H et les codes de Reed-Solomon. Dans ce type de codes on choisit leur capacité de correction e avant leur construction et cela en choisisant un polynôme générateur admettant $2e$ racines de puissaces successives de l'une de ces racines niemes primitive de l'unité.

Les codes B.C.H.

Proposition 2.2.25 *Soit C un code cyclique de longueur n sur $F_q = F_{p^r}$ (corps des racines nièmes de l'unité) de générateur $g(X)$ tel que $d^\circ(g(X)) = t$ qui admet pour racines $\alpha_1,\ \alpha_2,\ ...,\ \alpha_t \in F_{p^r}$. Alors la matrice*

$$H = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{n-1} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & \alpha_t & \alpha_t^2 & \cdots & \alpha_t^{n-1} \end{pmatrix}$$

est une matrice de contrôle du code C.

Preuve. Un élément $c = (c_0, c_1, ..., c_{n-1}) \in (F_{p^r})^n$ est alors un mot du code C si et seulement si le polynôme $c(X)$ est un multiple de $g(X)$, c-à-d s'il admet les α_i pour racines. Donc $c \in C \Leftrightarrow c(X) \in \theta(C) \Leftrightarrow \forall i = 1, ..., t : c(\alpha_i) = 0 \Leftrightarrow cH^t = 0$ tel que H est la matrice donnée par : $H = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{n-1} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & \alpha_t & \alpha_t^2 & \cdots & \alpha_t^{n-1} \end{pmatrix} \Leftrightarrow H$ est une matrice de contrôle de C. ■

Théorème 2.2.26 *Soit C un code cyclique de longueur n sur $F_q = F_{p^r}$ le corps des racines n-ièmes de l'unité sur F_p , de distance minimale d et de polynôme générateur*

$g(X)$, b et δ deux entiers tel que b non nul et $\delta \geq 2$.
Si $g(X)$ admet $\delta - 1$ racines de puissances successives $\beta^b, \beta^{b+1}, ..., \beta^{b+\delta-2}$ dans F_q (où β est une racine nième primitive de l'unité). Alors $d \geq \delta$.

Preuve. On va supposer $b = 1$ pour la preuve (sans perte de généralité), et on pose $\alpha_1 = \beta,\ \alpha_2 = \beta^2, ..., \alpha_{\delta-1} = \beta^{\delta-1}$. D'après la proposition ci-dessus on trouve que C admet comme matrice de contrôle la matrice

$$H = \begin{pmatrix} 1 & \beta & \beta^2 & \dots & \beta^{(n-1)} \\ 1 & \beta^{b+1} & \beta^{2(b+1)} & \dots & \beta^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \beta^{\delta-2} & \beta^{2(\delta-2)} & \dots & \beta^{(n-1)(\delta-2)} \\ 1 & \beta^{\delta-1} & \beta^{2(\delta-1)} & \dots & \beta^{(n-1)(\delta-1)} \end{pmatrix}$$

. Il faut montrer que tout choix de $\delta-1$ colonnes de H définit une matrice carrée M de rang plein (c-à-d : $\det(M) \neq 0$). Prenons M la matrice carrée d'ordre $\delta - 1$ extraite des colonnes $C_{j_1}, C_{j_2}, ..., C_{j_{\delta-1}}$ correspondant aux puissances $j_1, j_2, ..., j_{\delta-1}$ des $\beta^{\,i}$, tel que $i \in \{1, \delta - 1\}$.

$$M = \begin{pmatrix} \beta^{\,j_1} & \beta^{\,j_2} & \dots & \beta^{\,j_{\delta-1}} \\ \beta^{2\,j_1} & \beta^{2\,j_2} & \dots & \beta^{2\,j_{\delta-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \beta^{(\delta-1)\,j_1} & \beta^{(\delta-1)\,j_2} & \dots & \beta^{\,(\delta-1)j_{\delta-1}} \end{pmatrix}$$

. L'expression de droite indique que M est une matrice de Vandermonde, dont le déterminant vaut :
$\det(M) = \beta^{\sum_{k=1}^{k=\delta-1} j_k} \prod_{1 \leq u < v \leq \delta-1} (\beta^{\,jv} - \beta^{\,ju}) \neq 0$. Donc les $\delta - 1$ colonnes de H sont linéairement indépendantes donc le code C est de distance $d > \delta - 1$ alors $d > \delta$.

■

Définition 2.2.27 *La borne inférieur $d \geq \delta$ pour la distance minimale d est dite borne B.C.H et l'entier δ est dit distance construite du code C.*

Définition 2.2.28 *Un code B.C.H de distance construite δ, est un code cyclique dont le générateur $g(X)$ est le produit (sans répétition de facteurs) des polynômes minimaux de $\delta-1$ racines de puissances successives $\beta^b, \beta^{b+1}, ..., \beta^{b+\delta-2}$ de $g(X)$.*

Remarque 2.2.29

1. *Si $b = 1$ le code B.C.H est dit code B.C.H au sens strict.*

2. *Si la longueur du code B.C.H est $n = p^r - 1$ alors le code B.C.H est dit code B.C.H primitif. Si α est une racine primitive de $\mathbb{k}$ et si β est une racine nième primitive de l'unité alors $\beta = \alpha$.*

3. *On définit aussi un code B.C.H de longeur n sur $\mathbb{k}$, un code cyclique dont le générateur est le PPCM des polynômes minimaux des racines $\beta^b, \beta^{b+1}, ..., \beta^{b-\delta-2}$.*

Matrice de controle d'un code B.C.H

Soit C un code B.C.H de générateur $g(X)$ qui admet $\delta - 1$ racines successives $\beta^b, \beta^{b+1}, ..., \beta^{b+\delta-2}$. On a : $g(\beta^r) = g(\beta^{r+1}) = ... = g(\beta^{r-\alpha-2}) = 0$.

Comme dans la proposition ci-dessus on trouve que la matrice de controle d'un code B.C.H de distance construite δ est une matrice de type $(\delta - 1) \times n$ de la forme :

$$H = \begin{pmatrix} 1 & \beta^b & \beta^{2b} & \dots & \beta^{(n-1)b} \\ 1 & \beta^{b+1} & \beta^{2(b+1)} & \dots & \beta^{(n-1)(b+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \beta^{b+\delta-3} & \beta^{2(b+\delta-3)} & \dots & \beta^{(n-1)(b+\delta-3)} \\ 1 & \beta^{b+\delta-2} & \beta^{2(b+\delta-2)} & \dots & \beta^{(n-1)(b+\delta-2)} \end{pmatrix}.$$

Construction d'un code B.C.H.

La réalisation d'un code B.C.H de longueur n de capacité de correction e peut se faire de la manière suivante :

1. Construite le corps F_q des racines nièmes de l'unité.

2. On détemine le polynôme primitif de F_q.

3. On choisit $2e = \delta - 1$ puissance successives de β c-à-d : $\beta^b, \beta^{b+1}, ..., \beta^{b+\delta-2}$.

4. On construit le polynôme générateur $g(X) = \prod_{j=b}^{b+\delta-2} M_{\beta^j}(X)$ (Sans répitition de facteurs) $= PPCM(M_{\beta^j}(X),\ j \in \{b, ..., b+\delta-2\})$.

Les codes Reed-Solomon.

Définition 2.2.30 *Soit m un entier tel que $m \geq 2$. Un code de Reed-Solomon de longueur $n = 2^m - 1$ est un code B.C.H primitif de longueur $n = 2^m - 1$ sur le corps de Galois $\Bbbk = F_{2^m}$.*

Remarque 2.2.31

1. *Tout les éléments non nuls de F_{2^m} sont des racines de $X^{2^m-1}-1$ car dans ce cas le groupe des racines niemes de l'unité $G_n(\Bbbk) = \Bbbk^*$.*

2. *En conséquence la décomposition sur $\Bbbk$ de $X^{2^m-1}-1$ est $X^{2^m-1}-1 = \prod_{u\in\Bbbk^*}(X-u)$. Si α est une racine primitive de $\Bbbk$ on obtient :*

$$X^{2^m-1}-1 = (X-1)(X-\alpha)...(X-\alpha^i)...(X-\alpha^{2^m-2}) .$$

3. *Le générateur $g(X)$ de degré t d'un code de Reed-Solomon (qui admet donc t racines de puissances successives) est donc de la forme : $g(X) = (X-\alpha^i)(X-\alpha^{i+1})...(X-\alpha^{i+t-1})$.*

propriétés d'un code de Reed-Solomon.

1. La dimension de code Reed-Solomon est $k = n - t = 2^m - 1 - t$.
2. La distance construite δ. on a $\delta - 1 = t$ racines successive donc $\delta = t + 1$.
3. On a $d \leq n - k + 1$, d'autre part on a $d \geq \delta = t + 1$ donc on obtient $d = t + 1$. Ce qui signifie que le code Reed-Solomon est un code M.D.S.

Exemple 2.2.32 *Soit le corps $\Bbbk = F_8 = F_{2^3}$. La longueur du code R-S est $n = 2^m-1 = 2^3-1 = 7$. Le code R-S au sens strict est engendré par $g(X) = (X-\alpha)(X-\alpha^2)(X-\alpha^3)$ on trouve $g(X) = X^3 + \alpha^6X^2 + \alpha X + \alpha^6$ et admet comme matrice génératrice :* $G = \begin{pmatrix} \alpha^6 & \alpha & \alpha^6 & 1 & 0 & 0 & 0 \\ 0 & \alpha^6 & \alpha & \alpha^6 & 1 & 0 & 0 \\ 0 & 0 & \alpha^6 & \alpha & \alpha^6 & 1 & 0 \\ 0 & 0 & 0 & \alpha^6 & \alpha & \alpha^6 & 1 \end{pmatrix}$.

2.2.6 Quelques méthodes de décodages des codes cycliques.

Décodage par syndrome polynômiale.

Proposition 2.2.33 *Soit* $y(X) \in F_q[X] \,/\, < X^n - 1 >$, $y(X) \in \theta(X)$ *si et seulement si* $g(X)$ *divise* $y(X)$ *dans* $F_q[X]$ *c-à-d* $y(X) \in \theta(X)$ *si et seulement si* $S(y(X) = 0$.

Preuve. $y(X) \in \theta(X) \Leftrightarrow y(X) \in < g(X) > \Leftrightarrow \exists\, q(X) \in F_q[X] \,/\, < X^n - 1 >$ tel que $y(X) = g(X)q(X) \Leftrightarrow$ le reste de la division Euclidienne de $y(X)$ par $g(X)$ est nul

$\Leftrightarrow S(y(X)) = 0$. ■

L'algorithme de décodage d'un code cyclique :

Soit C un code cyclique de paramètre $(n,\ k,\ d)$ sur le corps F_q de générateur $g(X)$ et de capacité de correction e, et soit $C(X)$ le mot envoyé et $y(X)$ le mot reçu dont l'erreur est de poids $w(\varepsilon(X)) \leq e$ alors l'algorithme de décodage est le suivant :

1. Calculer le syndrome du mot reçu.
2. Trouver l'erreur $\varepsilon(X)$ qui correspond au syndrome de $y(X)$ et $w(\varepsilon(X)) \leq e$.
3. Le mot envoyé est $C(X) = y(X) - \varepsilon(X)$.

Exemple 2.2.34 *Soit le code cyclique systématique* $C(7,\ 4)$ *de générateur* $g(X) = X^3+X+1$ *et soit le mot reçu* $y(X) = X^5+X^4+X^2$.

1.*Calculer le syndrome du mot* $y(X)$:

Le reste $r(X) = S(y(X)) = X^2+1$.

2.*Cherchons l'erreur* $\varepsilon(X)$ *de poids* $w(\varepsilon(X)) = 1$ *et de syndrome* $X^2 + 1$ *alors l'erreur* $\varepsilon(X)$ *est de la forme* $\varepsilon(X) = X^i$ *on remarque que* $S(X^6) = X^2 + 1$. *Donc le mot erreur est* $\varepsilon(X) = X^6$ *d'où le mot envoyé est :* $C(X) = X^6 + X^5 + X^4 + X^2$.

Proposition 2.2.35 *Si* $y_1(X)$ *et* $y_2(X)$ *deux mots tel que* $S(y_1(X)) = S(y_2(X))$ *et* $w(y_1(X)) \leq e$ *et* $w(y_2(X)) \leq e$ *alors* $y_1(X) = y_2(X)$.

Méthode de décodage de Méggit.

La méthode de décodage de Meggit s'applique aux codes cycliques binaires, mais elle peut se généraliser au cas non binaire. L'idée de base consiste en l'utilisation de la

cyclicité du code pour resteindre (reduire) la table des syndromes et permettre des calculs récursifs. Le décodeur de Meggit effectue un décodage symbole par symbole. On corrige d'abord une composante erronée du mot reçu au moyen de la méthode décrite ci-dessous, puis on applique de nouveau la méthode au nouveau mot reçu ainsi obtenu.

Un autre avantage de cette méthode réside dans le fait qu'on remplace le tableau de déchiffrement qui comporte tous les mots erreurs et tous les syndromes, par un tableau où ne figurent que les syndromes des mots erreurs donts le dernier symbole est erronné. On gagne ainsi beaucoup de place en mémoire et temps.

Proposition 2.2.36 *Soit $c(X)$ le mot envoyé, $y(X)$ le mot reçu et $\varepsilon(X)$ le mot erreur associé. Alors pour tout entier j tel que $0 \leq j \leq n-1$.*
1. *Le mot $X^j y(X)$ est un mot reçu dont l'erreur est $X^j \varepsilon(X)$.*
2. *$S(X^j \varepsilon(X)) = S(X^j y(X))$.(Tous les produits sont calculés dans l'anneau $F_2[X] \,/ < X^n - 1 >$).*

Preuve. On a : $y(X) = c(X) + \varepsilon(X)$ avec $c(X) \in C$, on déduit $X^j y(X) = X^j c(X) + X^j \varepsilon(X) ...(*)$, tel que $0 \leq j \leq n-1$. Le code C étant cyclique, on sait que $X^j c(X) \in C$. D'autre part $w(X^j \varepsilon(X)) = w(\varepsilon(X))$ car la multiplication par X^j ne modifie pas le poids d'un mot. L'égalité $(*)$ montre donc que $X^j \varepsilon(X)$ est le mot erreur du mot reçu $X^j y(X)$. Il existe $c(X)$ multiple de $g(X)$ dans $F_2[X] \,/ < X^n - 1 >$ tel que $y(X) = c(X) + \varepsilon(X)$. On obtien donc dans $F_2[X] \,/ < X^n - 1 >$ une relation de la forme $X^j y(X) = X^j c(X) + X^j \varepsilon(X)$. Ce ci implique dans $F_2[X]$ une égalité de la forme $X^j y(X) = X^j c(X) + X^j \varepsilon(X) + b(X)(X^n - 1)$. Puisque $g(X)$ divise $X^n - 1$ dans $F_2[X]$, on voit que $X^j y(X) \equiv X^j \varepsilon(X) \bmod g(X)$, ce qui montre que $X^j y(X)$ et $X^j \varepsilon(X)$ ont le même reste dans la division par $g(X)$, c-à-d le même syndrome. ■

Remarque 2.2.37 *Si on trouve $S(X^j y(X))$ dans une table de sydrome indiquant l'erreur correspondante, on peut trouver $X^j \varepsilon(X)$ et donc aussi $\varepsilon(X)$.*

Proposition 2.2.38 *Avec les notations de la proposition précédente, soit $S_j(X)$ la suite*

de polynômes de $F_2[X] / < X^n - 1 >$ *définie par :* $\begin{cases} S_0(X) = S(y(X)), \\ S_{j+1}(X) = S(XS_j(X)). \end{cases}$

Alors pour tout entier j *tel que* $0 \leq j \leq n-1$ *on a :* $S_j(X) = S(X^j y(X))$.

Preuve. Montrons par récurrence.

Pour $j = 0$ on trouve $S_0(X) = S(y(X) = S(X^0 y(X))$.

Pour $j = 1$ on trouve $S_1(X) = S(XS_0(X)) = S(XS(y(X))) = S(Xy(X))$.

On suppose vraie pour $j = k$ et on montre vraie pour $j = k+1$, soit

$S_k(X) = S(X^k y(X))$. De la définition de la suite $S_j(X)$ ona :

$S_{k+1}(X) = S(XS_k(X)) = S(XS(X^k y(X))) = S(X.X^k y(X))$ donc $S_{k+1}(X) = S(X^{k+1} y(X))$.

■

Algorithme de décodage de Meggit.

Soit T la table des syndromes des erreurs dont la composante d'indice $n-1$ est erronée.

Soit $C(X)$ le mot envoyé, $y(X)$ le mot reçu, et $\varepsilon(X)$ le mot erreur avec $w(\varepsilon(X)) \leq e$.

$\varepsilon(X)$	X^{n-1}
$S(\varepsilon(X))$	$S(X^{n-1})$

.

L'algorithme de décodage de Meggit est le suivant :

1. Calculer $S(y(X))$.
2. Si $S(y(X) = 0$ alors $C(X) = y(X)$ et l'algorithme se termine.
3. Sinon $(S(y(X) \neq 0)$

   a. Rechercher le plus petit entier j tel que $S(X^j y(X)) \in T$

   (c-à-d $S(X^j y(X) = S(X^{n-1}))$.

   b. Trouver l'erreur $\varepsilon(X) = X^{n-j-1}$ (c'est à dire corriger la composante d'indice $n-j-1$).

   c. Calculer le mot obtenu $y'(X) = y(X) - \varepsilon(X)$

   - Si $y'(X) \in \theta(X)$ alors $C(X) = y'(X)$.
   - Sinon répartir au début de l'algorithme avec $y'(X)$.

Exemple 2.2.39

1. *Soit* $C(7,\ 4,\ 3)$ *un code cyclique binaire sur le corps* F_2*, et soit* $g(X)$ *le polynôme générateur,* $g(X) = X^3+X+1$*. Soit* $y(X) = X^5+X^4+X^3+X^2+X+1$*.*

2. $S(y(X)) = X^2 + 1 \neq 0$*. Le tableau réduit est* $T =$

$\varepsilon(X)$	X^6
$S(\varepsilon(X))$	X^2+1

on trouve $S(y(X)) \in T$ *donc l'erreur est* $\varepsilon(X) = X^6$ *et*

$C\ (X) = y(X) + \varepsilon(X) = X^6 + X^5 + X^4 + X^3 + X^2 + X + 1.$

Décodage par piégeage d'erreur.

Supposons $C(n,\ k)$ un code cyclique $e-$correcteur sur le corps F_q de polynôme générateur $g(X)$.

Supposons que $C(X) \in C$ est le mot transmis et $y(X) = c(X) + \varepsilon(X)$ est le mot reçu, où $\varepsilon(X)$ est le mot erreur avec $w(\varepsilon(X)) \leq e$. La méthode de décodage par piégeage d'erreur est une modification de la méthode de Meggit, il s'agit de déplacer par décalage circulaire, c'est à dire "piéger" en quelque sorte, les composantes non nulles de l'erreur sur certaines positions. On considère un code non nécessairement binaire, et on suppose que le nombre d'erreurs ne dépasse pas la capacité de correction e. Le principe du décodage par piégeage d'erreur s'appuie sur les résultats suivants.

Lemme 2.2.40 *Soit* $\varepsilon(X)$ *le mot erreur du mot reçu* $y(X)$*.Si* $\deg \varepsilon(X) \leq n-k-1$ *alors* $\varepsilon(X) = S(\ y(X))$.

Preuve. Dans $F_q[X]\ /\ < X^n - 1 >$ on a $y(X) = c(X) + \varepsilon(X)$ avec $C(X) \in C$ soit encore dans $F_q[X]$ on a $y(X) = a(X)g(X) + \varepsilon(X) + b(X)(X^n - 1)$. Puisque $g(X)$ divise X^n-1 on trouve $y(X) = d(X)g(X)+\varepsilon(X)$. Si $\deg \varepsilon(X) \leq n-k-1$ alors d'après l'unicité du reste dans la division par $g(X)$, on obtient $S(\ y(X)) = \varepsilon(X)$. ■

Lemme 2.2.41 *Soit* $\varepsilon(X)$ *le mot erreur du mot reçu* $y(X)$*, alors* $w(S(\ y(X))) \leq e$ *si et seulement si* $S(\ y(X)) = \varepsilon(X)$.

Preuve. La division dans $F_q[X]$ de $y(X)$ par $g(X)$ s'exprime par :
$y(X) = g(X)a(X) + \varepsilon(X)$. Les conditions sur le degré des polynômes intervenant dans cette égalité, font que celle ci est également vérifiée dans $F_q[X] / < X^n - 1 >$.
La décomposition d'un mot reçu comme somme d'un mot du code et d'un mot de poids inférieur ou égal à e est unique. Donc si $w(S(\ y(X))) \leq e$ alors
$S(y(X)) = \varepsilon(X)$. Réciproquement si $S(\ y(X)) = \varepsilon(X)$ alors $w(S(\ y(X))) \leq e$ puisque le poids de l'erreur est au plus e. ■

Théorème 2.2.42 *Soit $\varepsilon(X)$ le mot erreur du mot reçu $y(X)$. Si $\varepsilon(X) = X^j\varepsilon_1(X)$ avec $\deg(\varepsilon_1(X)) \leq n - k - 1$, alors $w(S(X^{-j}y(X))) \leq e$.*

Preuve. D'après le lemme 2.2.40 comme $\deg(\varepsilon_1(X)) \leq n-k-1$ alors
$\varepsilon_1(X) = X^{-j}\varepsilon(X) = S(X^{-j}y(X))$. On pose $y_1(X) = X^{-j}y(X)$ donc $\varepsilon_1(X) = S(y_1(X))$ d'après le lemme 2.2.41 on trouve $w(S(y_1(X))) \leq e$ alors $w(S(X^{-j}y(X))) \leq e$. ■

Algorithme de décodage par piégeage d'erreur.

Soit $y(X)$ le mot reçu, $\varepsilon(X)$ le mot erreur avec $w(\varepsilon(X)) \leq e$.

1. Calcul de $S(y(X))$.
2. Si $S(y(X)) = 0$ alors $\varepsilon(X) = 0$. Sinon
   - *i)* Si $w(\varepsilon(X)) \leq e$ alors $\varepsilon(X) = S(y(X))$.
   - *ii)* Sinon on cherche le plus petit entier j tel que $w(S(X^j y(X))) \leq e$ alors $\varepsilon(X) = X^{-j}S(X^j y(X))$.
3. Le mot envoyé est $c(X) = y(X) - \varepsilon(X)$.

Exemple 2.2.43

1. *Soit le code C de Hamming $(15, 11, 3)$ sur F_2 tel que la capacité de correction $e = 1$, et soit $g(X) = X^4 + X + 1$. Soit $y(X) = X^5 + X^4 + X^2 + X + 1$ le mot reçu. Le syndrome est le reste de la division de $y(X)$ par $g(X)$ on trouve*
$X^5+X^4+X^2+X+1 = (X^4+X+1)(X+1)+X$, *donc* $S(y(X)) = X$ *et*
$w(S(y(X))) = 1 = e$. *Alors* $\varepsilon(X) = S(y(X))$ *et le mot transmis est*
$c(X) = y(X)-\varepsilon(X) = X^5+X^4+X^2+X+1$.

2. *Soit le code cyclique* $(7,\ 4,\ 3)$ *de générateur* $g(X) = X^3+X+1$. *Soit*
$y(X) = X^5 + X^4 + X^2$ *le mot reçu. Le syndrome est le reste de la division de* $y(X)$ *par* $g(X)$ *on trouve* $X^5 + X^4 + X^2 = (X^3 + X + 1)(X^2 + X + 1) + X^2 + 1$. *Ona* $S(y(X)) = X^2 + 1 \neq 0$ *et* $w(S(y(X))) = 2$. *On cherche le mot erreur* $\varepsilon(X)$ *tel que :*

• $S(\varepsilon(X)) = S(y(X)) = X^2+1$.

• $w(\varepsilon(X)) \leq e = 1 \Leftrightarrow w(\varepsilon(X)) = 1$, *alors* $\varepsilon(X) = X^i \setminus \overline{i = 0, 6}$.

• *Si* $i \in \{0,\ 1,\ 2\}$ *on a* $S(\varepsilon(X)) = \varepsilon(X)$.

• *Si* $i \in \{3,\ 4,\ 5\}$ *on a* $X^i \neq X^2+1$ *et* $S(\varepsilon(X)) = S(X^i) \neq X^2+1$. *On trouve que pour :* $\varepsilon(X) = X^6$ *et* $S(\varepsilon(X)) = S(X^6) = X^2+1$. *Alors le mot envoyé est*
$c(X) = X^6 + X^5 + X^4 + X^2$.

Décodage des codes B.C.H par la Méthode de Newton.

On considère un code B.C.H de longueur n de capacité e et de générateur $g(X)$ admettant $\delta-1 = 2e$ racines $\beta, \beta^2, ..., \beta^{\delta-1}$tel que β une racine n-ième primtive. Supposons que $C(X) \in C$ est le mot transmis et $y(X) = c(X) + \varepsilon(X)$ est le mot reçu, où supposons que $c(X) \in C$ est le mot transmis et $y(X) = c(X) + \varepsilon(X)$ est le mot reçu, où $\varepsilon(X)$ est le mot erreur avec $w(\varepsilon(X)) \leq e$. Si on suppose qu'il y a v erreurs (tel que $v \leq e$) comises en positions $i_1, i_2, ..., i_v$ alors $\varepsilon(X) = x^{i_1} + x^{i_2} + ... + x^{i_v}$.

Algorithme de décodage.

1. Calcul du syndrome.

$S = y.H^t = (s_1, s_2, ..., s_{\delta-1})$. Les s_i se calculent par : $\forall\ 1 \leq i \leq \delta-1 : s_i = y(\beta^i)$ ...(1)

On a d'autre part : $y(\beta^i) = c(\beta^i) + \varepsilon(\beta^i) = \varepsilon(\beta^i)$ car $c(\beta^i) = 0$.

Donc $\forall\ 1 \leq i \leq \delta - 1 : s_i = \varepsilon(\beta^i) = \beta^{i_1} + \beta^{i_2} + ... + \beta^{i_v}$ ...(2).

En posant $\forall\ 1 \leq j \leq v : X_j = \beta^{i_j}$, dits localisateurs de l'erreur. Le système (2) devient : $\forall\ 1 \leq i \leq \delta - 1 : s_i = X_1^i + X_2^i + ... + X_v^i$ ...(3). Afin de trouver les positions i_j il suffit de trouver les X_j pour tout $1 \leq j \leq v$.

2. Calcul des localisateurs.

On détermine le polynôme $\sigma(X) = (1 - X_1X)(1 - X_2X)...(1 - X_vX)$ ...(4) dit

polynôme localisteur dont les X_j^{-1} sont ses racines.

$\sigma(X)$ est de dgré v qui s'écrit de la forme $\sigma(X) = \sigma_v X^v + \sigma_{v-1}X + ... + \sigma_1 X + 1$ ...(5).

Par identification de (4) et (5) on obtient donc au maximum $2e$ équations, à partir des quelles on détermine les σ_j, $1 \leq j \leq v$. Ces equations dites relations de Newton sont données par le système (S) suivant :

$$\forall\ 1 \leq j \leq 2e : s_j + \sum_{i=0}^{i=j-1} \sigma_i s_{j-i} + j\sigma_j = 0\ ....(S).$$

3. Correction des erreurs.

On cherche les racines du polynôme $\sigma(X)$ en testant les valeurs des β^i, $1 \leq i \leq \delta - 1$, possibles. Lorsque l'on connaît les racines de $\sigma(X)$, on trouve les positions auxquelles une erreur s'est produite en prenant les inverses de ces racines.

Exemple 2.2.44 *Considérons le code B.C.H* $C(n = 15,\ k = 7,\ d = 5)$ *de longueur* $n = 15$.

Le corps des racines $15^{èmes}$ *de l'unité sur* F_2 *est* $\Bbbk = F_{16}$ *de racine primitive* α *et de polynôme primitif*

$M_\alpha(X) = X^4 + X + 1$. *On a* $\Bbbk = \{0,\ \alpha^i : 0 \leq i \leq 14\}$, *avec* $\alpha^4 = \alpha + 1$. *On suppose que son générateur est :*

$g(X) = (X^4+X+1)(X^4+X^3+X^2+X+1) = X^8+X^7+X^6+X^4+1$.

En calculant $g(\alpha^i)$ *pour* $0 \leq i \leq 14$ *on trouve tous les racines de g(X) :* $\alpha,\ \alpha^2,\ \alpha^3,\ \alpha^4,\ \alpha^6,\ \alpha^8,\ \alpha^9,\ \alpha^{12}$, *parmis eux il y a quatres racines successives :* $\alpha,\ \alpha^2,\ \alpha^3,\ \alpha^4$. *Supposons que l'on reçoite le vecteur* $y = 000100010100100$ *qu'on veut décoder et qui s'écrit en représentation polynômiale par :* $y(X) = X^3 + X^7 + X^9 + X^{12}$ *contenant* $v = e = 2$ *deux erreurs.*

1. *Calcul du syndrome.*

$s_1 = r(\alpha) = \alpha^3 + \alpha^7 + \alpha^9 + \alpha^{12} = \alpha^2 + \alpha = \alpha^5$.

$s_2 = r(\alpha^2) = \alpha^6 + \alpha^{14} + \alpha^{18} + \alpha^{24} = \alpha^2 + \alpha + 1 = \alpha^{10}$.

$s_3 = r(\alpha^3) = \alpha^9 + \alpha^{21} + \alpha^{27} + \alpha^{36} = \alpha^9 + \alpha^6 + \alpha^{12} + \alpha^6 = \alpha^2 + 1 = \alpha^8$.

$s_4 = r(\alpha^4) = \alpha^{12} + \alpha^{28} + \alpha^{36} + \alpha^{48} = \alpha^{12} + \alpha^{13} + \alpha^6 + \alpha^3 = \alpha^2 + \alpha = \alpha^5$.

2. *Calcul du polynôme localisateur.*

$v = 2$ *donc* $d^\circ(\sigma(X)) = 2$ *et* $\sigma(X) = \sigma_2 X^2 + \sigma_1 X + 1$ *et on resout le système :*

$$\begin{cases} s_1 + \sigma_1 = 0 \quad ...(1) \\ s_2 + \sigma_1 s_1 + 2\sigma_2 = 0 \quad ...(2) \\ s_3 + \sigma_1 s_2 + \sigma_2 s_1 + 3\sigma_3 = 0 \quad ...(3) \\ s_4 + \sigma_1 s_3 + \sigma_2 s_2 + \sigma_3 s_1 + 4\sigma_4 = 0 \quad ...(4) \end{cases} ...(S).$$

Comme le code est binaire et $s_1^2 = s_2$ *alors les équqtions* (1) *et* (2) *sont equivalentes . De plus* $\sigma_4 = \sigma_3 = 0$ *et l'équation* (4) *est toujours verifiée* $(0 = 0)$ *le système* (S) *devient :*

$$\begin{cases} s_1 + \sigma_1 = 0 \quad ...(1) \\ s_3 + \sigma_1 s_2 + \sigma_2 s_1 = 0 \quad ...(2) \end{cases} \Longleftrightarrow \begin{cases} \alpha^5 + \sigma_1 = 0 \quad ...(1) \\ \alpha^8 + \sigma_1 \alpha^{10} + \sigma_2 \alpha^5 = 0 \quad ...(2), \end{cases}$$

de (1) *on déduit :* $\sigma_1 = \alpha^5$ *et en substituant en* (2) *on obtient* $\sigma_2 = \alpha^{12}$ *et le polynôme localisateur est :* $\sigma(X) = \alpha^{12} X^2 + \alpha^5 X + 1$.

3. *Calcul des localisateurs et de l'erreur.*

$\sigma(X)$ *admet comme racines* α^8 *et* α^4 *donc les localisateurs d'erreurs sont les inverses de ces racines c.à.d.* $X_1 = \alpha^7$ *et* $X_2 = \alpha^{11}$. *En fin il y a deux erreurs en* $7^{ème}$ *et* $11^{ème}$ *positions et donc le mot erreur est* $\varepsilon(X) = X^7 + X^{11}$

4. *Correction du mot reçu.*

Le mot envoyé est : $c(X) = y(X) + \varepsilon(X) = X^3 + X^9 + X^{11} + X^{12}$ *qui correspond au mot* $c = 000100000101100$.

Chapitre 3

Cryptosystème basés sur les codes correcteurs.

3.1 Notions cryptographiques.

Définition 3.1.1 *La cryptographie est l'ensemble des techniques permettant d'écrire un message de façon brouillée afin que seul son destinataire légitime soit capable de comprendre la teneur du message.*

Définition 3.1.2

1. *Le chiffrement est la fabrication du message chiffré à partir du message clair et de la clé de chiffrement.*

2. *Le déchiffrement est l'extraction du message clair à partir du chiffré en utilisant la clé de déchiffrement.*

3. *Le décryptage ou l'attaque est l'extraction du message clair à partir du chiffré sans connaître la clé de déchiffrement.*

4. *La cryptanalyse est l'étude théorique d'un système de chiffrement en vue de mettre au point des algorithmes de décryptage.*

Définition 3.1.3 *Un système de cryptographie ou cryptosystème est constitué de :*

1. *Un ensemble fini A appelé alphabet. Par exemple, $A = \{0, 1\}$, l'alphabet binaire, est un alphabet fréquemment utilisé dans la pratique.*

2. *Un ensemble M composé de chaînes de symboles de l'alphabet A appelé espace de messages. Un élément de M est appelé un message clair ou tout simplement un text clair. Par exemple, M peut être constitué de chaînes binaires, texte anglais, code informatique, etc.*

3. *Un ensemble C appelé l'espace des messages chiffrés. C'est constitué de chaînes de symboles d'un alphabet B, qui peut être différente de l'alphabet A. Un élément de C est aussi appelé un cryptogramme.*

4. *Un ensemble K dit espace des clés. Un élément e de K est dit clé.*

5. *Pour chaque clé e de K, on définit une bijection f_e de M dans C, dite fonction de chiffrement. Si $m \in M$ alors $f_e(m) = c \in C$.*

6. *Pour chaque clé d de K, on définit une bijection f_d de C dans M, dite fonction de déchiffrement. Si $c \in C$ alors $f_d(c) = m \in M$.*

Remarque 3.1.4

1. $f_d = f_e^{-1}$ *et si* $m \in M$ *alors* $f_d(f_e(m)) = m \in M$.

2. *Les clés e et d dans la définition précédente sont désignées comme une paire de clés et parfois notée (e, d). On notera que e et d pourront être les mêmes.*

Définition 3.1.5

1. *Une entité est une personne ou un ordinateur qui envoie, reçoit ou manipule l'information. Qu'on représente souvent par Alice et Bob.*

2. *Un expéditeur est une entité dans une communication entre deux parties qui est l'émetteur légitime d'information.*

3. *Un récepteur est une entité dans une communication entre deux parties qui est le destinataire prévu d'information.*

4. *Un adversaire est une entité dans une communication entre deux parties qui ne sont ni l'expéditeur ni récepteur, et qui tente de vaincre le service de sécurité de l'information fournie entre l'émetteur et le récepteur.*

Remarque 3.1.6 *Un bon cryptosystème doit permettre un chiffrement et un déchiffre-ment rapide, tout en interdisant toute possibilité de décryptage. Pour alléger les charges en mémoire informatique, on préfère aussi que les clés soient de petite taille.*

Cryptographie symétrique.

Définition 3.1.7 *Le système de cryptographie symétrique, également appelée cryptographie à clé secrète, est un système où une seule clé suffit pour le chiffrement et le déchiffrement. Le cryptosystème est dit symétrique car pour chaque paire de clés de chiffrement /déchiffrement* $(e;\ d)$*, il est systématiquement "facile" de déterminer* d *en connaissant* e *et de déterminer* e *à partir* d*.*

Le principal inconvénient de ce type de cryptosystème est que la clé doit rester secrète pour toute personne autre que Alice et Bob ; elle ne doit notamment pas être captée par un espion Charlie. Cependant, un algorithme symétrique permet d'assurer simultanément confidentialité et intégrité, à condition.

Les problèmes de distribution des clés sont résolus par la cryptographie définit ci-dessous.

Exemple 3.1.8

1. *Parmis les cryptosystèmes symétriques, il ya le célèbre chiffrement de Cesar dont la clé est un entier k qui représente le nombre de décalage des lettres de la langue Grec. Si k=3 la lettre A devient D, le B devient E, ..., le Z devient C.*

2. *Le DES (1977-1999) (Data Encryption Standard) est un chiffrement symétrique utilisant des clés de 56 bits. Son emploi n'est plus recommandé aujourd'hui, du fait de sa lenteur à l'exécution et de son espace de clés trop petit* $(2^{56}$ *clés possibles) permettant une attaque systématique en un temps raisonnable.*

3. *AES, dernier l'algorithme symétrique standard choisi par l'institut de standardisation américain NIST en décembre 2001, utilise des clés dont la taille est au moins de 128 bits soit 16 octets, autrement dit il y en a* 2^{128}*, cela fait environ* $3,4\times10^{38}$ *clés possibles ; l'âge*

de l'univers étant de 1010 années, si on suppose qu'il est possible de tester 1000 milliards de clés par seconde (soit $3,2 \times 1019$ *clés par an), il faudra encore plus d'un milliard de fois l'âge de l'univers. Il est actuellement le plus utilisé et le plus sûr.*

3.1.1 Cryptographie asymétrique.

Définition 3.1.9 *Le système de cryptographie asymétrique, également appelée cryptographie à clé publique, est un système qui utilise deux clés, une dite publique p utilisée pour le chiffrement et une autre dite secrète s utilisée pour le déchiffrement de telle sorte que la connaissance de p ne permette pas de retrouver facilement s.*

L'avantage de ce cryptosystème c'est que la clé de chiffrement n'est plus secrete mais à la possetion de tous.

Dans le but d'envoyer un message confidentiel à Alice, Bob utilise la clé publique de Alice pour chiffrer ce message ; Alice étant le seul possesseur de la clé secrète (seule clé autori-sant le déchiffrement), il est le seul à pouvoir déchiffrer le message envoyé par Bob.

3.1.2 Exemples de cryptosystème asymétrique

Cryptosystème RSA.

Inventé en 1977 par Rivest, Shamir et Adleman. Il est très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Il repose sur le problème de factorisation d'un grand entier en produit de deux entier premiers.

- La clé secrete deux entiers premiers p, q.

- La clé publique : $(n,\ e)$ tel que $n = pq$ et e un entier non nul premier avec $\varphi(n)$ tel que φ est la fonction d'Euler.

- le chiffrement de m$\in \mathbb{Z}/n\mathbb{Z}$ est c=m$^e[n]$

- le déchiffrement de c est m=c$^d[n]$ tel que $ed = 1[\varphi(n)]$.

Exemple 3.1.10 $p = 5$ *et* $q = 17$, $n = p \times q = 85$, $\varphi(n) = (p-1) \times (q-1) = 64$.

Clé secrete $(p,\ q) = (5,\ 17)$.

Clé publique $(n,\ e) = (85,\ 5)$.

L'inverse de e $mod(\varphi(n))$ *est* $d = 13$.

Le chiffrement de $m = 10$ *est* $c = 10^5 = 40[85]$ *et le déchiffrement de c est :* $m = c^{13} = 10[85]$.

3.2 Cryptosystème de McEliece.

Le Cryptosystème de McEliece est un schéma de chiffrement asymétrique, inventé en 1978 par Robert McEliece. Ce système est le plus ancien Cryptosystème à clef publique utilisant des codes correcteurs d'erreurs basé sur les codes de Goppa. Pourtant le Cryptosystème de McEliece possède des propriétés intéressantes : la sécurité croit beaucoup plus rapidement avec la taille des clés que pour le système RSA, et le chiffrement est plus rapide. Comme tous les Cryptosystèmes à clef publique, ce système est constitué de trois algorithmes :

1. La génération de clefs.
2. Le chiffrement (utilisant la clef publique).
3. Le déchiffrement (utilisant la clef secrète).

Génération des clés.

On commence par générer un code cyclique $C(n,\ k,\ d)$ et de capacité de correction e, on calcul la matrice génératrice G de taille $k \times n$ sur le corps F_q des racines n-ième de l'unité sur F_2. Ce code doit posséder un algorithme de décodage efficace :

1. Choisir un polynôme générateur $g(X)$ (un diviseur de $X^n - 1$) de degré $t = n - k$ et déduire sa matrice génératrice G.

2. Sélectionner aléatoirement une matrice inversible binaire S de taille $k \times k$ (S est dite matrice brouilleur).

3. Sélectionner aléatoirement une matrice de permutation $P_\sigma$de taille $n \times n$ tel que $\sigma \in S_n$.

4. Calculer la matrice $G^{'} = S.G.P_\sigma$. Celle- ci est une matrice de type $k \times n$.

5. La clef publique est $(G^{'}, t)$ et la clef privée est $(S,\ G,\ P_\sigma)$.

Chiffrement.

Soit $m \in F_q^k$ un message de k bits que l'on veut chiffrer. On ne dispose pour cela que de la clef publique $G^{'}$:

1. On commence par calculer le mot de code C de longueur n associé à m : $c^{'} = m.G^{'}$.

2. Génère une erreur aléatoire ε de longueur n et de poids $v = w(\varepsilon) \leq e$.

3. Le chiffré sera simplement le mot de code bruité : $c = c^{'} + \varepsilon$.

Donc $c = m.G^{'} + \varepsilon$.

Déchiffrement.

Pour déchiffrer en connaissant P_σ, S et G il suffit de calculer :

1. Calculer $c.P_\sigma^{-1} = m.G^{'}.P_\sigma^{-1} + \varepsilon.P_\sigma^{-1} = m.S.G + \varepsilon.P_\sigma^{-1}$.

2. $m.S.G$ est un mot du code C et $\varepsilon.P_\sigma^{-1}$ est une erreur de poids v (car P est une permutation et conserve donc le poids des mots), donc on peut corriger cette erreur en utilisant la méthode de décodage par McElice ou par piégeage d'erreurs, ensuite retrouver le message initial $m^{'} = m.S$.

3. Pour trouver le message m on va multiplier $m^{'}$ par S^{-1}c-à-d : $m = m^{'}.S^{-1}$.

Et avoir fini de déchiffrer.

Exemple 3.2.1 *Soit $C(n,\ k,\ d)$ un code cyclique binaire $e = 1-$correcteur de longueur $n = 7$ et de générateur $g(X) = X^3+X^2+1$ et de matrice génératrice*

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$ *et de matrice génératrice normalisée.*

$G_N = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$ *et comme matrice de controle* $H_N = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$

On choisit clé secrete la matrice G_N*, une matrice inversible* S *d'ordre* $k = 4$*,*

$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ *et une matice de permutation* $P_\sigma = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$,

et la clé publique sera la matrice $G' = S.G_N.P_\sigma = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$ *et un entier non nul* $t \leq e = 1$ *donc* $t = 1$.

Le chiffrement d'un mot $m = 1010$ *donne le mot* $c = mG' + \varepsilon$ *où* $\varepsilon = 100000$ *est un mot erreur choisi de poids* $t = 1$. *On trouve* $c = 1101100$.

Pour le déchiffrement de c. On calcule $c' = cP_\sigma{}^{-1} = 0101101$*, puis on calcul son syndrome* $h(c') = c'.H_N^t = 011 = C_7^t$ *et l'erreur associée est* $\varepsilon_1 = 0000001$ *et le mot corrigé (transmit) est* $m = c' + \varepsilon_1 = 0101100$ *(le mot information associé en enlevant la redondance) on trouve* $m' = m.S = 1100$ *et donc* $m = m'.S^{-1} = m'.S = 1010$.

3.3 Cryptosystème de Niederreiter.

Ce cryptosystème est variante du cryptosystème de McEliece qui a été mise au point par Niederreiter en 1986. Du point de vue de la sécurité ce cryptosystème est équivalent à celui de McEliece et il est un peu plus efficace en temps de calcul. Il fonctionne comme le

chiffrement de McEliece, mais au lieu d'utiliser la matrice génératrice, il utilise la matrice de contrôle.

3.3.1 Génération des clés.

On commence par générer un code cyclique $C(n,\ k,\ d)$ de longueur n de générateur $g(X)$ (diviseur de $X^n - 1$) et de capacité de correction e, et on détermine sa matrice de controle normalisée H_N de taille $(n-k) \times n$.

1. On génére une matrice de permutation aléatoire P de taille $n \times n$.
2. On génére une matrice inversible aléatoire S de taille $(n-k) \times (n-k)$.
3. On calcule la matrice $H_{pub} = S.H_N.P$ de taille $(n-k) \times n$.
4. La clef publique est la matrice H_{pub} et la clef privée est le triplet $(S,\ H_N,\ P)$.

3.3.2 Chiffrement.

1. Pour chiffrer Bob commence par choisir un message $m \in F_q^n$ de poids e.

2. En utilisant la clef publique H_{pub} Bob calcule le mot chiffré c : $c = m.^tH_{pub}$ (où encore $c^t = H_{pub}.m^t$).

3.3.3 Déchiffrement.

Pour déchiffrer le méssage $c^t = H_{pub}.m^t$ de Bob, Alice procède comme avec le système de McEliece :

1. Alice commence par calculer : $S^{-1}.c^t = H_N.P.m^t$.

2. Alice retrouve le mot $m^{'}$correspondante au syndrome $S^{-1}.c^t$, il retrouve donc $m' = P.m^t$.

3. Alice en déduit alors le message clair m à partiène $m = P^{-1}.m'$. et le déchiffrement est fini.

Exemple 3.3.1 *Considérons le code B.C.H* $C(n = 15,\ k = 7,\ d = 5)$ *de longueur* $n = 15$ *Le corps des racines* $15^{èmes}$ *de l'unité sur* F_2 *est* $\Bbbk = F_{16}$ *de racine primive* α *et*

de polynôme primitif $M_\alpha(X) = X^4+X+1$. *On a* $\mathbb{k} = \{0, \alpha^i : 0 \leq i \leq 14\}$, *avec* $\alpha^4 = \alpha+1$.
On suppose que son générateur est :
$g(X) = (X^4 + X + 1)(X^4 + X^3 + X^2 + X + 1) = X^8 + X^7 + X^6 + X^4 + 1$. *En calculant* $g(\alpha^i)$ *pour* $0 \leq i \leq 14$ *on trouve toutes les racines de g(X) :* $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^6, \alpha^8, \alpha^9, \alpha^{12}$
parmis eux il y a quatre racines successive : $\alpha, \alpha^2, \alpha^3, \alpha^4$.
Le code C admet comme polynôme de contrôle $h(X) = \frac{X^n-1}{g(X)} = X^7+X^6+X^4+1$ *et le générateur de son orthogonal est* $h_1(X) = 1+X+X^3+X^6$.
Donc C admet comme matrice de controle la matrice :

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Qu'on peut mettre sous forme systématique :

$$H_N = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

On considère la matrice :

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$ *et la matrice de permutation :*

$$P = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

La clé publique $H' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$.

Soit le texte clair qu'on veut chiffrer et envoyer :

$m = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$ *de poids* $e = 2$.

Le chiffrement de m *est le mot* c *tel que :*

$c^t = H'.m^t = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$. *Pour le déchiffremnt on calcule* s *tel que :*

$s^t = S^{-1}.c^t = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \neq 0$. *Théoriquement* $s^t = H_N.Pm^t$ *c.à.d.* s^t *représente le syndrome du mot* $m'^t = P.m^t$ *et comme* $s^t \neq 0$ *donc* m'^t *n'est pas un mot de* C.

Pour la correction de m'^t. On va utiliser la méthode de décodage par syndrome. On a $s^t = c_1 + c_2$ (s^t est la somme de première et la deuxième colonne de H) donc $s^t = H_N.m'^t$ tel que $m' = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$ avec $w(m'^t) = 2 = w((P.m^t)$. Donc $P.m^t$ et m'^t ont le meme syndrome et tous deux de poids égale à la capacité de correction $e = 2$ alors, d'après la Proposition 2.2.35 on déduit que $P.m^t = m'^t$ d'où $m^t = P^{-1}.\, m'^t = P^t m'^t$

d'où $m = m'.P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$. Qui est bien le texte clair transmis.

Comparaison des cryptosystèmes McEliece, Niederreiter et du RSA.

	McEliece	**Niederreiter**	**RSA**
Taille de la clé publique	kn	k(n-k)	2n
	67072 octets	32750 octets	256 octets
Nombre de bits d'information	k	a=$\log_2(C_n^e)$	n
transmis par chiffrement	512	276	1024
Taux de transmission	$\frac{k}{n}$	$\frac{log_2(C_n^e)}{n-k}$	1
	51,17 %	56,81%	100%
Nombre d'opérations binaires du	$\frac{n}{2} + \frac{n}{k}$	$\frac{(n-k)ke}{an} + \frac{n}{a}$	$125.\frac{3^{m-1}}{2^m}$
chiffrement par bit d'information	513,9	50,1	2402,7
Nombre d'opérations binaires du	$\frac{W_1}{k}$	$\frac{W2}{a}$	$\frac{25}{2}3^{m-1}$
déchiffrement par bit d'information	5140	7863,3	738 112,5

$W_1 = n + mnt + 4m^2t^2 + 2m^2t + mn(2t+1) + \frac{k^2}{2}$ et

$W_2 = 2n + 4m^2t^2 + 2m^2t + mn(2t+1) + \frac{(n-k)^2}{2}$.

Avantages et désavantages du cryptosystème Niederreiter.

Comme on peut le voir, les systèmes de McEliece et de Niederreiter sont tès proches, l'un utilisant les matrice génératrice et l'autre les matrices de controle. Y. Xing a montré dans [11] que la sécurité des deux systèmes est équivalente c.à.d. toute attaque structurelle sur l'un se traduit par une attaque structurelle sur l'autre. D'après le tableau ci dessus

on remarque que la variante de Niederreiter présente cependant les avantages suivants :

1. Réduction de la taille de la clé publique.

2. Réduction du cout de la multiplication matrice-vecteur.

3. Un chiffrement très rapide, il est 10 fois plus rapide que McEliece et plus de 40 fois plus rapide que RSA.

4. Taux de transmission, c'est à dire le rapport $\frac{k}{n} = \frac{\text{dimension}}{\text{longueur}}$, du systéme de Niederreiter supèrieur à celui du système de McEliece.

Le désavantage du cryptosystème de Niederreiter est qu'il est plus lent au déchiffrement que le cryptosytème de McEliece (15 fois moins rapide), mais il est presque 100 fois plus rapide que RSA.

Conclusion générale

Le cryptosystème de Niederreiter est un cryptosystème à clé publique qui utilise les codes correcteurs. Ce cryptosystème est une variante améliorée du cryptosystème de McEliece. Niederreiter dans [1] a utilisé les codes de Goppa qui sont des codes linéaires basés sur les courbes Elliptiques. Dans notre travail on a utilisé des codes cycliques dits codes B.C.H basés sur les polynômes et en utilisant la méthode du syndrome polynomial pour la correction d'erreurs.

Bibliographie

[1] J. A. Buchmann, Introduction to Cryptography, Published by Springer, New York, 2004.

[2] M. Demazure, Cours d'algèbre, Cassini, Paris, 2008.

[3] G. Dubertret, Initialisation à la cryptographie, EMS S.A.S , Paris, Novembere 2012.

[4] W. Diffie and M. Hellman, New directions in cryptography. IEEE. Transaction On Information Theory, vol. tt-22, n°. 6, p 644-645, November 1976.

[5] N. Hadj-said, Ali Bacha, A.Beloraf, A.M'hamed. Hal, Cryptosystème à clé publique de McEliece basé sur les codes cycliques de Hamming, STIC, p 384-388, 2005.

[6] R. J. McEliece, A puplic-key cryptosystem based on algebraic coding theory. DSN Prog. Rep, Jet Prop. Lab, California Inst. Technol. Pasadena, CA, p 114-116, January 1978.

[7] P. Meunir, Algèbre avec application à l'algorithmiques et à la cryptographie. Imprimé en décembre 2009, Normandie, roto impression S.A.S, 61250 lonrai (orne). Imprimé en france.

[8] H. Niederreiter, Knapsack-type Cryptosystems and Algebraic Coding Theory, *Problems of Control and Information Theory* 15, vol. 1, n° 6, 1986, p. 159-166.

[9] R. L. Pivest, A. Shamir, and L. M. Adleman, A method for obtaining digital signatures and public - key cryptosystems, Communications of the ACM, 21(2) : 120-126, February 1978.

[10] O. Pipini, J. Wolfmann. Algèbre discrète et codes correcteurs, Springer-Verlag

Berlin, Heidelberg 1995. Imprimé en Allemagne.

[11] Y. Xing Li, R. H. Deng et X. M. Wang, On the equivalence of McEliece's and Niederreiter's public-key cryptosystems, *IEEE Transactions on Information Theory, vol.* 40, 1994, p. 271-273.

[12] G. Zémor. Cours de cryptographie. Imprimé en Grande-Bretagne par Cambridge université Press. Dépot légal novembre 2000.