UNIVERSITE CHEIKH ANTA DIOP DE DAKAR

FACULTE DES SCIENCES ET TECHNIQUES

LABORATOIRE D'ALGEBRE, DE CRYPTOLOGIE, DE GEOMETRIE ALGEBRIQUE ET APPLICATIONS (LACGAA)

MASTER 2 TRANSMISSION DE DONNEES ET SECURITE DE L'INFORMATION

MEMOIRE DE FIN D'ANNEE

Chiffrement par blocs et Matrice de diffusion MDS

Soutenu par :
Aliou Samba Wélé

Membres de jury :

Pr.Cheikh T. Gueye
Pr. Mamadou Sanghare
Pr. Ismaila Diouf
Dr. Demba Sow
Dr. Amadou Tall

Sous la direction :
Pr.Cheikh T. Gueye

ANNEE ACADEMIQUE : 2015-2016

Dédicace

Qu'ALLAH soit loué.
Je rends grâce à DIEU,
PAIX et SALUT sur le PROHETE MOUHAMED
Je dédie ce mémoire à :

- **Mes chers parents pour l'éducation et le soutient permanent qu'ils m'ont toujours apporté.**
- **Toute ma famille.**
- **Mes professeurs.**
- **Tous mes camarades de classe.**
- **Tous mes amis.**
- **Tous ceux qui m'ont soutenu de près ou de loin.**

REMERCIEMENT

Je rends grâce à ALLAH Le Tout Puissant
Le Clément et Le Miséricordieux
Paix et Salut sur notre bien-aimé le Prophète
Je remercie :

- **Mes chers Parents.**
- **Monsieur le Professeur Cheikh Thiécoumba Guèye pour l'honneur qu'il m'a fait en acceptant mon encadrement.**
- **Tous les membres du Laboratoire d'Algèbre, de Cryptologie,de Géométrie Algébrique et Application (LACGAA).**
- **Tout le personnel administratif et technique du Département de Mathématique et Informatique (DMI) de la Faculté des Sciences et Technique de l'Université Cheikh Anta Diop de Dakar.**
- **Tous mes frères et soeurs du Master 2 Algèbre Cryptologie et de la Géométrie Algébrique promotion 2015-2016.**
- **Tous mes professeurs de l'élémentaire à l'université.**
- **Tous ceux qui, de prés ou de loin, ont participé à la réalisation de ce mémoire.**

Table des matières

Résumé

Ce mémoire porte sur les liens entre les codes correcteurs d'erreurs et les matrices de diffusion linéaires utilisées en cryptographie symétrique. L'objectif de ce mémoire ce n'est pas de construire des matrices MDS optimales destinées à des applications spécifiques, mais de présenter un cadre de travail général pour une telle recherche. Ce mémoire commence par l'étude des codes définis sur un anneau de polynômes du type $\mathbb{F}_2[x]/(f(x))$. Elle se poursuit par l'étude des codes additifs systématiques définis sur $(\mathbb{F}_2^m; +)$ et leur lien avec la diffusion linéaire en cryptographie symétrique. Un point important du mémoire est l'introduction de codes à coefficients dans l'anneau des endomorphismes de $\mathbb{F}_2^m$. Le lien entre les codes qui sont des sous-modules à gauche et les codes additifs est mis en évidence.

Introduction

la cryptographie, une science s'attachant à protéger des messages (assurant la confidentialité, l'authentification, la non-répudiation et l'intégrité) en s'aidant souvent de secrets ou clés, est composée par le chiffrement symétrique et asymétrique Le chiffrement symétrique lui même est composé par le chiffrement par bloc (exemple : DES, AES) et le chiffrement par flux (exemple : RC4). Dans ce présent mémoire nous allons nous intéressé aux chiffrements par blocs notamment aux chiffrements par blocs itératifs en étudiant en détail le principe de diffusion en utilisant, dans leurs processus, des matrices MDS qui seront optimales et résistantes faces aux attaques différentielles et linéaires.
Ce document est structuré en deux parties. Dans la première partie on commence par faire un rappel sur quelques concepts de base en algèbre et ensuite introduire quelques notions de bases de la théorie des codes correcteurs d'erreurs et de la cryptographie. La seconde partie est consacrée à l'étude des chiffrements par blocs itératifs et les matrices de diffusion MDS.

Première partie

Préliminaires

Chapitre 1

Rappels mathématiques

1.1 Anneaux et Corps

Dans cette section, nous nous proposons de décrire les structures d'anneaux et de corps d'une manière générale mais nous ne donnons ici que quelques définitions et quelques résultats élémentaires de la théorie des anneaux et des corps.

1.1.1 Structure d'anneaux

Définition 1. *On appelle anneau un ensemble A muni de deux lois de composition internes : une addition $(x, y) \longrightarrow x + y$ et une multiplication $(x, y) \longrightarrow xy$, satisfaisant aux axiomes suivants :*
(A_1) L'addition est une loi de groupe abélien.
(A_2) La multiplication est associative et admet un élément neutre, noté 1_A ou 1, et appelé élément unité.
(A_3) La multiplication est distributive par rapport à l'addition.

Si de plus la multiplication est commutative, c'est-à-dire si on a $xy = yx$ quels que soient $x, y \in A$, on dit que l'anneau est commutatif.

Si A est un anneau quelconque, on dit que deux éléments x et y de A commutent ou sont permutables si l'on a $xy = yx$.

On appelle pseudo-anneau, un ensemble A muni d'une addition et d'une multiplication satisfaisant aux axiomes des anneaux mais tel que la multiplication n'ait pas d'élément neutre.
L'élément neutre pour l'addition dans un anneau A est noté 0 et est appelé l'élément nul de A. Toutes les règles de calcul valables dans un groupe abélien s'appliquent évidemment au groupe additif de A qui est l'ensemble A considéré comme groupe abélien. Par exemple, l'opposé d'un élément $x \in A$ se note $-x$ et on note $x + (-y) = x - y$.

1° *Pour tout élément x d'un anneau A, on a :*

$$(a) \qquad x \cdot 0 = 0 \cdot x = 0.$$

2° *Pour tout $x \in A$ et tout $y \in A$, on a :*

$$(b) \qquad x(-y) = (-x)y = -(xy).$$

3° *Pour tout élément $x \in A$, on définit par récurrence sur l'entier $n \in \mathbf{N}$ les éléments x^n et $n \cdot x$, en posant :*

$$x^0 = 1, x^n = x^{n-1} \cdot x$$

$$0 \cdot x = 0, n \cdot x = (n-1) \cdot x + x.$$

On a alors les propriétés suivantes que l'on vérifie aisément par récurrence :

$$(d) \qquad x^m.x^n = x^{m+n}$$

$$(e) \qquad (m+n)x = m \cdot x + n \cdot x$$

quels que soient $m, n \in \mathbf{N}$ et $x \in A$.

4° *Pour tout $x \in A$ et pour tout $n \in N$ on a :*

$$(f) \qquad n \cdot x = (n \cdot 1)x = x \cdot (n \cdot 1).$$

Ces règles de calcul nous permettent de développer les produits de sommes d'éléments d'un anneau A en tenant compte de l'ordre des termes. Si A est commutatif, on peut procéder à des simplifications.

Définition 2. *Soit A un pseudo-anneau. Sur la somme directe externe $\mathbb{Z} \oplus A$ des groupes abéliens additifs $\mathbb{Z}$ et A, on définit une multiplication en posant :*

$$(m+\alpha)(n+\beta) = mn + n\alpha + m\beta + \alpha\beta \text{ pour tous } m,\ n \text{ dans } \mathbb{Z} \text{ et tous } \alpha, \beta \text{ dans } A.$$

On vérifie que cette multiplication est associative, et distributive par rapport à l'addition. De plus 1 en est un élément neutre : on a donc construit un anneau unitaire.
Enfin l'inclusion $\mathcal{I}$ de A dans $\mathbb{Z} \oplus A$ est un morphisme de pseudo-anneaux : on a donc construit un anneau unitaire qui contient A. On l'appelle l'extension de Dorroh de A.

Remarque 1. *Certains auteurs appellent anneaux les objets que nous avons appelés pseudo anneaux; ils appellent anneaux unitaires, ou unifères, les triplets que nous appelons anneaux. Notre point de vue est justifié par le fait que tout pseudo-anneau peut être plongé dans un anneau avec élément unité.*

Exemple 1. *a) Munis de l'addition et de la multiplication ordinaires, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ et $\mathbb{C}$ sont des anneaux commutatifs.*
b) Un anneau est dit non nul s'il n'est pas réduit à $\{0\}$.
Si A est un anneau non nul on note $A^* = A\text{-}\{0\}$.

1.1.2 Binôme de Newton

Théorème 1. *Soit A un anneau. Soient a et b deux éléments permutables de A. Pour tout entier n>1, on a la formule suivante dite binôme de Newton :*

$$(a+b)^n = \Sigma_{k=0}^{n} C_n^k a^{n-k} b^k$$

Raisonnons par récurrence sur n. Pour n = 1, le résultat est trivial car la formule se réduit alors à $(a+b)^1 = 1 \cdot a + 1 \cdot b$.
Supposons le théorème 1 soit vraie pour l'entier n > 1, et montrons qu'elle est vraie pour n + 1. On a donc d'après l'hypothèse de récurrence

$$(a+b)^n = \sum_{k=0}^{n} C_n^k a^{n-k} b^k$$

En multipliant cette relation par a + b, il vient

$$(a+b)^{n+1} = (a+b)^n(a+b) = (a+b)^n a + (a+b)^n b = (\sum_{k=0}^{n} C_n^k a^{n-k} b^k)a + (\sum_{k=0}^{n} C_n^k a^{n-k} b^k)b$$

Or, on montre facilement que, puisque a et b sont permutables, il en est de même que a^p et b^q quels que soient p, q $\in \mathbb{N}$.
Donc :

$$(a+b)^{n+1} = \sum_{k=0}^{n} C_n^k a^{n+1-k} b^{k+1}$$

Le coefficient de $a^{n+1-k}b^k$ dans le second membre de cette relation est

$$C_n^k + C_n^{k-1} = C_{n+1}^k \qquad pour \qquad 1 \leq k \leq n$$

Comme d'autre part

$$C_n^0 = C_{n+1}^0 = 1 \qquad et \qquad C_n^n = C_{n+1}^{n+1} = 1$$

On obtient :

$$(a+b)^{n+1} = \sum_{k=0}^{n+1} C_{n+1}^k a^{n+1-k} b^k$$

1.1.3 Propriétés élémentaires des anneaux

Soit A un anneau et soit a ∈ A. Alors nous savons que a·0 = 0·a = 0. On voit ainsi que dans un anneau, le produit de deux facteurs est nul lorsque l'un des facteurs est nul. La réciproque est inexacte comme le montre l'exemple suivant.

Exemple 2. *Prenons $A = \mathbb{R} \times \mathbb{R}$. Pour (a, b) ∈ A et (c, d) ∈ A, posons :*

$$(a, b) + (c, d) = (a + c, b + d), (a, b)(c, d) = (ac, bd).$$

Alors A est un anneau commutatif, l'élément unité étant (1,1) et l'élément nul étant 0 = (0,0). On a (1,0) (0,1) = (0,0) = 0 et pourtant (1,0) ≠ 0 et (0,1)≠0. Cette remarque permet de poser la définition suivante.

Définition 3. *Soit A un anneau non réduit à {0}. On dit qu'un élément a ∈ A est un diviseur de zéro à gauche (resp. à droite) si a ≠ 0 et s'il existe un élément non nul b de A tel que ab = 0 (resp. ba = 0).*

Si A est commutatif, les notions de diviseur de zéro à gauche et à droite coïncident.
Dire que a est un diviseur de zéro à gauche, revient à dire que a ≠ 0 et que a n'est pas régulier à gauche pour la multiplication.

En effet, si a est non nul et est un diviseur de zéro à gauche, il existe b non nul dans A tel que ab = 0 ; alors la relation ab = 0 = a. 0n montre que a n'est pas régulier à gauche pour la multiplication de A.
Réciproquement, si ax = ay avec x ≠ y, alors on a

$$a(x - y) = 0 \ \ avec \ \ x - y \neq 0;$$

a est un diviseur de zéro à gauche.
De même, a est un diviseur de zéro à droite si et seulement si a ≠ 0 et n'est pas régulier à droite pour la multiplication de A.

Définition 4. *On dit qu'un anneau A est intègre ou est un anneau d'intégrité s'il est non nul, commutatif et s'il ne possède pas de diviseurs de zéro.*
Autrement dit l'anneau A est intègre si la relation ab = 0 implique a = 0 ou b = 0.

Définition 5. *Soit A un anneau. On dit qu'un élément x ∈ A est nilpotent s'il existe un entier n ≥ 1 tel que $x^n = 0$.*

Remarque 2. *On notera que si A possède un élément nilpotent a non nul, alors A possède des diviseurs de zéro car $a \cdot a^{n-1} = 0 = a^{n-1} \cdot a$.*

Définition 6. *Soit A un anneau et soit a ∈ A. On dit que a est un élément inversible de A si a possède un symétrique pour la multiplication.*
Nous noterons $A^\times$ l'ensemble des éléments inversibles de A.

Théorème 2. *Soit A un anneau non nul. L'ensemble $A^\times$ des éléments inversibles de A est un groupe pour la multiplication de A.*

1.1.4 Sous-anneaux. Idéaux. Anneaux quotients

Définition 7. *Soient A un anneau et B une partie non vide de A. On dit que B est un sous-anneau de A si les conditions suivantes sont vérifiées :*
a) B est un sous-groupe du groupe additif A.
b) Les relations $x \in B$ et $y \in B$ impliquent $xy \in B$.
c) L'élément unité 1 de A appartient à B.

On vérifie facilement que l'ensemble B, muni des deux lois de composition

$$(x, y) \longrightarrow x + y \qquad et \qquad (x, y) \longrightarrow xy$$

induites par celles de A, est un anneau. Le théorème suivant donne une caractérisation des sous-anneaux.

Théorème 3. *Soient A un anneau et B une partie de A. Les conditions suivantes sont équivalentes :*
a) B est un sous-anneau de A.
b) $1 \in B$ et quelques soient $x, y \in B$, on a $x - y \in B$ et $xy \in B$.

Exemple 3. *a) $\mathbb{R}$ est un sous-anneau de $\mathbb{C}$.*
b) Soit A un anneau, A est un sous-anneau de A mais $\{0\}$ n'est pas un sous anneau de A si $A \neq 0$.

Nous allons introduire maintenant la notion d'idéal dont le rôle en théorie des anneaux est l'analogue de celui des sous-groupes distingués en théorie des groupes.

Définition 8. *Soit A un anneau et I une partie de A. On dit que I est un idéal à gauche (resp. à droite) de A si :*
a) I est un sous-groupe du groupe additif A.
b) Quel que soit $a \in A$ et quel que soit $x \in I$, on a $ax \in I$ (resp. $xa \in I$).
On dit que I est un idéal bilatère ou simplement un idéal de A si I est à la fois un idéal à gauche et un idéal à droite de A.

Notons que dans un anneau commutatif, tous les idéaux sont bilatères.

Exemple 4. *Dans tout anneau A, les sous-groupes triviaux A et $\{0\}$ sont des idéaux. Tout idéal de A autre que A et l'idéal nul 0 s'appelle un idéal propre de A.*

Définition 9. *Soit I un idéal de l'anneau A. On dit que I est un idéal maximal si $I \neq A$ et si, pour tout idéal J diffèrent de I,$I \subset J$ implique $J = A$.*

Définition 10. *Soit A un anneau. On dit qu'un idéal I de A est un idéal principal s'il existe $a \in A$ tel que $I = Aa = Aa$.*

On dit qu'un anneau A est principal s'il est commutatif, intègre, et si tout idéal de A est principal.

Exemple 5. *Les idéaux de $\mathbb{Z}$ sont les ensembles de la forme $n\mathbb{Z}$, $n \in \mathbb{N}$. On sait déjà que tout sous-groupe de $\mathbb{Z}$ est de la forme $n\mathbb{Z}$, $n \in \mathbb{N}$ et il est clair que si $a \in \mathbb{Z}$ et $x \in n\mathbb{Z}$, alors $ax \in n\mathbb{Z}$. Donc $\mathbb{Z}$ est un anneau principal.*

Théorème 4. *Soient A un anneau et I un idéal bilatère de A. Alors la relation définie par $x\mathcal{R}y \iff x-y \in I$ est une relation d'équivalence sur A, compatible avec les deux lois de A. L'ensemble quotient, noté A/I, muni des deux lois est un anneau appelé anneau-quotient de A par I.*

Si, de plus, A est commutatif, l'anneau A/I est commutatif.

1.2 Définitions et Propriétés Fondamentales(Corps)

Définition 11. *On appelle corps tout anneau K non nul dans lequel tout élément non nul est inversible.*
On dit qu'un corps est commutatif si sa multiplication est commutative.
Ainsi pour un corps K on a, $K^\times = K^*$. Si 1 est l'élément unité du groupe multiplicatif K, alors 1 est l'élément unité de K. Ainsi un corps possède toujours au moins les deux éléments 0 et 1.

Exemple 6. *Les anneaux $\mathbb{Q}$, $\mathbb{R}$ et $\mathbb{C}$ sont des corps commutatifs de caractéristique 0.*

Exemple 7. *L'ensemble $\mathbb{Q}[\sqrt{2}]$ muni de l'addition et de la multiplication ordinaires est un corps commutatif.*
Dans un corps commutatif, on écrit souvent $xy^{-1} = y^{-1}x = x/y$. On vérifie facilement que toutes les règles de calculs habituelles dans $\mathbb{R}$ et $\mathbb{C}$ sont valables dans un corps commutatif.

Théorème 5. *1. Soit $a \in \mathbb{Z}$. Dans l'anneau $\mathbb{Z}/n\mathbb{Z}$, l'élément $\bar{a}$ est inversible si et seulement si a et n son premiers entre eux.*

2. L'anneau $\mathbb{Z}/n\mathbb{Z}$ est un corps si et seulement si n est un nombre premier.

1. On a les équivalences :
   a inversible $\iff$ il existe u $\in \mathbb{Z}$ tel que $\bar{a}\bar{u} = \bar{1}$
   $\iff au \equiv 1 mod(n)$
   $\iff$ il existe v $\in \mathbb{Z}$ tel que au - l = vn soit au - vn = 1, ce qui, d'après l'identité de Bezout, signifie que a et n sont premiers entre eux.
2. Supposons que n soit premier et soit p $\in \mathbb{N}$ tel que $0 < p < n$; alors n et p sont premiers entre eux, donc d'après a), $\bar{p}$ est inversible et $\mathbb{Z}/n\mathbb{Z}$ est un corps.
   Inversement, si $\mathbb{Z}/n\mathbb{Z}$ est un corps, tout élément $\bar{m}$ $(0 < m < n)$ de $\mathbb{Z}/n\mathbb{Z}$ est inversible ; donc d'après a), m et n sont premiers entre eux. n n'admet donc pas d'autres diviseurs positifs que 1 et lui-même. Donc n est un nombre premier.

Théorème 6. *1. Tout corps K est intègre.*

2. Dans un corps K, tout élément non nul est régulier pour la multiplication de K.

1. Soient a, b $\in$ K tels que ab = 0 et a $\neq$ 0. Alors a^{-1} existe et a $\in$ K. La relation ab = 0 implique
$$0 = a^{-1}0 = a^{-1}(ab) = (a^{-1}a)b = 1.b = b$$

2. On sait que $K^* = K\backslash\{0\}$ est un groupe multiplicatif ; donc tout élément de K^* est régulier pour la multiplication de K.

Définition 12. *Soit K un corps. On dit qu'une partie K' de K est un sous-corps de K si :*
a) K' est un sous-anneau de K.
b) Les relations $x \neq 0$ et $x \in K'$ impliquent $x^{-1} \in K'$.

Théorème 7. *Soit M un idéal d'un anneau commutatif A. Alors M est maximal si et seulement si l'anneau quotient A/M est un corps.*

Supposons M maximal et soit a un élément de A n'appartenant pas à M. L'ensemble aA + M est un idéal de A contenant M et différent de M ; donc aA + M = A. Alors, il existe un élément z de A et un élément m de M tels que az + m = 1. En prenant les classes modulo M, on obtient $\bar{a}\bar{z} = \bar{a}\bar{z} = \bar{1}$, ce qui montre que la classe de a est inversible dans A/M ; autrement dit l'anneau quotient A/M est un corps.
Réciproquement, supposons que A/M soit un corps. Soit I un idéal contenant M. Prenons un élément quelconque a de I, n'appartenant pas à M. La classe de a n'est pas la classe nulle et est donc inversible dans A/M. Il existe un élément u de A tel que $\bar{1} = \bar{a}\bar{u} = \bar{a}\bar{u}$, ou $1 - au = \bar{0}$; donc 1 - au ∈ M et par suite 1 = (1 - au) + au ∈ I. On en déduit que I = A. Donc l'idéal I est maximal.

Théorème 8. *Soit K un anneau commutatif non nul. Pour que K soit un corps il faut et il suffit que les seuls idéaux de K soient $\{0\}$ et K.*

Théorème 9. *Soient K et K' deux corps commutatifs et $f : K \longrightarrow K'$ un morphisme de corps. Alors f est une application injective.*

Définition 13 (Corps premier)**.** *Un corps est dit premier s'il ne contient aucun sous corps strict.*

Proposition 1. *Il n'existe que deux types de corps premiers.*
1. Un corps premier de caractéristique 0 est isomorphe à $\mathbf{Q}$
2. Un corps premier de caractéristique p est isomorphe à $\mathbb{Z}_p$

Dans la suite de ce document nous manipulerons la structure des corps finis (également appelé corps de Galois). Il existe plusieurs manières de construire ces objets.

Théorème 10 (Wedderburn)**.** *Tout corps fini est commutatif*

Proposition 2. *Il existe un unique corps de Galois d'ordre donné, à un isomorphisme près.*

Proposition 3. *La caractéristique du corps à q éléments est p(premier), si $q = p^n$*

Définition 14 (Polynôme minimal)**.** *Soit $\mathcal{K}$, un corps commutatif et $\mathcal{L}$ une extension algébrique de $\mathcal{K}$. Le polynôme minimal d'un élément a de $\mathcal{L}$ sur $\mathcal{K}$ est le polynôme unitaire à coefficients dans $\mathcal{K}$ de plus bas degré qui admet a pour racine.*

Caractérisation 1. *Le corps de Galois $\mathbb{F}_q$ d'ordre $q = p^n(ppremier)$ est un corps fini de la forme $\mathbb{Z}_p[X]/(f)$ où $f \in \mathbb{Z}_p[X]$ est le polynôme minimal de degrés n sur $\mathbb{F}_p$ d'un élément primitif de $\mathbb{F}_q$.*

Définition 15 (Corps des racines)**.** *Le corps des racines d'un polynôme à coefficients dans $\mathcal{K}$ est la plus petite extension de $\mathcal{K}$ contenant toutes les racines du polynôme.*

Caractérisation 2. *Le corps de Galois $\mathbb{F}_q$ d'ordre q est l'ensemble des racines du polynôme $X^q - X$ dans son corps des racines.*

Exemple 8. *Le corps $\mathbb{Z}_p$ des entiers modulo p(premier) est isomorphe au corps de Galois $\mathbb{F}_p$.*

Proposition 4. *Soit ϕ la fonction d'indicatrice d'Euler.Le groupe multiplicatif $\mathbb{F}_q^\times$ est cyclique,il possède $\phi(p^n - 1)$ éléments primitifs*

1.3 Corps de Galois

1.3.1 Corps de Galois contenant un nombre premier d'éléments

On a déjà vu dans le chapitre précédent qu 'un corps de Galois est un corps contenant un nombre fini d'éléments.

Définition

Considérons l'ensemble des entiers relatifs $\mathbb{Z}$ et un nombre q premier appartenant à $\mathbb{N}$: l'ensemble des entiers relatifs modulo q est un corps appelé corps de Galois : il contient q éléments .

Exemples

- L'ensemble $\mathbb{Z}/q\mathbb{Z}$ est formé des éléments :$\{\bar{0}, \bar{1}, \bar{2}, ..., q - 1\}$.Les éléments du corps sont les classes des nombres 0,...q-1 modulo q.
  Pour tout a appartient à $\mathbb{Z}$ a peut s'écrire $a = pq + r$ r étant le reste de la division de a par q $0 \leq q - 1$.
  a appartient à la classe de r ,notée $\bar{r}$
  $\mathbb{Z}/q\mathbb{Z}$ contient q éléments. On peut le noter $\mathbf{GF}(q)$, corps de Galois à q éléments.
- l'ensemble $\mathbb{Z}/2\mathbb{Z}$ ou $\mathbf{GF}(2)$ contient les classes de 0 et 1 : $\mathbf{GF}(2) = \{\bar{0}, \bar{1}\}$.
  L'ensemble $\mathbb{Z}/2\mathbb{Z}$ ou $\mathbf{GF}(2)$ contient les classes 0 et 1 : $\mathbf{GF}(2) = \{\bar{0}, \bar{1}\}$.
  Les tables d'addition et de multiplication sont :

+	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

  Un corps de Galois défini comme une classe de nombres modulo q ne peut contenir qu'un nombre premier d'éléments.
  Nous allons donc montrer comment on peut étendre la définition d'un corps de Galois à partir du $\mathbf{GF}(q)$ ou définir des extensions de $\mathbf{GF}(q)$.

Ces corps contiennent q^m éléments et nous allons en exposer la construction sans justification théorique.

1.3.2 Corps de Galois défini comme ensemble de polynômes à coefficients dans $\mathbb{Z}/q\mathbb{Z}$

Définition

Considérons les polynômes en x dont les coefficients sont dans $\mathbf{GF}(q)$. Soit $f(x)$ un tel polynôme : il s'écrit :

$$f(x) = f_{n-1}x^{n-1} + f_{n-2}x^{n-2} + ... + f_1x + f_0$$

Les coefficients $f_0, f_1, ..., f_{n-1}$ sont des élément de $\mathbf{GF}(q)$
L'ensemble de ces polynômes est un corps de Galois noté par $\mathbf{GF}(q)[X]$.
Pour construire le $\mathbf{GF}(q^m)$ on va procéder de la même que pour construire les $\mathbb{Z}/q\mathbb{Z}$ mais en raisonnant non plus sur les entiers relatifs mais sur les polynômes de $\mathbf{GF}(q)[X]$. Pour cela on suppose qu'on connait un polynôme $p(x)$ de $\mathbf{GF}(q)[X]$ de degré m irréductible.
Irréductible veut dire qui n'est divisible par aucun autre polynôme de degré inférieur.
L'ensemble des polynômes modulo $p(x)$ à coefficients dans $\mathbf{GF}(q)$ est un ensemble fini noté $\mathbf{GF}(q)[X]/(p(x))$: on peut démontrer que c'est un corps. Pour trouver à quelle classe de polynômes modulo $p(x)$ appartient un polynôme quelconque $f(x)$ on effectue la division euclidienne de $f(x)$ par $p(x)$:

$$f(x) = q(x)p(x) + r(x)\ , 0 \leq degres(r(x)) \leq degres(p(x)) - 1 = m - 1$$

Alors $f(x)$ appartient à la classe de $r(x) = r(\bar{x})$.
L'ensemble $\mathbf{GF}(q)[X]/(p(x))$ est formé de l'ensemble des classes de polynômes de degré inférieur ou égal à m-1 (puisque ce sont les restes par divisions euclidiennes par p(x))
combien contient-il d'éléments ? chaque polynôme contient m élément de $\mathbf{GF}(q)$ qui peuvent prendre q valeurs : donc le nombre d'éléments de $\mathbf{GF}(q)[X]/(p(x))$ est q^m .
A une condition prés sur p(x) que nous verrons dans le paragraphe suivant, cet ensemble est le corps de Galois à q^m éléments :$\mathbf{GF}(q^m)$. De plus la caractéristique du corps est q.

Exemple

Très souvent le corps de départ est $\mathbf{GF}(2)$ ou $\mathbb{Z}/2\mathbb{Z}$

- **Construction**

  Sur $\mathbf{GF}(2)$, $p(x) = x^3 + x + 1$ est irréductible.
  En effet :

  x+1 ne divise pas p(x) puisque 1 n'est pas racine

x ne divise pas p(x) puisque 0 n'est pas racine

aucun polynôme de degré 2 ne divise p(x) sinon p(x) serait le produit d'un polynôme de degré 2 et d'un polynôme de degré 1 ce qui est impossible puisque aucun polynôme de degré 1 ne divise p(x).

Tous les polynômes de degré inférieur à 3 forment une classe de polynôme modulo p(x)

Ils sont

Polynômes	notation utilisant les valeurs des coefficients en x^2 x x^0
$f_0(x) = 0$	0 0 0
$f_1(x) = 1$	0 0 1
$f_2(x) = x$	0 1 0
$f_3(x) = x + 1$	0 1 1
$f_4(x) = x^2$	1 0 0
$f_5(x) = x^2 + 1$	1 0 1
$f_6(x) = x^2 + x$	1 1 0
$f_7(x) = x^2 + x + 1$	1 1 1

Cet ensemble contient 2^3 éléments : c'est le $\mathbf{GF}(2^3)$ ou $\mathbf{GF}(8)$.

On voit que les éléments du $\mathbf{GF}(q^m)$ défini de cette façon ne sont pas très commode à écrire. Pour les utiliser, on peut les représenter sous forme de m-uplets du corps de départ soit m bits si celui-ci est $\mathbf{GF}(2)$.

Nous avons défini l'ensemble qui n'est pas très original...Pour définir le corps,il faut définir les opérations d'addition et de multiplication.

• Addition

L'addition de deux éléments du $\mathbf{GF}(q^m)$ en général est l'addition des polynômes terme à terme. Il faut se souvenir de la propriété de l'addition dans le corps de départ où sont pris les coefficients.

Dans l'exemple que nous avons pris,dans le $\mathbf{GF}(2)$ corps des coefficients : 1+1 = 0 (+ ou - désigne la même chose)

Par exemple : $f_4(x) + f_5(x) = x^2 + x^2 + 1 = 1 = f_1(x)$

• Multiplication

Pour sa définition, on fait enfin appelle à la définition du corps.

Cherchons par exemple : $f_2(x)f_4(x) = x^3$

Pour trouver à quel élément il est égal, il faut trouver à quelle classe appartient x^3.

On effectue la division de x^3 par $p(x) = x^3 + x + 1$: on trouve rapidement : $x^3 = (x^3 + x + 1) + x + 1$

La classe de x^3 est $x + 1 = f_3(x)$: donc $f_2(x)f_4(x) = f_3(x)$

NB 1. *Pour trouver le reste d'un polynôme par la division par* $x^3 + x + 1$*, donc pour trouver à quelle classe de polynôme il appartient, il faut remplacer* $x^3 + x + 1$ *par 0 dans le polynôme de départ : quelle est la classe de* x^3 *? on sait que* $x^3 + x + 1 = 0$ *donc* $x^3 = x + 1$

Corps d'extension

Dans l'exemple précédent, le $\mathbf{GF}(8)$ est le corps d'extension du corps de départ $\mathbf{GF}(2)$. il contient le corps de départ. En effet, toutes les constantes de $\mathbf{GF}(2) = \{0, 1\}$ sont des éléments de $\mathbf{GF}(8)$ puisque ce sont des polynômes de degré 0.
Le polynôme $x^3 + x + 1$ est irréductible sur $\mathbf{GF}(2)$ mais $x^3 + x^2 + 1$ l'est aussi. On peut montrer que ces deux corps sont isomorphes. Donc on considère que c'est le même corps. Cette représentation sous forme de polynômes est assez limitée pour des développements théoriques : nous allons voir une autre méthode de construction de $\mathbf{GF}(q^m)$

1.3.3 Représentation cyclique des corps finis

Définition

Pour définir le $\mathbf{GF}(q^m)$, on choisit un polynôme irréductible p(x) de degré m, à coefficients dans $\mathbf{GF}(q)$. Ce polynôme étant premier n'a pas de racine dans $\mathbf{GF}(q)$. Mais il existe des racines ailleurs que dans $\mathbf{GF}(q)$, on peut en trouver m : on note α une racine qui est l'élément primitif de $\mathbf{GF}(q^m)$
L'ensemble de toutes les puissances de α :$\{\alpha^0, \alpha, ...\alpha^{q^m-2}\}$ est un corps appelé corps de Galois.

Exemple

On va construire le $\mathbf{GF}(8)$ avec cette nouvelle définition.
Soit toujours $p(x) = x^3 + x + 1$ le polynôme irréductible.
On appelle α une de ses racines, et on va exprimer les différentes puissances de α en fonction de $\alpha^0 = 1, \alpha, \alpha^2$
En effet dès α^3 on peut exprimer en fonction des puissances inférieures parce que $\alpha^3 + \alpha + 1 = 0$, donc $\alpha^3 = \alpha + 1$. On continue ...

$$\alpha^4 = \alpha^2 + \alpha$$

$$\alpha^5 = \alpha^3 + \alpha^2 = \alpha + 1 + \alpha^2$$

Etc...
De façon générale, si on effectue la division de α^m par $\alpha^3 + \alpha + 1$, $\alpha^m = q(\alpha)(\alpha^3 + \alpha + 1) + r(\alpha)$,comme $\alpha^3 + \alpha + 1 = 0$, $\alpha^m = r(\alpha)$. Continuons de cette façon :

$$\alpha^6 = (\alpha^3 + \alpha + 1)(\alpha^3 + \alpha + 1) + \alpha^2 + 1 = \alpha^2 + 1$$

$$\alpha^7 = (\alpha^4 + \alpha^2 + \alpha)(\alpha^3 + \alpha + 1) + 1 = 1$$

On voit que $\alpha^7 = 1$, donc $\alpha^7 + 1 = 0$
Ceci est général : si le polynôme p(x) est bien choisi $\alpha^{2^m-1} + 1 = 0$ ou pour un corps de

caractéristique q $\alpha^{q^m-1} - 1 = 0$.
Donc on trouve bien 2^m éléments constituant le corps de Galois $\mathbf{GF}(2^m)$.

NB 2. *Le polynôme est bien choisi veut dire qu'il n'existe pas de puissance $j \leq 2^m - 2$ (ou $q^m - 2$) telle que $\alpha^j + 1 = 0$ (ou $\alpha^j - 1 = 0$). On dit que le polynôme est* **primitif**.

Les correspondances entre la notation des éléments à l'aide de puissance de α et la notation à l'aide de polynômes s'effectue en remplaçant α par x.

Opérations

- multiplication :on utilise la notation en puissance α :
  $\alpha^i \alpha^j = \alpha^{(i+j) \bmod (2^m-1)}$ ou de façon générale $\alpha^i \alpha^j = \alpha^{(i+j) \bmod (q^m-1)}$
- addition : on utilise la notation en polynômes de α. On fait la somme bit par bit modulo 2 ou q

1.4 Modules

1.4.1 Définition et exemple

Définition 16. *Soit A un anneau (non nécessairement commutatif). On appelle A-module à gauche (un module à gauche) tout groupe abélien (M, +) muni d'une multiplication*

$$\begin{array}{ccc} A \times M & \to & M \\ (a, m) & \mapsto & am \end{array}$$

vérifiant
- *a(m + m') = am + am' $\forall$ a $\in$ A, $\forall$ m, m' $\in$ M*
- *(a + b)m = am + bm $\forall$ a, b $\in$ A, $\forall$ m $\in$ M*
- *1_A.m = m $\forall$ m $\in$ M*

On le note ${}_AM$.
De la même manière, on définit un A-module à droite

$$\begin{array}{ccc} A \times M & \to & M \\ (a, m) & \mapsto & ma \end{array}$$

On le note M_A

Exemple 9.
1. *Tout anneau A a une structure de A-module à gauche (resp. à droite) par :*
$$\begin{array}{ccc} A \times A & \to & A \\ (a, b) & \mapsto & ab \end{array}$$
On le note par $A_s (resp.\ A_d)$
2. *Tout $\mathbb{K}$-e-v est un $\mathbb{K}$-module à gauche. Soit V un $\mathbb{K}$-e-v*
$$\begin{array}{ccc} \mathbb{K} \times V & \to & V \\ (s, v) & \mapsto & sv \end{array}$$

3. Tout groupe abélien G est $\mathbb{Z}$-module

$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (n, g) & \mapsto & ng \end{array}$$

Remarque 3. *Si A est un anneau commutatif A-module à gauche coïncide avec A-module à droite*

1.4.2 Sous-modules

Définition 17. *Soit M un A-module et N une partie de M.*
N est dit sous-module de M si :

1. *(N,+) est un sous groupe de (M,+)*
2. *$\forall$ $a \in A$, $\forall$ $n \in N$, $an \in N$.*

Exemple 10. *Soit M un A-module alors M et $\{0\}$ sont des sous-modules de M appelés sous modules triviaux de M.*

1.4.3 Modules-Quotients

Soit A un module et R une relation d'équivalence sur un A-module M compatible avec la structure de M définie par $\forall$ x, $y \in$ M, $xRy \iff x - y \in$ N.
Posons

$$N = \bar{0_R} = \{x \in M, xR0\} = \{x \in M, x \in N\}$$

Montrons que N est un sous-module de M.
N$\neq$ 0, 0 $\in$ N car 0R0
Soit x, $y \in$ N alors : $xR0$ et $yR0 \implies xR0$ et $-yR0 \implies (x-y)R0$
d'où $x - y \in$ N.
$x \in$ N, a $\in$ A on a $xR0 \implies x - 0 \in$ N $\implies ax - 0 \in$ N donc $axR0 \implies ax \in$ N.
N est un sous module de M.
Considérons la relation R' définie par $xRy \iff x - y \in$ N' $\forall$ x, $y \in$ M.
R' est une relation d'équivalence. Soit x, y, x', $y' \in$ M tels que $xR'y$ et $x'R'y'$ alors :
$x - y \in$ N' et $x' - y' \in$ N' $\implies (x + x') - (y + y') \in$ N' $\implies (x + x')R'(y + y')$.
$xR'y \implies x - y \in$ N' $\implies a(x - y) \in$ N' $\forall$ a $\in$ N' (car N' sous-module) $\implies axR'ay$
Donc R' est une relation d'équivalence compatible avec la structure de M.

$$\bar{O'_R} = \{x \in M, xR'0\} = \{x \in M, x \in N'\} = N'$$

Donc N' = $\bar{O_{R'}}$
D'où si R est une relation d'équivalence sur M compatible avec la structure de M et N est un sous-module de M qui la définit (i.e $\forall$ x, $y \in$ M, $xRy \iff x - y \in$ N).
Dans ce cas on note M/N au lieu de M/R l'ensemble des classes d'équivalence modulo R.
Un élément x de M/N se note par $\bar{x}$, $x \in$ M.

$$\bar{x} = \{y \in M, xRy\}$$

$= x + N$.
On définit sur M/N deux opérations : l'addition + et la multiplication $\cdot$ par :

1. $\forall \bar{x}, \bar{y} \in M/N \bar{x} + \bar{y} = x \bar{+} y$
2. $\forall a \in A, a \cdot \bar{y} = \bar{ay}$

M/N muni de ces deux opérations a une structure de A-module appelé module quotient de M par N.

Proposition 5. *Soit N un sous-module de M (sur A-module) alors :*

1. *Tout sous module de M/N est de la forme S/N avec S un sous module de M contenant N.*
2. *Posons $\mathcal{F}$ l'ensemble des sous-modules de M contenant N, m l'ensemble des sous-modules de M/N alors l'application :*

$$\begin{array}{rccc} \varphi : & \mathcal{F} & \to & m \\ & S & \mapsto & S/N \end{array}$$

1.4.4 Homomorphismes de modules

Définition 18. *Soient M et N deux A-modules un homomorphisme (morphisme) de modules de M dans N est une application :*

$$f : M \to N \qquad verifiant$$

1. $\forall\ x,\ y \in M,\ f(x+y) = f(x) + f(y)$
2. $\forall\ x \in A,\ \forall\ x \in M,\ f(ax) = af(x)$

Proposition 6. *Soit $f : M \to N$ un morphisme de modules sur un anneau A.*

1. *$f(M)$ un sous-module de N et noté*

$$Imf = \{f(x), x \in M\}$$

2. *Si N' est un sous-module de N alors $f^{-1}(N)$ qui est l'image réciproque de N' par f est un sous module de M.*
3. *$M/kerf \cong Imf$*

1.4.5 Sous modules engendrés

Définition 19. *Soit X une partie d'un A-module M. Il existe un plus petit sous-module de M contenant X. Ce sous-module est appelé sous-module de M engendré par X on le note [x].*

Définition 20. *Un A-module M est dit de type fini s'il peut être engendré par une partie finie($\subset$ M)*
Un A-module est dit cyclique s'il est engendré par l'un de ses éléments. $M = [\{x\}_{x \in M}]$

1.4.6 Sous-module maximal

Définition 21. *Soit M un A-module. Un A-sous-module N de M est dit maximal si $N \subsetneq M$ et pour tout A sous-module K de M, $N \subseteq K \Longrightarrow N{=}K$*

Proposition 7. *Soit M un A-module de type fini alors M admet un sous-module maximal.*

1.4.7 Somme Directe de A-modules et Produit Direct de A-modules

Définition 22. *Soit $(M_\alpha)_{\alpha\in\wedge}$ une famille de A-modules. On appelle produit direct de M_α, $\alpha \in \wedge$, tout couple $(P,(p_\alpha)_{\alpha\in\wedge})$ d'un A-module P et une famille d'homomorphisme de A-modules $\{p_\alpha : P \to M_\alpha, \alpha \in \wedge\}$ vérifiant la propriété universelle suivante :*
Pour tout A-module N et pour toute famille d'homomorphisme $\{f_\alpha : N \to M_\alpha\}_{\alpha\in\wedge}$, il existe un unique homomorphisme $f : N \to P$ tel que $\forall\ \alpha \in \wedge$, $p_\alpha \circ f \sim f$.

,

Proposition 8. *Soit $(M_\alpha)_{\alpha\in\wedge}$ une famille de A-module. Si $(M_\alpha)_{\alpha\in\wedge}$ possède un produit direct, alors elle est unique à isomorphisme près.*

Proposition 9. *Soit $(M_\alpha)_{\alpha\in\wedge}$ une famille quelconque de A-modules. Alors $(M_\alpha)_{\alpha\in\wedge}$ admet un produit direct*

Définition 23. *Soit $(M_\alpha)_{\alpha\in\wedge}$ une famille de A-modules. On appelle coproduit ou somme directe de M_α, $\alpha \in \wedge$, tout couple $(S,(\mu_\alpha)_{\alpha\in\wedge})$ d'un A-module S et une famille d'homomorphisme de A-modules $\{\mu_\alpha : M_\alpha \to S\}$ vérifiant la propriété universelle suivante :*
Pour tout A-module N et pour toute famille d'homomorphisme $\{f_\alpha : M_\alpha \to N\}_{\alpha\in\wedge}$, il existe un unique homomorphisme $f : S \to N$ tel que $\forall\ \alpha \in \wedge$, $f \circ \mu_\alpha \sim f_\alpha$.

Proposition 10. *Soit M_α une famille de A-modules*

1. *$(M_\alpha)_{\alpha\in\wedge}$ admet une somme directe*
2. *$(S,(\mu_\alpha)_\alpha)$ est unique à isomorphisme près*

Chapitre 2

Cryptographie et codes correcteurs d'erreurs

2.1 Cryptographie

2.1.1 Généralités

Depuis l'antiquité, l'homme a cherché à communiquer des informations de façon confidentielle malgré l'exposition potentielle à des regards indiscrets. L'essor des télécommunications a accru le besoin d'outils assurant la confidentialité, l'authenticité et l'intégrité des informations. Des secrets d'État à la protection de la vie privée, en passant par la sécurité des transactions commerciales. Jusqu'aux années 1970, les systèmes de chiffrement se basaient sur une information secrète, partagée entre les interlocuteurs. Ces systèmes, dits à clé secrète, ont pour avantage un débit élevé mais leur utilisation implique un partage antérieur de cette information secrète. Ce scénario est envisageable à petite échelle pour des besoins ponctuels mais ne l'est pas dans le monde actuel où chacun communique quotidiennement avec des centaines d'entités distinctes potentiellement inconnues (via courrier électronique, navigateur web ; téléphonie mobile ; matériel réseau,...).
En 1976, Dieffie et Hellman publient ce qui deviendra la base de la cryptographie à clé publique. Ils énoncent les propriétés nécessaires à de tels systèmes et donnent un protocole permettant à deux interlocuteurs de se partager une information secrète uniquement à partir de données publiques. En pratique, les systèmes respectant ce protocole, dits à clé publique, ont souvent des débits faibles ; ils sont donc souvent utilisés :afin de démarrer une communication sécurisée par un chiffrement à clé secrète. Ce procédé est connu sous le nom de cryptographie hybride. En 1977 est né l'algorithme RSA de Rivest, Shamir et Adelman, le premier cryptosystème à clé publique.
Depuis ce jour, la recherche sur ce sujet n'a cessé de proposer de nouveaux systèmes et d'affaiblir les existants. Cryptograhes et cryptanalystes s'affrontent afin de concevoir et d'évaluer des systèmes à la fois rapides et dignes de confiance.
Ces systèmes sont constitués des éléments suivants :

- Une fonction de génération de clé qui génère un couple $(\mathbf{K}_{sec}, \mathbf{K}_{pub})$ aléatoirement.
- Une fonction de chiffrement $\mathcal{ENC}$ qui en utilisant la clé publique $\mathbf{K}_{pub}$, associe à un

message clair m un message chiffré c.

$$c = \mathcal{ENC}(\mathbf{K}_{pub}, m)$$

Une fonction de déchiffrement $\mathcal{DEC}$ qui, en utilisant la clé secrète $\mathbf{K}_{sec}$, calcule le message clair associé à un message chiffré c.

$$m = \mathcal{DEC}(\mathbf{K}_{sec}, c)$$

Il existe actuellement trois familles de cryptosystèmes à clé publique se basant sur trois domaines différents : la théorie des nombres, les réseaux euclidiens et les codes correcteurs d'erreurs. Les systèmes les plus répandus aujourd'hui sont basés sur la théorie des nombres et reposent sur deux problèmes supposés difficiles, le problème de la factorisation et celui du logarithme discret. Ce quasi-monopole est inquiétant car il n'existe aucune preuve mathématique de la réelle difficulté de ces problèmes si ce n'est la non-existence de preuve opposée. Autre faille de ces systèmes, Shor a montré que ces deux problèmes pouvaient être résolus en temps polynomial dans le modèle de l'ordinateur quantique. Certes, celui ci est loin d'être opérationnel mais la menace est bien réelle et il faudra, le jour venu, disposé d'alternatives crédibles afin de ne pas se retrouver dépourvu. Voilà pourquoi depuis plusieurs années la recherche examine les systèmes basés sur les réseaux euclidiens et les codes correcteurs d'erreurs.
Les fonctions de chiffrement asymétriques se basent sur des fonctions à sens unique munies d'une trappe. Une fonction à sens unique doit être évaluable efficacement pour tout message clair, et trouver la préimage d'un élément généré par cette fonction doit être une opération difficile. La trappe permet au destinataire légitime de simplifier l'inversion et donc de déchiffrer le message. Elle doit en conséquence rester secrète pour conserver le caractère à sens unique de la fonction. Une opération sera considérée difficile lorsqu'elle est considérée déraisonnable, en termes de temps ou de moyens et tenant compte du bénéfice potentiel, par l'entité adverse de tenter d'effectuer cette opération. Un système cryptographique dispose de b bits de sécurité si un ordinateur doit effectuer au moins 2^b opérations pour résoudre le plus simple des problèmes sur lequel se base le système. Étant donné l'évolution perpétuelle de la technologie, il faut régulièrement réévaluer le nombre de bits de sécurité nécessaire pour considérer une opération difficile. Il est considéré aujourd'hui qu'un minimum de 128 bits de sécurité est nécessaire pour protéger une information d'ordre gouvernementale.

2.1.2 Cryptographie basée sur les codes correcteurs d'erreurs

L'ordinateur quantique

Un ordinateur quantique est un ordinateur qui repose sur les propriétés quantiques de la matière pour résoudre des problèmes hors de portée d'un ordinateur classique. En 1994,

Peter Shor a proposé un algorithme, utilisant cet ordinateur, qui permettrait de factoriser un entier R.S.A en temps polynomial par rapport à la taille de la clé. L'algorithme de Shor ainsi que les progrès techniques sur la mémoire des ordinateurs ont amené à un développement de l'étude de systèmes cryptographiques alternatifs aux systèmes basés sur la théorie des nombres et résistant aux algorithmes utilisant l'ordinateur quantique.

Les systèmes connus

Il existe donc une grande variété de problèmes difficiles en théorie des codes. Même si certains semblent mieux adaptés que d'autres, chacun de ces problèmes pourraient servir de base à un cryptosystème : il suffit de pouvoir y cacher une trappe. Cependant, la sécurité de presque tous les cryptosystèmes liés aux codes connus à ce jour repose en partie sur un même problème appelé :***syndrome decoding*** (SD). Ce problème correspond au problème du décodage d'un code quelconque. La raison de ce choix vient du fait que le problème (SD) est à la fois pratique d'utilisation (il est défini dans un contexte général rendant facile l'ajout d'une trappe) et d'une sécurité remarquable (même les meilleurs algorithmes sont très couteux, et il est rarement possible de les améliorer, même dans des cas très particuliers, avec des paramètres précis). Cependant, rien n'empêche d'imaginer faire reposer la sécurité d'un système sur la difficulté de trouver un mot de poids donné, ou n'importe quel autre problème. Par ailleurs, même si le problème SD est pratique du point de vue de la sécurité, il a quand même quelques inconvénients : la manipulation d'une matrice souvent de taille relativement importante va en général faire que tous les systèmes auront besoin, à un moment ou à un autre, de manipuler des objets de grande taille. Ceci fait que dans les systèmes à clef publique utilisant des codes, la taille de la clef sera toujours assez importante et, de façon générale, de tels systèmes ne seront pas très adaptés à une implantation sur des architectures limitées en mémoire(les cartes à puce par exemple).
Il existe plusieurs cryptosystemes basés sur les codes mais dans ce document on cite les deux les plus célèbres que sont : celui proposé par Mc Eliece et la variante de Niederreiter. Cette dernière a été mise au point par Niederreiter en 1986. Sa sécurité est équivalente au cryptosysteme Mc Eliece. Elle fonctionne comme le chiffrement de Mc Eliece, mais en utilisant la matrice de parité du code et en utilisant l'erreur pour contenir le message.

2.2 Codes correcteurs d'erreurs

Les codes correcteurs d'erreurs ont pour objectif de permettre la transmission d'information malgré l'ajout éventuel d'erreurs lors de la transmission. Afin d'y parvenir, on ajoute une redondance au message à transmettre qui, lorsqu'un nombre suffisamment faible d'éléments de ce message étendu est perdu ou altéré, permettra de reconstituer le message initialement envoyé. Cette reconstitution est appelé le décodage. On s'intéressera principalement aux codes en blocs ; codes découpant le message en blocs de taille fixe, et les traitant indépendamment l'un après l'autre et plus précisément aux codes linéaires.

2.2.1 Codes linéaires

Lors de l'envoi d'un message composé de lettres d'un alphabet $\mathcal{A}$, celui-ci est découpé en bloc de k lettres auxquels sont ajoutés une redondance via une application linéaire

transformant un bloc de k lettres en un bloc de n lettres (où n sera évidement choisi supérieur à k). Ce nouveau bloc est transmis à travers un canal de communication, ce qui altérera potentiellement le bloc. Puisque n est supérieur à k, le message reçu ne fera peut-être pas parti de l'image par l'application linéaire appliquée (on en déduit la présence d'au moins une erreur). Le destinataire devra donc trouver l'élément de l'image qui a vraisemblablement été envoyé à l'origine. Cependant, si le nombre d'erreurs ajoutées est trop important, il se peut qu'un autre élément de l'image apparaisse plus vraisemblablement comme étant le bloc d'origine ; voire que le bloc reçu devienne un autre élément de l'image, ce qui nous empêcherait de deviner la présence d'erreurs. Dans ce cadre, un bon code correcteur d'erreur est un code qui disperse suffisamment les mots de codes (afin de pouvoir corriger plus d'erreurs) tout en ayant un rendement, le ratio $\frac{k}{n}$, le plus haut possible (afin de limiter le sur coût du codage). Les blocs de k lettres avant transmission sont des vecteurs de k éléments de l'alphabet $\mathcal{A}$ et sont appelés des mots d'information. En pratique $\mathcal{A}$ sera un corps fini $\mathcal{F}$, ce qui permet de créer l'espace vectoriel, de dimension k, des mots d'information. L'application linéaire appliquée aux mots d'information associe à chacun de ces mots un élément d'un espace vectoriel de dimension n. Puisque n est supérieur à k, l'image de l'application linéaire est un sous espace de dimension k de l'espace $\mathcal{F}^{\mathrm{n}}$. Cette image est appelée un code linéaire. La dimension du sous-espace, k, est appelée dimension du code et la dimension de l'espace d'arrivée, n, est appelée longueur du code. Les éléments d'un code linéaire sont appelés mots de code. Une matrice formée des vecteurs d'une base d'un code $\mathcal{C}$ est appelée matrice génératrice de $\mathcal{C}$. Un mot d'information peut être codé en le multipliant par une matrice génératrice puis le mot de code obtenu peut être décodé en le multipliant par l'inverse de cette même matrice. La matrice génératrice d'un code dont les k premières colonnes forment la matrice identité $k \times k$ est dite sous forme systématique. Il n'existe qu'une matrice génératrice sous forme systématique pour un code linéaire donné. Un mot d'information codé via une telle matrice est simple à décoder puisqu'il suffit d'extraire les k premières coordonnées du mot de code pour retrouver le mot d'information. Le code $\mathcal{C}^{\perp}$ d'un code $\mathcal{C}$ de dimension k et de longueur n sur $\mathcal{F}$ est le sous-espace vectoriel orthogonal à $\mathcal{C}$ c'est-à-dire le sous-espace vectoriel défini par

$$\mathcal{C}^{\perp} = \{\mathrm{c'} \in \mathcal{F}^{\mathrm{n}} \mid \mathrm{c.c'} = 0, \mathrm{c} \in \mathcal{C}\}$$

où l'opérateur. est le produit scalaire qui à $x = x_0, ..., x_{n-1}$ et $y = y_0, ..., y_{n-1}$ associe

$$x.y = \sum_{i=0}^{n-1} x_i y_i$$

Il s'agit d'un code de longueur n et de dimension n-k.
Les matrices génératrices de $\mathcal{C}^{\perp}$ sont dites matrices de parité de $\mathcal{C}$. Le produit $\mathcal{H}\mathrm{m}^{\mathrm{t}}$ où $\mathcal{H}$ est une matrice de parité de $\mathcal{C}$ et m un mot de $\mathcal{F}^{\mathrm{n}}$ est appelé le syndrome de m. Le syndrome d'un mot de $\mathcal{C}$ est nul et tout mot de $\mathcal{F}^{\mathrm{n}}$ ayant un syndrome nul appartient à

$\mathcal{C}$; c'est-à-dire

$$\mathrm{c} \in \mathcal{C} \Longleftrightarrow \mathcal{H}\mathrm{m}^{\mathrm{t}} = 0$$

2.2.2 Décodage

Le canal le plus utilisé est le canal binaire symétrique. Il s'agit d'un canal qui transmet des éléments binaires et qui, indépendamment les uns des autres, peut modifier la valeur de chaque bit avec probabilité p. Lors de la réception d'un mot de code bruité, il faut trouver le mot de code qui est vraisemblablement celui qui a été envoyé, c'est-à-dire celui qui a le plus de coordonnées en commun avec ce premier. Dans le cas de mots binaires la distance de Hamming apportent une notion de distance entre les mots et permet donc d'exprimer la notion de "mot le plus proche".
Soit F un corps fini et $x = x_0, ..., x_{n-1}$ un mot de F^{n}, le poids de Hamming de x est défini par

$$W(x) = \#\{x_i | x_i \neq 0.\}$$

Il s'agit du nombre de coordonnées non-nulles de x.
Soit $y = y_0, ..., y_{n-1}$ un mot de F^{n},la distance de Hamming entre x et y est définie par

$$d(x, y) = \#\{i | x_i \neq y_i\}$$

Il s'agit du nombre de coordonnées en lesquelles x et y diffèrent. Cette distance peut également s'écrire

$$d(x, y) = W(x - y)$$

Décoder un mot $x \in F^{\mathrm{n}}$ d'un code $\mathcal{C}$ de longueur n consiste à trouver le mot de $\mathcal{C}$ le plus proche de x c'est-à-dire $c \in \mathcal{C}$ tel que il n'existe pas $c' \in \mathcal{C}$,d(c',x)<d(c,x). Il est à noter que selon cette définition ; le décodage n'est pas forcément unique.
La distance minimale d'un code $\mathcal{C}$ est définie par

$$d(\mathcal{C}) = min\{d(x, y) | x \in \mathcal{C}, y \in \mathcal{C}, x \neq y.\}$$

Il s'agit de la plus petite distance séparant deux mots distincts du code $\mathcal{C}$. De par la linéarité du code, elle est également le poids du mot non-nul de $\mathcal{C}$ ayant le plus petit poids et s'écrit donc

$$d(\mathcal{C}) = min\{W(x) | x \in \mathcal{C}\}$$

Soient c un mot de $\mathcal{C}$ et e un mot de poids $\mid \frac{d(\mathcal{C})-1}{2} \mid$, alors c'est l'unique mot de code le plus proche de $c + e$.
On parlera de succès de décodage lorsque le décodage d'un mot de code bruité $c+e$ donne de façon unique le mot de code non-bruité c.
Un code linéaire $\mathcal{C}$ permet de décoder avec succès tout mot $m = c + e$ où $c \in \mathcal{C}$ et e est

un mot de poids inférieur à $\mid \frac{d(\mathcal{C})-1}{2} \mid$.
Cependant l'existence de cette possibilité de décoder ne donne pas d'algorithme de décodage, si ce n'est un parcours exhaustif des mots de codes.
Puisqu'il est difficile de décoder un code aléatoire, des codes particuliers possédant des structures particulières ont été créés afin de générer des familles de codes munies d'algorithme de décodage efficace.
Les codes suivant sont des codes sur $\mathbb{F}_{\not=}$

2.2.3 Quelques exemples de codes

Dans cette section on présentes exemples de codes par blocs les plus célèbres et les algorithmes utilisés pour les décoder.

codes à répétitions

Ce code est la forme la plus simple de code par blocs : chaque bit à transmettre est simplement répété d fois dans le message transmis. Ainsi pour d = 3 sur $\mathbf{F}_2$ le message 1101 devient 111111000111. Ce code peut aussi être défini sur n'importe quel autre corps et sa matrice génératrice est toujours la matrice $1 \times d$ remplie de 1. Cette construction donne une distance minimale de d mais une dimension de 1 pour une longueur d aussi. On construit donc des codes de la forme [d, 1, d].
Le décodage est ensuite très simple puisqu'il suffit de garder le bit majoritaire dans chaque bloc. La capacité de correction ainsi obtenue est donc de $\mid \frac{d-1}{2} \mid$. Le principal problème est qu'avec ce code le taux de transmission est de $\frac{1}{d}$ ce qui est très faible. On peut toutefois noter que ce code est parfait quand on choisit d impair.

Codes de Hamming

Ce code est lui aussi très simple mais a un bien meilleur taux de transmission. Pour le définir il est nécessaire d'introduire la notion de *matrice de parité* : c'est une matrice C de taille $(n-k) \times n$ qui a pour noyau le code $\mathcal{C}$ tout entier. Ainsi si $c \in \mathcal{C}$, on aura toujours $\mathcal{H}c^T = 0$. Remarquons qu'un même code $\mathcal{C}$ peut avoir plusieurs matrices de parité différentes de même qu'il peut avoir plusieurs matrices génératrices différentes. On appellera *syndrome* d'un mot c l'élément $\mathcal{S}$ obtenu en calculant $\mathcal{S} = \mathcal{H}c^T$. Ainsi les mots de $\mathcal{C}$ seront tous les éléments de $\mathbb{F}_q^n$ ayant un syndrome nul.
Le code de Hamming est donc un code binaire défini par sa matrice de parité plutôt que par sa matrice génératrice. C'est la matrice de dimension rx$(2^r - 1)$ qui contient toutes les colonnes non nulles distinctes que l'on peut écrire sur r bits. Ainsi pour r =3 on a :

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

Le code de Hamming est l'ensemble des mots de longueur $2^r - 1$ dans le noyau de $\mathcal{H}$. C'est donc un espace de dimension $2^r - 1 - r$. De plus, la distance minimale de ce code est 3 car il n'existe aucun mot de code de poids 1 ou 2 puisque cela signifierait qu'il y a une colonne nulle ou deux colonnes égales dans $\mathcal{H}$. On a donc un code $[2^r - 1, 2^r - 1 - r, 3]$ dans lequel on doit donc pouvoir décoder une erreur. Effectivement, on peut facilement décoder une erreur seule en utilisant la matrice de parité : si on reçoit un mot bruite $c' = c + e$ où c est un mot du code et e une erreur de poids 1 on calcule $\mathcal{S} = \mathcal{H}\mathrm{c'}^{\mathrm{T}} = \mathcal{H}\mathrm{c}^{\mathrm{T}} + \mathcal{H}\mathrm{e}^{\mathrm{T}}$ puisque c est dans le code et donc dans le noyau de $\mathcal{H}$. L'erreur e étant de poids 1, $\mathcal{S}$ est donc égale à la colonne de $\mathcal{H}$ où se situe l'erreur et comme toutes les colonnes sont distinctes on peut retrouver e et donc c.
Cette technique de décodage passant par le calcul d'un syndrome est la méthode utilisée pour décoder quasiment tous les codes par blocs et est un moyen facile d'obtenir une information ne dépendant que de e et pas du message transmis contenu dans c.
Pour r = 1 le code de Hamming n'est pas intéressant puisqu'il ne contient que le mot nul et pour r = 2 c'est le code a répétition de longueur 3. En revanche, hpour n'importe quelle autre valeur de r c'est un code parfait.

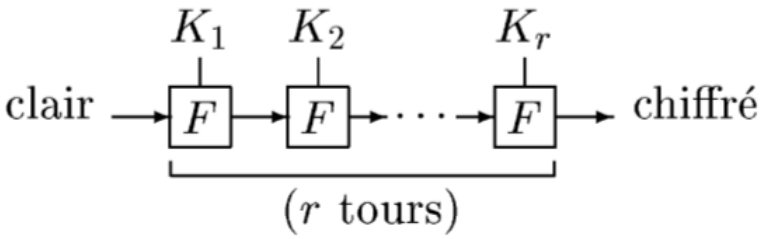
Deuxième partie

Chiffrement par blocs et Matrice de diffusion MDS

Chapitre 3

Chiffrement par blocs

Dans un système de chiffrement par blocs, chaque texte clair est découpé en blocs de même longueur et chiffré bloc par bloc.
La longueur l des clés doit être suffisante pour que l'attaque exhaustive consistant à déchiffrer le chiffré avec toutes les clés possibles jusqu'à l'obtention du clair soit irréaliste (l $\geq$128).
Le principe général d'un chiffrement itératif par blocs est le suivant : pour chaque bloc, on itère r fois une fonction interne F ; à chacun des r tours, la fonction F est paramétrée par une clef K_i (1$\leq$ i $\leq$ r), et la fonction du tour i peut être notée F_{Ki} . Comme on veut que le chiffrement soit inversible (pour pouvoir déchiffrer), il faut que les fonctions F_{Ki} soient bijectives.



3.1 Principes

3.1.1 Principes fondamentaux des chiffrements par blocs

Ils ont été introduits par Shannon.

Définition 24. *La confusion vise à cacher n'importe quelle structure algébrique dans le système.*

Définition 25. *La diffusion doit permettre à chaque bit de texte clair d'avoir une influence sur une grande partie du texte chiffré. Ce qui signifie que la modification d'un bit du bloc d'entrée doit entraîner la modification de nombreux bits du bloc de sortie correspondant.*

Remarque 4. *La confusion est assurée par une substitution non-linéaire.*
La diffusion est assurée par une permutation linéaire.

3.1.2 Principe de la Fonction F

On utilise une combinaison de substitutions et de permutations.
Le texte clair et le texte chiffré sont des suites de bits de longueur l_m : x = $(x_1, ..., x_{lm})$.
La substitution (appelée aussi "S-boîte") est notée

$$\pi_s : \{0,1\}^l \longrightarrow \{0,1\}^l$$

La permutation

$$\pi_p : \{1, ..., l_m\}^l \longrightarrow \{1, ..., l_m\}^l$$

La fonction F_{Ki} se compose donc de plusieurs phases.
- on ajoute la clé au message : $x \bigoplus K_i$
- on découpe $x \bigoplus K_i$ en m sous-chaînes de longueur l, auxquelles ont fait subir la substitution π_s
- on recolle les sous-chaînes et on applique la permutation π_p.

La seule transformation non-linéaire est la substitution π_s. Il est nécessaire que l soit petit, pour assurer que l'implémentation de π_s soit réalisable, avec une petite mémoire.

3.1.3 Chiffrement de Feistel

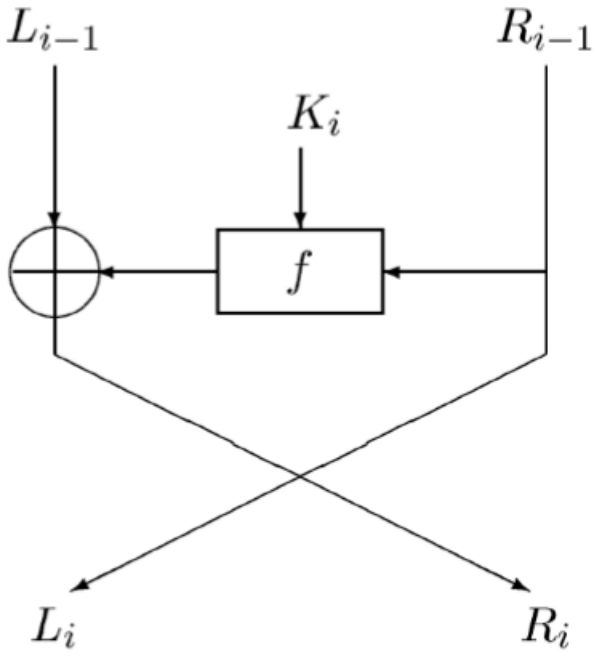
Le système de chiffrement par blocs le plus utilisé jusqu'à l'an 2000 est le DES. Il fait partie de la classe plus générale des chiffrements de Feistel.

Définition 26. *Un chiffrement de Feistel est un chiffrement itératif par blocs opérant sur des blocs de 2n bits. La fonction itérée F est définie par*

$$\begin{array}{rccc} F & : & \mathbf{F_2^n} \times \mathbf{F_2^n} & \rightarrow & \mathbf{F_2^n} \times \mathbf{F_2^n} \\ & & (L_{i-1}, R_{i-1}) & \mapsto & (L_i, R_i) \end{array}$$

avec $L_i = R_{i-1}$ *et* $R_i = L_{i-1} + f(R_{i-1}, K_i)$.

Quelle que soit la fonction f utilisée, un chiffrement de Feistel est inversible.
Pour déchiffrer, il suffit d'utiliser le même processus à r tours en inversant l'ordre des clefs K_i (la fonction F est involutive par construction).



$L_i = R_{i-1}$ et $R_i = L_{i-1} + f(R_{i-1}, K_i)$

3.2 DES

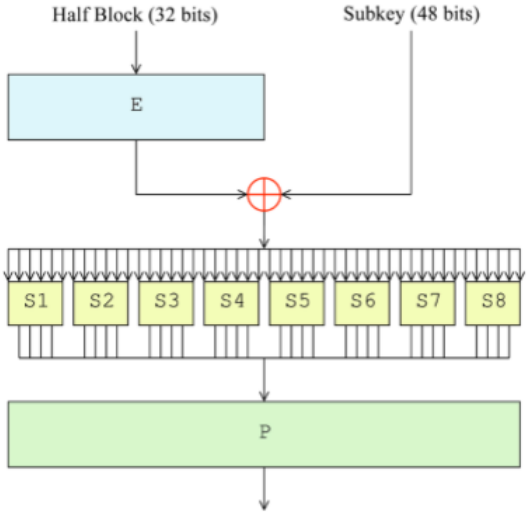
Pour le DES, la taille des blocs est de 64 bits (donc 8 octets), et la taille des clés est de 56 bits (7 octets). Il comporte 16 itérations. C'est un chiffrement de Feistel avec n=32.

3.2.1 La fonction f

C'est une fonction

$$\begin{array}{rcccc} f & : & \{0,1\}^{32} \times \{0,1\}^{32} & \rightarrow & \{0,1\}^{32} \times \{0,1\}^{32} \\ & & R_{i-1}K_i & \mapsto & f(R_{i-1}, K_i) \end{array}$$

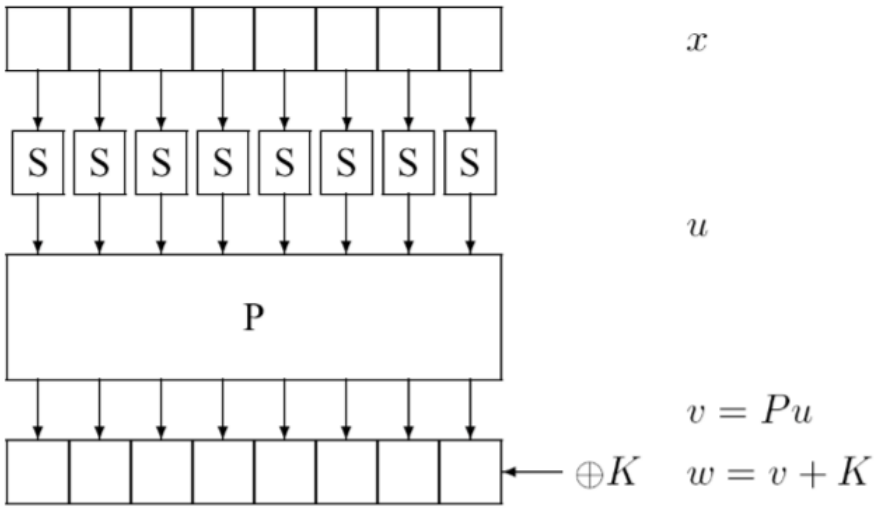
qui se compose de :
- une augmentation E de R_{i-1} pour en faire un bloc de 48 octets, c'est-à-dire que E($Ri-1$) est composé de tous les bits de R_{i-1}, 16 d'entre eux apparaissant deux fois ;
- on calcule $E(R_{i-1}) \bigoplus K_i$, et on le découpe en 8 sous-chaînes de 6 bits.
- chacune des sous-chaînes de 6 bits est transformée par une fonction non linéaire fixée en une sous-chaîne de 4 bits.
- les sous-chaînes de 4 bits sont réordonnées suivant une permutation fixée.



Le DES a été choisi comme norme au États-Unis en 1975 et est devenu le système cryptographique le plus utilisé dans le monde.
Le DES a été critiqué à cause de la taille trop faible de ses clés.
Effectivement, en 1998, un défi a été résolu en quelques jours par une machine spécialement construite pour retrouver la clé par une attaque exhaustive.

3.2.2 Les schémas de substitution/permutation (S/P)

On appelle ainsi les schémas itératifs dont les tours sont construits de la façon suivante :



Si x est l'entrée du tour, x est découpé en sous-blocs d'égale longueur x_1,x_2,... auxquels on applique une substitution S (éventuellement il y a autant de substitutions que de sous-blocs). Si u_i =Sx_i, on concatène les u_i pour former u, à qui on applique une permutation P (éventuellement une transformation linéaire plus complexe). On note v = Pu. La dernière étape du tour est l'ajout de la clé de tour K à v. Notons w=v+K.

Le premier tour est précédé de l'ajout d'une clé supplémentaire K_0 pour éviter que l'attaquant déchiffre le premier tour jusqu'à l'ajout de K_1.
L'algorithme RIJNDAEL, choisi pour être le nouveau standard du chiffrement par blocs AES, est de ce type.

3.3 AES

L'AES (Advanced Encryption Standard) est, comme son nom l'indique, un standard de chiffrement symétrique évolué, destiné à remplacer le DES (Data Encryption Standard) qui est devenu trop faible au regard des attaques actuelles.

3.3.1 Présentation générale

- Le développement de l'AES a été instigué par le NIST (National Institue of Standards and Technology) le 2 janvier 1997.
- Il a été adopté comme standard en Novembre 2001, sur la base de l'algorithme original Rijndael conçu par Joan Daemen et Vincent Rijmen.
- Contrairement à DES, AES est issu d'un processus de consultation et d'analyse ouvert à des experts mondiaux.
- Il s'agit d'un cryptosystème par blocs itératifsss (comme le DES) mais il n'utilise pas de schéma de Feistel.
- Techniques donc semblables au DES (substitutions, transpositions, XOR,...) complétées par des opérations algébriques simples et très performantes.
- La structure générale ne comprend qu'une série de substitutions et de transpositions.
- Avec l'algorithme original "Rijndael", il est possible d'utiliser plusieurs longueurs de blocs clairs et de Clefs :

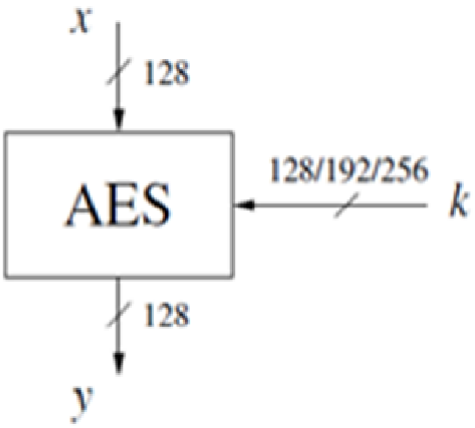
$$128,\ 192,\ \textit{ou}\ 256\ \textit{bits}.$$

- Le nombre de rounds (itérations) est fonction de ces longueurs : 10,12, 14 .
- Il existe, de ce fait, 9 versions possibles pour "Rijndael" .
- Pour conclure sur cet aspect, on voit que cet algorithme de chiffrement répond aux mêmes exigences que le DES mais il est également beaucoup plus sûr et flexible que son prédécesseur.

3.3.2 Structure de l'AES : Chiffrement et Déchiffrement

L'algorithme standard AES adopté par NIST opère sur des blocs clairs (x) de 128 bits qu'il transforme en blocs chiffrés (y) de 128 bits par une séquence de N_r opérations ou "rounds", à partir d'une clef K de 128, 192 ou 256 bits.

Suivant la taille de la clef, le nombre de rounds diffère : respectivement 10, 12 et 14 rounds.



Calcul sur les octets(Le Corps fini $GF(2^8)$)

Un octet, composé des 8 bits

$$(b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0)$$

peut être considéré comme un polynôme b(x) de degré 7 avec des coefficients dans {0,1} :

$$b(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

• **Addition $\oplus$ des polynômes** :L'addition de deux de ces polynômes revient à additionner les coefficients de chacun, modulo 2 (1+1=0). Cette addition correspond au XOR ($\oplus$) au niveau des bits.

• **Multiplication $\otimes$ des polynômes** : Pour la multiplication de deux polynômes a(x) et b(x), c'est la multiplication usuelle (double distributivité) suivie d'une réduction modulo un polynôme binaire irréductible de degré 8.

$$c(x) = a(x) \otimes b(x) \ \ mod \ \ m(x)$$

Dans l'AES, ce polynôme est :

$$m(x) = x^8 + x^4 + x^3 + x + 1$$

• **Structure du Corps fini $GF(2^8)$** : En utilisant $m(x)$ irréductible, les $2^8 = 256$ octets binaires possibles, vus comme des polynômes, ont une structure de corps fini (Corps de Galois) $GF(2^8)$ notée aussi $\mathbb{F}_2^8[x]/(m(x))$. En particulier, pour tout polynôme binaire de degré ≤ 8, on peut calculer $b(x)$ tel que $a(x)b(x) \equiv 1 \ \ mod \ \ m(x)$ autrement l'inverse de $a(x)$ c'est $b(x)$.

Calcul sur les mots de 32 bits (Anneau $\mathbb{F}_2^8[x]$)

- Un mot de 32 bits (4 octets) est considéré comme un polynôme B(x) de degré ≤ 3 (avec 4 coefficients).
- Les coefficients des polynômes utilisés sont des octets dans $GF(2^8)$.
- Addition + des polynômes : elle se fait comme précédemment.
- Multiplication des polynômes : elle se fait comme précédemment mais pour revenir au degré 3, il faut réduire modulo un polynôme $M(x)$ de degré 4.
- Dans l'AES, ce polynôme est $M(x) = 1 + x^4$ (il n'est pas irréductible).
- Les mots de 32 bits possibles (4 octets), vus comme des polynômes, ont une structure d'Anneau de polynômes $\mathbb{F}_2^8[x]$.

Les différentes étapes du Chiffrement et du Déchiffrement

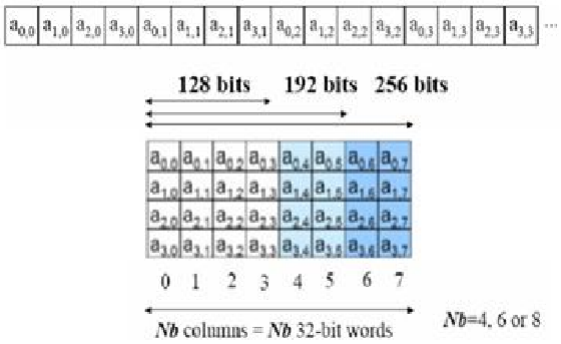
L'ordonnancement des différentes étapes est décrit comme suit :

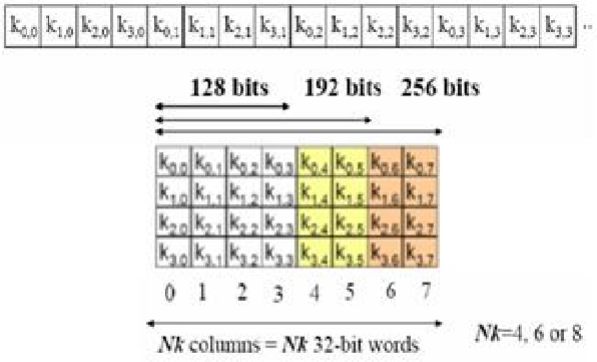
1. **Tableaux d'état du clair et des clefs**

   - Le texte clair et la clef sont conservés sous forme de tableaux d'état. On appelle état un résultat intermédiaire de l'algorithme, représenté sous forme d'un tableau d'octets rectangulaire de 4 lignes, et d'un nombre de colonnes égal à $N_b = L_{bloc}/32$ (L_{bloc} est la longueur ou la taille du Bloc)
   - On représente la clé de la même façon, le nombre de colonnes étant : $N_k = L_{clef}/32$
   - Le nombre de colonnes dépend donc des tailles des blocs clairs et clefs. Une colonne du tableau correspond à un mot de 32 bits et chaque petit bloc représente 8 bits = 1 octet.
     Au début de l'algorithme, on remplit l'état avec le bloc à chiffrer dans l'ordre

     $$a_{0,0}, a_{1,0}, a_{2,0}, a_{3,0}, a_{0,1}, ...$$

     et de la même façon pour la clef.



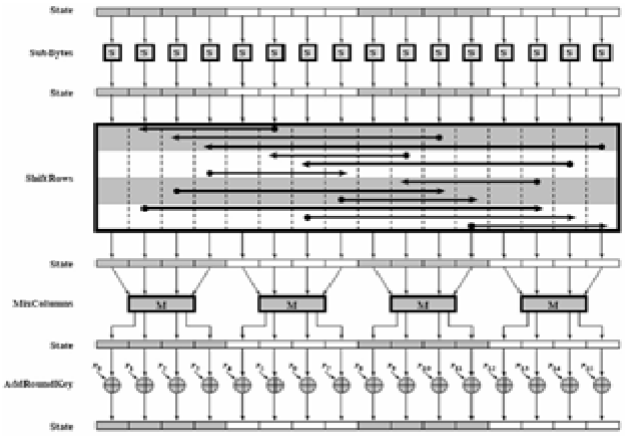


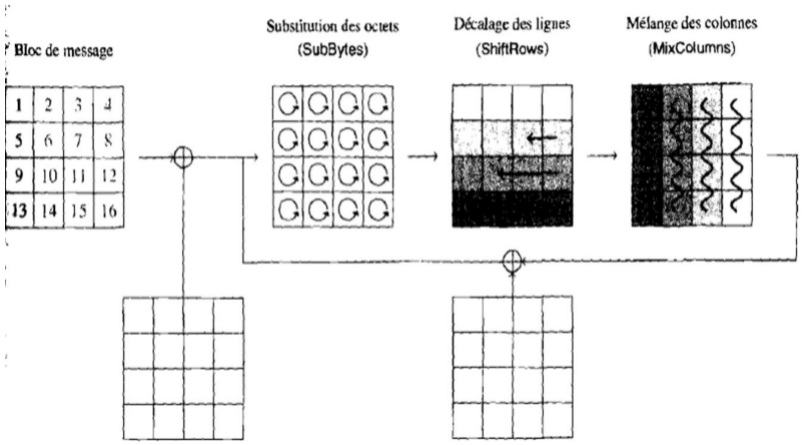
(Tableau d'état de la Clef)

2. Chiffrement/Déchiffrement

   A chaque round, quatre transformations sont appliquées (sauf à la dernière où 3 sont appliquées) :

   – Substitution d'octets dans le tableau d'état (ByteSub)
   – Décalage d'octets dans le tableau d'état (ShiftRow)
   – Déplacement de colonnes dans le tableau d'état (MixColumn)
   – Addition d'une clef de round qui varie à chaque round (AddRoundKey)





Le schéma suivant décrit succinctement le déroulement du chiffrement :

(a) On calcule la clef étendue

(b) On effectue un AddRoundKey initial (”tour 0”)

(c) On effectue (N_{r-1}) tours :

i. ByteSub (Etat) ;

ii. ShiftRow (Etat) ;

iii. MixColumn (Etat) ;

iv. AddRoundKey (Etat, K_i) ;

(d) On effectue un tour final :

i. ByteSub (Etat) ;

ii. ShiftRow (Etat) ;

iii. AddRoundKey (Etat, K_i) ;

– BYTE_SUB (Byte Substitution) est une fonction non-linéaire opérant indépendamment sur chaque bloc à partir d'une table dite de substitution.

– SHIFT_ROW est une fonction opérant des décalages (typiquement elle prend l'entrée en 4 morceaux de 4 octets et opère des décalages vers la gauche de 0, 1, 2 et 3 octets pour les morceaux 1, 2, 3 et 4 respectivement).

– MIX_COL est une fonction qui transforme chaque octet d'entrée en une combinaison linéaire d'octets d'entrée et qui peut être exprimée mathématiquement par un produit matriciel sur le corps de Galois (28).

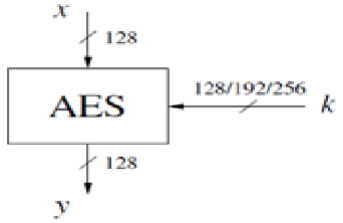
– K_i est la ième sous-clé calculée par un algorithme à partir de la clef principale K.

Le déchiffrement consiste à appliquer les opérations inverses, dans l'ordre inverse et avec des sous-clés également dans l'ordre inverse.

(a) **Les Iterations** Le nombre de tours dépend de la longueur du bloc et de la clef selon le tableau suivant :

	Key length		
Block length	128 bits Nk=4	192 bits Nk=6	256 bits Nk=8
128 bits Nb=4	10	12	14
192 bits Nb=6	12	12	14
256 bits Nb=8	14	14	14

Remarque 5. *L'AES est un peu différent de Rinjdael qui autorise plus de variabilité de la taille du bloc clair et de la taille de la clef.*



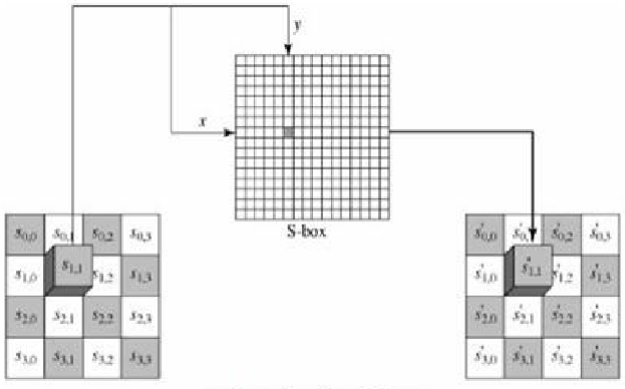
Paramètres de l'AES(Blocs clair et chiffré de 128 bits ; Clefs de 128/192/256 bits) • Chaque round utilise une sous-clé K_i différente (générée à partir de la clé principale) et est composé des quatre transformations qui agissent sur l'état actuel :

ByteSub(Etat) ;
ShiftRow(Etat) ;
MixColumn(Etat) ;
AddRoundKey (Etat, Ki) ;

(b) La transformation ByteSub La transformation ByteSub est une fonction non linéaire qui opère une substitution des bytes de l'état () en cours de traitement :

$$ByteSub(A + B) \neq ByteSub(A) + ByteSub(B)$$

Chacun des bytes est substitué, indépendamment, par son image dans une table de substitution (S-BOX) inversible. C'est une fonction bijective d'où sa réciprocité pour le déchiffrement.



Fonction ByteSub

(a) S-box

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

(b) Inverse S-box

		y														
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

• Génération et propriétés des S-Box :

On considére chacun des bytes x de l'état comme un octet (8 bits) et on lui applique la même substitution comme suit :

1. On considére x comme un polynôme dans $GF(2^8)$,

et on calcule son inverse multiplicatif dans $GF(2^8)$

2. On applique ensuite au résultat la substitution mono alphabétique affine

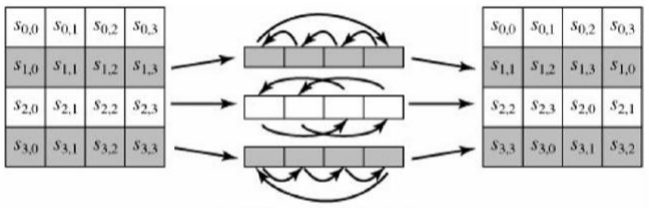
$$y = Ax^{-1} + b$$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Cette substitution forme la S-Box, qui est appliquée à chacun des bytes de l'état.

(c) La transformation ShiftRow C'est une transformation qui augmente la diffusion dans le round traité.



(a) Shift row transformation

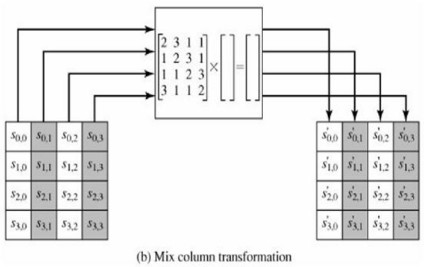
– La transformation ShiftRow effectue un décalage cyclique des lignes de l'état Selon la taille des blocs clairs (la valeur de N_b) , les décalages ne seront pas toujours identiques :
  – la ligne 0 n'est jamais décalée,
  – la ligne 1 l'est de C_1 octets,
  – la ligne 2 de C_2 octets
  – et la ligne 3 de C_3 octets.
– Les valeurs de C_1, C_2 et C_3 dépendent de la longueur du bloc, selon la table suivante :

N_b	C_1	C_2	C_3
4	1	2	3
6	1	2	3
8	1	3	4

2A	64	D5	CA	⟶	pas de décalage	⟶	2A	4C	5A	CA
E4	4C	AA	ED	⟶	décalage de 1	⟶	4C	AA	ED	E4
1F	35	5A	37	⟶	décalage de 2	⟶	5A	37	1F	35
94	4E	F0	84	⟶	décalage de 3	⟶	84	94	4E	F0

(d) La transformation MixColumn La transformation MixColumn est une permutation qui consiste à prendre chaque colonne de l'état obtenu après le décalage et à la multiplier par la matrice suivante, définie par Rinjdael.
– Les colonnes forment des mots de 32 bits.
– Chaque colonne considérée comme un polynôme sur $\mathbb{F}_2^8[x]$ est multipliée par le polynôme fixe $c(x)(mod\ x^4+1)$
– $c(x) =' 03'x^3 \oplus' 01'x^2 \oplus' 01'x \oplus' 02'$
Dans la pratique, on fera la multiplication avec la matrice ci dessous ; Elle contiendra toujours ces mêmes valeurs et le calcul se fera dans $GF(2^8)$.

$$\begin{pmatrix} b_{0,x} \\ b_{1,x} \\ b_{2,x} \\ b_{3,x} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} a_{0,x} \\ a_{1,x} \\ a_{2,x} \\ a_{3,x} \end{pmatrix}$$


(b) Mix column transformation

Etape du MixColumn

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

$$\begin{aligned} s'_{0,j} &= (2 \bullet s_{0,j}) \oplus (3 \bullet s_{1,j}) \oplus s_{2,j} \oplus s_{3,j} \\ s'_{1,j} &= s_{0,j} \oplus (2 \bullet s_{1,j}) \oplus (3 \bullet s_{2,j}) \oplus s_{3,j} \\ s'_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus (2 \bullet s_{2,j}) \oplus (3 \bullet s_{3,j}) \\ s'_{3,j} &= (3 \bullet s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \bullet s_{3,j}) \end{aligned}$$

Remarque : A ce niveau, on a également de la diffusion puisqu'une différence sur un byte d'entrée se propage sur 4 bytes de sortie.

(e) La transformation AddRoundKey

$$a_{i,j} + k_{i,j} = b_{i,j}$$

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} + \begin{bmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \end{bmatrix} = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}$$

Chapitre 4

Cryptanalyse

Les deux principales méthodes connues de cryptanalyse des chiffrements par blocs symétriques sont la cryptanalyse différentielle et la cryptanalyse linéaire.

Elles exploitent toutes deux des comportements statistiques non uniformes dans le processus de chiffrement. La cryptanalyse différentielle date de 1990 et est due à Biham et Shamir. La cryptanalyse linéaire date de 1992 et est due à Matsui.

Appelons x le texte clair, y son chiffré.
En posant x_0=x et $x_i = F(x_{i-1}, k_i)$ pour tout i =1,...,r, on a y=x_r.

Le but de l'attaque est de déterminer la valeur de k_r,
puis, en utilisant le chiffrement de x en x_{r-1}=$F^{-1}(y, k_r)$ qui est un chiffrement à r-1 tours, de calculer k_{r-1}, et ainsi de suite. Nous nous concentrons donc sur le calcul de k_r.

4.1 Cryptanalyse différentielle

Il s'agit d'une attaque à clairs choisis.

La cryptanalyse différentielle s'intéresse à l'évolution les différences $x_i + x'_i$ pour deux clairs x, x'.

On montre que, si $x + x' = \alpha$, alors $x_{r-1} + x'_{r-1} = \beta$ avec une grande probabilité.

On utilise cela pour déterminer la clé inconnue k_r à partir de plusieurs messages x et de leurs cryptogrammes x_r obtenus par E_k.

Notons

$$P_{\alpha,\beta} := P(x_{r-1} + x'_{r-1} = \beta | x + x' = \alpha)$$

et supposons avoir déterminé une valeur de (α, β) telle que cette probabilité soit particulièrement élevée.

On itère, pour un grand nombre de couples clairs-chiffrés(x, y) et (x', y'), chiffrés avec l'algorithme de chiffrement utilisant la clef inconnue K, tels que $x + x' = \alpha$, les étapes suivantes :

Pour toutes les valeurs possibles $\mathcal{K}$ de k_r :
• Calculer z :=$F^{-1}(y, \mathcal{K})$ et z' :=$F^{-1}(y', \mathcal{K})$.
• Si z+z'=β, incrémenter un compteur associé à $\mathcal{K}$.

Le compteur le plus souvent incrémenté est celui de k_r. Notons que l'attaque est d'autant plus efficace que la probabilité P s'éloigne de la probabilité uniforme, soit $\frac{1}{2^n}$.

Exemple 11. *On considère*

$$\begin{array}{ccccccccccccc} & & k^1 & & & & k^2 & & & & k^3 & & \\ & & \downarrow & & & & \downarrow & & & & \downarrow & & \\ x \rightarrow & \oplus & \rightarrow & x^0 & \xrightarrow{S} & S(x^0) \rightarrow & \oplus & \rightarrow & x^1 & \xrightarrow{S} & S(x^1) \rightarrow & \oplus & \rightarrow & y \end{array}$$

où les x représentent trois bits. La boîte S est une substitution sur $\mathbf{F}_2^3$.

On suppose que

$$P((S(x) + S(x') = \beta | x + x' = \alpha) > \frac{1}{8}$$

Si $x + x' = \alpha$, *on a* $x^0 + x^{0'} = (x + k^1) + (x' + k^1) = x + x' = \alpha$,
et la probabilité que $S(x^0) + S(x^{0'}) = \beta$ *est importante.*

Si c'est réalisé, $x^1 + x^{1'} = (S(x^0) + k^2) + (S(x^{0'}) + k^2) = S(x^0) + S(x^{0'}) = \beta$

D'où

$$P_{\alpha,\beta} = P(x^1 + x^{1'} = \beta | x + x' = \alpha) > \frac{1}{8}$$

Pour tous couples clairs-chiffrés (x, y) *et* (x', y') *donné tel que* $x + x' = \alpha$, *et pour chaque valeur possible* $\mathcal{K}$ *de* k_3, *on calcule*

$$z^1 = S^{-1}(y + \mathcal{K}) et z^{1'} = S^{-1}(y' + \mathcal{K}).$$

Si $z^1 + z^{1'} = \beta$, *on sait que* $\mathcal{K}$ *est une valeur probable pour* k^3.

Si $z^1 + z^{1'} \neq \beta$, *on sait que* $\mathcal{K}$ *est une valeur probable pour* k^3.

Calcul de $P_{\alpha,\beta}$.

Supposons que $S : \mathbf{F}_2^s \longrightarrow \mathbf{F}_2^s$.
On calcule les coefficients de la matrice D *de taille* $2^s \times 2^s$ *:*

$$D[\alpha, \beta] = Card\{(x, x') \in (\mathbf{F}_2^s)^2 | x + x' = \alpha et Sx + Sx' = \beta\}$$

Alors, pour les entrées et sorties de S*, on a :*

$$P_{\alpha,\beta} := P(Sx + Sx' = \beta | x + x' = \alpha) = \frac{D[\alpha, \beta]}{2^s}.$$

On peut donc suivre l'évolution des différences dans l'algorithme, en découpant par sous-blocs :

$$P_{\alpha,\beta} = P(x_{r-1} + x'_{r-1} = \beta | x + x' = \alpha) = \prod P(x_i + x'_i = \beta_i | x_{i-1} + x'_{i-1} = \alpha_i)$$

et en tenant compte de l'action de la permutation sur les β_i*.*

4.2 Cryptanalyse linéaire

La cryptanalyse linéaire consiste à simplifier l'algorithme de chiffrement en faisant une approximation linéaire.
En augmentant le nombre de couples disponibles, on améliore la précision de l'approximation et on peut en extraire la clé. C'est une attaque à clair connu.
La cryptanalyse linéaire s'intéresse aux relations linéaires entre les bits au cours de l'algorithme. La méthode consiste à attaquer le dernier tour comme dans le cas de l'attaque différentielle.
Pour a $\in \mathbf{F}_2^n$, notons $a.x = a_1x_1 + ... + a_nx_n$.
Notons

$$P_{a,b} := P(a.x + b.x_{r-1} = 0)$$

pour $x \in M$ et supposons que $P_{a,b}$ soit assez éloignée de $\frac{1}{2}$

On itère, pour un grand nombre de couples clairs-chiffrés (x, y) les étapes suivantes : pour toutes les valeurs possibles $\mathcal{K}$ de k_r :
- Calculer $z := F^{-1}(y, \mathcal{K})$
- Si $a.x + b.z = 0$, incrémenter un compteur associé à $\mathcal{K}$.

Pour un assez grand nombre de couples, un compteur est particulièrement élevé ou bas, c'est celui de k_r.

Chapitre 5

Matrice de Diffusion MDS

Notation 1. *- $E = GF(2)^m$ est un GF(2) espace vectoriel des m-uplets binaires.*
- $\mathcal{L} = \mathcal{L}(E,E)$ est l'anneau des endomorphismes GF(2) linéaire.
- $\mathcal{M}_{s,t}(\mathcal{R})$ est un $\mathcal{R}$-module de matrices de taille $s \times t$ sur un anneau $\mathcal{R}$.
- $\mathcal{M}_k(\mathcal{R}) = \mathcal{M}_{k,k}(\mathcal{R})$
- Si $\varphi \in \mathcal{L}$ est un endomorphisme linéaire de E, $M_\varphi \in \mathcal{M}_m(GF(2))$ est sa matrice binaire associée et par convention on a : Si $x = (x_1, ..., x_m) \in E$ alors $\varphi(x) = xM_\varphi$.

5.1 Codes par Blocs additifs sur GF(2^m) et Matrices de Diffusion MDS

La plupart du temps pour l'alphabet A on prend une structure mathématique. Dans la suite, on va s'intéresser à trois d'entre elles :
-(A,+) est un groupe additif. Un code additif C de longueur r sur A est un sous groupe de $(A, +)^r$.
-A = F est un corps fini. Un code linéaire sur F est un F-s.e.v de l'espace vectoriel F^r.
-A = R est un anneau R. Un code linéaire sur R est un R-sous module de R^r.

Dans tous les cas, l'alphabet A possède une structure de groupe additif qui est commutative.
Dans ce cas, la distance minimale du code devient le poids minimal des éléments non nuls.

5.1.1 Codes par Blocs sur E=GF(2^m)

Dans ce mémoire, nous nous intéressons aux codes additifs définis sur E, c'est-à-dire que l'alphabet est le groupe additif constitué par l'ensemble des m-uplets binaires munis de l'addition composante par composante.

Définition 27. *Un code par blocs C de longueur r sur E est un code additif sur l'alphabet (E, +).*

A partir de l'isomorphisme de l'espace vectoriel $E^r \simeq GF(2^m)^r$, un code par blocs C de longueur r est également un code linéaire binaire de longueur n = mr sur GF(2).
Cependant, nous ne sommes pas intéressés par ses propriétés binaires, mais par ses propriétés de bloc. En particulier, nous ne regardons pas le poids binaire des mots de code,

mais le poids par blocs des mots de code.
Dans le reste de ce mémoire, à moins qu'il ne soit explicitement indiqué, w(c) désigne le poids par bloc d'un élément $c \in E^r$ et dC désigne la distance minimale du code par bloc C.

5.1.2 Codes par Blocs systématique

Définition 28. *Supposons que $|C|$ est de la forme 2^{km} pour un entier k. Un code C est un code par bloc au sens strict de pseudo-dimension k s'il existe une fonction de codage linéaire systématique*
$\Phi : E^k \longrightarrow E^r$ tels que $\Phi(x)=(x_1, ..., x_k, \Phi_1(x), ..., \Phi_{r-k}(x))$,
$\Phi_i \in (E^k, E^r)$, $x =(x_1, ..., x_k) \in E^k$ et $\Phi(E^k)=C$.

Autrement dit, il existe une fonction de codage telle que les k premiers blocs représentent le message x encodé.

Cette définition est équivalente au fait que l'image binaire de C est de dimension k_2 = mk et admet une matrice génératrice binaire systématique G = (I | M) où I est la matrice identité de taille mk et M est une matrice binaire de taille Mk × m(r-k).

Définition 29. *Un code par blocs systématique C est un code équivalent à un code par blocs au sens strict par permutation des blocs.*

Suivant les notations de la Définition 26, les applications linéaires $\Phi_i : E^k \longrightarrow E^r$ peuvent être décomposées en k éléments de $\mathcal{L}$: $\Phi_i(x) = \sum_{j=1}^{k} \varphi_{i_j}(x_j) \in \mathcal{L}$
Si $M_{i,j} \in M_m(GF(2))$ représente une matrice binaire m×m correspondant à l'endomorphisme $\varphi_{i,j}(i.e \varphi_{i,j}(a) = aM_{i,j}$ pour tout a ∈ E), la matrice génératrice systématique binaire de C est :

$$G = \begin{pmatrix} I_m & 0_m & \cdots & 0_m & M_{1,1} & \cdots & M_{1,r-k} \\ 0_m & \ddots & \ddots & \vdots & \vdots & & \vdots \\ \vdots & \ddots & \ddots & 0_m & \vdots & & \vdots \\ 0_m & \cdots & 0_m & I_m & M_{k,1} & \cdots & M_{k,r-k} \end{pmatrix} \tag{5.1}$$

où I_m et 0_m sont respectivement une matrice identité et une matrice nulle de taille m×m.
Dans ce sens nous pouvons construire une matrice $\mathcal{L} - generatrice\ \mathcal{G} \in \mathcal{M}_{k,r}(\mathcal{L})$

$$\mathcal{G} = \begin{pmatrix} Id & 0 & \cdots & 0 & \varphi_{1,1} & \cdots & \varphi_{1,r-k} \\ 0 & \ddots & \ddots & \vdots & \vdots & & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & & \vdots \\ 0 & \cdots & 0 & Id & \varphi_{k,1} & \cdots & \varphi_{k,r-k} \end{pmatrix} \tag{5.2}$$

Où Id et 0 sont respectivement une fonction identité et une fonction nulle de $\mathcal{L}$. Les

mots de code c de C sont de la forme c = x$\mathcal{G}$, x=$(x_1, ..., x_k) \in E^k$. Par convention, $a\varphi = \varphi(a)$ et pour tout φ et $\psi \in \mathcal{L}$, $\varphi\psi$ représente $\varphi o \psi$.

Définition 30. *Soit C un code par bloc systématique général sur E. Une matrice $\mathcal{L}$-génératrice de C est une matrice k×r $\mathcal{G} = (\varphi_{i,j})$ sur $\mathcal{L}$ telle que la matrice $G = (M_{\varphi_{i,j}})$ de taille km×rm est une matrice génératrice binaire du code C considéré comme un code linéaire sur GF(2) de dimension km et de longueur rm.*

5.1.3 Codes par Blocs Systématiques Équivalents

Dans la situation classique des codes linéaires, deux codes C et C' sont équivalents s'il existe une isométrie ψ (c'est-à-dire un endomorphisme linéaire ψ de $\mathbf{F}_n$ en préservant la distance de Hamming) telle que C' = ψ(C). Un résultat majeur sur les isométries dans le contexte des codes linéaires sur un corps fini est le fait que les isométries correspondent aux matrices monomiales, c'est-à-dire les n × n matrices avec un et un seul élément non nul par ligne et par colonne.
En pratique, une telle isométrie consiste à multiplier chaque coordonnée d'un mot de code par un scalaire non nul, puis en permutant ces coordonnées.
Ces propriétés peuvent être facilement généralisées au cas des codes par blocs. Tout d'abord, nous allons caractériser certaines isométries sur E^r.

Soit S_r le groupe de permutations agissant sur l'ensemble des indices [1 ; r]. Une permutation $\sigma \in S_r(r)$ agit sur E^r de façon naturelle : Si $x = (x_1, ..., x_r) \in E^r$, on définit $x' = \sigma(x) = (x_{\sigma^{-1}(1)}, ..., x_{\sigma^{-1}(r)})$. Clairement, σ est une isométrie de E^r pour la distance(par bloc) de Hamming. Si P_σ est la matrice de permutation associée à σ, on a $\sigma(x) = xP_\sigma$. La multiplication scalaire dans le cas de codes linéaires est remplacée par l'action d'éléments inversibles de $\mathcal{L}$, c'est-à-dire d'éléments du groupe linéaire GL(m, 2). Si $\lambda = (\lambda_1, \lambda_2, ..., \lambda_r)$ est un r-tuple d'éléments de GL(m, 2), il agit sur E^r comme suit : $\lambda(x) = (\lambda_1(x_1), ..., \lambda_r(x_r)) = (x_1\lambda_1, ..., x_r\lambda_r)$. Une telle application est clairement une isométrie pour la distance (par bloc) de Hamming sur E_r. De plus, la matrice diagonale D_λ avec des éléments diagonaux λ_i, est la matrice de cette isométrie : $\lambda(x) = xD_\lambda$. La proposition suivante donne la caractérisation des isométries de E_r.

Proposition 11. *Le groupe isométrique de E^r pour la distance de hamming (code par blocs) est le groupe constitué de matrices carrées de taille r avec un et un seul élément inversible non nul sur chaque ligne et colonne.*

Si C est un code par blocs avec $\mathcal{G}$ sa matrice $\mathcal{L}$-generatrice et $\mathcal{M}$ une matrice monomiale dans Mon(GL(m,2)) alors la matrice $\mathcal{G}' = \mathcal{G}\mathcal{M}$ est une matrice $\mathcal{L}$-génératrice de l'image C' de C par $\mathcal{M}$.

Définition 31. *Deux codes par blocs C et C' sont équivalents s'il existe une transformation $\mathcal{M}$ tel que C' soit l'image de C par $\mathcal{M}$.*

5.1.4 Code par Blocs Systématique MDS et Matrices MDS

Si un code par bloc additif C est MDS, alors $|C| = 2^{m(r-d+1)}$, donc sa taille est nécessaire une puissance de 2^m. De plus, à la suite des résultats, on peut montrer que C admet une matrice $\mathcal{L}$-génératrice systématique $\mathcal{G}$ et aussi $|C| = 2^{mk}$, où k est la pseudo-dimension de C et la condition MDS devient k + d = r + 1.
En outre, C est MDS si et seulement si la restriction de $\mathcal{G}$ à k colonnes conduit à une matrice inversible dans $\mathcal{M}_k(\mathcal{L})$.
Pour les applications cryptographiques, nous nous intéressons particulièrement à la partie de redondance des matrices systématiques $\mathcal{L}$-génératrice des codes par bloc MDS.

Définition 32. *Un code par bloc additif est MDS si et seulement si s'il admet une matrice $\mathcal{L}$-génératrice telle que toute sous matrice carrée e×e de $\mathcal{M}$ soit une matrice qui appartient à $\mathcal{A}ut(E^e)$.*

Notons que, puisque $\mathcal{L}$ n'est pas un anneau commutatif, nous n'utilisons pas la notion de déterminant pour les sous-matrices carrées de $\mathcal{M}$. Cependant, l'inversibilité d'une matrice carrée e × e est directement liée à l'inversibilité de la matrice binaire me × me correspondante obtenue en substituant à chaque entrée l'endomorphisme $\varphi_{i,j}$ qui correspond à la matrice binaire $\mathcal{M}_{\varphi_{i,j}}$ de taille m×m.

Définition 33. *Une matrice $\mathcal{M} \in \mathcal{M}_{k,s}(\mathcal{L})$ est MDS si le code par bloc systématique C ayant comme matrice $\mathcal{L}$-génératrice $\mathcal{G} = (I_k|\mathcal{M})$ est un code MDS de longueur $r = k + s$*

Proposition 12. *Soit $\mathcal{M} \in \mathcal{M}_{k,s}(\mathcal{L})$ une matrice MDS. Une matrice $\mathcal{M}'$ obtenue par toute permutation de lignes et de colonnes de $\mathcal{M}$ est aussi MDS.*

Preuve

Soient s = r - k et C un code par bloc systématique avec une matrice $\mathcal{L}$-génératrice $\mathcal{G} = (I_k|\mathcal{M})$. Pour préserver la structure systématique de la matrice, nous appliquons à C une permutation séparant les k premières positions et les s dernières. Soit $\sigma = (\sigma_1, \sigma_2) \in S_r$, $\sigma_1 \in S_k$, $\sigma_2 \in S_s$. Soit C' = σ(C) et si la matrice $\mathcal{L}$-génératrice de C' est $\mathcal{G}' = (I_k|\mathcal{M}')$, on a $\mathcal{M}' = \Pi_{\sigma_1^{-1}}\mathcal{M}\Pi_{\sigma_2}$ où $\Pi_{\sigma_1^{-1}}$ et Π_{σ_2} sont respectivement les matrices $k \times k$ et $s \times s$ associées à σ_1^{-1} et σ_2.

Proposition 13. *Soit $\mathcal{M} \in \mathcal{M}_{k,s}(\mathcal{L})$ une matrice MDS. Une matrice $\mathcal{M}'$ obtenue en multipliant à gauche toute ligne de $\mathcal{M}$ et en multipliant à droite toute colonne de $\mathcal{M}$ par certains éléments de GL(m,2) est MDS.*

Preuve

Similaire, supposons que $\lambda = (\lambda_1, ..., \lambda_r) \in GL(m, 2)^r$ un r-uplets de scalaires "non nuls" (i.e elements inversible dans $\mathcal{L}$) On décompose $\lambda = (\lambda_{(1)}|\lambda_{(2)})$ en séparant les k premières composantes et les s dernières composantes. La matrice $\mathcal{L}$-génératrice de l'image C' de C

par la transformation λ est $\mathcal{G}' = (I_k|\mathcal{M}')$ avec $\mathcal{M}' = D_{\lambda_{(1)}^{-1}}\mathcal{M}D_{\lambda_{(2)}}$ où $D_{\lambda_{(1)}^{-1}}$ et $D_{\lambda_{(2)}}$ sont respectivement les matrices diagonales $k \times k$ et $s \times s$ avec sur les diagonales $\lambda_{(1)}^{-1}$ et $\lambda_{(2)}$.

Définition 34. *Deux matrices MDS $\mathcal{M}$ et $\mathcal{M}'$ dans $\mathcal{M}_{k,s}(\mathcal{L})$ sont équivalentes si $\mathcal{M}'$ peut être déduite de $\mathcal{M}$ en appliquant les transformations données dans les Propositions 6 et 7.*

5.1.5 Application à la Cryptographie des Matrices de Diffusion MDS

Classiquement, les algorithmes cryptographiques symétriques alternent les couches de confusion et les couches de diffusion dans leurs processus cryptographiques itératifs. La couche de confusion consiste à appliquer une fonction non linéaire, appelée S-box, qui agit généralement sur r blocs de taille m. Les valeurs typiques de m sont 4 ou 8. La couche diffusion assure la diffusion de toute différence d'entrée entre les différents blocs r. Pour l'efficacité, cette couche diffusion est en fait une application linéaire de E^r à E^r.
L'objectif de cette couche de diffusion n'est pas d'assurer une diffusion à l'intérieur de chaque bloc, mais une diffusion entre les blocs. En pratique, comme précédemment, on note $x = (x_1, ..., x_r)$, les r blocs d'entrée et par $y = xM$ les blocs de sortie où M peut être considéré comme une $\mathcal{L}$-matrice r×r ou une rm×rm matrice binaire.
Un exemple c'est le mixColumn la matrice de diffusion de l'AES.
Dans ce mémoire, on introduit la notion de numéro de branche, qui est une mesure de la résistance d'une matrice de diffusion contre la cryptanalyse linéaire et différentielle dans le contexte des chiffrements par blocs SPN.
Nous n'allons pas décrire en détail les concepts de nombres de branches linéaires et différentielles et leurs liens avec la cryptanalyse. Nous donnons simplement une définition de ces concepts adaptés à notre approche.

Nous avons besoin de quelques notations : soit $\mathcal{M} \in \mathcal{M}_{k,s}(\mathcal{L})$ alors M représente la matrice binaire km×ks. La notation $\mathcal{M}^T$ correspond à la matrice transposée de $\mathcal{M} \in \mathcal{M}_{k,s}(\mathcal{L})$ et la matrice $\mathcal{M}^{T_*}$ représente un élément de $\mathcal{M}_{k,s}(\mathcal{L})$ et est associée à la matrice binaire M^T. Notons que $\mathcal{M}^T$ et $\mathcal{M}^{T_*}$ ne sont pas égales.

Définition 35. *Soit $\mathcal{M} \in \mathcal{M}_{k,s}(\mathcal{L})$ une matrice de diffusion qui prend en entrée k blocs de m bits et en sortie s blocs de m bits.*
Le nombre de branches différentiables de $\mathcal{M}$ est la distance minimale du code par bloc additif généré par$(I_k|\mathcal{M}^T)$.
Le nombre de branches linéaires de $\mathcal{M}$ est la distance minimale du code par bloc additif généré par$(I_s|\mathcal{M}^{T^*})$.

Théorème 11. *Une matrice de diffusion binaire M de taille km×rm (ou équivalent à une matrice de diffusion $\mathcal{M} \in \mathcal{M}_{k,s}(\mathcal{L})$) a un nombre maximal de branches différentielles si et seulement si elle a un nombre maximal de branches linéaires. Dans cette situation les deux matrices sont MDS.*

Proposition 14. *Si $\mathcal{M}$ est une matrice carrée MDS alors $\mathcal{M}^{-1}$ est aussi MDS.*

Proposition 15. *Supposons que $\mathcal{M}$ soit une matrice (non nécessairement carrée) MDS tels que les éléments de $\mathcal{M}$ commutent alors la matrice $\mathcal{M}^T$ est MDS.*

5.1.6 Structure d'Anneau sur GF(2^m) des codes par blocs additifs

Dans cette section, on considère un anneau $\mathcal{R} = (GF(2)^m, +, \times) = (GF(2), +, \times)^m$

Pour i ∈ [0 ;m-1], on note par $e_i \in GF(2)^m$ un élément tel que $e_{i,j} = 0$ excepté $e_{i,i} = 1$ et π_i la projection x $=(x_0, ..., x_{m-1}) \longrightarrow x_i$.

Soit $\mathcal{C}$ est code additif de longueur r sur $\mathcal{R}$, nous considérons les codes dérivés suivants :
$e_i\mathcal{C} = \{e_ic = (e_ic_0, ..., e_ic_{r-1}) | c \in \mathcal{C}\} \subset \mathcal{R}^r$
et $\pi_i\mathcal{C} = \{\pi_i(c) = (\pi_i(c_0), ..., \pi_i(c_{r-1})) | c \in \mathcal{C}\} \subset GF(2)^r$.

Un $\mathcal{R}$-code linéaire de longueur r est un sous module de $\mathcal{R}^r$. Un code additif $\mathcal{C}$ est linéaire sur $\mathcal{R}$ si et seulement si les codes $e_i\mathcal{C}$ sont des sous codes de $\mathcal{C}$.

Proposition 16. *Un code additif $\mathcal{C}$ de longueur r sur $\mathcal{R}$ est linéaire si et seulement si il est somme directe des codes $e_i\mathcal{C}$, pour tout i ∈ [0 ;m-1]. Dans ce cas il est isomorphe à la somme directe des codes binaires $\pi_i(\mathcal{C})$.*

corollaire 1. *La distance minimale d'un code par blocs linéaire est le minimum des distances minimales des codes binaires $\pi_i(\mathcal{C})$.*

5.2 $\mathcal{L}$-codes

5.2.1 Définition de $\mathcal{L}$-codes

L'ensemble $\mathcal{L}$ est un anneau non commutative.

Définition 36. *Un $\mathcal{L}$-code linéaire à gauche $\mathcal{C}$ de longueur r sur $\mathcal{L}$ est un sous module à gauche de $\mathcal{L}^r$.*

Théorème 12. *Soit $\mathcal{C}$ un $\mathcal{L}$-code de longueur r. L'ensemble $C = \{a\varphi = (\varphi_1(a), ..., \varphi_r(a)) \in E^r \mid \forall a \in E$ et $\forall \varphi \in \mathcal{C}\}$ est un code par bloc additif. Réciproquement, Si C est un code par bloc additif, l'ensemble $\mathcal{C} = \{\varphi = (\varphi_1, ..., \varphi_r) \in \mathcal{L}^r \mid \forall a \in E, a\varphi \in C\}$ est un $\mathcal{L}$-code et leurs distances minimales sont égales.*

5.2.2 Dualité des $\mathcal{L}$-codes

Il n'y a pas de notion de dualité sur les codes par blocs additifs lorsqu'ils sont considérés comme des codes par blocs et non comme des codes binaires.

On peut définir une sorte de produit scalaire de la manière suivante : pour tout φ et ψ dans $\mathcal{L}^r$, on note $< \varphi, \psi >= \Sigma_{i=1}^r \varphi_i\psi_i = \Sigma_{i=1}^r \psi_i \circ \varphi_i \in \mathcal{L}$. Notons que $\mathcal{L}^r$ est un module non commutatif sur $\mathcal{L}$. Pour tout $\lambda \in \mathcal{L}$ et $\varphi \in \mathcal{L}$, on représente respectivement

par $\lambda\varphi = (\lambda\varphi_1, ..., \lambda\varphi_r)$ et $\varphi\lambda = (\varphi_1\lambda, ..., \varphi_r\lambda)$ le produit à gauche et à droite.La fonction bilinéaire est linéaire comme module à gauche sur le composant gauche et linéaire comme module à droite sur le composant droit. En particulier $< \lambda\varphi, \psi >= \lambda < \varphi, \psi >$ et $< \varphi, \lambda\psi >=< \varphi, \psi > \lambda$

De plus, cette fonction bilinéaire est non dégénérée en ce sens que si, pour un $\varphi \in \mathcal{L}^r$, $< \varphi, \psi >= 0$ pour tout $\psi \in \mathcal{L}^r$ alors $\varphi = 0$ Nous sommes en mesure de définir le dual d'un $\mathcal{L}$-code.

Définition 37. *Soit $\mathcal{C}$ un $\mathcal{L}$ code linéaire (à gauche). Le dual $\mathcal{C}^\perp$ de $\mathcal{C}$ est un sous ensemble de $\mathcal{L}^r$ défini par $\mathcal{C}^\perp = \{\psi \in \mathcal{L}^r \mid <\varphi,\psi>=0, \forall\varphi \in \mathcal{C}\}$*

Théorème 13. *Soit $\mathcal{C}$ un code linéaire systématique de rang k. Le dual $\mathcal{C}^\perp$ de $\mathcal{C}$ est un $\mathcal{L}$ sous module linéaire à droite de rang r-k.*

Notez que, puisque $\mathcal{C}^\perp$ n'est pas un module à gauche, nous ne pouvons pas associer à ce code un code par bloc additif. Ainsi, cette notion de code dual ne peut pas être étendue aux codes par blocs additifs. Cependant, il reste beaucoup de propriétés utiles en relation avec cette notion de dualité. On peut notamment définir une matrice génératrice de $\mathcal{C}^\perp$ pour sa structure de module à droite. De plus, si $\mathcal{G}$ est une matrice génératrice de $\mathcal{C}$ et $\mathcal{H}$ une matrice génératrice de $\mathcal{C}^\perp$, alors $\mathcal{G}\mathcal{H}^T = 0$ (mais pas nécessaire $\mathcal{H}\mathcal{G}^T = 0$).
Un élément c $\in E^r$ est dans le code par bloc additif $\mathcal{C}$ si et seulement si $c\mathcal{H}^T = 0$. Ainsi, la matrice $\mathcal{H}$ est également appelée matrice $\mathcal{L}$-contrôle code $\mathcal{C}$. De plus, si $\mathcal{G} = (I_k|\mathcal{M})$ est une matrice génératrice de $\mathcal{C}$ sous forme systématique, alors H = $(\mathcal{M}^T|I_{r-k})$ est une matrice génératrice (droite) de $\mathcal{C}^\perp$. Une attention particulière doit être portée au fait que $\mathcal{M}^T$ désigne la transposition de $\mathcal{M}$ au niveau de $\mathcal{L}$ et ne correspond pas à la matrice obtenue par la transposition de son image binaire M (c'est-à-dire la matrice binaire m (r-k) × mk).

De plus, il existe un équivalent du théorème des codes linéaire qui traite le lien entre l'indépendance des colonnes d'une matrice de contrôle et la distance minimale d'un code.

Théorème 14. *Soit $\mathcal{C}$ un code par bloc additif et H une $\mathcal{L}$-matrice de contrôle de $\mathcal{C}$. La distance minimale(par bloc) de $\mathcal{C}$ est d si et seulement si toutes d-1 colonnes de H définissent une application de rang d-1 et certains d colonnes de H définissent une application linéaire de rang strictement inférieur à d.*

En identifiant E_r et $GF(2)^{mr}$, il est possible de définir la notion de dualité binaire d'un code par bloc additif. Cette approche peut être définie en utilisant une sorte de «produit scalaire hermitien» sur $\mathcal{L}$. Nous devons utiliser la transposition d'une application linéaire. Si φ est un élément de $\mathcal{L}$ avec une matrice binaire associée M_φ, la transposition de φ est l'application linéaire $\varphi^T \in \mathcal{L}$ avec comme matrice binaire M_φ^T.
On définit une fonction bilinéaire $< \varphi, \psi >_T = \Sigma_{i=1}^r \varphi_i\psi_i^T \in \mathcal{L}$. On peut remarquer que : $< \lambda\varphi, \psi >_T = \lambda < \varphi, \psi >_T$ et $< \varphi, \lambda\psi >_T =< \varphi, \psi >_T \lambda^T$.
Nous sommes capables de définir le dual binaire d'un $\mathcal{L}$-code.

Définition 38. *Soit $\mathcal{C}$ un $\mathcal{L}$-code linéaire. Le dual $\mathcal{C}^{\perp *}$ binaire de $\mathcal{C}$ est sous ensemble de $\mathcal{L}^r$ défini par :*
$\mathcal{C}^{\perp *} = \{\psi \in \mathcal{L}^r \mid <\varphi, \psi>_T = 0, \forall \varphi \in \mathcal{C}\}$

Théorème 15. *Soit $\mathcal{C}$ un code linéaire systématique de rang k, le dual $\mathcal{C}^{\perp *}$ binaire de $\mathcal{C}$ est un $\mathcal{L}$ sous module linéaire de $\mathcal{L}^r$ de rang r-k i.e un $\mathcal{L}$-code systématique.*

La preuve de ce théorème vient directement de la relation $<\varphi, \lambda\psi>_T = <\varphi, \psi>_T \lambda^T$ Ce qui implique en particulier que, si $<\varphi, \psi>_T = 0$ alors $<\varphi, \lambda\psi>_T = 0$ pour tout λ dans $\mathcal{L}$, Donc $\mathcal{C}^{\perp *}$ est un sous-module gauche.

Pour introduire la relation entre les deux types de dualité, nous introduisons la notation suivante : si $\mathcal{M} = (\varphi_{i,j})$ est une matrice avec des coefficients dans $\mathcal{L}$, on represente $\mathcal{M} = (\varphi_{i,j}^T)$ obtenue en remplaçant chaque coefficient de la matrice par son application transposée. Notez que nous ne transposons pas la matrice elle-même. En particulier, si $\varphi = (\varphi_1, ..., \varphi_r)$, on note $\varphi^* = (\varphi_1^T, ..., \varphi_r^T)$

Proposition 17. *$\varphi \in \mathcal{L}$ est dans $\mathcal{C}^\perp$ si et seulement si φ^* est dans $\mathcal{C}^{\perp *}$*

La preuve c'est une conséquence directe des définitions 24 et 25.

En conséquence de ces résultats, $\mathcal{H}$ est une matrice génératrice du $\mathcal{L}$-code à droite $\mathcal{C}^\perp$ si et seulement si la matrice $\mathcal{H}^*$ est une matrice génératrice du $\mathcal{L}$-code à gauche de $\mathcal{C}^{\perp *}$. En particulier, si $\mathcal{G} = (I_k|\mathcal{M})$ est une matrice génératrice de $\mathcal{C}$ sous forme systématique, alors $\mathcal{H}^* = (\mathcal{M}^{T*}|I_{r-k})$ est une matrice génératrice de $\mathcal{C}^{\perp *}$.

Conclusion

Les travaux de ce mémoire n'était pas de construire des matrices MDS optimales destinées à des applications spécifiques, mais de présenter un cadre de travail général pour une telle recherche. Cette nouvelle approche nous permet de construire à partir des codes par blocs des matrices de diffusions MDS.

Nous avons montré qu'il existe d'autres directions de recherches non explorées notamment les sous-modules non commutatifs sur $\mathcal{L}$.

Bibliographie

[**1**] Nora EL Amrani Thierry P. Berger. Mds diffusion matrices and cryptographic applications. 2015.

[**2**] JUND HWAN SONG SANGWOO PARK DAESUNG KWON, SOO HAK SUNG. Design of block ciphers and coding theory. 2005.

[**3**] Bora Aslan Tolga Sakalli. Algebraic construction of 16*16 binary matrices of branch number 7 with one ?xed point.

[**4**] ANNEAUX ET CORPS. 2007.

[**5**] Pr. Omar Dianka. Anneaux et Modules. 2015.

[**6**] Dr. Abdoulaye Mbaye. Corps fini. 2015.

[**7**] Dr Gregory Landais. Mise en oeuvre de cryptosystèmes basés sur les codes correcteurs d'erreurs et de leurs cryptanalyses. 2014.

[**8**] Dr Babacar Alassane Ndaw. L'algorithme de chiffrement AES(Advanced Encryption Standard). 2015.