$N°$ d'ordre : 24/2021-C/MT

Ministère de L'enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene

Faculté des Mathématiques

THÈSE DE DOCTORAT

Présentée pour l'obtention du grade de doctorat

En : MATHÉMATIQUES

Spécialité : Mathématiques Fondamentales et Cryptographie

Par : BOUMEZBEUR Mouna

Sujet

Les codes correcteurs d'erreurs et l'analyse des critères de sécurité des fonctions Booléennes et vectorielles

Soutenue publiquement le jeudi 04 mars 2021, devant le jury composé de :

MAMACHE Fatiha	Maître de Conférences A	à l'USTHB	Présidente
GUENDA Kenza	Professeur	à l'USTHB	Directrice de thèse
BATOUL Aicha	Maître de Conférences A	à l'USTHB	Examinatrice
NOUI Lemnouar	Professeur	à U.Batna	Examinateur
SENOUCI Abdelkader	Maître de Conférences A	à U.Jijel	Examinateur
MERABET Brahim	Maître de Conférences B	à U.Ghardaia	Invité
ALLAILOU Boufeldja	Maître de Conférences A	à IRD.Reghaia	Invité

À la mémoire du Dr. Aini Laoudi.
À mes parents.

Remerciements

C'est avec plaisir que je remercie ici toutes les personnes qui m'ont aidé ou accompagné durant cette thèse.

En première instance, je tiens à exprimer mes remerciements les plus sincères à Madame Guenda Kenza, ma directrice de thèse. Son admirable générosité alliée à son enthousiasme scientifique intarissable a accompagné et soutenu sans faillir ces longues années de thèse. Elle a toujours été disponible et présente pour m'accompagner à toutes les étapes. Qu'elle trouve ici l'expression de ma profonde reconnaissance et de mon admiration.

J'aimerais ensuite exprimer ma gratitude à Madame Mamache Fatiha, maître de conférences à l'U.S.T.H.B pour m'avoir fait l'honneur d'accepter de présider le jury de ma thèse.

Je tiens à remercier Monsieur le professeur Noui Lemnouar, Madame Batoul Aicha, Monsieur Senouci Abdelkade, Monsieur Merabet Brahim et Monsieur Allailou Boufeldja de me faire l'honneur de constituer mon jury de thèse.

Ensuite, je voudrais remercier chaleureusement Madame Mesnager Sihem pour m'avoir proposé le sujet, pour sa grande disponibilité et ces judicieux conseils. Je suis extrêmement redevable du temps qu'elle m'a consacré, d'avoir lu, relu et corrigé plusieurs fois mon manuscrit avec une incroyable attention et patience.

Durant ma thèse, j'ai également séjourné pendant un mois à l'université de Primorska, dans le cadre d'un stage de court durée. Je remercie Monsieur Pasalic Enes pour m'y avoir chaleureusement accueillie. Ces remerciements s'adressent aussi à tous les membres de l'équipe qui ont contribué, par leur disponibilité et leur gentillesse, à rendre ce séjour plus qu'agréable. Une pensée toute particulière s'adresse à Bapic Amar.

Les dernières lignes de ces remerciements vont à mon bien le plus précieux, ma famille. Je suis infiniment reconnaissante et immensément redevable à mes parents et mes frères Mohamed Lamine et Walid de m'avoir toujours inconditionnellement encouragée et soutenue, pour leur intérêt dans mon travail et leur affection. Hélas, j'ai beau réécrire ce paragraphe, ces quelques mots restent bien trop faibles face à la reconnaissance que je leur porte.

Résumé

La cryptographie, ou l'art de la communication sécurisée, est considérée comme un axe de recherche publique de plusieurs applications. Cette branche s'intéresse au traitement de l'information. De la même manière, les codes correcteurs d'erreurs sont les outils qui permettent une communication numérique à travers un canal bruité, de façon à ce que les erreurs dans la transmission des bits soient détectées et corrigées par le récepteur. Dans ces deux cadres, les fonctions Booléennes et vectorielles jouent un rôle important. Cependant, la fonction utilisée doit être équilibrée, ait un haut degré algébrique, une haute non−linéarité et une immunité algébrique élevée.

Parmi les nombreux domaines reliés en partie à la cryptologie, il y a les travaux présentés dans cette thèse, ou en s'intéressant tout particulièrement aux liens entre les codes correcteurs et les fonctions booléennes et vectorielles. Dans cette thèse un nouveau lien entre les annulateurs des fonctions vectorielles et les codes LCD est établie, et alors une borne inférieure sur l'immunité algébrique des fonctions vectorielles et Booléennes est donnée en fonction de la distance minimale des codes LCD associés. Ensuite, une étude des propriétés de ces codes nous a menée à donner un lien entre l'immunité algébrique et l'immunité spectrale des fonctions vectorielles, ainsi qu'un lien entre ces deux critères et le poids des codes LCD. Nous avons aussi donné une caractérisation des propriétés cryptographiques du produit de Dirichlet des fonctions booléennes, ainsi qu'une généralisation de cette notion aux fonctions vectorielles.

Mots−clés : Fonctions Booléennes, Fonctions vectorielles, codes cycliques, codes LCD, attaques algébriques, annulateurs, immunité algébriques.

Table des matières

Table des figures

Notations

Nous donnons ici la liste des notations utilisées dans cette thèse ainsi que la page de leur définition.

$SI(z)$	Immunité spectrale de la suite z, page 12
$\mathbb{F}_2^n$	Corps fini à 2^n éléments, page 16
$supp(f)$	Support d'une fonction Booléenne f, page 18
$wt(x)$	Poids de Hamming de x, page 18
$\mathcal{B}_n$	Ensemble des fonctions Booléennes à n variables, page 18
$d(x, y)$	Distance de Hamming de x et y, page 18
$\deg(f)$	Degré de la forme algébrique normale d'une fonction Booléenne f, page 20
$D_a(f)$	Dérivée d'une fonction Booléenne f relativement au vecteur a, page 21
tr_n^m	La fonction trace de F_{2^m} sur F_2, page 21
$F_f(a)$	Transformée de Fourier discrète de f, page 22
$W_f(a)$	Coefficient de Walsh de f en a, page 22
$\{W_f(a),\ a \in \mathbb{F}_2^n\}$	Spectre de Walsh de f, page 22
$nl(f)$	Non−linéarité de f, page 34
$nl_r(f)$	Non−linéarité d'ordre r de f, page 35
$FAI(F)$	Immunité algébrique rapide de f, page 49
$AI(f)$	Immunité algébrique de f, page 54
$\rho(1, m)$	Le rayon de recouvrement de $RM(1, m)$, page 63
$RM(r, m)$	Le code de Reed−Muller d'ordre r et de longueur 2^m, page 68
k_m	Le code de Kerdock de longueur 2^m, page 72

Introduction générale

Aujourd'hui l'information transite presque entièrement sous une forme numérisée sur une très grande variété de support, et cela, avec des quantités, des débits et des services impressionnants. Tout cela met en jeu énormément de disciplines pour lesquelles la recherche est encore très active, comme la théorie des codes correcteurs ou la cryptographie.

L'objectif fondamental de la cryptographie est de permettre à deux personnes de communiquer à travers un canal peu sûr d'une manière à ce qu'une tierce personne ne puisse pas comprendre ce qui est échangé. Ainsi, l'un de ses buts premiers est de garantir la confidentialité des données en chiffrant l'information, de telle sorte que les données échangées ne puissent pas être modifiées ou manipulées.

De la même manière que la cryptographie, les codes correcteurs d'erreurs sont les outils qui permettent une communication numérique à travers un canal bruité, d'une façon ou les erreurs dans la transmission des bits soient détectées et corrigées par le récepteur.

Dans ces deux cadres, les fonctions Booléennes jouent un rôle important :

— En codage : chaque code de longueur 2^n, pour certain entier positif n, peut être interprété comme un ensemble de fonctions Booléennes, les codes importants, comme les codes cycliques, les codes Reed−Muller et les codes de Kerdock, peuvent être définis de cette manière comme des ensembles des fonctions Booléennes.

— En cryptographie : les fonctions Booléennes sont devenues importantes pour la conception et la sécurité de certains systèmes de chiffrement symétriques, les fonctions utilisées dans ces systèmes doivent avoir des propriétés très particulières pour résister à des attaques spécifiques.

Parmi les nombreux domaines reliés en partie à la cryptologie, les travaux présentés dans cette thèse s'intéressent tout particulièrement à établir un lien entre la théorie des codes et la théorie des fonctions Booléennes et vectorielles en cryptographie. Ainsi deux

objectifs différents sont alors visés.

La présente thèse est organisée de la manière suivante :

Le chapitre 1 est une introduction à la cryptographie et les codes correcteurs d'erreurs.

Le chapitre 2 est un rappel des notions indispensables à la compréhension de cette thèse, principalement les corps finis, les fonctions Booléennes et vectorielles utilisées en cryptographie.

Dans le chapitre 3, nous commençons par donner une description du contexte d'utilisation des fonctions Booléennes et des fonctions vectorielles dans les systèmes de chiffrement symétriques, ensuite nous définissons les propriétés cryptographiques suivantes : l'équilibre ; le degré algébrique ; la non−linéarité et l'immunité algébrique.
Les attaques correspondantes seront également mentionnées, mais pas décrites. Par contre, un état de l'Art sur les attaques algébriques est traité. Enfin, nous présentons la notion de l'immunité algébrique des fonctions Booléennes et vectorielles qui est un critère nécessaire pour résister à ces attaques.

Le chapitre 4 est consacré aux codes correcteurs d'erreurs (les codes linéaires, les codes cycliques et LCD). Nous donnerons notamment une description du contexte d'utilisation des fonctions Booléennes et vectorielles dans la théorie des codes.

Nous exposons dans le chapitre 5, nos premiers résultats : "Linear codes from vectorial Boolean functions in the context of algebraic attacks". Le premier résultat est un lien direct entre les annulateurs des fonctions vectorielles et une classe des codes LCD. Comme deuxième résultat nous avons donné un lien entre l'immunité algébrique et l'immunité spectrale des fonctions vectorielles, ainsi qu'un lien entre ces deux critères et le poids des codes associés. Enfin, nous proposons une généralisation de la notion du complément algébrique des fonctions Booléennes aux fonctions vectorielles. Cette généralisation nous a mené à donner une borne sur le degré algébrique le plus bas des annulateurs des fonctions vectorielles.

Enfin, dans le (dernier) chapitre 6, nous présentons : "Cryptographic properties of Dirichlet product for vectorial Boolean functions". Nous étudions d'abord les propriétés cryptographiques du produit de Dirichlet pour les fonctions Booléennes, ensuite, nous proposons une généralisation de cette notion aux fonctions vectorielles.

Chapitre 1

Introduction aux codes correcteurs et à la cryptographie

Pendant longtemps, les rois et les généraux ont dû se doter de moyens de communication efficaces pour gouverner leur pays et diriger leur armées, en étant conscients des risques engendrés si les messages tombaient entre les mains de l'ennemi. Ce fut la crainte de ces interceptions des messages qui fut à l'origine du développement des techniques utilisées pour que seuls les vrais destinataires puissent les lire, dans le même temps où les ennemis essayaient de briser ces techniques. Le problème de confidentialité a mené les nations à créer des services secrets, pour assurer la sécurité des communications.

Aujourd'hui, l'information prend une importance croissante et la révolution des communications a bouleversé les sociétés modernes, aussi, le codage des messages, dit cryptage, joue un rôle très important dans la vie quotidienne. L'information transite presque entièrement sous une forme numérisée, sur une très grande variété de supports, et cela, avec des quantités, de débits et des services impressionnants, comme nos e-mails qui passent par différents ordinateurs et nos appels téléphoniques qui rebondissent sur les satellites. Ces voies peuvent être facilement infiltrées, alors le cryptage est le seul moyen de protéger notre intimité ainsi que mettre à l'abri les entreprises et leurs clients.

Ce fut l'article de Shannon, publié en 1949, intitulé "*Communication Theory of Secrecy Systems*" [157] qui est la première publication dans le domaine de la cryptographie où les bases de la théorie de l'information ont été introduites.

1.1 Introduction aux codes correcteurs

La théorie des codes traite de la forme de l'information elle−même quand elle doit être stockée, ou transmise à travers un canal de communication. L'information doit être reçue par le destinataire en sécurité, dans son intégralité et le plus rapidement possible [14, 117, 156, 162]. Or, quel que soit le canal considéré, avec des variations selon le support, on ne peut jamais le supposer *sûr*, dans plusieurs sens du terme, alors il y a toujours des erreurs au cours de la transmission. C'est−à−dire que tous les canaux permettant de transmettre une information numérique peuvent être soumis à des perturbations qui sont susceptibles d'altérer une partie des messages, et donc modifier le sens (le message est susceptible d'être lu, voire altéré, par des tiers plus ou moins bien intentionnés). Cependant, l'information peut être de n'importe quel type pourvu qu'on puisse en donner une représentation numérique : textes, images, vidéos, sons ... par exemple. Les canaux de transmission peuvent également être de tous types, réseaux de câbles ou d'ondes, via éventuellement un support de stockage.

La communication de l'information commence par sa formulation par l'émetteur, se poursuit par un transit via un canal, et se termine par la reconstitution du message par le destinataire. Alors, ce dernier, qui effectue généralement l'opération inverse de l'émetteur, reçoit le signal. Le schéma de communication proposé par Shannon est décrit par la figure 1.1.

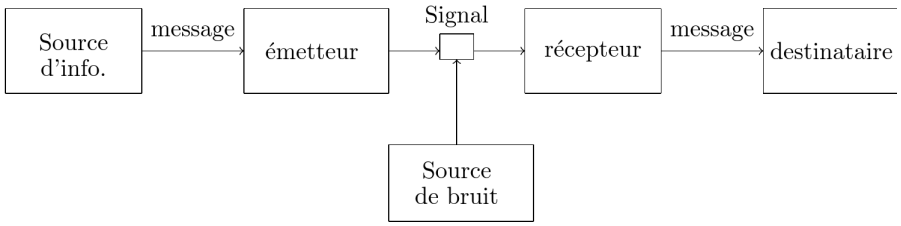

FIGURE 1.1 – Schéma de communication

Le but de la détection et la correction des erreurs est réalisé en utilisant un algorithme d'encodage qui transforme l'information (une suite de symboles choisis dans un alphabet fini A), avant de l'envoyer dans un canal. Si l'alphabet avec lequel les messages sont construits est dans l'espace vectoriel $\mathbb{F}_2$ d'ordre 2, alors le code est dit binaire. Si le code n'est pas binaire, alors les symboles de l'alphabet doivent être transmis en vecteurs binaires avant d'être envoyés sur un canal binaire.

Le modèle du canal de communication le plus simple, et aussi le plus utilisé, est *le*

canal binaire symétrique (CBS). Dans ce canal, chaque bit transmis peut être changé avec une probabilité p et inchangé avec une probabilité $1+p$ (le 0 devient 1 et vice versa). Il existe de nombreux autres modèles comme le canal à bruité blanc gaussien ou le canal à effacements.
Le modèle du canal à effacements (CBE), introduit en 1955 par Elias [75], est un canal tel qu'après la transmission d'un mot de code certaines positions sont effacées, c'est−à−dire que la position de l'erreur est connue, mais on ne sait pas au juste quel symbole il y avait dans cette position.
Sur un canal d'effacement, la probabilité d'erreur est indépendante du mot de code transmis et ne dépend que de la position des erreurs, désigné par *motif d'effacements*.

Définition 1. *On peut coder les positions effacées comme des* 1 *dans un mot binaire de longueur n, on parle alors de motif d'effacements. Pour un motif d'effacements fixé, on notera $\mathcal{I}$ l'ensemble des mots du code dont le support est inclus dans ce motif. $\mathcal{I}$ est un sous−espace vectoriel et l'ensemble des mots de code compatibles avec le mot reçu est égal à $c+\mathcal{I}$ où c est le mot de code envoyé.*

En 1948, Claude Shannon a mis au point la "*théorie mathématique de l'information*" [156], permettant de calculer la *quantité d'information* dans n'importe quel type de message, ce qui revient à déterminer son taux de redondance. Réduire cette redondance est ce qu'aujourd'hui nous appelons la *compression*, ou, par fidélité historique, la *théorie de l'information*.

Ce sont ces trois préoccupations qui retiennent l'attention des concepteurs de méthodes de transmission de l'information : l'*efficacité*, la *sécurité* et l'*intégrité*.
Dans un souci d'*efficacité*, qui relève de l'optimisation de l'espace de stockage, et plus encore de la rapidité de transmission, il s'agit de rendre le message le plus court possible, en ne gardant que ce qui est indispensable à sa compréhension, ou mieux, en le reformulant sans redondance.
La *confidentialité* dont on veut entourer la transmission d'une information est une préoccupation beaucoup plus ancienne que celle de l'efficacité. Puisque le canal de transmission n'est pas sûr et le message peut être intercepté au cours de sa transmission, alors il faut transformer le texte en le décorrélant de sa signification, et en ne laissant qu'à ses destinataires la clef de déchiffrement.

Shannon contribua également à ce domaine apportant pour la première fois, en 1948, une preuve théorique de confidentialité. Une discipline scientifique est aujourd'hui consacrée aux codes secrets, *la cryptologie*.

1.2 Introduction à la cryptographie

La cryptologie, ou la science du secret est composée de la cryptographie, qui est l'art de la communication sécurisée, et de la cryptanalyse, qui est l'analyse de celle−ci.
Dans la problématique du secret, on se sert d'un *bruit* pour perturber le message et cacher le contenu. Le rôle de l'émetteur est alors de construire une perturbation du message dans le but d'empêcher un ennemi, le *cryptanalyste*, de cryptanalyser le message à partir du signal, pour permettre au destinataire légitime de retrouver le message d'origine.

La cryptographie fait partie d'un ensemble de théories et techniques liées à la transmission de l'information comme la théorie des ondes électro−magnétiques, la théorie du signal, la théorie des codes correcteurs d'erreurs et la théorie de la complexité ...

La cryptographie classique est utilisée depuis longtemps par des citoyens ordinaires. Ainsi, son objectif traditionnel est de permettre à deux personnes de communiquer à travers un canal peu sûr, de telle sorte qu'une tierce ne puisse pas comprendre les messages échangés, et que les données échangées ne puissent pas être manipulées ou modifiées. Autrement dit, le but est d'élaborer des méthodes permettant de transmettre des données de manière confidentielle.

Durant ces vingt dernières années, il y a eu une explosion de la recherche académique publique en cryptographie. Et si le but de la cryptographie est d'assurer toujours la confidentialité des échanges, elle est devenue multiforme, et ainsi, le but de la cryptographie moderne est de garantir :

1- **La confidentialité** des informations stockées ou manipulées par le biais des algorithmes de chiffrement.
2- **L'authentification** des communicants d'une communication. Il faut pouvoir détecter une usurpation d'identité, et seul le destinataire connaît le message puisque c'est la seule personne à connaître la clef de chiffrement, autrement dit, l'émetteur est sûr de l'identité du destinataire.
3- **L'intégrité** des informations stockées ou manipulées. Il s'agit de vérifier que le message n'a pas subi d'*altérations, falsification* ou *destruction accidentelle ou volontaire sans qu'on s'en aperçoive* lors de son traitement, conservation ou transmission.
4- **La non−répudiation** des informations. C'est une protection des communicants d'un échange, entre eux, et non plus contre un tiers. D'un côté, l'émetteur du

message ne peut nier avoir écrit ou envoyé le message, et de l'autre côté, le receveur ne peut nier avoir reçu le message.

5- **Rapidité** des calculs de chiffrement et déchiffrement, pour les opérations portant sur de grandes quantités de données, le critère d'efficacité est très important afin de pouvoir chiffrer *à la volée* des flux audio ou vidéo par exemple en utilisant au minimum la bande passante.

Ces critères sont utilisés pour créer des algorithmes complexes dits cryptosystèmes. Un cryptosystème est l'ensemble des algorithmes cryptographiques, de tous les textes en clair, les textes chiffrés et les clefs possibles. Généralement, on distingue deux familles de cryptosystèmes : les systèmes de chiffrement asymétriques et les systèmes de chiffrement symétriques.

1.2.1 Chiffrements asymétriques

Les chiffrements asymétriques ou les chiffrements à clef publique ont été introduits en 1976 quand Diffie et Hellman ont voulu résoudre le problème de l'échange des clefs dans le but d'envoyer des messages chiffrés. Ils étaient persuadés de l'existence d'une solution dont le concept est donné par le scénario suivant :

— Alice envoie à Bob une clef placée dans un coffre verrouillé avec son cadenas.
— Bob lui envoie la boîte en y ajoutant son propre cadenas.
— Alice enlève son cadenas et envoie à Bob la boîte qu'il peut enfin ouvrir.

Alors, de cette manière une tierce personne ne peut pas intercepter leur clef.

Le but est de trouver des fonctions mathématiques permettant un tel codage, Hellman a trouvé une solution utilisant l'arithmétique modulaire :
Alice et Bob choisissent un groupe G et un élément g de grand ordre n dans G. Alice choisit un entier $a \in [1, n-1]$ et envoie à Bob $\alpha = g^a$, de même, Bob choisit un entier $b \in [1, n-1]$ et envoie à Alice $\beta = g^b$. Alors, Alice calcule β^a, tandis que Bob calcule α^b, et les deux partagent un secret commun $g^{ab} = \beta^a = \alpha^b$. Une tierce personne doit alors résoudre le problème du logarithme discret pour trouver a et b à partir de g^a et g^b.

En 1978, Rivest, Shamir et Adleman mettent au point la première réalisation pratique du chiffrement à clef publique, avec un système connu sous le nom de RSA [148]. Le principe est d'avoir deux clefs, une clef publique de chiffrement p et une clef secrète de déchiffrement s, de telle sorte qu'il n'existe pas d'algorithme efficace permettant de trouver s à partir de p.

1.2.2 Chiffrements symétriques

Les chiffrements symétriques, ou les chiffrements à clef secrète, sont héritiers des méthodes anciennes des cryptographies comme les transpositions, les substitutions et le chiffre de Vigenère. L'émetteur et le récepteur disposent chacun d'un algorithme pour chiffrer et déchiffrer respectivement, ces algorithmes sont inverses l'un de l'autre, d'où le terme *symétrique*.

L'émetteur et le récepteur doivent échanger une clef (la clef doit rester secrète sous peine qu'un tiers parvienne à déchiffrer les correspondances), qui leur permet à la fois de chiffrer et déchiffrer. Cet échange de clefs à travers un canal sécurisé où au moyen d'autres techniques cryptographiques, est souvent un point faible de ces méthodes de chiffrement. Cependant, elles possèdent un avantage considérable par rapport à la cryptographie asymétrique, notamment quant au temps de traitement et à la consommation énergétique.
Les systèmes de chiffrement symétriques se divisent à leur tour en chiffrement à flot et chiffrement par blocs.

Chiffrements à flot et chiffrements par blocs

Chiffrements à flot : L'archétype de tous les chiffrements à flot est le célèbre chiffrement de Vernam, appelé aussi *masque−jetable*, introduit et publié pour la première fois par Gilbert Vernam [[98], pp. 394−403] pour protéger les communications télégraphiques pendant la première guerre mondiale. Ce chiffrement peut être vu comme un chiffre de Vigenére, mais où la clef doit être aussi longue que le message à chiffrer, parfaitement aléatoire et jamais réutilisée. Le principe de ce chiffrement est de combiner par l'opération Booléenne XOR (OU EXCLUSIF), bit par bit, le message clair avec une suite de bits aléatoire appelée suite chiffrante, qui correspond à la clef secrète ; cette suite est engendrée par un générateur aléatoire.

Définition 2. *Un générateur aléatoire de bits est un algorithme permettant de produire une suite de bits à la fois statistiquement indépendants et non biaisés*

$$(Pr(0) = Pr(1) = \frac{1}{2}).$$

Si on veut chiffrer un message clair $m = m_1 \dots m_n$ avec la clef $k = k_1 k_2 \dots$, l'opération de chiffrement est simplement, pour tout i : $c_i = m_i \oplus k_i$; le déchiffrement est alors défini par $m_i = c_i \oplus k_i$, où $c = c_1 \dots c_n$ est le message chiffré.

FIGURE 1.2 – Chiffre de Vernam

En 1949, Claude Shannon a prouvé que ce chiffrement est parfaitement sûr, de plus, tout chiffrement parfaitement sûr est nécessairement une variante du chiffre de Vernam. Cependant, le chiffre de Vernam n'est pas particulièrement pratique : non seulement il exige la création d'une suite binaire totalement aléatoire aussi longue que le message clair qui serve qu'une seule fois (puisque si on utilise la même clef deux fois, alors on peut extraire beaucoup d'informations des messages chiffrés), mais il nécessite aussi la transmission de celle−ci entre les deux parties communicantes.

Dans les chiffrements à flot [70, 159], la résolution du problème de la taille et de la transmission de la clef est possible, et cela, si on remplace la clef aléatoire par une clef pseudo−aléatoire générée par un générateur pseudo−aléatoire. C'est−à−dire, qu'à partir d'une clef de petite taille (qui est l'entrée du générateur pseudo−aléatoire), appelée graine, on aura en sortie une suite déterministe ayant des propriétés proches d'une suite aléatoire, qui peut être ainsi utilisée pour chiffrer des messages aussi longs.

Il existe plusieurs types de générateurs pseudo−aléatoires, comme le générateur $X.9.17$, RSA-Rondom, Micali-Schnorr et autres. Les générateurs les plus connus sont les générateurs basés sur les registres à décalage à rétroaction linéaire, LFSR [1]. Ce système produit une suite infinie de bits satisfaisant une relation de récurrence linéaire.

Soit un LFSR possédant L cellules contenant chacune un bit (figure 1.3). Ces L bits forment ce qu'on appelle l'état interne du registre, alors le principe de fonctionnement de ce LFSR est le suivant :

— Les L cellules sont initialisées avec les valeurs $s_{L-1}, \ldots, s_0$.
— à chaque instant t, les L bits du registre sont décalés vers la sortie produisant ainsi le bit le plus ancien du registre, tandis que la cellule la plus à gauche prend comme

1. "Linear Feedback Shift Register" en anglais.

valeur le résultat calculé grâce à la relation de rétroaction :

$$s_{t+L} = \sum_{i=1}^{L} c_i s_{t+L-i},$$

où les coefficients de rétroaction $(c_i)_{1\leq i\leq L}$ sont des éléments de $\mathbb{F}_2$.

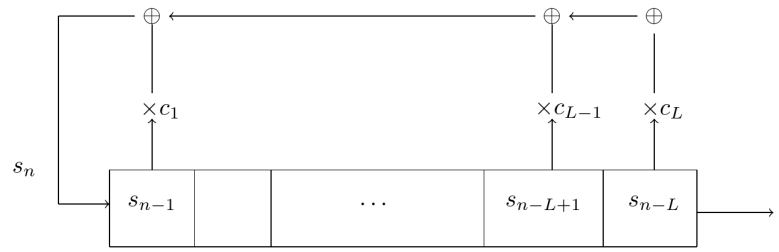

FIGURE 1.3 – Registre à décalage à rétroaction linéaire (LFSR)

La suite $(s_t)_{t\geq 0}$ est une suite périodique, c'est−à−dire qu'il existe un entier positif T tel que $s_{t+T} = s_t$. On associe à un LFSR un polynôme appelé polynôme de rétroaction [140], dont les coefficients sont les coefficients de rétroaction $(c_i)_{1\leq i\leq L}$:

$$P(X) = 1 + \sum_{i=1}^{L} c_i X^i.$$

Si $P(x)$ est de bas degré, alors ce polynôme est appelé le polynôme minimal de s et son degré est la complexité linéaire de la suite, noté $lc(s)$.

En 1969, Massey a montré que l'algorithme de Berlekamp [120], pour le décodage des codes BCH (Bose, Ray−Chaudhuri et Hocquenghem), permet avec une complexité en $O(L^2)$ de retrouver le polynôme de rétroaction d'un LFSR à partir de seulement $2L$ bits consécutifs de la suite produite s et cela, sans la connaissance préalable de L. Pour se mettre à l'abri d'une attaque par l'algorithme de Berlekamp−Massey, la suite chiffrante doit être de complexité linéaire élevée et de grande période en utilisant des LFSRs de taille raisonnable, pour leur simplicité d'implémentation, est alors d'introduire la non−linéarité au système pour le rendre plus complexe. Cette composante peut être obtenue par deux techniques qui sont les registres combinés et les registres filtrés.

— **Registre combiné** : on utilise n LFSRs en parallèle et on combine leurs sorties avec une fonction Booléenne à n variables, dite fonction de combinaison, afin de produire la suite chiffrante. On note $(u_t^i)_{t\geq 0}$ la suite produite par chacun de ces

LFSR pour tout $1 \leq i \leq n$, alors la suite chiffrante $(s_t)_{t\geq 0}$, engendrée par ce générateur, est donnée par :

$$s_t = f(u_t^1, u_t^2, \ldots, u_t^n),\ \forall t \geq 0.$$

La complexité linéaire de ce type de générateur est liée au degré algébrique de la fonction Booléenne utilisée [96, 153]. Le schéma général d'un générateur combiné est donné par la figure (1.4).

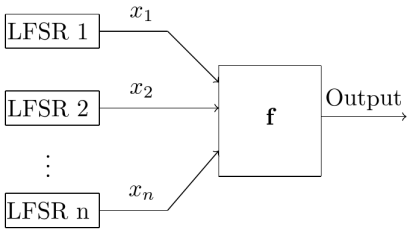

FIGURE 1.4 – Générateur combiné.

— **Registre filtré** : est un générateur pseudo−aléatoire composé d'un unique LFSR de taille n, que l'on filtre par une fonction Booléenne à n variables, ainsi, la suite pseudo−aléatoire générée satisfait la relation suivante :

$$s_t = f(u_{t+\gamma_1}, \ldots, u_{t+\gamma_n}),$$

où $(u_t)_{t\geq 0}$ est la suite produite par le LFSR et $(\gamma_i)_{1\leq i\leq n}$ une suite décroissante d'entiers de $\{0, \ldots, n-1\}$. Le schéma général d'un générateur filtré est donné par la figure 1.5.

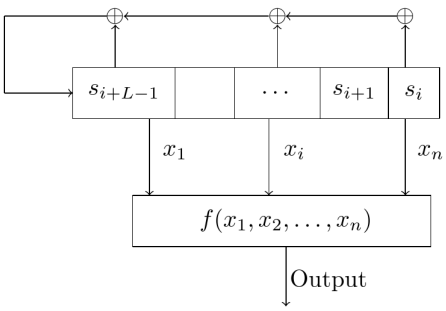

FIGURE 1.5 – Générateur filtré.

Remarque 3. *Dans le but d'accélérer l'opération de chiffrement et de déchiffrement, on peut combiner les n LFSRs par une fonction vectorielle ayant n entrées (les sorties des LFSRs) et m sorties.*

La notion de l'immunité spectrale a été introduite dans [91]. Si l'immunité spectrale d'une fonction Booléenne est faible alors on peut trouver l'état initial d'un générateur filtré dans lequel la fonction Booléenne est utilisée. On reprend la définition, donnée dans [95], de l'immunité algébrique (définition (67)) qui généralise modérément la définition originale.

Définition 4. *(Immunité spectrale) Soient α un élément primitif du groupe multiplicatif $\mathbb{F}_{2^n}^*$ et $\beta \in \mathbb{F}_{2^n}^*$. Pour tout $t \in \mathbb{N}$, on note $z_t = f(\alpha^t \beta)$.*
L'immunité spectrale de la suite binaire $(z_t)_{t\in\mathbb{N}}$ est la complexité linéaire minimale des suites 2^n-aires $(u_t)_{t\in\mathbb{N}}$ qui vérifient l'une des équations suivantes :

$$z_t u_t = 0, \text{ pour tout } t \in \mathbb{N},$$
$$(1 + z_t)u_t = 0, \text{ pour tout } t \in \mathbb{N},$$

où la complexité linéaire d'une suite binaire est définie comme la longueur du LFSR le plus court qui génère la suite.

Définition 5. *(Complexité linéaire) La complexité linéaire d'une suite infinie $z = (z_t)_{t\geq 0}$ de bits est :*

- *$lc(z) = 0$ si $z_i = 0$ pour tout i ;*
- *$lc(z) = \infty$ si aucun registre à décalage linéaire ne produit z ;*
- *$lc(z) = n$ si le plus petit registre à décalage linéaire produisant z a pour longueur n.*

Définition 6. *[167](Annulateur de suite binaire) Pour une suite binaire $z = (z_t)_{t\geq 0}$ de période T, la suite binaire $\boldsymbol{a} \neq 0$ de période T qui satisfait $\boldsymbol{a}\, z = 0$ est dite un annulateur de z.*

Alors, l'immunité spectrale peut être aussi définie comme suit,

Définition 7. *[95](Immunité spectrale) L'immunité spectrale d'une suite binaire $z = (z_t)_{t\in\mathbb{N}}$, notée $SI(z)$, est la complexité linéaire minimale de tous les annulateurs binaires non$-$nuls de z,*

$$SI(z) = \min_{\{\boldsymbol{a}\neq 0, \boldsymbol{a}\, z=0 \text{ ou } a\,(z+1)=0\}} lc(a).$$

Chiffrements par blocs : L'autre grande famille d'algorithmes de chiffrement symétriques largement utilisée actuellement est la famille des chiffrements par blocs. Dans [123], Matsui a introduit les cryptanalyses linéaires pour les chiffrements par blocs, et dans ces derniers, les bits du message sont premièrement divisés en blocs de n bits, où chaque bloc x_i de n bits est chiffré pour obtenir un bloc y_i de même taille. Cette opération est paramétrée par une clef secrète, appelée sous−clef, de longueur k qui change pour chaque tour. Ensuite, on remet ensemble les blocs chiffrés indépendamment (ou presque), pour obtenir enfin le message chiffré. Les bits de chiffrement obtenus sont transmis vers le canal.
De même, pour le déchiffrement on fragmente à nouveau le message en blocs de n bits, ensuite, on applique l'algorithme de déchiffrement pour chaque bloc. Les tailles des blocs les plus répandues sont 64 et 128 bits, mais certains contextes particuliers acceptent des blocs de 32, 48 ou 96 bits.

Parmi les exemples classiques les plus connus de chiffrement par blocs on trouve IDEA (International Data Encryption Algorithm, $k = 128, n = 64$) et RC5 (RC5−32 pour des mots de 32 bits), mais les exemples les plus célèbres sont le DES [82] (Data Encryption Standard) et le AES [83] (Advanced Encryption Standard), qui est le standard actuel (FIPS 197) de chiffrement par blocs [62].

— **DES** : jusqu'au début de l'an 2000, ce système de chiffrement à clef secrète est resté le plus célèbre et le plus utilisé. Avec le temps et les progrès de l'informatique, il est désormais possible, même avec des moyens modestes, de percer les messages chiffrés par DES.
— **AES** : choisi en octobre 2000, cet algorithme, appelé RIJNDAEL, qui a été conçu par Rijmen et Daemen [62], supporte des clefs de tailles : 128, 192 et 256 bits et opère sur des blocs de taille fixe de 128 bits. La taille de la clef utilisée spécifie le nombre de cycles de transformation qui convertissent l'entrée.

Pour une clef de taille 128 bits, RIJNDAEL génère 10 sous−clefs de 128 bits. Ainsi, le premier tour est précédé d'un "*ou exclusif*" bit à bit entre le message clair et la sous-clef numéro 0 ; de même, le dernier tour est légèrement différent des tours précédents. On y retrouve trois étapes, conformément aux principes fondamentaux de confusion et de diffusion énoncées par Shannon. La première étape, consiste à appliquer à chaque bit de l'entrée la même boîte−S (définition (2.3)). Ensuite, une permutation linéaire qui assure la diffusion, puis la sous−clef du tour est ajoutée bit à bit au résultat.

Les chiffrements à flot possèdent cependant des avantages considérables par rapport aux chiffrements par blocs, notamment quant au temps de traitement : le chiffrement et le déchiffrement sont en temps réel et simultanés. Et alors le processus de déchiffrement ne propage pas les erreurs de transmission.
Si les chiffrements à flot sont souvent plus rapides que les chiffrements par blocs, leur sécurité est souvent plus faible.

Un système de chiffrement idéal devrait résoudre simultanément tous ces problèmes : utiliser des clefs publiques, assurer le secret, l'authentification et l'intégralité, le tout le plus rapidement possible. Les systèmes conventionnels comme le DES sont très efficaces mais utilisent des clefs privées ; les systèmes à clef publique peuvent assurer l'authentification mais sont inefficaces pour le chiffrement de grandes quantités de données car trop coûteux. Cette complémentarité a motivé le développement de protocoles cryptographiques *hybrides*, à l'instar de PGP (Pretty Good Privacy). C'est−à−dire, on combine les deux systèmes de chiffrement, d'abord, pour échanger la clef secrète on utilise le chiffrement asymétrique, ensuite, on passe à un chiffrement symétrique pour le reste de la communication.

Chapitre 2

Les fonctions Booléennes et vectorielles

Le but de ce chapitre est de présenter un aperçu sur les aspects les plus intéressants des fonctions Booléennes et vectorielles en cryptographie.

Nous commencerons ce chapitre par faire appel aux outils mathématiques nécessaires, également un certain nombre de propriétés que possèdent ces outils. Ensuite, dans la deuxième partie de ce chapitre nous donnerons quelques−unes des représentations des fonctions Booléennes dans le contexte de la cryptographie : la forme algébrique normale, la représentation polynomiale univariée et la représentation trace. Ainsi que l'outil puissant appelé *la transformée de Walsh*. Cet outil va servir essentiellement à caractériser les critères cryptographiques des fonctions Booléennes. Dans la troisième partie de ce chapitre, nous discutons les propriétés et les définitions des fonctions vectorielles. Nous donnerons quelques−unes des représentations des fonctions vectorielles dans le contexte de la cryptographie : la forme algébrique normale, la représentation polynomiale univariée et la représentation trace. Ainsi que la transformée de Walsh.

2.1 Résultats préliminaires

Avant de rentrer dans le vif du sujet, nous avons besoin de définir plusieurs notations autour des vecteurs de n bits que l'on va constamment manipuler tout au long de cette thèse.

Un mot sur les éléments de $\mathbb{F}_2^n$

Soit $\mathbb{F}_q$ ($q = p^n$ la puissance d'un nombre premier) le corps fini à q éléments. Le corps que nous utiliserons le plus est sans conteste le corps fini à 2 éléments $\mathbb{F}_2$, qui est l'ensemble formé des bits 0 et 1.

Rappelons que l'opération X0R notée $\oplus$, ou simplement $+$ s'il n'y a pas un risque de confusion, n'est rien d'autre que l'addition dans le corps à deux éléments $\mathbb{F}_2$. Elle est définie par :

$$0+0=0,\ 0+1=1,\ 1+0=1,\ 1+1=0.$$

Pour un entier naturel $n \in \mathbb{N}$, soit $\mathbb{F}_2^n = \mathbb{F}_2 \times \mathbb{F}_2 \times \ldots \times \mathbb{F}_2$ (n fois) l'espace vectoriel des vecteurs binaires de longueur n.

Un vecteur de longueur n (ou bien de n bits), sera toujours noté x ou y. On aura de temps en temps à manipuler les bits d'un tel vecteur, ils seront notés par la même lettre que le vecteur, et indexés de 1 à n, par exemple : $x = (x_1, x_2, \ldots, x_n)$. On note par $\overline{x}$ le complement de x, c'est−à−dire $\overline{x} = (\overline{x_1}, \ldots, \overline{x_n}) = (1, \ldots, 1) + (x_1, x_2, \ldots, x_n)$.

Nous utiliserons deux notions importantes sur les vecteurs, qui sont le support et le poids de Hamming. Ils sont définis pour des vecteurs de longueur quelconque par :

Définition 8. *Soit un vecteur $x = (x_1, x_2, \ldots, x_n)$ de $\mathbb{F}_2^n$. Le support de x que nous noterons $supp(x)$ est l'ensemble des positions des bits non nuls de x :*

$$supp(x) = \{i,\ x_i = 1\},\ 1 \leq i \leq n.$$

Le poids de Hamming de x est égal à son nombre de composantes non nulles. On le notera $wt(x)$. C'est−à−dire :

$$wt(x) = \sum_{i=1}^{n} x_i \overset{def}{=} |Supp(x)| = \texttt{cardinal du support de x.}$$

Nous aurons également besoin d'ordonner les éléments de $\mathbb{F}_2^n$. L'ordre que nous utiliserons est l'ordre lexicographique inverse et nous le désignerons par ordre usuel. En particulier les notations $<$, $\leq$, $>$, $\geq$ entre éléments de $\mathbb{F}_2^n$ se réfèreront toujours à cet ordre. Il est défini par :

Définition 9. *(Ordre lexicographique inverse) Soit* $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$ *et* $x \neq y$, *alors*

$$\exists i_0,\ 1 \leq i_0 \leq n\ :\ x_{i_0} < y_{i_0} \text{ et } \forall i < i_0,\ x_i = y_i.$$

Cet ordre présente de nombreuses propriétés qui vont le rendre très utile. En particulier, si l'on voit un $n-$uplet de $\mathbb{F}_2$ comme la représentation binaire d'un entier, l'ordre lexicographique inverse est le même que l'ordre usuel sur les entiers. Il faut pour cela choisir la convention suivante :

Définition 10. *(L'écriture* $2-$*adique) On associe à un vecteur* $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$ *un entier de* 0 *à* $2^n - 1$ *par*

$$\sum_{i=1}^{n} x_i 2^{i-1}.$$

Nous aurons également besoin de définir l'ordre lexicographique partiel, que nous désignerons par $\preceq$ dans $\mathbb{F}_2^n$.

Définition 11. *[137](Ordre lexicographique partiel) Soit* $u = (u_1, \ldots, u_n) \in \mathbb{F}_2^n$ *et* $x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$, *on définit l'ordre lexicographique partiel* $x \preceq y$ *par :*

$$x_i \leq y_i, \text{ pour tout } i \text{ tel que } 1 \leq i \leq n,$$

au sens arithmétique habituel ($0 \leq 0$, $0 < 1$, $1 \leq 1$).

Lemme 12. *[137] Soit* $x = (x_1, x_2, ..., x_n) \in \mathbb{F}_2^n$, *alors il y a* $2^{wt(x)}$ *vecteurs* $u = (u_1, \ldots, u_n)$ *tels que* $u \preceq x$,

où $wt(x)$ est le poids de Hamming de x.

2.2 Généralités sur les fonctions Booléennes

Une fonction Booléenne à n variables est une fonction de l'espace vectoriel $\mathbb{F}_2^n$ à valeurs dans l'espace vectoriel $\mathbb{F}_2$. On notera par $\mathcal{B}_n$ l'ensemble des fonctions Booléennes à n variables (contient exactement 2^{2^n} éléments), et on munit cet ensemble des opérations induites par les opérations du corps $\mathbb{F}_2^n$, alors l'ensemble des fonctions Booléennes sur $\mathbb{F}_2^n$ est un espace vectoriel de dimension 2^n sur $\mathbb{F}_2$.

n	4	5	6	7	8
Nombre de fonctions Booléennes	2^{16}	2^{32}	2^{64}	2^{128}	2^{256}
$\approx$	$6 \cdot 10^4$	$4 \cdot 10^9$	10^{19}	10^{38}	10^{77}

TABLE 2.1 – Nombre de fonctions Booléennes à n variables

Définition 13. *[31](Vecteur des valeurs) Le vecteur des valeurs, ou table de vérité, d'une fonction Booléenne f est le vecteur binaire $[f(0,\ldots,0),\ldots,f(1,\ldots,1)]$ de longueur 2^n composé des valeurs de la fonction : $f(x)$ pour $x \in \mathbb{F}_2^n$.*

Exemple 14. *La table (2.2) donne le vecteur des valeurs d'une fonction Booléenne f à 3 variables.*

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
$f(x_1,x_2,x_3)$	0	1	1	0	0	1	1	1

TABLE 2.2 – La table de vérité d'une fonction Booléenne à 3 variables

Le support et le poids de Hamming d'une fonction Booléenne sont définis comme suit :

Définition 15. *(Support) On appelle support d'une fonction Booléenne f l'ensemble des vecteurs de F_2^n pour lesquels la valeur par la fonction f est non nulle, c'est−à−dire :*

$$supp(f) = \{x \in \mathbb{F}_2^n,\ f(x) = 1\}.$$

Le cardinal du support de f est appelé poids de Hamming de f et noté $wt(f)$.

Dans l'exemple (14), on a $supp(f) = \{(0,0,1),(0,1,0),(1,0,1),(1,1,0),(1,1,1)\}$.

On munit $\mathcal{B}_n$ de la métrique de Hamming. On définit la distance de Hamming, notée $d(f,g)$, entre f et g comme le nombre de différences entre leurs vecteurs des valeurs respectifs, i.e. $wt(f+g)$.

Les fonctions Booléennes les plus élémentaires après les fonctions 0 et 1 sont les fonctions coordonnées $x = (x_1, x_2, \ldots, x_n) \to x_i$, que l'on notera x_i. Le produit de ces fonctions sont aussi des fonctions Booléennes appelées les *monômes*. On utilise la notation suivante pour désigner les monômes.

Notation 16. *Pour tout* $u = (u_1, \ldots, u_n) \in \mathbb{F}_2^n$, x^u *désigne le monôme dans l'anneau* $\mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2 + x_1, \ldots, x_n^2 + x_n)$ *défini par :*

$$x^u = \prod_{i=1}^{n} x_i^{u_i}.$$

Le degré de ce monôme est le poids de Hamming du mot u. L'ensemble des monômes constitue une base de l'espace vectoriel des fonctions Booléennes sur $\mathbb{F}_2^n$, et nous pouvons alors représenter une fonction Booléenne comme un polynôme multivarié.

2.2.1 Représentations des fonctions Booléennes

Il existe de multiples représentations des fonctions Booléennes [59]. Nous allons nous intéresser à celle que nous utiliserons par la suite. Ainsi qu'aux outils importants, utilisés dans l'étude des fonctions Booléennes en cryptographie.

Représentation sous forme algébrique normale : On introduit la représentation la plus utilisée d'une fonction Booléenne en cryptographie et codage, à savoir, la représentation de polynôme multivarié à n variables sur $\mathbb{F}_2$. On appelle aussi cette représentation la Forme Algébrique Normale ou ANF (*Algebraic Normal Form*).

Définition 17. *[31](La forme algébrique normale) L'écriture (unique) d'une fonction Booléenne* f *comme combinaison linéaire à coefficients dans* $\mathbb{F}_2$ *de monômes s'appelle la forme algébrique normale (ANF). Elle s'écrit :*

$$f(x) = \sum_{u \in \mathbb{F}_2^n} a_u x^u,$$

où, pour tout u*, le coefficient* a_u *est dans* $\mathbb{F}_2$*, et* $f(x) \in \mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2+x_1, \ldots, x_n^2+x_n)$.

La *transformée de Möbius* de f est définie comme suit :

Théorème 18. *[97] La transformée de Möbius d'une fonction Booléenne* f *à* n *variables est la fonction Booléenne obtenue par la formule suivante :*

$$a_u = \sum_{x \in \mathbb{F}_2^n,\, x \preceq u} f(x).$$

La transformée de Möbius est involutive, c'est−à−dire qu'elle est sa propre inverse. Le théorème suivant a été introduit par Hall dans [92].

Théorème 19. *Soit f et g deux fonctions Booléennes à n variables et soit $\mathbb{F}_2^n$ partiellement ordonné par l'ordre lexicographique partiel $\preceq$, alors les déclarations suivantes sont équivalentes.*

(i) $f(u) = \sum_{v \preceq u} g_v$ *pour tout* $v \in \mathbb{F}_2^n$.

(ii) $g(u) = \sum_{v \preceq u} f_v$ *pour tout* $v \in \mathbb{F}_2^n$.

Preuve. En remplaçant le membre à droite de (ii) par (i), on obtient,

$$\sum_{v \preceq u} f(v) = \sum_{v \preceq u} \sum_{w \preceq v} g(w) = \sum_{w \preceq v \preceq u} g(w) = \sum_{w \preceq u} 2^{wt(u-v)} g(w) = g(u).$$

où la dernière égalité est une conséquence du fait que $\mathbb{F}_2$ est de caractéristique 2. Ainsi, (i) implique (ii). En remplaçant f par g nous aurons (ii) implique (i). ■

Définition 20. *[31](Degré algébrique d'une fonction Booléenne) Le degré algébrique d'une fonction Booléenne f, noté* $\deg(f)$*, est le nombre maximal de variables distinctes apparaissant dans les monômes de f représentée sous forme algébrique normale.*

Le degré algébrique de f correspond au poids de Hamming maximal du vecteur u tel que le coefficient a_u de la forme algébrique normale de f est non-nul :

$$\deg(f) = \max_{u \in F_2^n,\, a_u \neq 0} wt(u).$$

Une fonction Booléenne de degré algébrique 1 est appelée fonction *affine* et lorsque sa valeur en 0 est nulle, la fonction est alors *linéaire*.

En reprenant l'exemple (14), on a,

$$\begin{aligned}
a_{000} &= f(000) = 0 \\
a_{001} &= f(000) + f(001) = 1 \\
a_{010} &= f(000) + f(010) = 1 \\
a_{011} &= f(000) + f(001) + f(010) + f(011) = 0 \\
a_{100} &= f(000) + f(100) = 0 \\
a_{101} &= f(000) + f(001) + f(100) + f(101) = 0 \\
a_{110} &= f(000) + f(010) + f(100) + f(110) = 0 \\
a_{111} &= \sum_{x \in \mathbb{F}_2^3} f(x) = wt(f) \mod 2 = 1
\end{aligned}$$

L'ANF de f est $f(x_1, x_2, x_3) = x_2 + x_3 + x_1x_2x_3$, et elle est de degré algébrique 3.

La dérivée d'une fonction Booléenne est définie comme suit :

Définition 21. *(Dérivée) Soit f une fonction Booléenne à n variables, pour tout $a \in \mathbb{F}_2^n$, on définit $D_a f$, la dérivée de f relativement au vecteur a par :*

$$D_a f(x) = f(x) \oplus f(x+a).$$

Représentation univariée et la représentation trace : Par définition, il existe un unique isomorphisme entre les éléments de l'espace vectoriel $\mathbb{F}_2^n$ et le corps $\mathbb{F}_{2^n}$ d'ordre 2^n. L'espace vectoriel $\mathbb{F}_2^n$ peut être muni d'une structure du corps $\mathbb{F}_{2^n}$ et alors $\mathbb{F}_2$ est un sous−corps de $\mathbb{F}_{2^n}$.

Comme il a été montré par Carlet dans [31], toute fonction Booléenne sur $\mathbb{F}_{2^n}$ est un cas particulier d'une fonction vectorielle de $\mathbb{F}_{2^n}$ dans $\mathbb{F}_{2^n}$, alors elle admet une unique représentation comme polynôme à une variable sur $\mathbb{F}_{2^n}$, dite *la représentation univariée* de f.

$$f(x) = \sum_{i=0}^{2^n-1} \delta_i x^i, \ \delta_i \in \mathbb{F}_{2^n}.$$

Ce polynôme univarié est la représentation univariée d'une fonction Booléenne si et seulement si $(\sum_{i=0}^{2^n-1} \delta_i x^i)^2 = \sum_{i=0}^{2^n-1} \delta_i^2 x^{2i} [\mod (x^{2^n}+x)]$, autrement dis, $\delta_0, \delta_{2^n-1} \in \mathbb{F}_2$ et, pour tout $i \in [1,\, 2^n-2]$, $\delta_{2i} = \delta_i^2$ où l'indice $2i$ est calculé $\mod (2^n-1)$.

Autres représentations : Il existe d'autres représentations des fonctions Booléennes dont la représentation *trace* d'un polynôme à coefficients dans $\mathbb{F}_{2^n}$, cette représentation permet de travailler avec des valeurs dans le corps $\mathbb{F}_{2^n}$, et ainsi d'utiliser la structure plus aisément.

La fonction $tr_n(u) = u + u^2 + u^{2^2} + \ldots + u^{2^{n-1}}$, qui est définie sur $\mathbb{F}_2^n$, est une forme linéaire sur $\mathbb{F}_2$ qui satisfait $(tr_n(u))^2 = tr_n(u^2) = tr_n(u)$.

La fonction $(u, v) \longrightarrow tr_n(uv)$ est le produit intérieur dans $\mathbb{F}_{2^n}$ (autrement dit, elle est symétrique et pour tous les $v \neq 0$, la fonction $u \longrightarrow tr_n(uv)$ est une forme linéaire non nulle sur $\mathbb{F}_{2^n}$). Chaque fonction Booléenne peut être écrite sous la forme $f(x) = tr_n(F(x))$ où F est une application de $\mathbb{F}_{2^n}$ dans $\mathbb{F}_{2^n}$. Ainsi, chaque fonction Booléenne peut être aussi

représentée sous la forme :

$$f(x) = tr_n(\sum_{i=0}^{2^n-1} \beta_i x^i),$$

où $\beta_i \in \mathbb{F}_{2^n}$. Une telle représentation n'est pas unique.

2.2.2 Transformée de Walsh

La transformée de Walsh (ou Walsh−Hadamard) est un outil assez puissant pour analyser les propriétés cryptographiques des fonctions Booléennes. C'est en fait une transformée de Fourier discrète qui s'applique aux fonctions de $\mathbb{F}_2^n$ à valeurs dans $\mathbb{C}$. Du point de vue espace vectoriel, les caractères sont obtenus à partir du produit scalaire sur $\mathbb{F}_2^n$.
La transformée de Walsh représente une corrélation entre les fonctions Booléennes et les fonctions affines et elle est liée aux attaques sur les systèmes de chiffrement à flot utilisant les LFSRs.

La transformée de Fourier discrète d'une fonction Booléenne, en $a \in \mathbb{F}_2^n$, est définie dans ce contexte par :

Définition 22. *(Transformée de Fourier discrète) La transformée de Fourier discrète d'une fonction Booléenne f à n variables est définie par*

$$F_f(u) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{x \cdot u},$$

où $x \cdot u$ désigne le produit scalaire usuel des vecteurs x et u ($x \cdot u = x_1 \cdot u_1 + \ldots + x_m \cdot u_m$).

La transformée de Walsh est alors la fonction W_f définie sur $\mathbb{F}_2^n$ et à valeurs dans $\mathbb{Z}$ donnée par :

Définition 23. *(Transformée de Walsh) La transformée de Walsh d'une fonction Booléenne f est la transformée de Fourier discrète de sa fonction signe $(-1)^f$,*

$$\forall u \in \mathbb{F}_2^n,\ W_f(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+u \cdot x}.$$

L'ensemble des valeurs de la transformée de Walsh est appelée le *spectre* de Walsh.

$$\{W_f(u),\ u \in \mathbb{F}_2^n\}.$$

La relation entre la transformée de Fourier discrète et la transformée de Walsh [86] est donnée comme suit :

$$W_f(a) = 2^n\delta_0(a) - 2F_f(a),$$

où δ_0 est la *fonction Kronecker*, telle que $\delta_0(u) = 1$ pour tout vecteur $u \neq 0$ et $\delta_0(u) = 0$ ailleurs.

Par exemple, le spectre de Walsh de la fonction f de l'exemple (14) est le suivant :

u	(0,0,0)	(0,0,1)	(0,1,0)	(0,1,1)	(1,0,0)	(1,0,1)	(1,1,0)	(1,1,1)
$W_f(u)$	-2	2	2	6	2	-2	-2	2

TABLE 2.3 – Le spectre de Walsh d'une fonction Booléenne à 3 variables

Proposition 24. *[117] égalité de Parseval :*

$$\sum_{x\in\mathbb{F}_2^n} W_f(x)^2 = 2^{2n}.$$

Enfin, rappelons une formule qui est à l'origine de nombreux résultats et qui nous serait très utile dans le chapitre (6). Ce résultat est un lien entre la transformée de Walsh d'une fonction Booléenne f et la restriction de f à un sous−espace arbitraire de $\mathbb{F}_2^n$ que Lechner [[108], théorème (2.6), page 147] appelle la *formule de sommation de Poisson*. La formule de sommation de Poisson a été utilisée pour prouver de nombreuses propriétés cryptographiques dans [36, 108, 114], et plus tard dans [25, 26].

Nous rappelons que pour toute fonction Booléenne f et tout sous−espace vectoriel $E \subset \mathbb{F}_2^n$, la restriction de f, notée f_v, est la fonction Booléenne à $wt(v)$ variables sur E, définie comme $f_v(x) = f(v + x)$, pour tout $x \in E$.

Théorème 25. *[108, 117](Formule de sommation de Poisson). Soit la fonction $f : \mathbb{F}_2^n \to \mathbb{R}$ et W_f sa transformée de Walsh. Soit S un sous−espace arbitraire de $\mathbb{F}_2^n$ et $S^\perp$ un annulateur de S, c'est−à−dire,*

$$S^\perp = \{x \in \mathbb{F}_2^n : x \cdot s = 0 \, \textit{pour tout} \, s \in S\}.$$

Alors,

$$\sum_{u\in S} W_f(u) = 2^{\dim S} \sum_{u\in S^\perp} f(u).$$

Soit $E = \{u \in \mathbb{F}_2^n,\ u \preceq v\}$ un sous−espace vectoriel de $\mathbb{F}_2^n$, alors

$$W_{1_E}(u) = |E|1_{E^\perp}(u). \tag{2.1}$$

En utilisant la formule (2.1), nous avons le corollaire suivant :

Corollaire 26. *[46] Pour toute fonction Booléenne $f \in \mathbb{F}_2^n$, pour tout sous−espace vectoriel E, nous avons :*

$$\sum_{u \in a+E} (-1)^{b\cdot u} W_f(u) = |E|(-1)^{a\cdot b} \sum_{x \in b+E^\perp} (-1)^{a\cdot x + f(u)}.$$

Le corollaire suivant a été découvert indépendamment par Duvall et Mortick [74].

Corollaire 27. *Soit f une fonction Booléenne à n variables, alors*

$$\sum_{u \in E} W_f(u) = 2^{wt(v)} \sum_{u \preceq \overline{v}} f(u) = 2^n - 2^{wt(v)+1} wt(f_{\overline{v}}).$$

2.3 Généralités sur les fonctions vectorielles

Les boîtes de substitutions (boîtes−S) sont une partie fondamentale des systèmes de chiffrement par blocs, non seulement elles garantissent la confidentialité des données en chiffrant l'information, mais elles jouent un rôle très important dans la robustesse des systèmes de chiffrement, étant la seule source de non−linéarité. Une boîte−S remplace un petit bloc de bits d'entrée $(x_1, \ldots, x_n)$ par un autre bloc de bits de sortie $(y_1, \ldots, y_m)$, où le bloc $(y_1, \ldots, y_m)$ est la juxtaposition des sorties des m fonctions Booléennes qui ont en commun l'entrée $(x_1, \ldots, x_n)$.

Mathématiquement une boîte−S est une fonction vectorielle, ou une (n, m)−fonction, c'est−à−dire, une fonction de l'espace vectoriel $\mathbb{F}_2^n$ dans l'espace vectoriel $\mathbb{F}_2^m$, où n et m sont deux entiers positifs. Alors cette fonction peut être représentée par le vecteur $(f_1, f_2, \ldots, f_m)$, où, pour $1 \leq i \leq m$, les f_i sont des fonctions Booléennes de $\mathbb{F}_2^n$ dans $\mathbb{F}_2$, qu'on appelle *fonctions coordonnées*.

Si F est une (n, m)−fonction, alors les fonctions composantes de $F = (f_1, f_2, \ldots, f_m)$ sont les fonctions $v \cdot F = v_1 f_1 + \ldots + v_m f_m$, où $v \in \mathbb{F}_2^m$, $v \neq 0$.

Exemple 28. *Le tableau suivant définit une fonction vectorielle de $\mathbb{F}_2^4$ dans $\mathbb{F}_2^4$ (avec une notation hexadécimal).*

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$F(x)$	e	b	c	6	f	d	0	8	7	3	9	a	4	2	5	1
$f_1(x)$	1	1	1	0	1	1	0	1	0	0	1	1	0	0	0	0
$f_2(x)$	1	0	1	1	1	1	0	0	1	0	0	0	1	0	1	0
$f_3(x)$	1	1	0	1	1	0	0	0	1	1	0	1	0	1	0	0
$f_4(x)$	0	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1

TABLE 2.4 – Exemple d'une $(4,4)-$fonction

La transformée de Möbius (définition (18)) nous permet de donner l'ANF des quatre fonctions coordonnées de cette $(4,4)-$*fonction :*

$$f_1(x) = 1 + x_4 + x_1x_2 + x_2x_3 + x_2x_4 + x_1x_2x_4.$$
$$f_2(x) = 1 + x_1 + x_2x_4 + x_2x_3 + x_1x_3 + x_1x_2 + x_1x_3x_4 + x_1x_2x_3.$$
$$f_3(x) = 1 + x_2 + x_1x_2 + x_1x_3 + x_3x_4 + x_2x_3x_4.$$
$$f_4(x) = x_1 + x_3 + x_4 + x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_1x_2x_3 + x_1x_3x_4.$$

2.3.1 Représentations des fonctions vectorielles

Plusieurs notions sur les fonctions Booléennes ont été généralisées aux fonctions vectorielles dont la notion de la forme algébrique normale.

En cryptographie, la représentation la plus utilisée d'une $(n,m)-$fonction est ce qu'on appelle la forme algébrique normale (ANF). Soit F une $(n,m)-$fonction, cependant, chacune de ses coordonnées est une fonction Booléenne, donc elle admet une représentation sous forme algébrique normale, et alors la fonction F peut être représentée d'une façon unique sous la forme algébrique normale.

$$F(x_1, \ldots, x_n) = \sum_{u \in \mathbb{F}_2^n} c(u)(\prod_{i=1}^{n} x_i^{u_i}),$$

où $c(u) \in \mathbb{F}_2^m$.

Définition 29. *[59](Degré algébrique) Le degré algébrique d'une* $(n,m)-$*fonction, noté* $\deg(F)$*, est défini par le degré de son ANF*

$$\deg(F) = \max\{|supp(u)|,\, c(u) \neq (0,\ldots,0) \in \mathbb{F}_2^m,\, u \in \mathbb{F}_2^n\}.$$

Le degré algébrique de F est égale au degré algébrique maximal de ces fonctions co-

ordonnées, par conséquent, le degré algébrique maximal de ces fonctions composantes.

Une autre notion du degré est le degré algébrique minimal de toutes les fonctions composantes de F, appelé le *degré minimal* et défini comme suit :

$$\deg_{min}(F) = \min\{\deg(v \cdot F) : v \in \mathbb{F}_2^m, v \neq 0\} \leq \deg(F).$$

Représentation polynomiale univariée : Toute $(n, m)-$fonction F peut être vu en tant qu'une fonction de $\mathbb{F}_{2^n}$ dans lui−même, puisque $\mathbb{F}_{2^m}$ est un sous−corps de $\mathbb{F}_{2^n}$. Ainsi, il existe une seconde représentation des $(n, m)-$fonctions quand $n = m$ ou quand m est un diviseur de n. Toute $(n, n)-$fonction F admet une unique représentation appelée *représentation polynomiale univariée* sur $\mathbb{F}_{2^n}$.

$$F(x) = \sum_{i=0}^{2^n-1} \delta_i x^i, \ \delta_i \in \mathbb{F}_{2^n}, \tag{2.2}$$

où $\delta \in \mathbb{F}_{2^n}$, pour $1 \leq i \leq 2^n - 1$.

Dans ce cas, les fonctions composantes de la fonction F sont les fonctions de la forme $tr_1^n(v \cdot F(x))$, où $v \in \mathbb{F}_{2^n}^*$ et $tr_1^n = \sum_{i=0}^{n-1} x^{2^i}$ est la fonction trace de $\mathbb{F}_{2^n}$ dans $\mathbb{F}_2$.

Le degré algébrique de F est égale au $\deg(F) = \max\{wt(j), \ \delta_j \neq 0, \ 0 \leq j \leq 2^n - 1\}$. C'est−à−dire, le nombre de coefficients non−nuls j_s dans l'expression binaire de j :

$$\deg(F) = \max_{j=0,\ldots,2^n-1, \, \delta_j \neq 0} |\{s, \ s \neq 0 / j = \sum_{s=0}^{n-1} j_s 2^s\}|.$$

Par l'interpolation de Lagrange, toute $(n, n)-$fonction F peut être écrite comme suit :

$$F(x) = \sum_{\alpha \in \mathbb{F}_{2^n}} F(\alpha)(1 + (x + \alpha)^{2^n-1}).$$

La simplification de cette expression nous mène à l'obtention de la représentation (2.2) et alors les coefficients $\delta_0 = F(0)$ et $\delta_i = \sum_{\alpha \in \mathbb{F}_{2^n}} F(\alpha)\alpha^{-i}$ pour tout $1 \leq i \leq 2^n - 1$.

En reprenant l'exemple (28), la représentation univariée de la $(n, m)-$fonction F est : $F(x) = (\alpha^2 + \alpha + 1)x^{13} + (\alpha + 1)x^{12} + (\alpha^2 + 1)x^{11} + (\alpha^2 + \alpha)x^{10} + \alpha^2 x^9 + (\alpha + 1)x^8 + (\alpha^2 + \alpha + 1)x^7 + \alpha^3 x^5 + (\alpha^3 + \alpha^2 + 1)x^4 + (\alpha^3 + \alpha)x^3 + \alpha^3 x^2 + (\alpha^2 + 1)x + \alpha^3 + \alpha^2 + \alpha$.

Cette représentation des $(n, n)-$fonctions peut être calculée à l'aide de la transformée de Fourier discrète [18, 88, 117].

Plus généralement, si m est un diviseur de n, alors F admet une représentation polynomiale univariée de la forme :

$$F(x) = tr_m^n(\sum_{j=0}^{2^n-1} \delta_j x^j),$$

où $tr_m^n(x) = x + x^{2^m} + x^{2^{2m}} + x^{2^{3m}} + \ldots + x^{2^{n-m}}$ est la fonction trace de $\mathbb{F}_{2^n}$ dans $\mathbb{F}_{2^m}$.

2.3.2 Transformée de Walsh

En choisissant un produit scalaire dans $\mathbb{F}_2^n$ et un produit scalaire dans $\mathbb{F}_2^m$, alors la transformée de Walsh d'une $(n, m)-$fonction F, notée W_F, calculée en $(u, v) \in \mathbb{F}_2^n \times \mathbb{F}_2^m$ est la transformée de Walsh de la fonction Booléenne $v \cdot F$ en u, c'est−à−dire :

$$W_F(u, v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{v \cdot F(x) + u \cdot x}.$$

Dans le cas où la fonction F est représentée sous forme polynomiale univariée, la transformée de Walsh est donnée par :

$$W_F(u) = \sum_{x \in \mathbb{F}_{2^n}} (-1)^{F(x) + tr_1^m(ux)}.$$

Remarque 30. *Si on note par G_F le graphe $\{(x, y) \in \mathbb{F}_2^n \times \mathbb{F}_2^m;\ y = F(x)\}$ de F, et par 1_{G_F} la fonction Booléenne dite fonction indicatrice, qui vaut 1 sur le graphe et 0 ailleurs, alors la transformée de Walsh d'une $(n, m)-$fonction est la transformée de Fourier de l'indicatrice du graphe de F, c'est−à−dire $W_F(u, v) = \widehat{1_{G_F}}(u, v)$.*

Chapitre 3

Fonctions Booléennes et vectorielles en cryptographie

Tandis que la sécurité des systèmes de chiffrement asymétriques repose sur des problèmes difficiles, pour lesquels on ne connaît pas d'algorithmes de résolution en temps polynomiale, c'est−à−dire que ce sont des problèmes $\mathcal{NP}-$difficiles, comme la factorisation, le logarithme discret ou la recherche du plus petit vecteur dans un réseau euclidien, la mesure de la sécurité des systèmes de chiffrement symétriques repose soit sur des algorithmes de théorie de l'information, soit sur l'étude de la complexité des meilleures attaques connues à ce jour.

Nous commencerons ce chapitre par l'introduction du contexte d'utilisation des fonctions Booléennes et vectorielles en cryptographie, puis, nous donnerons l'ensemble des définitions des critères cryptographiques de ces fonctions comme : l'équilibre, le degré algébrique, la non−linéarité et l'immunité algébrique, ensuite, un état de l'Art sur les attaques algébriques est traité, enfin, nous présenterons la notion de l'immunité algébrique des fonctions Booléennes et vectorielles, qui est un critère nécessaire pour résister à ces attaques.

3.1 Contexte de l'utilisation

Les fonctions Booléennes sont une partie fondamentale des systèmes cryptographiques symétriques, et encore plus dans les systèmes de chiffrement à flot, ou la fonction Booléenne peut être utilisée comme fonction de filtrage d'un LFSR, ou fonction de combinaison de plusieurs LFSRs.

Afin de résister aux nombreuses attaques connues et aux attaques futures, les fonctions Booléennes utilisées doivent satisfaire aux différents critères de façon simultanée [31, 70]. Les principaux critères sont :

— Pour éviter une dépendance statistique entre les entrées et la sortie de la fonction Booléenne utilisée, cette dernière doit être équilibrée.

— Pour résister à l'attaque de Rønjom−Helleseth [149] et à l'attaque de Berlekamp−Massey [120, 153], la fonction Booléenne utilisée doit avoir un haut degré algébrique.

— Pour résister à l'attaque par la meilleure approximation affine (BAA) [70], à l'attaque par corrélation rapide [126] sur les chiffrements à flot [27] et à l'attaque linéaire sur les systèmes de chiffrement par blocs [123], la fonction Booléenne doit avoir une non−linéarité élevée.

— Pour résister aux attaques par corrélation [158] et par corrélation rapide [126], il est nécessaire d'utiliser une fonction Booléenne ayant un ordre de résilience suffisamment élevé.

— Afin de résister à l'attaque algébrique [57] et notamment, un bon comportement contre l'attaque algébrique rapide, la fonction Booléenne doit avoir une haute immunité algébrique.

L'utilisation des fonctions Booléennes dans les systèmes de chiffrement à flot peut être schématisée par la figure (3.1).

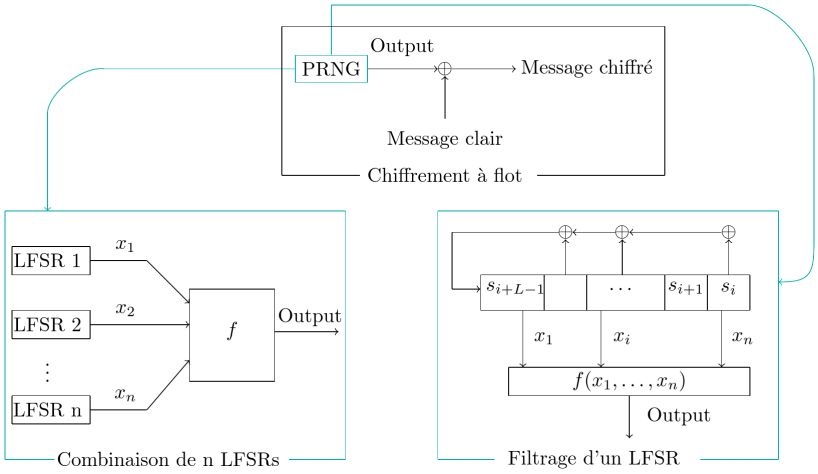

FIGURE 3.1 – Les utilisations d'une fonction Booléenne dans les systèmes de chiffrement à flot

Dans les générateurs (de nombres) pseudo−aléatoires (PRNG [1]) des systèmes de chiffrement à flot, les fonctions vectorielles sont utilisées pour combiner les sorties des n

1. "Pseudorandom number generator" en anglais.

LFSRs, ou pour filtrer un seul LFSR, en gérant les m bits à chaque cycle d'horloge au lieu d'un seul. Cela augmente la vitesse du chiffrement, mais risque de diminuer sa robustesse. Les fonctions vectorielles sont souvent plus efficaces puisque l'attaquant peut combiner les m sorties de la fonction de n'importe quelle manière. L'utilisation des fonctions vectorielles dans les systèmes de chiffrement à flot peut être schématisée par la figure (3.2).

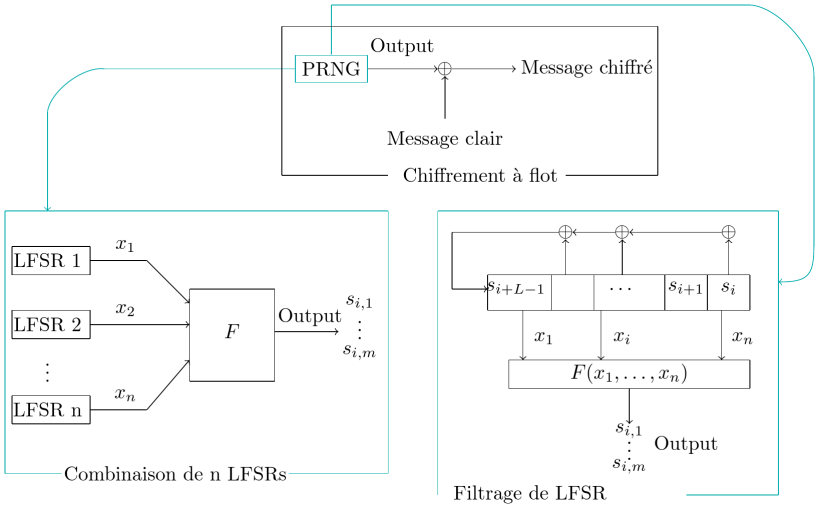

FIGURE 3.2 – Les utilisations d'une fonction vectorielle dans les systèmes de chiffrement à flot

Dans les systèmes de chiffrement par blocs, les fonctions vectorielles utilisées doivent également satisfaire ces critères.

Les fonctions Booléennes, à leur tour, jouent un rôle important dans les systèmes de chiffrement par blocs, ou chaque bloc admet en entrée un vecteur binaire $(x_1, \ldots, x_n)$ et en sortie le vecteur binaire $(y_1, \ldots, y_m)$, les coordonnées $y_1, \ldots, y_m$ sont les sorties de n fonctions Booléennes, figure (3.3).

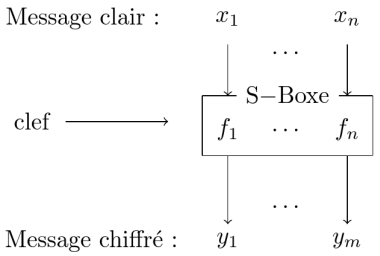

FIGURE 3.3 – Schéma d'un chiffrement par blocs

Nous allons maintenant détailler les différents critères cryptographiques des fonctions Booléennes qui nous intéressent.

3.2 Critères cryptographiques des fonctions Booléennes

Le rôle de la fonction Booléenne dans la construction des systèmes de chiffrement est l'application concrète des principes de confusion et diffusion [157]. Compte−tenu de ces principes et de nombreuses attaques découvertes au cours de l'évolution de la cryptographie [158], des critères de résistance ont été alors définis afin d'être satisfaits au mieux lors de l'élaboration d'un système de chiffrement et du choix des fonctions Booléennes cryptographiques à utiliser.

Le degré Algébrique des fonctions Booléennes : Il est important qu'une fonction Booléenne utilisée en cryptographie ait un haut degré algébrique, en particulier, dans le cas du chiffrement à flot. Le degré de la fonction Booléenne à n variables doit être proche de n à cause de l'attaque algébrique rapide (section (3.4.1)), cependant, il suffit que le degré soit plus grand que 2 dans le cas de chiffrement par blocs.

Dans le cas d'une combinaison de plusieurs LFSRs de longueurs $L_1, \ldots, L_n$, la suite de valeurs retournée par la fonction de combinaison f est modélisable par un LFSR de longueur inférieure à $L = f(L_1, \ldots, L_n)$. Cet LFSR peut en particulier être reconstruit à partir d'au plus $2L$ valeurs via l'algorithme de Berlekamp−Massey [120].

La relation entre le degré algébrique d'une fonction Booléenne et sa transformée de Walsh peut être caractérisée par la proposition suivante :

Proposition 31. *[106, 36] Soit f une fonction Booléenne à n variables ($n \geq 1$) et $1 \leq k \leq n$. Nous assumons que les valeurs de la transformée de Walsh de f sont divisibles par 2^k, c'est−à−dire que les valeurs de sa transformée de Fourier sont divisibles par 2^{k-1}, ce qui est équivalent à dire que toutes les distances de Hamming entre f et les fonctions affines sont divisibles par 2^{k-1}. Alors le degré algébrique de f est au plus égal à $n-k+1$.*

Fonctions Booléennes équilibrées : Un distingueur est une machine de Turing probabiliste avec oracle qui fournit une réponse binaire après un nombre de requêtes, basé sur les travaux de Leveiller et al. [109], Englund et Johansson dans [76], et Molland [133]. Ensuite, combiné par des résultats mathématiques de [134] par Molland et Helleseth. Le but des attaques par distingueur est de distinguer la clef de chiffrement d'une véritable

séquence aléatoire, notamment, dans le cas d'un chiffrement par blocs, où on considère un ensemble de permutations sur $\mathbb{F}_2^n$ et on essaye de distinguer un tel ensemble d'un ensemble de permutations tirées aléatoirement.

Afin d'éviter la cryptanalyse à clair connu [1], le premier critère que l'on demande aux fonctions Booléennes utilisées en cryptographie est *l'équilibre*. Une fonction Booléenne à n variables est dite équilibrée si elle prend autant de fois la valeur 0 que la valeur 1 sur l'ensemble de ses entrées.

Définition 32. *[31] Une fonction Booléenne f à n variables est dite équilibrée si sa sortie est uniformément distribuée, i.e.*

$$|\{x \in F_2^m,\, f(x) = 0\}| = |\{x \in F_2^m,\, f(x) = 1\}| = 2^{m-1}.$$

On peut déduire de la définition de la transformée de Walsh (définition (23)) qu'une fonction Booléenne f est équilibrée si et seulement si $W_f(0) = 0$. Cela limite le degré de la fonction à $m - 1$. Cette propriété est particulièrement importante en cryptographie puisqu'une telle fonction se comporte comme une variable aléatoire suivant une loi uniforme.

La fonction Booléenne f de l'exemple (14) n'est pas équilibrée.

Une fonction Booléenne utilisée comme fonction de combinaison doit être équilibrée, pour éviter que la connaissance du chiffré donne une information statistique sur le clair (en effet, si la fonction donne plus de 0 que de 1, on sait que, dans leur majorité, les bits du chiffré sont égaux aux bits correspondants du clair). Cependant, une généralisation mise en évidence par [158] est que dans le cas d'une fonction de combinaison de plusieurs LFSR, cela doit rester vrai quand on fixe certaines coordonnées à des valeurs constantes en entrée. Dans le cas contraire, en fixant ces coordonnées, on est à même d'observer une *corrélation* entre les entrées et les sorties de la fonction, et ceci permet de réduire la complexité d'une recherche exhaustive pour retrouver les initialisations des LFSRs.

Fonctions Booléennes sans corrélations et résilientes : L'*attaque par corrélation classique* a été d'abord proposée par Siegenthaler [158], ensuite, elle a été améliorée par Meier et Staffelbach dans [125, 126] et est devenue ce qu'on appelle *l'attaque par corrélation rapide*. Cette classe d'attaques entre dans la catégorie plus générale des attaques du type *diviser pour mieux régner* [159] qui exploitent l'existence d'une approximation non nécessairement linéaire, mais avec moins de variables. Alors, il est nécessaire d'utiliser

une fonction Booléenne ayant un ordre de résilience suffisamment élevé, au sens de la définition suivante :

Définition 33. *Une fonction Booléenne f à n variables est dite sans corrélation d'ordre $t \geq 1$ si sa distribution de valeurs ne change pas lorsqu'on fixe au plus t variables quelconques en entrée. De plus, une fonction Booléenne équilibrée et sans corrélation d'ordre t est dite résiliente d'ordre t.*

Un résultat établi par Xiao et Massey [171] permet d'obtenir la caractérisation de la résilience à l'aide des coefficients de Walsh :

Théorème 34. *[171] Une fonction Booléenne f à n variables est $t-$résiliente si et seulement si $W_f(u) = 0$ pour tout $u \in \mathbb{F}_2^n$ tel que $wt(u) \leq t$. Ceci est équivalent à dire que f est $t-$résiliente si et seulement si elle est équilibrée et $W_f(u) = 0$ pour tout $u \in \mathbb{F}_2^n$ tel que $0 < wt(u) \leq t$.*

L'ordre d'immunité aux corrélations des fonctions Booléennes à n variables n'est pas borné. L'ordre de résilience de ces fonctions n'est borné que par $n-1$, puisque la fonction Booléenne $\sum_{i=1}^{n} x_i$ est une fonction résiliente d'ordre $(n-1)$. Cependant, la situation est différente dans le cas des fonctions Booléennes non$-$équilibrées, non$-$constantes et sans corrélation.

Proposition 35. *[85] Soit f une fonction Booléenne non$-$constante, non$-$équilibrée et sans corrélation d'ordre t, alors $t \leq \frac{2n}{3} - 1$.*

L'attaque par corrélation est efficace si la fonction Booléenne de combinaison n'est pas hautement non$-$linéaire [126]. Plus précisément, Canteaut et Trabbia dans [27] et Canteaut dans [24] ont montré que l'attaque par corrélation sur le générateur de combinaison est aussi inefficace que possible si le coefficient de Walsh $W_f(u)$ de la fonction f est petit pour tout vecteur u de poids de Hamming plus grand et proche de m. Cette condition est satisfaite si la fonction f est hautement non$-$linéaire.

La non$-$linéarité des fonctions Booléennes : Introduit par Pieprzyk et Finkelstein [144], la non$-$linéarité des fonctions Booléennes utilisées en cryptographie est un paramètre important. Dans le cas de chiffrement par blocs comme ceux à flot [27, 50], l'existence d'une approximation affine d'une fonction permet très souvent de construire des attaques efficaces.

Définition 36. *La non$-$linéarité d'une fonction Booléenne f, notée $nl(f)$, est la distance minimale entre f et toutes les fonctions affines.*

Une fonction Booléenne est de haute non−linéarité si toutes les valeurs du spectre de Walsh sont de petite valeur absolue.

Proposition 37. *[31] La non−linéarité d'une fonction Booléenne f est égale à*

$$nl(f) = 2^{n-1} - \frac{1}{2}\max_{a\in\mathbb{F}_2^n} |W_f(a)|.$$

La moyenne des valeurs prises par W_f^2 est égale à 2^n, puisque l'identité de Parseval de la proposition (24) énonce que pour toute fonction Booléenne à n variables, $\sum_{x\in\mathbb{F}_2^n} W_f(x)^2 = 2^{2n}$. Ainsi, $\max_{a\in\mathbb{F}_2^n} |W_f(a)| \geq 2^n$. Ce qui nous donne une borne sur la non−linéarité des fonctions Booléennes (Lobanov) :

$$nl(f) \leq 2^{n-1} - 2^{\frac{n}{2}-1}. \tag{3.1}$$

L'identité de Parseval implique que la non−linéarité optimale d'une fonction Booléenne vaut $2^{\frac{n}{2}}$, et est atteinte par les fonctions *courbes*[2] [151], qui n'existent que si n est pair. Lorsque n est impair, $2^{\frac{n}{2}}$ n'est pas un entier et alors la borne n'est pas atteinte.

Proposition 38. *(Fonction courbe). Une fonction Booléenne à n variables est dite fonction courbe si et seulement si pour tout $x \in \mathbb{F}_2^n$, $|W_f(x)| = 2^{\frac{n}{2}}$.*

Notons qu'une fonction Booléenne f est courbe si et seulement si, toutes ses fonctions dérivées sont des fonctions courbes, au sens de la proposition suivante :

Proposition 39. *Une fonction Booléenne f à n variables est courbe si et seulement si, toutes ses fonctions dérivées $D_a f(x) = f(x) + f(a+x)$, $a \neq 0$, sont des fonctions équilibrées, c'est−à−dire, si et seulement si f satisfait PC.*

La non−linéarité optimale n'est pas connue pour un nombre impair de variables supérieur ou égal à 9. Une fonction courbe possédant une non−linéarité maximale ne peut être équilibrée et la linéarité optimale pour une fonction équilibrée n'est pas connue dés que $n \geq 8$.

La non−linéarité de la fonction Booléenne de l'exemple (14) est égale à 1. Cette fonction n'est pas une fonction courbe.

La définition (20) du degré algébrique d'une fonction Booléenne à n variables nous donne le théorème suivant :

2. "Bent functions" en anglais.

Théorème 40. *[65, 151] Soit $n \geq 4$ un entier pair, alors le degré algébrique de toutes les fonctions courbes est au plus égal à $\frac{n}{2}$.*

Cette borne est appelée la *Borne de Rothaus*. Une reconnaissance des fonctions courbes de bas degrés algébrique peut être trouvée dans le chapitre (6.2) de [31]. Cependant, une caractérisation complète des fonctions courbes de degré algébrique $d \geq 3$ est toujours un problème ouvert.

Lien entre le degré algébrique, la non−linéarité et la résilience. La proposition suivante donne un lien entre le degré algébrique, la non−linéarité et la résilience d'une fonction Booléenne.

Proposition 41. *(Degré algébrique, non−linéarité et résilience). Soit f une fonction Booléenne $t-$résiliente.*

- *Si $t \leq n-1$, alors* $\deg(f) \leq n-(t-1)$ *[158].*
- $nl(f) \leq 2^{n-1} - 2^{t-1}\left\lceil \frac{2^{n-t-2}}{\sqrt{2^n - \sum_{i=0}^{t}\binom{i}{t}}} \right\rceil$ *[29]*

La non−linéarité d'ordre r : La non−linéarité d'ordre r est un critère plus général que la non−linéarité d'ordre 1 (le cas $r = 1$ correspond à la non−linéarité usuelle (définition (36)) notée $nl(f)$), qui joue un rôle important dans la sécurité des systèmes cryptographiques.

Définition 42. *(non−linéarité d'ordre r) Soit f une fonction Booléenne à n variables, alors pour tout entier positif $r \leq n$, on appelle la non−linéarité d'ordre r de f, notée $nl_r(f)$, la distance de Hamming minimale entre f et toutes les fonctions Booléennes de degrés au plus r.*

Cette distance est égale au nombre de bits à changer dans le vecteur des valeurs de la fonction f pour obtenir une fonction Booléenne de degré algébrique au plus r. La non−linéarité d'ordre r permet de connaître le degré algébrique de la fonction Booléenne f, mais il n'est pas utile d'accorder la même importance à tous les ordres [102]. La non−linéarité d'ordre 1 doit être optimale pour éviter les attaques linéaires, mais quand l'ordre augmente, la complexité des attaques augmente. De nombreux articles s'intéressent au problème :

1) Sous un angle cryptographique :
   - D'un point de vue général dans [35].
   - En le reliant à d'autres critères comme, la résilience [19, 104] et l'immunité algébrique [30, 32].

2) D'un point de vue de la théorie des codes [22, 106, 107]

Le calcule de la non−linéarité d'ordre r d'une fonction Booléenne de degré algébrique plus grand que r est un problème difficile pour $r > 1$, même la non−linéarité d'ordre 2 n'est connue que pour peu de fonctions particulières et pour des fonctions Booléennes de petit nombre de variables. Un bon algorithme a été donné par G. Kabatiansky et C. Tavernier, ensuite amélioré et implémenté par Fourquet et Tavernier [87], cet algorithme est efficace seulement pour $r = 2$ et $n \leq 11$.

Avalanche Stricte et Propagation : Le critère *Avalanche Stricte* (SAC) est une autre propriété importante des fonctions cryptographiques (les algorithmes de chiffrement par blocs et les fonctions de hachage). Ce critère a été introduit par Wabster et Tavares [169] puis généralisé au critère de *Propagation* (PC) par Preneel et al dans [146]. Le SAC et ses généralisations sont basés sur les propriétés des dérivées des fonctions Booléennes. L'idée est d'obtenir des modifications de plus en plus importantes au fur et à mesure que les données se propagent dans la structure de l'algorithme, et ainsi d'obtenir des résultats en sortie totalement différents avec très peu de perturbation en entrées ; par conséquent, elles sont liées aux propriétés de diffusion des systèmes cryptographiques utilisant cette fonction.

Définition 43. *Soit $E \subset \mathbb{F}_2^n$, une fonction Booléenne f à n variables satisfait le critère PC de propagation par rapport a E si, pour tout $a \in E$, la dérivée de f en a, $D_a f :\rightarrow f(x+a) \oplus f(x)$, est équilibrée.*

Nous avons la définition suivante :

Définition 44. *Une fonction Booléenne f à n variables satisfait le critère $PC(l)$ de propagation de degré l ($1 \leq l \leq n$) si pour tout $a \in \mathbb{F}_2^n$ de poids $0 < wt(a) \leq l$, la fonction dérivée de f en a, $D_a f :\rightarrow f(x+a) \oplus f(x)$, est équilibrée.*

Cela correspond en fait à l'annulation des coefficients d'auto−corrélation de f en ces même points. Le cas $l = 1$ correspond au critère (SAC).

L'immunité algébrique : Alors que le degré algébrique des fonctions Booléennes était considéré comme un bon critère pour résister aux attaques algébriques, il est apparu en 2003 [57] que ce n'était pas un critère pertinent pour résister aux attaques algébriques et que le critère nécessaire et approprié (mais non−suffisant) contre ces attaques est une immunité algébrique élevée. Le principe général des attaques algébriques, qui sont efficaces (sous certaines conditions) contre les chiffrements à flot [57], réside dans la possibilité

de retrouver la clef par la résolution d'un système d'équations polynomiales multivariées surdéterminé reliant les bits de sortie et l'état initial dés que le degré de ces relations est faible, via l'utilisation de *bases de Gröbner* par exemple [79].

Considérons un LFSR dans le cas d'un générateur filtré ou n LFSRs dans le cas d'un générateur de combinaison, de partie linéaire de taille N et avec une fonction Booléenne f à n variables comme fonction de filtrage ou de combinaison. S'il existe une fonction Booléenne non nulle g et une fonction Booléenne h de petit degré tel que $f(x)g(x) = h(x)$, $\forall x \in \mathbb{F}_2$, alors l'attaque algébrique est réalisable.

Ainsi, toute relation de ce type est équivalente à l'existence d'un annulateur de petit degré pour la fonction f ou pour la fonction $(1 + f)$, alors l'immunité algébrique de f [79, 124], notée $AI(f)$, est définie comme le degré minimal de toutes les fonctions non nulles g vérifiant $f(x)g(x) = 0$ ou $(1 + f(x))g(x) = 0$.

3.3 Critères cryptographiques des fonctions vectorielles

Les fonctions vectorielles sont utilisées comme des fonctions de combinaison ou de filtrage dans les systèmes de chiffrement à flot, ou comme des boîtes$-S$ dans les systèmes de chiffrement par blocs. Ces deux situations sont différentes, mais quelques$-$uns des critères de sécurité que ces fonctions doivent satisfaire sont les mêmes. Essentiellement, la non$-$linéarité [140], la non$-$linéarité d'ordre supérieur [101], l'immunité algébrique, l'équilibre et le degré algébrique.

Fonctions vectorielles équilibrées : Tout comme les fonctions Booléennes, le critère d'équilibre est préférable pour les fonctions vectorielles utilisées en cryptographie. Une $(n, m)-$fonction est dite *équilibrée* si elle prend chaque valeur de $\mathbb{F}_2^m$ un même nombre de fois, c'est$-$à$-$dire que pour tout $b \in \mathbb{F}_2^m$, on a $|F^{-1}(b)| = 2^{n-m}$.

Proposition 45. *[110] Une $(n, m)-$fonction est dite équilibrée si et seulement si toutes ses fonctions composantes $v \cdot F$, $v \in \mathbb{F}_2^m$, $v \neq 0$, sont équilibrées, c'est$-$à$-$dire que F est équilibrée si et seulement si, pour tout élément non nul $v \in \mathbb{F}_2^m$, $W_F(0, v) = 0$.*

D'une manière équivalente, F est une fonction équilibrée si et seulement si

$$\sum_{a \in \mathbb{F}_2^n} W_{D_a(v \cdot F)}(0) = 0 \text{ pour tout } v \in \mathbb{F}_2^m,\ v \neq 0.$$

Une fonction vectorielle utilisée comme une fonction de combinaison ou de filtrage doit être équilibrée parce que n'importe quelle combinaison de ses sorties peut être faite, alors pour éviter qu'une telle combinaison donne des informations statiques permettant de distinguer quand une paire de texte est une paire (texte clair, texte chiffré), cela nécessite que la fonction soit équilibrée.
Dans les systèmes de chiffrement par blocs, il est préférable que la fonction vectorielle soit équilibrée.

Le degré algébrique : Le degré algébrique (définition (29)) est un paramètre important, puisque les sorties de la fonction utilisée dans les systèmes de chiffrement à flot (ces sorties sont aussi les sorties du PRG (figure (3.2))), peuvent être combinées et utiliser dans l'attaque de Berlekamp−Massey.

Dans les systèmes de chiffrement par blocs, le degré algébrique est un paramètre de sécurité contre les attaques structurées, comme les attaques *intégrales* [103], l'attaque d'ordre différentiel élevé ou les attaques basées sur la propriété de division [161].
L'attaque d'ordre différentiel élevé [28, 101, 105] exploite le fait que le degré algébrique d'une $(n, m)-$fonction est bas, ou plus généralement l'existence d'un sous−espace V de $\mathbb{F}_2^n$ telle que la fonction $D_V F(x) = \sum_{v \in V} f(x + v)$ est constante (i.e., $D_{a_1} \dots D_{a_k} F(x)$, où $\{a_1, \dots, a_k\}$ est une base de V).

La non−linéarité : Dans les systèmes de chiffrement à flot, puisque la structure des générateurs pseudo−aléatoires utilisés pour combiner ou filtrer les fonctions n'est pas itérative, alors toutes les m séquences produites par une $(n, m)-$fonction peuvent être combinées par une fonction Booléenne à m variables g, linéaire ou non−linéaire mais non−constante, pour réaliser l'attaque par corrélation.

La non−linéarité des fonctions vectorielles est un critère très important pour résister contre les attaques par corrélation rapide [126]. Plusieurs résultats sur la non−linéarité des fonctions vectorielles sont des généralisations de celles des fonctions Booléennes, ainsi, la non−linéarité des $(n, m)-$fonctions a été introduite par Nyberg [139] et étudiée plus tard par Chabaud et Vaudenay [48], au sens de la définition suivante :

Définition 46. *[139, 48] La non−linéarité d'une $(n, m)-$fonction F, notée $nl(F)$, est la non−linéarité minimale de toutes ses fonctions composantes $v \cdot F$, $v \in \mathbb{F}_2^m$, $v \neq 0$.*

Comme dans le cas des fonctions Booléennes, la non−linéarité d'une $(n, m)-$fonction peut être exprimée en fonction de la transformée de Walsh :

$$nl(F) = 2^{n-1} - \frac{1}{2} \max_{v \in \mathbb{F}_2^{m*}, u \in \mathbb{F}_2^n} |\sum_{x \in \mathbb{F}_2^n} (-1)^{v \cdot F(x) + u \cdot x}|.$$

Remarque 47. *Notons que* $\max_{v \in \mathbb{F}_2^{m*}, u \in \mathbb{F}_2^n}$ *peut être remplacé par* $\max_{(u,v) \in \mathbb{F}_2^n \times \mathbb{F}_2^m; (u,v) \neq (0,0)}$, *puisque* $\sum_{x \in \mathbb{F}_2^n} (-1)^{u \cdot x} = 0$, *pour tout élément non nul* u.

La borne $2^{n-1} - 2^{\frac{n}{2}-1}$ (équation (3.1)) sur la non−linéarité des fonctions Booléennes à n variables est valide pour les (n, m)−fonctions, alors la notion des fonctions courbes a été généralisée aux fonctions vectorielles comme suit :

Définition 48. *Soit* n *et* m *deux entiers (*n *pair), une* (n, m)−*fonction est dite courbe si*

$$nl(F) = 2^{n-1} - 2^{\frac{n}{2}-1}.$$

Une condition nécessaire pour l'existence des (n, m)−fonctions courbes est donnée par la proposition suivante :

Proposition 49. *Les* (n, m)−*fonctions courbes n'existent que si* n *est pair et* $m \leq \frac{n}{2}$.

Clairement, une (n, m)−fonction F est courbe si et seulement si toutes ses fonctions composantes $v \cdot F$, $v \neq 0$, sont courbes, par conséquent, le degré algébrique d'une fonction courbe est au plus $\frac{n}{2}$. Notons aussi que d'après la proposition (39), une (n, m)−fonction F est courbe si et seulement si, pour tout $v \in \mathbb{F}_2^m$, $v \neq 0$, et pour tout $a \in \mathbb{F}_2^n$, $a \neq 0$, la fonction $v \cdot (F(x) + F(x + a))$ est équilibrée, c'est−à−dire, par la proposition (45) on a, pour tout $a \in \mathbb{F}_2^n$, $a \neq 0$, la fonction $F(x) + F(x + a)$ est équilibrée.

Proposition 50. *Une* (n, m)−*fonction est courbe si et seulement si toutes ses fonctions dérivées* $D_a F(x) = F(x) + F(x + a)$, $a \in \mathbb{F}_2^n$, $a \neq 0$, *sont des fonctions équilibrées.*

Dans notre exemple (28), la $(4, 4)$−fonction est une fonction équilibrée, de linéarité égale à 12, et de non−linéarité égale à 2, c'est−à−dire que cette fonction n'est pas courbe.

L'existence et la non−existence des résultats sur la non−linéarité élevée des (n, m)−fonctions ont été déduites dans [165], ensuite des bornes sur la non−linéarité des (n, m)−fonctions ont été données dans [40].

La proposition (49) a mené à la redécouverte d'une borne supérieure meilleure que la borne du rayon de recouvrement si $n > \frac{n}{2}$ [48].

Théorème 51. *(Borne de Sidelnikov−Chabaud−Vaudenay). Soient n et m deux entiers positifs tels que $m \geq n-1$. Soit F une (n,m)−fonction, alors*

$$nl(F) \leq 2^{n-1}\frac{1}{2}\sqrt{3\times 2^n - 2 - 2\frac{(2^n-1)(2^{n-1}-1)}{(2^m-1)}}.$$

Notons que pour $m = n-1$, cette borne coïncide avec la borne donnée par l'équation (3.1). Pour $m > n$, la racine carrée ne peut pas être un entier [48]. Par conséquent, la borne de Sidelnikov−Chabaud−Vaudenay ne peut être étroite que si $n = m$ avec n impair, dans ce cas :

$$nl(F) \leq 2^{n-1} - 2^{\frac{n-1}{2}}. \tag{3.2}$$

Définition 52. *[48] Une (n,n)−fonction F qui atteint la borne (3.2) avec égalité, c'est−à−dire, $nl(F) = 2^{n-1} - 2^{\frac{n-1}{2}}$ (n impair), est appelée fonction presque courbe (AB[3]).*

Notons que les fonctions presque courbes n'existent que si $n \geq 3$. Ces fonctions sont les (n,n)−fonctions tels que, pour tout $u, v \in \mathbb{F}_2^n$, $v \neq 0$, la somme $\sum_{x\in\mathbb{F}_2^n}(-1)^{v\cdot F(x)+u\cdot x} = W_F(u,v)$ est égale à 0 ou à $\pm 2^{\frac{n+1}{2}}$.

Dans le cas des systèmes de chiffrement par blocs dont la structure est itérative, la connaissance d'une combinaison non−linéaire des sorties de la fonction F d'une non−linéarité faible ne mène pas forcément à une attaque, sauf peut être dans le cas où le degré algébrique de la fonction est très faible.

La non−linéarité des (n,m)−fonctions est un critère très important pour résister contre les attaques linéaires [123] et différentielles [48]. Par conséquent, une deuxième généralisation de ce critère a été introduite dans [45].

Définition 53. *[175] Soit F une (n,m)−fonction, alors la non−linéarité restreinte de F, notée $unl(F)$, est la distance de Hamming minimale entre toutes les fonctions affines non−constantes et les fonctions Booléennes $g \circ F$, où g est une fonction Booléenne non−constante à m variables.*

La non−linéarité restreinte de F peut être liée aux valeurs de la transformée de Fourier discrète de la fonction $\phi_b = 1_b \circ F$, alors une borne dépend de $nl(F)$ peut être directement déduite comme suit :

3. "Almost bent" en anglais.

Proposition 54. *[175] Pour toute $(n,m)-$fonction F, on a :*

$$unl(F) = 2^{n-1} - \frac{1}{2}\max_{u\in\mathbb{F}_2^n,\neq 0}\sum_{b\in\mathbb{F}_2^m}|\hat{\phi}_b(u)| \geq 2^{n-1} - 2^{m/2}(2^{n-1} - nl(F)).$$

Cette borne ne donne pas une idée sur la meilleure non−linéarité restreinte, même si $nl(F)$ est proche de la non−linéarité des fonctions courbes, alors $unl(F)$ est approximativement plus large que $2^{n-1} - 2^{\frac{n+m}{2}-1}$.
Si F est équilibrée, alors la distance minimale entre $v \cdot F$, $v \in \mathbb{F}_2^m$, $v \neq 0$, et les fonctions affines ne peut pas être atteinte par les fonctions affines constantes, car la distance entre $v \cdot F$ (qui est équilibrée) et les fonctions constantes est 2^{n-1}.

Proposition 55. *(Borne du rayon de recouvrement) Pour toute $(n,m)-$fonction équilibrée F :*

$$unl(F) \leq nl(F) \leq 2^{n-1} - 2^{\frac{n}{2}-1}.$$

Dans [45], une autre borne a été donnée :

$$unl(F) \leq 2^{n-1} - \frac{1}{2}\left(\frac{2^{2m}-2^m}{2^n-1} + \sqrt{\frac{2^{2n}-2^{2n-m}}{2^n-1} + \left(\frac{2^{2m}-2^m}{2^n-1} - 1\right)^2} - 1\right).$$

Cette borne améliore la borne du rayon de recouvrement seulement pour $m \geq \frac{n}{2} + 1$.

La non−linéarité d'ordre r : Pour tout entier positif r, la non−linéarité d'ordre r d'une fonction vectorielle F est définie comme la non−linéarité d'ordre r minimale de ces fonctions composantes. La définition (42) peut être généralisée aux fonctions vectorielles en considérant la non−linéarité minimale d'ordre r des fonctions coordonnées :

$$nl_r(F) = \min_{v\neq 0} nl_r(v \cdot F),\ v \neq 0.$$

L'immunité algébrique : L'existence d'équations multivariées de bas degré entre les bits d'entrées et les bits de sorties d'une boîte$-S$ peut être exploitée dans les attaques algébriques sur les schémas par bloc. Contrairement au cas d'un chiffrement à flot, le système d'équations qu'on obtient est généralement difficile à résoudre. Un critère nécessaire mais non−suffisant contre ces attaques est que la fonction vectorielle ait une immunité algébrique élevée. Plusieurs notions d'immunité algébrique d'une fonction vectorielle sont liées à ces attaques : l'immunité algébrique standard (définition (75)), l'immunité algébrique par composantes (définition (78)) et l'immunité algébrique du graphe (définition (77)).

3.3.1 Dans les systèmes de chiffrement à flot

La corrélation et la résilience : La notion de résilience des fonctions Booléennes a été généralisée aux $(n,m)-$fonctions, au sens de la définition suivante :

Définition 56. *Soit n et m deux entiers positifs et t un entier tel que $0 \leq t \leq n$, alors une $(n,m)-$fonction F est dite sans corrélation d'ordre t si la distribution de valeurs de ses sorties ne change pas lorsqu'on fixe au plus t variables quelconques x_i de x en sorties. De plus, une $(n,m)-$fonction équilibrée et sans corrélation d'ordre t est dite résiliente d'ordre t.*

Notons par $(n,m,t)-$fonction toute $(n,m)-$fonction résiliente d'ordre t. Si cette fonction existe, alors $m \leq n-1$, puisque une $(n,m)-$fonction équilibrée n'existe que si $m \leq n$. Il est montré dans [51] que si une $(n,m,t)-$fonction existe, alors

1) $m \leq n - \log_2 \left[\sum_{i=0}^{t/2} \binom{n}{i} \right]$ si t est pair.

2) $m \leq n - \log_2 \left[\binom{n-1}{(t-1)/2} + \sum_{i=0}^{t-1/2} \binom{n}{i} \right]$ si t est impair.

Une meilleure borne a été donnée dans [63], si une $(n,m,t)-$fonction existe, alors

$$t \leq \lfloor \frac{2^{m-1}n}{2^m - 1} \rfloor - 1 \ et \ t \leq 2 \lfloor \frac{2^{m-2}(n+1)}{2^m - 1} \rfloor - 1.$$

3.3.2 Dans les systèmes de chiffrement par blocs

Dans la plupart des systèmes de chiffrement par blocs, les fonctions vectorielles sont généralement la seule source de non$-$linéarité, alors la résistance du chiffrement dépend largement de la puissance cryptographique des fonctions vectorielles utilisées.

l'uniformité différentielle des fonctions vectorielles : Les attaques différentielles ont été introduites par Biham et Shamir en 1991 [16]. Ces attaques assument l'existence des chaînes binaires ordonnées (α, β), $\alpha \neq 0$, du même longueur que les blocs, tel qu'un bloc m du texte clair choisi aléatoirement, et c, c' des textes chiffrés liés à m et $m+\alpha$, la différence entre les bits $c+c'$ a une probabilité plus large pour être égale à β que si c et c' sont choisis aléatoirement. Le couple (α, β), qui correspond a un biais dans la distribution de la sortie, est appelé *différentiel*.

L'idée de l'attaque différentielle est d'étudier les propriétés de la différentielle du chiffrement et de comparer ces propriétés à celle attendues pour une fonction aléatoire. Plus

la probabilité de la différentielle est large, plus l'attaque est efficace.

L'existence de l'attaque différentielle mène à un critère sur la fonction vectorielle F utilisée comme une boîte$-S$ dans les fonctions de tour dans le chiffrement. Le critère est que la sortie de $D_aF(x) = F(x)+F(x+a)$, $x, a \in \mathbb{F}_2^n$, $a \neq 0$ soit uniformément distribuée.

Définition 57. *[139, 140, 141] Soient n, m et γ des entiers positifs, une $(n, m)-$fonction F est dite d'uniformité différentielle si, pour tout élément non$-$nul $a \in \mathbb{F}_2^n$, et tout $b \in \mathbb{F}_2^m$, l'équation $F(x) + F(x + a) = b$ admet au plus γ solutions. La valeur minimale de γ qui vérifie une telle propriété, c'est$-$à$-$dire, le nombre maximal des solutions de telle équation, noté par γ_F, est appelée l'uniformité différentielle de F.*

L'uniformité différentielle γ_F est nécessairement paire, puisque si x est une solution de $D_aF(x) = b$, alors $x + a$ est aussi une solution. Cependant, l'uniformité différentielle d'une $(n, m)-$fonction est bornée par 2^{n-m} puisque D_aF est aussi une $(n, m)-$fonction, alors il existe au moins un élément de $\mathbb{F}_2^n$ qui a au moins 2^{n-m} pré$-$images par D_aF. Si F est une fonction équilibrée, alors l'uniformité différentielle est égale à 2^{n-m}. Dans ce cas, la fonction est dite *parfaitement non$-$linéaire* (PN [4]).

Le théorème (39) montre que les fonctions courbes sont aussi des fonctions parfaitement linéaires. Le comportement des $(n, m)-$fonctions a été étudié dans [61, 93, 142, 164].

Dans [138, 141], les auteurs ont montré que pour une meilleure contribution de la boîte$-S$ F dans la résistante contre l'attaque différentielle, la valeur γ_F doit être le plus bas possible. Quand $m \geq n$, la plus petite valeur de γ_F est 2, alors les $(n, m)-$fonctions d'uniformité différentielle 2 n'existent que si $m \geq n$.

Définition 58. *[13, 138, 141] Une $(n, m)-$fonction F est dite presque parfaitement non$-$linéaire (APN)[5] si et seulement si elle est d'uniformité différentielle égale à 2, c'est$-$à$-$dire, si pour tout $a \in \mathbb{F}_2^n$, $a \neq 0$, et tout $b \in \mathbb{F}_2^n$, l'équation $F(x) + F(x + a) = b$ a 0 ou 2 solutions, i.e. $|\{D_aF(x), x \in \mathbb{F}_2^n\}| = 2^{n-1}$.*

Remarque 59. *On préfère que les boîtes$-S$ soient APN ou d'uniformité différentielle égale à 4, ou au moins d'uniformité différentielle égale à 6.*

4. "Perfect nonlinear" en anglais.
5. "Almost perfect nonlinear" en anglais.

3.4 Les attaques algébriques

Les générateurs de combinaison et de filtrage sont tous les deux sujets des *attaques algébriques*. Ces attaques sont essentiellement de nature algébrique contrairement aux attaques par corrélations qui sont plus de nature statistique. L'idée est que les bits de la clef d'un cryptosystème peuvent être caractérisés comme des solutions d'un système d'équations multivariées traduisant les spécifications du cryptosystème viennent de Claud Shannon ([157], pp. 49) :

"*Thus, if we could show that solving a certain system requires at least as much work as solving a system of simultaneous equations in a large number of unknowns, of a complex type, then we would have a lower bound of sorts for the work characteristic.*"

Jusqu'à l'invention des attaques algébriques, cette observation brillante conduisait plus à un critère de conception qu'une attaque réelle. Ce système est trop complexe pour être résolu puisque ses équations sont hautement non−linéaires et le nombre d'inconnues est trop important pour un système d'équations non−linéaires.

Les attaques algébriques ont été introduites en 2003 [55, 57, 79], réalisables sur les systèmes de chiffrement par blocs et à flot [6], elles ont changé la situation des fonctions Booléennes dans les systèmes de chiffrement à flot.

Les attaques algébriques sont des attaques à clair connues qui exploitent des relations algébriques entre les bits du clair, ceux du chiffré et ceux de la clef secrète. La connaissance de plusieurs couples clairs−chiffrés fournit un système d'équations dont les inconnues sont les bits de la clef secrète. Différentes techniques peuvent être utilisées pour la résolution :

— Si le système comporte un nombre réduit de monômes de degré supérieur ou égal à 2 on pourra utiliser une technique de linéarisation consistant à remplacer chaque monôme de degré plus grand que 1 par une nouvelle variable et résoudre le système homogène obtenu par l'élimination de Gausse. Ceci permet d'obtenir un système linéaire plus simple à résoudre.
  Notons que la linéarisation n'est pas la meilleure méthode pour résoudre les systèmes d'équations non−linéaires [10], mais cette méthode donne une bonne approximation.
— Pour d'autres situations on pourra envisager d'utiliser les bases de Gröbner [78] ou des méthodes *ad−hoc* comme l'algorithme XL [56].

3.4.1 Attaques algébriques sur les schémas à flot

Dans le cas des chiffrements à flot décrits par la figure (3.1) et la figure (3.2), le principe général est de retrouver la clef par résolution des systèmes d'équations polynomiales multivariées surdéterminés, c'est−à−dire, un système avec un nombre d'équations indépendantes strictement plus grand que le nombre des inconnues.

En considérant un générateur de combinaison ou de filtrage de taille N avec une fonction Booléenne f à n variables comme fonction de filtrage, alors il existe une permutation linéaire $L : \mathbb{F}_2^N \to \mathbb{F}_2^N$ qui exprime l'état interne du registre en fonction de l'état précédent et une fonction linéaire $L^{'} : \mathbb{F}_2^N \to \mathbb{F}_2^n$, ainsi, l'attaque algébrique de base consiste à écrire toutes les équations entre l'état initial du LFSR, noté $(u_1, \ldots, u_N)$, et la suite chiffrante pseudo−aléatoire $(z_i)_{i\geq 0}$ fournie par le générateur, pour chaque $i \geq 0$:

$$\begin{cases} z_0 = f(L^{'} \circ L^0(u_1, \ldots, u_N)) \\ z_1 = f(L^{'} \circ L^1(u_1, \ldots, u_N)) \\ z_2 = f(L^{'} \circ L^2(u_1, \ldots, u_N)) \\ \qquad\qquad\qquad\qquad\qquad\qquad \vdots \end{cases} \tag{3.3}$$

Puisque L^t est une fonction linéaire pour tout $t \geq 1$, alors toutes les équations ont un degré égal au degré algébrique de la fonction Booléenne f. Ainsi, le nombre de ceux exploitable par l'attaquant est égale au nombre des bits z_i connus par ce dernier. Cependant, le nombre de bits z_i peut être plus grand que le nombre des inconnues, ce qui réduit la complexité de la résolution du système en utilisant les bases de Gröbner, le calcul de ces bases peut se faire d'une manière efficace grâce aux algorithmes $F4$ et $F5$ de Jean−Charles Faugère [77, 78, 79].

Dans [10, 11], des résultats récents sur la complexité de $F4$ et $F5$ sont donnés. Ces résultats ne s'appliquent que sous certaines hypothèses difficiles a vérifié pour les systèmes issus de la cryptographie.

Citons que l'algorithme FGLM [80] permet de calculer efficacement une base de Gröbner si l'on en dispose d'une pour un autre ordre sur les monômes, et qu'il est difficile d'évaluer la complexité des algorithmes de calcule de base de Gröbner.
Pour faciliter l'analyse de la complexité, on peut aussi résoudre le système d'équations en question par la linéarisation. Il s'agit de l'algorithme XL [56] mais dont la complexité est moins bonne que celle de $F4$ [8, 64] et de l'algorithme XSL [58] dont la complexité est

discutée dans [52].

Par la méthode de linéarisation (section (3.4)), on obtient donc un système linéaire dont le nombre d'inconnues est approximativement : $D^w = 1 + \left(\sum_{i=1}^{deg(f)} \binom{N}{i} \right)^w$.
L'attaque nécessite alors la connaissance de D bits de suite chiffrante et D^w opérations, avec $w \simeq 3$.

Pour retrouver la clef k il nous faut au moins $L_{deg(f)}$ équations ce qui nous donne un système que l'on sait résoudre en $O((D^w)^3)$.

L'amélioration de Meier et Courtois : En 2003, Courtois et Meier [57] ont proposé une amélioration de l'attaque, une idée simple mais très efficace. L'attaque fonctionne dès lors qu'il existe des relations de petit degré entre la sortie de la fonction et ses entrées [124]. Plus précisément, supposons l'existence des deux fonctions h et $g \neq 0$ de bas degré au plus d, telles que $f \times g = h$ (où $f \times g$ est le produit de Hadamard, et $supp(f \times g) = supp(f) \cap supp(g)$). Donc, pour chaque $i \geq 0$, la relation (3.3) implique :

$$z_i g(L' \circ L^i(u_1, \ldots, u_N)) = h(L' \circ L^i(u_1, \ldots, u_N)). \tag{3.4}$$

Cette équation en $u_1, \ldots, u_N$ est de degré au plus d, puisque L, L' sont linéaires et le système d'équations obtenu après linéarisation a au plus $\sum_{i=0}^{d} \binom{N}{i}$ inconnues et peut être résolu par l'élimination de Gauss.

Dans [84], les auteurs ont tenté de dégager d'autres caractéristiques qui peuvent faciliter la résolution de ces systèmes.

Attaques algébriques standards

Supposons qu'on a une fonction de filtrage f ou de combinaison de haut degré. L'attaque algébrique contre le générateur filtré est réalisable si l'attaquant cherche l'existence des fonctions h et $g \neq 0$ de bas degrés qui vérifient :

C1 Il existe une fonction non nulle g de bas degré tel que $fg = h$, où h est une fonction non nulle de degré au plus d.

C2 Il existe une fonction non nulle g de bas degré tel que $fg = 0$ ou $g(f + 1) = 0$.

Les conditions **C1** et **C2** sont équivalentes.

Les relations algébriques de bas degrés existent pour quelques chiffrements à flot qui étaient résistants à toutes les attaques connues précédemment, on cite parmi ces systèmes

Toyocrypt et LILI-128, ces deux derniers ayant été attaqués par Courtois et Meier dans leur article [57].

Généralement, dans une attaque algébrique pour chercher des annulateurs de la fonction f, on utilise le scénario **C2**. Ainsi, l'attaque algébrique standard repose sur la notion d'annulateur d'une fonction Booléenne,

Définition 60. *(Annulateur) Soit f une fonction Booléenne à n variables. Toute autre fonction Booléenne non nulle à n variables g telle que $f(x)g(x) = 0$, $\forall x \in \mathbb{F}_2^n$, est appelée annulateur de f.*

L'ensemble des annulateurs de f, noté $AN(f)$, forme un idéal de l'anneau des fonctions Booléennes à n variables. Le paramètre essentiel, qui influence la complexité des attaques algébriques, est le degré minimal des fonctions non nulles de $AN(f)$ et $AN(1+f)$. Ce paramètre est l'*immunité algébrique* de f.

$$AI(f) = \{g \in AN(f) \cup AN(1+f),\ g \neq 0\}. \tag{3.5}$$

Nous étudierons plus en détail cette notion dans la section (3.5).

Ainsi, pour un LFSR de longueur $N \geq 2k$, où k est la taille de la clef (cette condition est nécessaire sinon le système serait vulnérable aux attaques dites *time−memory−data trade−off* [17]), du générateur filtré par la fonction f à n variables, l'attaque sera plus performante que la recherche exhaustive de la clef dès que l'immunité algébrique de la fonction de filtrage vérifie :

$$AI(f) \geq 0.42 \left\lfloor \frac{k}{1+\log_2 k} \right\rfloor.$$

Tandis que l'immunité algébrique de la fonction Booléenne à n variables f est égal au plus à $\lceil \frac{n}{2} \rceil$, alors d'après [57], la complexité d'une attaque algébrique utilisant une telle fonction f est d'ordre

$$7\left(\binom{N}{0} + \ldots + \binom{N}{\frac{n}{2}}\right)^{\log_2(7)} \approx 7\left(\binom{N}{0} + \ldots + \binom{N}{\frac{n}{2}}\right)^{2.8}.$$

Si on choisit $k = 128$ et $N = 256$, alors pour $n \geq 15$, la complexité de l'attaque algébrique est supérieure à la complexité d'une attaque exhaustive qui est égale à 2^{128}.

L'utilisation d'une fonction de filtrage f de haute immunité algébrique ne garantit pas la résistance aux attaques algébriques, en effet, il existe une version plus évoluée appelée

attaques algébriques rapides.

Attaques algébriques rapides

L'attaque algébrique rapide (FAA [6]), qui est une amélioration de l'attaque algébrique standard, est beaucoup plus efficace que ce dernier. L'attaque algébrique rapide a été introduite par Nicolas Courtois [55], et améliorée plus tard par Armknecht dans [4] et par Hawkes et Rose dans [94].

Cette attaque est basée sur l'observation qu'il est possible d'obtenir une équation à bas degré à partir du système d'équations (3.3) en cherchant des relations entre l'état initial du LFSR et plusieurs bits de la fonction de la sortie simultanément, c'est−à−dire, les équations de la forme, pour chaque $i \geq 1$,

$$z_i g(L^{'} \circ L^t(u_1, \ldots, u_N)) = h(L^{'} \circ L^t(u_1, \ldots, u_N)), \tag{3.6}$$

où $(u_1, \ldots, u_N)$ est la clef secrète, g, h sont deux fonctions Booléennes de bas degrés e et d respectivement, avec $e < d$ pour $fg = h$.

En faisant une combinaison linéaire de plusieurs relations de ce type, on peut éliminer les monômes de degré plus grand que d dans l'ANF de h. Il n'est pas nécessaire de connaître les valeurs de z_i pour chercher de telles combinaisons.

Soit $D = \sum_{i=0}^{d} \binom{N}{i}$ et $E = \sum_{i=0}^{e} \binom{N}{i}$. Il a été prouvé par Armknecht et Kraus [5] l'existence des combinaisons linéaires $(\alpha_i)_{0 \leq i \leq D-1}$ tels que

$$h^*(L^{'} \circ L^t(u_1, \ldots, u_N)) = \sum_{i=0}^{D-1} \alpha_i h(L^{'} \circ L^{i+t}(u_1, \ldots, u_N)). \tag{3.7}$$

Par conséquent, le système d'équations (3.6) de degré d peut être transformé a un système d'équations de degré e,

$$h^*(L^{'} \circ L^t(u_1, \ldots, u_N)) = \sum_{i=0}^{D-1} \alpha_i z_{t+i} g(L^{'} \circ L^{i+t}(u_1, \ldots, u_N)). \tag{3.8}$$

Ce polynôme dépend de la suite chiffrante et ne peut pas être pré−calculé. Donc, on obtient une nouvelle attaque qui s'effectue en quatre étapes :

1) Calculer les fonctions g et h.

6. "Fast algebraic attacks" en anglais.

2) Trouver un polynôme $\alpha(x)$ tel que h soit de degré au plus r.
3) Calculer les h^*.
4) Résoudre le nouveau système algébrique (3.8) et retrouver l'état initial.

La complexité de l'attaque : La complexité de l'attaque algébrique rapide est approximativement d'ordre [94],

$$O(\min\{N^{\max[\deg(g)+\deg(fg),\, 3\deg(g)]},\ g \neq 0\}).$$

L'attaque algébrique rapide avec $AI(f) \leq \deg(g)$ n'est autre que l'attaque algébrique, ce qui a mené [47] a étudié *la complexité algébrique rapide*, notée FAC,

$$FAC(f) = \min\{\max[\deg(g) + \deg(fg),\, 3\deg(g)],\ 1 \leq \deg(g) < AI(f)\},$$

où les valeurs ne changent pas en remplaçant f par $1+f$, et $n \leq FAC(f) \leq FAI(f)$. $FAI(f)$ est l'immunité algébrique rapide de la fonction Booléenne f.

L'immunité algébrique rapide : L'immunité algébrique rapide est considérée comme un critère cryptographique très important pour les fonctions Booléennes utilisées dans les systèmes de chiffrement à flot et résistantes contre les attaques algébriques rapides. Ce critère a été introduit dans [112] et utilisé dans [47, 131, 152].

Une définition de l'immunité algébrique rapide a été proposée dans [112] comme suit :

Définition 61. *[112] Soit f une fonction Booléenne à n variables. L'immunité algébrique rapide de la fonction f, notée $FAI(f)$, est définie par :*

$$FAI(f) = \min\left(2AI(f),\ \min_{1\leq \deg(g)<AI(f)}(\deg(g) + \deg(fg))\right).$$

Par cette définition, on a $AI(f) + 1 \leq FAI(f) \leq \deg(f) + 2$ puisque $\deg(f) + 1 \leq \deg(g) + \deg(fg)$ pour toute fonction non$-$constante g de degré algébrique plus petit que $AI(f)$ et $\deg(l) + \deg(lf) \leq \deg(f) + 2$ pour toute fonction affine l.

L'efficacité de l'attaque algébrique rapide : L'efficacité de l'attaque algébrique rapide sur un générateur filtré par une fonction Booléenne f à n variables, dépend de l'existence de deux fonctions Booléennes à n variables, $g \neq 0$ de degré algébrique bas e, et h de degré algébrique d, telles que [55],

$$f(x)g(x) = h(x),\ \forall x \in \mathbb{F}_2^n. \tag{3.9}$$

On définit la notion d'immunité aux attaques algébriques rapides :

Définition 62 ([43], page 117)**.** *(Immunité aux attaques algébriques rapides) Une fonction Booléenne f à n variables est immune aux attaques algébriques rapides s'il est impossible de trouver deux fonctions $g \neq 0$ et h satisfaisant l'équation (3.9), telles que* $\deg(g) = e < \deg(h) = d$ *et* $d + e < n$.

Autrement dit, si f une fonction Booléenne à n variables, et s'il existe une fonction Booléenne $g \neq 0$ à n variables de bas degré algébrique telle que h soit de degré algébrique raisonnable, c'est−à−dire un degré algébrique non large en respectant n, alors une attaque algébrique rapide est faisable.

L'existence de g et h : Il a été prouvé par Courtois [55] que $e + d \geq n$, le nombre de monômes de degrés au plus e et le nombre de monômes de degrés au plus d est strictement plus grand que 2^n, et ça existe [90], alors $g \neq 0$ est de degré algébrique e et h est de degré algébrique au plus d, telle que $fg = h$. Puisque $fg = h$ implique que $fh = ffg = fg = h$, alors h est un annulateur de $(1 + f)$. Si de plus $h \neq 0$ alors son degré est au minimum égal à l'immunité algébrique de f.

On a $(1 + f)g = h$ donc $fg = h + g$. Une haute immunité algébrique n'est pas seulement un critère nécessaire pour résister aux attaques algébriques standard, mais aussi pour résister aux attaques algébriques rapides. Un exemple d'attaque algébrique rapide est celle qui permet de cryptanalyser le générateur par un LFSR filtré *Sfinks*, candidat à l'appel *eSTREAM* [54].

Résistance optimale et presque optimale contre les FAAs : Dans [42, 73], les auteurs ont mentionné qu'une fonction Booléenne f à n variables a une *résistance optimale* contre les attaques algébriques rapides si et seulement s'il n'existe pas une fonction Booléenne non nulle g de degré algébrique au plus e tel que $\deg(g) + \deg(h) < n$ et $e \leq \frac{n}{2}$. Quand on considère la résistance de f contre les attaques algébriques rapides, on doit déterminer si $\deg(h) \geq n - e$ est vérifiée pour toute fonction Booléenne non nulle g de degré au plus e [73, 143]. Si c'est vrai pour tout entier positif $e < \frac{n}{2}$, alors f a une résistance optimale.

La fonction Booléenne f est dite de résistance *presque optimale* si l'inégalité $\deg(h) \geq n - e - 1$ est vérifiée pour tout entier positif $e \leq \frac{n}{2}$.

Définition 63. *[72] Soit g une fonction Booléenne à n variables de degré algébrique au plus e. Une fonction Booléenne f a une résistance presque optimale contre les attaques algébriques rapides si* $\deg(fg) \geq n - e - 1$ *est vérifiée pour tout entier positif* $e < \frac{n}{2}$*, et a une résistance optimale si* $\deg(fg) \geq n - e$ *est vérifiée pour tout entier positif* $e < \frac{n}{2}$*.*

Attaque algébrique "univariée"

Plus récemment, les travaux de Gong, Helleseth et Rønjom [89, 95, 149, 150] ont montré que la représentation univariée est bien plus pertinente que les autres représentations pour évaluer correctement la sécurité des registres filtrés. Pour le cas des attaques algébriques, c'est cette représentation univariée qui donne exactement la valeur de la complexité linéaire de la suite produite, ce qui détermine précisément la sécurité de ce type de générateur.

Dans ces conditions, la complexité de l'attaque dépend du degré de la fonction de filtrage. Au lieu de linéariser le système d'équations multivariées (liant l'état initial et la suite chiffrante), on considère la sortie du générateur filtré comme la sortie d'un seul LFSR. La taille de ce LFSR est la complexité linéaire Λ de la suite chiffrante, définition (5). Cette valeur détermine exactement la complexité de résolution du plus petit système linéaire qui lie les bits de la suite chiffrante et l'état initial.

Le théorème de Blahut [18, 122] implique que la valeur de la complexité linéaire Λ est entièrement déterminée par le nombre de coefficients non nuls dans la représentation univariée de la fonction de filtrage. Plus précisément, soit $c_i \in \mathbb{F}_{2^n}$ pour $0 \leq i \leq 2^n - 2$, les coefficients de la représentation univariée de la fonction Booléenne f, alors la complexité linéaire de la suite sortante d'un LFSR de polynôme caractéristique primitive et filtré par f est donnée par :

$$\Lambda = |\{0 \leq i \leq 2^n - 2,\, c_i \neq 0\}|.$$

Dans le cas de la combinaison de registres, la complexité linéaire croît suffisamment vite avec le degré de la fonction [100, 152]. Rønjom et Helleseth ont observé dans [149] que la complexité linéaire est toujours plus petite que le nombre de monômes présents dans l'attaque algébrique standard.

3.4.2 Attaques algébriques sur les schémas par bloc

Le principe des attaques algébriques décrit par Caude Shannon s'applique aussi aux systèmes de chiffrement par blocs. L'idée a été proposée par Courtois et Pieprzyk [58],

mais contrairement au cas des chiffrements à flot, il n'est pas toujours possible de monter une attaque sur un chiffrement par blocs, puisque l'attaquant doit résoudre un système d'équations algébriques *non surdéterminé*.

L'attaque se fait en exploitant l'existence d'un système d'équations multivariées de bas degré impliquant les bits d'entrée et les bits de sortie des boîtes$-S$ utilisées dans le système de chiffrement, c'est$-$à$-$dire, si F est une boîte$-S$ à n entrées et m sorties, alors on cherche le degré algébrique le plus bas d des relations booléennes entre les entrées et les sorties de F telle que :

$$F(x_1, \ldots, x_n, f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)) = 0, \tag{3.10}$$

où pour tout $1 \leq i \leq m$, les f_i sont des fonctions Booléennes coordonnées de la boîte$-S$. La recherche d'une telle relation de degré d est équivalente à la recherche d'un annulateur de degré d de la fonction caractéristique ϕ_F de la fonction F. La fonction ϕ_F est une fonction Booléenne à $(n+m)$ variables est définie par :

$$\phi_F(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m) = 1 \Leftrightarrow y_i = f_i(x_1, x_2, \ldots, x_n),\ \forall i = 1, \ldots, m.$$

Rappelons qu'un annulateur g de la fonction caractéristique de F est défini par :

$$g(x, y)\phi_F(x, y) = 0,\ \forall (x, y) \in \mathbb{F}_2^n \times \mathbb{F}_2^m,\ \textit{tels que}\ \phi_F(x, y) = 1.$$

Pour une $(n, m)-$fonction, le nombre des inconnues dans l'équation (3.10) est égal à $\sum_{i=0}^{d} \binom{n+m}{i}$ et le nombre d'équations est 2^n, le nombre des relations indépendantes de degré d est au moins $\sum_{i=0}^{d} \binom{n+m}{i} - 2^n$.

L'efficacité réelle des attaques algébriques sur les systèmes de chiffrement par blocs est difficile à étudier, même si elles ont été mises en œuvre récemment sur des systèmes de petite taille [6, 53]. Contrairement aux attaques algébriques sur les systèmes de chiffrement à flot, le nombre global des variables dans le plus large système d'équations exprimant le chiffrement est plus large, et les systèmes d'équations obtenus sont *non surdéterminés*. Alors, personne n'est capable de prédire correctement la complexité de la résolution de tels systèmes polynomiaux.

Dans le cas AES, où la boîte$-S$ est la fonction puissance $x \in \mathbb{F}_{2^8} \rightarrow x^{2^8-2} \in \mathbb{F}_{2^8}$, un exemple de telle fonction est $x^2 y = x$, où $x, y \in \mathbb{F}_{2^8}$. On peut avoir des relations de degré

3 puisque $\sum_{i=0}^{3}\binom{16}{i} = 697 > 2^8 = 256$, la résolution du système d'équations obtenu donne les bits d'entrées de la boîte$-S$. L'attaque contre l'AES a montré mieux, l'existence de 39 relations quadratiques.

3.5 L'immunié algébrique

Avant 2003 un haut degré algébrique des fonctions Booléennes à n variables était considéré comme un bon critère suffisant pour assurer la sécurité du système (3.1) contre les attaques algébriques. En 2003, Courtois et Meier [57] ont montré que ce n'était pas le critère pertinent pour résister aux attaques algébriques et qu'il était parfois possible de mener une attaque même si le degré algébrique de la fonction Booléenne est élevé.

Le concept de l'immunité algébrique d'une fonction Booléenne a été d'abord proposé par Meier et al. dans [124]. Depuis, il était généralisé aux fonctions vectorielles par Armknecht et Krause dans [5]. Les auteurs de [57] ont prouvé que $\lceil \frac{n}{2} \rceil$ est la valeur maximale de l'immunité algébrique d'une fonction Booléenne à n variables. Dans [5] les auteurs ont donné une borne sur l'immunité algébrique des fonctions vectorielles de n entrées et m sorties.

Quelques constructions des fonctions Booléennes et vectorielles avec une immunité algébrique optimale ont été proposées dans la littérature et on note [5], [42] et [176]. Dans [42], Carlet et Feng ont donné une construction d'une classe infinie des fonctions Booléennes dont l'immunité algébrique est optimale, cette construction était obtenue à partir de la borne BCH des codes cycliques, cependant, ils ont aussi donné un lien entre ladite immunité des fonctions Booléennes et les codes cycliques.

3.5.1 L'immunité algébrique des fonctions Booléennes

Avant d'introduire la notion de l'immunité algébrique, on rappelle d'abord la définition des annulateurs.

Définition 64. *[81](Annulateur) Soit H un sous$-$ensemble de $\mathbb{F}_2^n$, une fonction Booléenne $g : \mathbb{F}_2^n \to \mathbb{F}_2$ de l'ensemble $AN(H) = \{g \neq 0; g(x) = 0, \forall x \in H\}$, est dite un annulateur de H.*

La définition de l'immunité algébrique d'un sous$-$ensemble quelconque de $\mathbb{F}_2^n$, notée par $AI(H)$, est alors donnée comme suit :

Définition 65. *[59](Immunité algébrique) Soit H un sous−ensemble de $\mathbb{F}_2^n$. Alors toute fonction $p \neq 0$ nulle sur H est appelée annulateur de H. L'immunité algébrique de H, notée $AI(H)$, est le degré minimal d de tous les annulateurs de H.*

$$AI(H) = \min\{d;\ il\ existe\ une\ fonction\ p \neq 0;\ deg(p) = d;\ p(x) = 0; \forall x \in H\}.$$

D'une manière équivalente, on définit l'immunité algébrique d'une fonction Booléenne à n variables comme suit :

Définition 66. *L'immunité algébrique de la fonction Booléenne f est le plus petit entier d tel que f ou $1+f$ admet un annulateur de degré d.*

$$AI(f) = \min\{deg(g) : 0 \neq g \in \mathbb{B}_n;\ fg = 0 ou\ (f+1)g = 0\}.$$

C'est−à−dire, l'immunité algébrique d'une fonction Booléenne f est le degré minimal atteint par une fonction non nulle de $AN(f) \cup AN(1+f)$. Suivant Meier, Pasalic et Carlet [124], l'immunité algébrique d'une fonction Booléenne est donc donnée par la définition suivante :

Définition 67. *[124](Immunité algébrique) L'immunité algébrique d'une fonction Booléenne f à n variables est définie comme le minimum entre $AI(f^{-1}(1))$ et $AI(f^{-1}(0))$.*

Remarque 68. *Soit f une fonction Booléenne à n variables, nous notons par $LDA(f)$ le degré algébrique le plus bas des annulateurs de f.*

Dans [57] et [79], les auteurs ont démontré que l'immunité algébrique d'une fonction Booléenne à n variables est inférieure ou égale à $\lceil \frac{n}{2} \rceil$. Par ailleurs, quand le nombre de variables n est impair, seules les fonctions équilibrées peuvent atteindre l'immunité algébrique maximale.

Proposition 69. *L'immunité algébrique maximale d'une fonction Booléenne à n variables f est égale à $\lceil \frac{n}{2} \rceil$. Ainsi,*

$$AI(f) \leq \max\{deg(f); \lceil \frac{n}{2} \rceil\}.$$

Preuve. Nous avons vu qu'il existe des annulateurs de f de degré d si $wt(f) < \sum_{i=0}^{d} \binom{n}{i}$. Puisque l'immunité algébrique de f est le degré minimal de tous les annulateurs de f et de $1+f$, alors si nous avons à la fois $wt(f) < \sum_{i=0}^{d} \binom{n}{i}$ et $wt(1+f) = 2^n - wt(f) < \sum_{i=0}^{d} \binom{n}{i}$, alors il existe des annulateurs de degré d de f et de $1+f$, on choisit la plus petite valeur

possible de d pour obtenir l'immunité algébrique de f. D'autre part, on sait que pour toute fonction Booléenne à n variables f :

$$\min\{wt(f),\, wt(1+f)\} \leq 2^{n-1}.$$

Il est facile de voir que $2^{n-1} < \sum_{i=0}^{d} \binom{n}{i}$ si et seulement si $d \geq \lceil \frac{n}{2} \rceil$, d'où le résultat. ■

Lien entre l'immunité algébrique, le poids de Hamming, la non−linéarité et la non−linéarité d'ordre r

La non−linéarité et le degré algébrique sont non corrélés. Il existe des fonctions Booléennes avec une non−linéarité élevée et un bas degré algébrique, avec une non−linéarité faible et un bas degré algébrique, avec une non−linéarité élevée et un degré algébrique élevé mais jamais une non−linéarité maximale et un degré algébrique élevé à cause de la borne de Rothaus (théorème (40)). Il existe aussi des fonctions Booléennes avec une non−linéarité faible et un degré algébrique élevé. Si on remplace le degré algébrique par l'immunité algébrique, ce dernier cas ne peut se faire.

Le résultat suivant lie le poids d'une fonction Booléenne à son immunité algébrique.

Théorème 70. *[39] Soit f une fonction Booléenne à n variables telle que $AI(f) > d$. Alors,*

$$\sum_{i=0}^{d} \binom{n}{i} \leq wt(f) \leq \sum_{i=0}^{n-d-1} \binom{n}{i}.$$

Ainsi,

$$\sum_{i=0}^{AI(f)-1} \binom{n}{i} \leq wt(f) \leq \sum_{i=0}^{n-AI(f)} \binom{n}{i}.$$

Preuve. L'inégalité de gauche doit être satisfaite sinon, le nombre d'équations $wt(f)$ est plus petit que le nombre d'inconnues ce qui implique l'existence d'un annulateur de degré $d < AI(f)$ ce qui est impossible. L'inégalité de droite est obtenue à partir de l'autre en remplaçant f par $1 + f$. ■

Corollaire 71. *Soit f une fonction Booléenne à n variables. S'il existe $d \in \mathbb{N}$, $1 \leq d \leq n$ tel que $wt(f) < \sum_{i=0}^{d} \binom{n}{i}$ ou $wt(f) > \sum_{i=0}^{n-d-1} \binom{n}{i}$, alors $AI(f) \leq d$.*

Les auteurs de [39] ont déduit qu'une non−linéarité faible implique une immunité algébrique faible (une immunité algébrique élevée n'implique pas une non−linéarité élevée, de même, une non−linéarité élevée n'implique pas une immunité algébrique élevée). Nous avons ce qui suit :

$$nl(f) \geq \sum_{i=0}^{AI(f)-2} \binom{n}{i}$$

plus généralement, pour la non−linéarité d'ordre r, nous avons la borne suivante :

$$nl_r(f) \geq \sum_{i=0}^{AI(f)-r-1} \binom{n}{i}.$$

Ces bornes inférieures ont été améliorées pour la non−linéarité d'ordre 1 comme suit :

$$nl(f) \geq 2 \sum_{i=0}^{AI(f)-2} \binom{n-1}{i}$$

La borne sur la non−linéarité d'ordre r a été amélioré par Lobanov [115, 116] comme suit :

$$nl_r(f) \geq 2 \sum_{i=0}^{AI(f)-r-1} \binom{n-r}{i}.$$

Une borne entre $nl_r(f)$ et $FAI(f)$ a été donné dans [168]

$$nl_r(f) \geq \sum_{i=0}^{\lfloor \frac{FAI(f)-r}{2} \rfloor} \binom{n}{i}.$$

En fait, une amélioration un peu plus forte mais plus compliquée est dans [33]. Enfin une borne meilleure dans [129] :

$$nl_r(f) \geq \sum_{i=0}^{AI(f)-r-1} \binom{n}{i} + \sum_{i=AI(f)-2r}^{AI(f)-r-1} \binom{n-r}{i}.$$

En 2008, Carlet et Freng [42] ont mis à jour une famille de fonctions Booléennes et ses bonnes propriétés cryptographiques. Ces fonctions constituent la version Booléenne d'une classe de fonctions vectorielles précédemment étudiée par Feng, Liao et Yang [81].

Lien entre l'immunité algébrique rapide et le degré algébrique

Un lien entre l'immunité algébrique rapide (définition (61)) et le degré algébrique des fonctions Booléennes a été donné dans [132] comme suit :

Proposition 72. *[132] Soit n un entier positif, soit f une fonction Booléenne à n variables. Soit g une fonction Booléenne atteignant $FAI(f)$, alors* $\deg(f) \leq \lfloor \frac{FAI(f)}{2} \rfloor$ *et* $\deg(fg) \geq \lceil \frac{FAI(f)}{2} \rceil$.

3.5.2 L'immunité algébrique des fonctions vectorielles

Comme c'est mentionné dans la section (3.4), les attaques algébriques concernent aussi les fonctions vectorielles utilisées dans les systèmes de chiffrement à flot et de chiffrement par blocs. Cependant, seules les attaques algébriques standard ont été considérées dans la littérature pour les chiffrements à flot utilisant les fonctions vectorielles.

Plusieurs notions de l'immunité algébrique des $(n, m)-$fonctions ont été liées à ces attaques. On rappelle d'abord la définition des annulateurs et on introduit la définition de l'immunité algébrique d'un ensemble.

Définition 73. *(Immunité algébrique d'un sous−ensemble) On appelle un annulateur d'un sous−ensemble E de $\mathbb{F}_2^n$, toute fonction Booléenne à n variables $p \neq 0$, nulle sur E. L'immunité algébrique de E, noté $AI(E)$ est le degré minimal d de tous les annulateurs non nuls de E.*

$$AI(E) = \min\{\deg(p) : 0 \neq p \in \mathbb{B}_n,\ p(x) = 0;\ \forall x \in E\}.$$

Dans [3], Armknecht a donné la définition des annulateurs d'une $(n, m)-$fonction F sous forme polynomiale univariée.

Définition 74. *Soient deux $(n, m)-$fonctions F et G, alors G est un annulateur de F si $G(x)F(x) = 0$, pour tout $x \in \mathbb{F}_{2^n}$.*

Les différentes notions de l'immunité algébrique des $(n, m)-$fonctions ont été étudiées dans [5, 7, 34]. La première généralisation de l'immunité algébrique aux $(n, m)-$fonctions a été introduite par Armknecht dans [4].

Définition 75. *(Immunité standard) L'immunité algébrique standard d'une $(n, m)-$fonction F, noté $AI(F)$, est définie comme*

$$\begin{aligned} AI(F) &= \min\{AI(F^{-1}(z)),\ z \in \mathbb{F}_2^m\}. \\ &= \min_{z \in \mathbb{F}_2^m} \min\{d,\ deg(p(x)) = d,\ p(x) = 0;\ \forall x \in F^{-1}(z)\}. \end{aligned}$$

Autrement dit, l'immunité algébrique d'une $(n, m)-$fonction est :

$$\min\{AI(F^{-1}(z)), \textit{pour tout } z \in \mathbb{F}_2^m\}.$$

Dans [5], nous trouvons notamment le résultat suivant.

Théorème 76. *L'immunité algébrique standard d'une $(n,m)-$fonction est au plus d, où d est le plus petit entier tel que $\sum_{i=0}^{d}\binom{n}{i} > 2^{n-m}$.*

Preuve. Il existe au moins un élément z tel que $|F^{-1}(z)| \leq 2^{n-m}$. Soit g une fonction de degré algébrique égal à d et de ANF

$$g(x) = a_0 + \sum_{i=0}^{n} a_i x_i + \sum_{1\leq i\leq j\leq n} a_{i,j}x_i x_j + \ldots + \sum_{i_1\leq i_2\leq \ldots i_d\leq n} a_{i_1,i_2,\ldots i_d}x_{i_1}x_{i_2}\ldots x_{i_d}.$$

Alors, g est un annulateur de $|F^{-1}(z)|$ si et seulement si les coefficients de son ANF satisfont un système d'équations linéaires homogènes correspondant au fait que $g(x) = 0$ pour tout $x \in F^{-1}(z)$. Dans ce système, on a $|F^{-1}(z)|$ équations linéaires à $\sum_{i=0}^{d}\binom{n}{i}$ inconnues. Comme le nombre d'équations est strictement inférieur au nombre d'inconnues, le système a nécessairement des solutions non triviales. ■

La deuxième notion de l'immunité algébrique des $(n,m)-$fonctions, et qui joue aussi un rôle important dans le cadre des chiffrements par blocs, est *l'immunité algébrique du graphe* [5, 34, 49, 58].

Définition 77. *(Immunité algébrique du graphe) L'immunité algébrique du graphe d'une $(n,m)-$fonction F, noté $Al_{gr}(F)$, est l'immunité du graphe $\{(x,(F(x))), x \in \mathbb{F}_2^n\}$ de la boîte$-S$*

La troisième notion de l'immunité algébrique des $(n,m)-$fonctions est *l'immunité algébrique par composantes*, introduite dans [34].

Définition 78. *(Immunité algébrique par composantes) L'immunité algébrique par composantes d'une $(n,m)-$fonction F, noté $Al_{comp}(F)$, est égale au minimum des immunités algébriques de ses fonctions composantes.*

$$Al_{comp}(F) = \min_{v\in\mathbb{F}_2^m\setminus\{0\}} AI(v\cdot F).$$

Plusieurs observations sur les trois notions de l'immunité algébrique des $(n,m)-$fonctions ont été faites dans [34]. L'immunité algébrique standard n'est intéressante que pour les petites valeurs de m et est efficace seulement dans ce cas : si $m \geq n$, l'immunité algébrique standard est au plus égale à 1 pour $m = n$, et si $m > n$ l'immunité algébrique est nulle. L'immunité algébrique par composantes peut être utilisée pour étudier les deux autres notions de l'immunité algébrique des $(n,m)-$fonctions. Notons que $AI(F) \leq Al_{comp}(F)$, puisque Al est non$-$décroissante et aussi puisque $AI(F) = AI(F^{-1}(H))$ pour un hyperplan affine H de $\mathbb{F}_2^m$. Nous avons aussi que $Al_{gr}(F) \leq Al_{comp}(F) + 1$, puisque si g est un

annulateur non nul de $v \cdot F$, $v \neq 0$, alors $h(x, y) = g(x)(v \cdot y)$ est un annulateur non nul du graphe de F, et si g est un annulateur non nul de $v \cdot (F+1)$, alors $h(x, y) = g(x)(v \cdot y) + g(x)$ est un annulateur non nul du graphe de F.

Dans [5], les auteurs ont observé que, pour une $(n, m)-$fonction F,

$$AI(F) \leq Al_{gr}(F) \leq AI(F) + m.$$

Théorème 79. *L'immunité algébrique standard d'une $(n, m)-$fonction est,*

$$AI(F) \leq d_{n,m} \leq d_{n,m-1} \leq \ldots \leq d_{n,1} = \lceil \frac{n}{2} \rceil,$$

où $d_{n,m}$ est le plus petit entier tel que $\sum_{i=0}^{d_{n,m}} \binom{n}{i} > 2^{n-m}$.

En effet, il existe au moins un élément z tel que $|F^{-1}(z)| \leq 2^{n-m}$ et par la relation (70), avec $\phi = 1_{|F^{-1}(z)|}$, on a $\sum_{i=0}^{AI(\phi)-1} \binom{n}{i} \leq 2^{n-m}$ et alors $AI(\phi) - 1 < d_{n,m}$. Puisque $AI(F) \leq AI(\phi)$. Cela donne la preuve de la première inégalité, observé dans [5] et prouvé dans [81].

D'une manière similaire, les auteurs de [5] ont aussi prouvé que :

Théorème 80. *L'immunité algébrique du graphe d'une $(n, m)-$fonction est,*

$$Al_{gr}(F) \leq D_{n,m} \leq D_{n,m-1} \leq \ldots \leq D_{n,1} = \lceil \frac{n+1}{2} \rceil,$$

où $d_{n,m}$ est le plus petit entier tel que $\sum_{i=0}^{D_{n,m}} \binom{n+m}{i} > 2^n$.

Notons que $D_{n,m} = d_{n+m,m}$. Nous avons $D_{n,m} \leq n - m$, ce qui implique $D_{n,m} \leq n$ puisque $\sum_{i=0}^{n-m} \binom{n-m}{i} = 2^{n-m}$.

Lien entre l'immunité algébrique standard, l'immunité algébrique de graphe, l'immunité algébrique par composantes et la non−linéarité d'ordre 1 et r : Comme prouvé dans [33], la borne sur la non−linéarité des fonctions Booléennes donnée par Lobanov a été généralisée aux $(n, m)-$fonctions comme suit :

$$nl(F) \geq 2^m \sum_{i=0}^{AI(F)-2} \binom{n-1}{i},$$

Pour la la non−linéarité d'ordre r, nous avons ce qui suit,

$$nl_r(F) \geq 2^m \sum_{i=0}^{AI(F)-r-1} \binom{n-r}{i},$$

Nous avons alors

$$nl_r(F) \geq 2^{m-1} \sum_{i=0}^{AI(F)-r-1} \binom{n}{i} + 2^{m-1} \sum_{i=AI(F)-2r}^{AI(F)-r-1} \binom{n-r}{i}.$$

En appliquant la borne de Lobanov aux fonctions composantes de F, on obtient la borne suivante :

$$nl(F) \geq 2 \sum_{i=0}^{Al_{comp}(F)-2} \binom{n-1}{i}.$$

Pour la non−linéarité d'ordre r, nous avons ce qui suit,

$$nl_r(F) \geq 2 \sum_{i=0}^{Al_{comp}(F)-r-1} \binom{n-r}{i}.$$

Alors

$$nl_r(F) \geq \sum_{i=0}^{Al_{comp}(F)-r-1} \binom{n}{i} + \sum_{i=Al_{comp}(F)-2r}^{Al_{comp}(F)-r-1} \binom{n-r}{i}.$$

L'inégalité $Al_{comp}(F) \geq Al_{gr}(F) - 1$ implique :

$$nl(F) \geq 2 \sum_{i=0}^{Al_{gr}(F)-3} \binom{n-1}{i}.$$

Pour la non−linéarité d'ordre r, l'inégalité $Al_{comp}(F) \geq Al_{gr}(F) - 1$ implique :

$$nl_r(F) \geq 2 \sum_{i=0}^{Al_{gr}(F)-r-2} \binom{n-r}{i}.$$

d'où

$$nl_r(F) \geq \sum_{i=0}^{Al_{gr}(F)-r-2} \binom{n}{i} + \sum_{i=Al_{gr}(F)-2r-1}^{Al_{gr}(F)-r-2} \binom{n-r}{i}.$$

Chapitre 4

Fonctions Booléennes, vectorielles et les codes correcteurs

Depuis longtemps, les fonctions cryptographiques ont été utilisées pour construire des codes linéaires. Comme les codes de Reed−Muller et les codes de Kerdock qui ont été construits à partir des fonctions Booléennes. Depuis, de nombreuses tentatives de constructions de codes linéaires avec de bons paramètres ont été faites, par conséquent, de nombreux progrès ont été réalisés dans ce sens.

Nous commencerons ce chapitre par donner un rappel sur les codes correcteurs d'erreurs : les codes linéaires, les codes cycliques et les codes LCD. Ensuite, nous donnerons une description du contexte d'utilisation des fonctions Booléennes et vectorielles dans la théorie des codes.

4.1 Les codes correcteurs d'erreurs

Les codes correcteurs d'erreurs ont été inventés pour permettre la transmission de l'information à travers un canal de communication peu sûr. La transmission de l'information binaire sur des lignes bruitées présentant un risque d'erreurs variable selon les cas, il s'agit de trouver un moyen de les corriger à la réception de l'information, au prix d'une certaine redondance, tout en minimisant dans chaque situation le temps d'occupation de la ligne.

L'idée générale pour parvenir à la détection et la correction de ces erreurs est d'ajouter des données supplémentaires, appelées *redondances*, au message pour être utilisée par le récepteur et pouvoir détecter les erreurs et donc reconstruire l'information originale. Si cette information est envoyée sans redondance, la plus petite altération peut en effet

entraîner des fausses interprétations à l'arrivée. Il s'agit alors pour l'émetteur d'envoyer des messages sous une forme de suite binaire divisée en blocs de taille fixe égale à k, chaque bloc est dit un *mot*, codé individuellement en *mot de code* de longueur n, où $n \geq k$. L'ensemble des mots de code est alors un *code*. Cependant, n symboles parmi les k bits du message sont transmis et nous appelons *taux de transmission* le rapport $\frac{k}{n}$, qui est la vitesse de transmission. Le taux de transmission doit être le plus possiblement élevé, puisque les messages doivent être envoyés le plus rapidement possible et aussi d'une manière fiable.

4.1.1 Les codes linéaires

Plaçons-nous dans le contexte des espaces vectoriels définis sur $\mathbb{F}_2$. Tout espace de ce type est muni de la métrique de Hamming, cette dernière étant définie par la distance suivante : soient u et v deux vecteurs de cet espace, la distance entre u et v est donnée par

$$d_H(u, v) = |\{i,\, u_i \neq v_i\}|.$$

De plus, à tout vecteur u on associe son poids de Hamming, $wt(u)$, qui est défini comme la distance entre u et le vecteur nul.

Un code linéaire $\mathcal{C}$ de longueur n défini sur $\mathbb{F}_2$ est un sous−espace vectoriel de $\mathbb{F}_2^n$. Chacun de ses éléments est appelé mot du code $\mathcal{C}$. On associe à $\mathcal{C}$, outre sa longueur, une dimension k correspondant à la dimension de l'espace vectoriel (le code contient alors 2^k mots de code) ainsi qu'une distance minimale d (c'est la plus petite distance séparant deux mots du code). On dit que $\mathcal{C}$ est un $[n, k, d]-$code linéaire binaire.

Matrice génératrice : Un code linéaire $\mathcal{C}$ est caractérisé par sa matrice génératrice, G : c'est une matrice à k lignes (linéairement indépendantes) et n colonnes, à coefficients dans $\mathbb{F}_2$, et telle que $\{Gu,\, u \in \mathbb{F}_2^k\} = \mathcal{C}$.
Un tel code est dit autocomplémentaire s'il contient le mot tout-à−un [117].

Distance minimale : Soit $\mathcal{C}$ un $[n, k, d]-$code linéaire, on remarque directement que $k \leq n - d + 1$ (cette inégalité est la *borne de Singleton*), cette borne impose à une $[n, k, d]-$code linéaire d'avoir au moins $r \geq d - 1$ chiffres de redondance. Un code qui atteint cette borne est dit *MDS*[1].
La distance de Hamming entre c et c', notée $d_H(c, c')$, est le nombre de coordonnées pour

1. maximum distance séparable

lesquelles c et $c^{'}$ diffèrent, i.e.,

$$d_H(c, c^{'}) = |\{i \in \{1, \ldots, n\},\, c_i \neq c_i^{'}\}|.$$

La distance minimale, notée $d_{\min}$, d'un code linéaire est

$$d_{min}(\mathcal{C}) = \min_{(c,c^{'}) \in \mathcal{C} \times \mathcal{C};\, c \neq c^{'}} d_H(c, c^{'}).$$

Puisque $\mathcal{C}$ est un code linéaire, donc $d_H(c, c^{'}) = wt(c - c^{'})$, alors

$$d_{\min}(\mathcal{C}) = \min_{c \in \mathcal{C}, c \neq 0} |supp(c)|.$$

Il existe des familles de codes pour lesquelles la distance minimale est connue, la plupart sont des codes algébriques. Dans le cas général, Vardy [163] a montré que l'estimation de la distance minimale d'un code linéaire est un problème difficile, et la décision de ce problème est dite $\mathcal{NP}-$complète, alors certaines méthodes et algorithmes ont été proposés, comme l'utilisation des bases de Gröbnner dans [9] et de l'algorithme de Brouwer et Zimmemann [15]. Ce dernier a été developé dans [111].

Code dual : La matrice génératrice d'un code linéaire est étudiée pour générer des mots de code, mais pour vérifier si le mot de longueur n reçu est un mot de code. Par contre, la caractérisation des mots de codes est obtenu grâce à la matrice génératrice du code *dual*.

Définition 81. *(Le code dual) Soit $\mathcal{C}$ un code linéaire binaire de paramètres $[n, k]$. On définit le code dual de $\mathcal{C}$, que l'on note $\mathcal{C}^{\perp}$, comme l'ensemble des mots $u \in \mathbb{F}_2^n$ tels que le produit scalaire $u \cdot v$ soit nul pour tout mot $v \in \mathcal{C}$. $\mathcal{C}^{\perp}$ est un code linéaire binaire de paramètres $[n, n-k]$.*

La matrice génératrice H du $\mathcal{C}^{\perp}$ est appelée la matrice de contrôle du code $\mathcal{C}$. Un autre paramètre important est le *rayon de recouvrement*, qui est défini comme suit.

Définition 82. *Soit $\mathcal{C}$ un $[n, k, d]$-code linéaire binaire. On appelle rayon de recouvrement de $\mathcal{C}$ le plus petit entier ρ tel que l'ensemble de toutes les boules de rayon ρ (pour la distance de Hamming) centrées en les mots de $\mathcal{C}$ recouvre tout l'espace $\mathbb{F}_2^n$.*
On peut également le voir comme

$$\max_{x \in \mathbb{F}_2^n} \min_{y \in \mathcal{C}} wt(x + y).$$

En s'intéressant aux codes linéaires, les performances de codage et décodage avec correction d'erreurs sont donc nettement améliorées. Cependant, il existe une classe de codes linéaires qui améliore encore la facilité du calcul, et la probabilité de l'implémenter simplement sur des circuits électroniques. Cette classe est appelée les *codes cycliques* [117].

4.1.2 Les codes cycliques

La notion des codes cycliques apparat pour la première fois par l'intermédiaire de Prange, [145]. Ils suscitent beaucoup d'intérêt de par leur structure algébrique. Ils sont utilisés à la fois pour corriger les erreurs isolées et les erreurs en rafale.

On appelle code cyclique de longueur n tout $\mathbb{F}_q$ sous−espace vectoriel $\mathcal{C}$ de $\mathbb{F}_q^n$ stable par décalage circulaire [[117], page. 188], c'est−à−dire, tel que :

$$(c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C} \Rightarrow (c_{n-1}, c_0, \ldots, c_{n-2}) \in \mathcal{C}.$$

Cependant, il est commode d'identifier les vecteurs de $\mathbb{F}_q^n$ avec l'ensemble des polynômes de $\mathbb{F}_q[X]/(X^n - 1)$ de degré inférieur ou égal à $n - 1$ par correspondance

$$(c_0, c_1, \ldots, c_{n-1}) \to c(X) = c_0 + c_1 X + \ldots + c_{n-1} X^{n-1}.$$

Le polynôme $c_0 + c_1 X + \ldots + c_{n-1} X^{n-1}$ est appelé *polynôme représentant* du mot de code $(c_0, c_1, \ldots, c_{n-1})$, alors le code $\mathcal{C}$ est un code cyclique si et seulement s'il est un idéal du $\mathbb{F}_q[X]/(X^n - 1)$, c'est−à−dire qu'il satisfait $f\mathcal{C} \subseteq \mathcal{C}$, pour tout élément non nul $f \in \mathbb{F}_q[X]/(X^n - 1)$.

On appelle *polynôme générateur* du code cyclique $\mathcal{C} \subset \mathbb{F}_q^n$ le polynôme non nul de plus petit degré représentant un élément de $\mathcal{C}$.

Théorème 83. *Soit $\mathcal{C}$ un code cyclique de longueur n et soit $g(X)$ son polynôme générateur. Alors*

1) *Le code $\mathcal{C}$ est l'idéal $(g(X))$ de l'anneau $\mathbb{F}_q[X]/(X^n - 1)$.*
2) *La dimension de $g(X)$ divise $X^n - 1$ dans $\mathbb{F}_q[X]$ et dans $\mathbb{F}_q[X]/(X^n - 1)$.*
3) *La dimension de $\mathcal{C}$ en tant que $\mathbb{F}_q[X]$ espace vectoriel, égale $n - \deg(g(X))$ et une base en est $(g(X), Xg(X), \ldots, X^{n-\deg(g)-1} g(X))$.*

Puisque le code dual d'un code cyclique est un code cyclique, le polynôme générateur de ce dernier est alors donné par le théorème suivant,

Théorème 84. *(polynôme générateur du code dual) Soit $g(X)$ le polynôme générateur d'un code cyclique $\mathcal{C}$ de longueur n sur $\mathbb{F}_q$ et de dimension $n - \deg(g)$. Soit $h(X) = (h_0 + h_1X + \ldots + h_{k-1}X^{k-1} + X^k) = (X^n - 1)/g(X)$. Alors le polynôme générateur du code dual $\mathcal{C}^\perp$ de $\mathcal{C}$ est le polynôme unitaire réciproque de $h(X)$,*

$$h_0^{-1}(h_0X^k + h_1X^{k-1} + \ldots + h_{k-1}X + 1) = h_0^{-1}X^k h(\frac{1}{X}),$$

Tandis que la dimension d'un code cyclique se déduit immédiatement de son polynôme générateur, c'est l'étude des *racines* du polynôme minimal qui permet d'évaluer la distance minimale, (le cas échéant dans un corps d'extension de $\mathbb{F}_q$).

Les zéros du code : Les zéros du polynôme générateur sont de la forme $\{\alpha^i,\ i \in I\}$, où α est un élément primitif de $\mathbb{F}_{q^m}$ et $I \in \mathbb{Z}/n\mathbb{Z}$ est l'ensemble de définition du code, c'est−à−dire, l'union des classes cyclotomiques de q modulo $n = q^m - 1$, où une classe cyclotomique de q modulo n est définie comme $\{(qn^j \mod n) \in \mathbb{Z}_n : j \in \mathbb{N}\}$.
Les éléments α_i, $i \in \mathbb{Z}/n\mathbb{Z}\backslash I$ sont les non−zéros du code cyclique. Le polynôme générateur de $\mathcal{C}^\perp$ est le polynôme unitaire réciproque du quotient de $X^n - 1$ par $g(X)$, et son ensemble de définition est l'ensemble $\{n - i;\ i \in \mathbb{Z}/n\mathbb{Z} \setminus I\}$.

4.1.3 Les codes LCD

Les codes LCD sont des codes linéaires dont l'intersection avec leurs duals est triviale. Ces codes ont été introduits par Massey dans [118]. Massey a notamment montré que les codes LCD asymptotiquement bons existent [118], ensuite, Sendrier a prouvé dans [155] que les codes LCD sont des codes asymptotiquement bons. Une condition suffisante et nécessaire pour qu'un code cyclique ai un dual complémentaire est donné par Yang et Massey dans [173]. Mutto et Lal ont construit un code réversible dans $\mathbb{F}_q$ [136].

Lorsque les codes LCD sont binaires, il a été montré récemment dans [44] qu'ils jouent un rôle important dans les mises en œuvre de blindage contre les attaques par canal auxiliaire [2] [44] et les attaques par injection de fautes [3]. Ils sont également utilisés dans les contre−mesures pour analyser des canaux latéraux passifs−actifs [4] sur les cryptosystèmes intégrés ([44] pour plus de détails).

2. "Side−channel attacks ou SCAs" en anglais.
3. "Fault injection attack ou FIA" en anglais.
4. "passive and active side channel" en anglais.

Les attaques par canaux cachés : Dans le but d'assurer la *confidentialité* et *l'intégrité* des données échangées, les informations sensibles sont protégées par des systèmes cryptographiques. Les algorithmes de sécurité, comme AES et RSA (qui ont été démontrés comme mathématiquement sûrs), sont implémentés au niveau matériel de ces systèmes. En effet, que ce soit sous forme logicielle ou matérielle, une implémentation d'un algorithme peut être sujet à des fuites d'information : un processeur émet des rayonnements, il consomme de l'énergie ; un algorithme a un temps d'exécution plus ou moins long ; la température d'un circuit ou son rayonnement magnétique. L'analyse de ces activités permet de déduire de nombreuses informations sur le fonctionnement de ces algorithmes (une sécurité mathématique ne garantit pas forcément une sécurité physique lors de l'utilisation du chiffrement).

Le système devient alors vulnérable à des attaques dites matérielles. Parmi ces attaques, on trouve les attaques appelées *attaques par canaux auxiliaires*, aussi connues par *attaques par canaux cachés*, qui sont considérées comme les plus redoutables et les plus efficaces contre les systèmes embarqués modernes. Elles permettent, en observant le fonctionnement des composants électroniques (consommation de courant, émissions électromagnétiques, émissions lumineuses ...), et en analysant les fuites d'énergie, l'obtention des informations sur le secret manipulé. Ces attaques se basent généralement sur une connaissance approfondie de l'architecture du matériel et portent sur différents types de données :

- **L'attaque par temps de calcul :** consiste à mesurer et analyser le temps mis pour effectuer certaines opérations cryptographiques dans le but de découvrir des informations secrètes.
- **L'attaque temporelle :** Elle est basée sur une comparaison du temps mis pour effectuer certaines opérations.
- **Les attaques par injection de fautes :** Elles consistent en l'introduction volontaire d'erreurs dans le système pour provoquer certains comportements révélateurs, comme obliger le processeur à faire une faute pour en déduire des informations.
- **Analyse de rayonnement électromagnétique :** Elle est basée sur le fait qu'une particule chargée de grande énergie émet un rayonnement électromagnétique. Ce rayonnement est analysé pour déduire l'information secrète.

Les codes LCD :

Définition 85. *Le code linéaire $\mathcal{C}$ est dit un code LCD si $\mathcal{C}$ et $\mathcal{C}^{\perp}$ sont complémentaires, c'est−à−dire, $\mathcal{C} \cap \mathcal{C}^{\perp} = \{0\}$.*

Nous avons alors $\mathbb{F}_2^n = \mathcal{C} + \mathcal{C}^\perp$.

Les prochaines caractérisations des codes LCD sont due à Massey [118].

Proposition 86. *Soit $\mathcal{C}$ un code linéaire. Soit G la matrice génératrice de $\mathcal{C}$. Alors les propriétés suivantes sont équivalentes :*

1) $\mathcal{C}$ est un code LCD.

2) La matrice GG^t est inversible.

3) La matrice HH^t est inversible.

Soit $f(x) = f_h x^h + f_{h-1}x^{h-1} + \dots f_1 x + f_0$ un polynôme dans $\mathbb{F}_q$ avec $f_h \neq 0$ et $f_0 \neq 0$, alors le polynôme réciproque $\tilde{f}$ de f est défini par :

$$\tilde{f}(x) = f_0^{-1} x^h f(x^{-1}).$$

Rappelons tout d'abord la définition d'un polynôme auto−réciproque.

Définition 87. *(polynôme auto−réciproque) Le polynôme $f(x)$ est un polynôme auto−réciproque si et seulement si $f(x) = f_0^{-1} x^h f(x^{-1})$.*

Les conclusions du théorème suivant sont connues dans la littérature, et sont faciles à prouver. Nous en utiliserons quelques−unes plus tard.

Théorème 88. *[119] Soit $\mathcal{C}$ un code cyclique de polynôme générateur $H(x)$, alors les propriétés suivantes sont équivalentes.*

(i) $\mathcal{C}$ est un code LCD.

(ii) $\mathcal{C}$ est généré par le polynôme auto-réciproque $H(x)$.

Par conséquent, le fait que $\mathcal{C}$ soit un code LCD d'une grande distance minimale améliore simultanément la résistance contre les attaques par canaux cachés et les attaques par injection de fautes.

4.2 Contexte de l'utilisation

Les fonctions Booléennes et vectorielles jouent un rôle très important, toute fonction Booléenne à n variables peut être spécifiée par un vecteur binaire de longueur 2^n représentant la liste des valeurs des fonctions Booléennes à n variables (définition (13)). Ainsi, chaque code de longueur 2^n peut être défini comme un sous−ensemble de fonctions Booléennes, telles que les codes de Reed−Muller.

4.2.1 Les codes de Reed−Muller et les fonctions Booléennes

Les codes de Reed−Muller ont été découverts par Muller dans [135], puis dans [147], Reed a donné un algorithme de décodage très performant, ainsi qu'une description très claire. Ces codes ont fait l'objet de nombreuses recherches, même si plusieurs questions à leur sujet restent encore ouvertes. Dans [121], Massey a motionné qu'en creusant profondément dans presque tous les problèmes algébriques de la cryptographie ou de la théorie des codes, on trouve les codes de Reed−Muller.

Les fonctions Booléennes sont fortement liées aux codes de Reed−Muller :

Soit f une fonction Booléenne, notant par ω_f le vecteur de valeurs de f (définition(13)), tel que pour tout $0 \leq i \leq 2^n - 1$:

$$\omega_f = (f(u_0), f(u_1), \ldots, f(u_{2^n-1})),\ u_i \in \mathbb{F}_2^n.$$

Définition 89. *(Code de Reed−Muller) Pour tout r compris entre 0 et n, on appelle code de Reed-Muller d'ordre r, et on note $RM(r,n)$, l'ensemble des vecteurs de longueur 2^n représentant le vecteur des valeurs des fonctions Booléennes de degrés algébriques au plus r :*

$$RM(r,n) = \{\omega_f | f \in \mathbb{B}_n,\ \deg(f) \leq r\}.$$

On appelle code de Reed−Muller raccourci d'ordre r et on note $RM^*(r,n)$, l'ensemble des mots du code $RM(r,n)$ privés de leur première coordonnée.

1) Pour $r = n$, $RM(n,n)$ est l'ensemble des fonctions Booléennes tout entier.

2) Pour $r = 0$, $RM(0,n)$ est la paire constituée des deux fonctions constantes (0 et 1).

3) Pour $r = 1$, $RM(1,n)$ est l'ensemble des formes affines sur l'espace vectoriel $\mathbb{F}_2^n$.

4) Pour $r = n-1$, $RM(n-1,n)$ est l'ensemble des fonctions Booléennes de poids pair.

La dimension du code $RM(r,n)$ est $k = \sum_{i=0}^{r} \binom{n}{i}$ (depuis que c'est le nombre des monômes de degrés au plus r), et de distance minimale $d_{min} = 2^{n-r}$, alors $RM(r,n)$ corrige $[\frac{1}{2}(d_{min} - 1)]$ erreurs.

Proposition 90. *Une fonction Booléenne à n variables appartient au code $RM(r,n)$ si et seulement si, pour tout $a \in \mathbb{F}_2^n$, la fonction $f(x)+f(x+a)$ appartient au code $RM(r-1,n)$.*

Remarque 91. *Cette propriété peut être étendue aux fonctions vectorielles.*

Un paramètre important du code $RM(r,n)$ est la distance minimale $d = d_{RM(r,n)}$.

Théorème 92. *[117] La distance minimale de $RM(r,n)$ est égale à 2^{n-r}.*

La preuve a été donné dans [[117], page 375].

Le code dual du code de Reed−Muller est un code de Reed−Muller.

Théorème 93. *Pour tout entier positif n et tout entier non−négatif $r < n$, alors*

$$RM(r,n)^{\perp} = RM(n-r-1,n),$$

où le code dual $RM(r,n)^{\perp} = \{f \in \mathcal{B}_n;\ \forall g \in RM(r,n),\ fg = \sum_{x\in\mathbb{F}_2^n} f(x)g(x) = 0\}$.

Preuve. Soit $\omega_f \in RM(n-r-n,n)$ et $\omega_g \in RM(r,n)$, alors $\deg(f) \leq n-r-1$ et $\deg(g) \leq r$. Ainsi $\deg(fg) \leq n-1$. Donc $\omega_{fg} \in RM(n-1,n)$ et le poids de Hamming du vecteur ω_{fg} est pair, c'est−à−dire que $wt(\omega_{fg}) \equiv 0(\mod 2)$ et $RM(n-r-1,n) \subseteq RM(r,n)^{\perp}$.
D'un autre coté,

$$\begin{aligned}&\dim RM(n-r-1,n) + \dim RM(r,n) = \\ &= 1 + \binom{n}{1} + \ldots + \binom{n}{n-r-1} + 1 + \binom{n}{1} + \ldots + \binom{n}{r} = 2^n,\end{aligned}$$

alors $RM(n-r-1,n) = RM(r,n)^{\perp}$. ■

Nous identifions l'espace−vectoriel $\mathbb{F}_2^n$ par le corps $\mathbb{F}_{2^n}$, on considère la représentation trace des fonctions Booléennes définie dans la section (2.2.1). Donc la famille de ces fonctions $tr_n(ax^j)$, tel que $a \in \mathbb{F}_{2^n}^*$ et $wt(j) \leq n-r-1$ génère $RM(n-r-1,n)$.

Une fonction Booléenne f appartient a $RM(r,n)$ si et seulement si, pour tout entier non nul j tel que $wt(j) \leq n-r-1$, on a $\sum\limits_{x\in\mathbb{F}_{2^n}} f(x)tr_n(ax^j) = tr_n(a\sum\limits_{x\in\mathbb{F}_{2^n}} f(x)x^j) = 0$ pour tout $a \in \mathbb{F}_{2^n}$, comme exprimer dans le corollaire suivant :

Corollaire 94. *Pour tout entier positif n et tout entier non−négatif $r < n$, une fonction Booléenne f dans $\mathbb{F}_{2^n}$ appartient a $RM(r,n)$ si et seulement si, pour tout entier non nul j tel que $wt(j) \leq n-r-1$, on a : $\sum_{x\in\mathbb{F}_{2^n}} f(x)x^j = 0$.*

L'immunité algébrique et les codes de Reed−Muller

L'immunité algébrique des fonctions Booléennes est reliée au comportement de $RM(n, m)$ sur le canal à effacements (section (1)) en présence du motif d'effacements f et $1 + f$.

Lemme 95. *Chercher les annulateurs de degré algébrique au plus r d'une fonction Booléenne f est la même chose que décoder $RM(r, n)$ en présence d'un motif d'effacements égale au vecteur des valeurs de $1 + f$. L'ensemble des annulateurs est alors exactement l'ensemble $\mathcal{I}$.*

Preuve. Supposons que l'on cherche des annulateurs de degré algébrique au plus r d'une fonction Booléenne f. On cherche donc une fonction g qui prend la valeur 0 en tous les points où f vaut 1. C'est équivalent à la recherche d'un mot de code de $RM(r, n)$ (associé à g) qui aurait pu être transmis lorsque l'on reçoit le mot $(0, \ldots, 0)$ avec les positions en dehors du support de f effacées. L'ensemble des annulateurs de f n'est donc rien d'autre que l'ensemble $\mathcal{I}$ (définition(1)) lorsque le motif d'effacements est $1 + f$. ■

Une fonction Booléenne f à n variables est d'immunité algébrique plus grande que r si et seulement si $supp(f)$ et $supp(1+f)$ contient un ensemble d'information pour $RM(r, n)$.

L'immunité algébrique des fonctions Booléennes dépend de l'ensemble des annulateurs de f et de $1 + f$ (définition (3.5)), mais ces deux ensembles ne sont pas complètement indépendants.

Proposition 96. *Soit f une fonction Booléenne à n variables et de poids $2^n - k$, où k est la dimension de $RM(r, n)$, alors si f n'admet pas d'annulateur de degré algébrique r, $1 + f$ n'admet pas d'annulateur de degré algébrique $n - r + 1$.*

Si n est impair et r correspondant à l'immunité algébrique maximale possible $\frac{n-1}{2}$.

Proposition 97. *Soit n impair, alors il y a une bijection entre les fonctions d'immunité algébrique maximale et les ensembles d'information du code auto−dual $RM(\frac{(n-1)}{2}, n)$.*

Preuve. Comme dans le cas $k = 2^{n-1}$, toute fonction Booléenne f d'immunité algébrique maximale, $supp(f)$ et $supp(1 + f)$ sont des ensembles d'information pour $RM(\frac{(n-1)}{2}, n)$. ■

Si n est pair, chaque ensemble d'information nous donne une fonction Booléenne d'immunité algébrique maximale.

La non−linéarité et les codes de Reed−Muller

La non−linéarité des fonctions Booléennes les codes de Reed−Muller : La non−linéarité d'une fonction Booléenne f est égale à la distance minimale du code $RM(1,n) \cup (f + RM(1,n))$. Étant donné que la distance minimale de $RM(r,n)$ est égale à 2^{n-r} pour tout $r \leq n$, alors $nl_r(f) \geq 2^{n-r-1}$ pour toute fonction Booléenne f de degré algébrique égal à $r+1 \leq n$.

Si f et h sont deux fonctions distinctes de $RM(r,n)$, alors

$$2^{n-r} \leq d(g,h) \leq d(g,f) + d(f,h),$$

ce qui implique :

$$2^{n-r} = \min_{g,h \in RM(r,n)} d(g,h) \leq \min_{g,h \in RM(r,n)} d(g,f) + \min_{g,h \in RM(r,n)} d(f,h) = 2nl_r(f),$$

d'où $nl_r(f) \geq 2^{n-r-1}$.

La non−linéarité des fonctions vectorielles les codes de Reed−Muller :
Dans [38, 165], les auteurs ont observé l'existence d'une relation entre la non−linéarité maximale des (n,m)−fonctions et les paramètres du code de Reed−Muller d'ordre 1.

Soit le code binaire $\mathcal{C}$ de paramètres $[2^n, K, D]$ dont le code de Reed−Muller $RM(1,n)$ est un sous−code. Soit $(b_1, \ldots, b_K)$ une base de $\mathcal{C}$ complémentaire de la base $(b_1, \ldots, b_{n-1})$ de $RM(1,n)$. Les fonctions Booléennes à n variables qui correspondent aux vecteurs $b_{n+2}, \ldots, b_K$ sont les fonctions coordonnées d'une $(n, K-n-1)$−fonction de non−linéarité D. Dans le cas inverse, si $D > 0$ est la non−linéarité d'une (n,m)−fonction, alors le code linéaire, qui est la réunion des translatés $v \cdot F + RM(1,n)$, où $v \in \mathbb{F}_2^m$, est de paramètres $[2^n,\ n+m+1,\ D]$.

4.2.2 Les codes de Kerdock et les fonctions Booléennes courbes

Les codes de *Kerdock*, noté K_n, sont des codes de longueur 2^n définis comme des sous−codes des codes de Reed−Muller. C'est−à−dire que le code de Kerdock de longueur 2^n est une réunion de translatés de $RM(1,n)$. Dans ce qui suit, nous donnons le lien entre les codes de Kerdock et les fonctions Booléennes.

Il existe plusieurs définitions du code de Kerdock [99], la plus simple suppose une

identification entre $\mathbb{F}_2^n$ et $\mathbb{F}_{2^{n-1}} \times \mathbb{F}_2$ et définit les éléments du code en tant que des fonctions Booléennes $f(x, y)$, $x \in \mathbb{F}_{2^{n-1}}$, $y \in \mathbb{F}_2$.

Définition 98. *Soit $n \geq 4$ un entier pair, le code de Kerdock de longueur 2^n est l'ensemble des fonctions Booléennes de la forme $f_u + h$, $h \in RM(1, n)$, où :*

$$f_u(x + y) = tr(\sum_{i=1}^{t} (ux)^{2^i+1}) + y\, tr(ux),$$

où $u \in \mathbb{F}_{2^{n-1}}$ et tr est la fonction trace sur $\mathbb{F}_{2^{n-1}}$, définie par : $x + x^{2^2} + \ldots + x^{2^{n-2}}$.

Le théorème suivant montre que le code de Kerdock est un sous−code du code $RM(2, n)$.

Théorème 99. *[99] Pour tout $r < n$, $RM(r, n)$ est l'ensemble des fonctions de la forme :*

$$f(x) = \sum_{i \in I} tr(a_i x^i),$$

où pour tout $i \in I$, $a_i \in \mathbb{F}_{2^n}$, $0 \leq i \leq 2^n - 1$ et $wt(i) \leq r$.

On montre que le code de Kerdock est non−linéaire, car si u et v sont linéairement indépendants, la fonction $f_u + f_v$ n'appartient pas à K_n.
En général, la distance minimale d'un code non−linéaire peut être différente de son poids, qui n'est pas le cas du code K_n, car ce code est distance−invariante [154].

Proposition 100. *Soit $\mathcal{C}$ un code de longueur 2^n qui est une réunion d'au moins 2 translatés du code $RM(1, n)$, alors la distance minimale de $\mathcal{C}$ est au plus $2^{n-1} - 2^{\frac{n}{2}-1}$.*

Preuve. La distance minimale entre deux translatés $\{f_1+h,\ h \in RM(1, n)\}$ et $\{f_2+h,\ h \in RM(1, n)\}$ est égale à la distance entre $f_1 + f_2$ au code $RM(1, n)$. Soit $f = f_1 + f_2$, alors pour toute fonction linéaire $l_a(x) = \sum_{i=1}^{n} a_i x_i$, on a $\min(d(f, l_a),\ d(f, l_a + 1)) = 2^{n-1} - \frac{1}{2}|\sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+l_a(x)}|$. On déduit $d(f,\ RM(1, n)) = 2^{n-1} - \frac{1}{2}\max_{a \in \mathbb{F}_2^n} |\sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+l_a(x)}|$. Puisque $(\sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+l_a(x)})^2 = 2^{2n}$, alors on déduit que la moyenne arithmétique de $(\sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+l_a(x)})^2$ est égale à 2^n et donc $\max_{a \in \mathbb{F}_2^n} |\sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+l_a(x)}| \geq 2^{\frac{n}{2}}$. Cela implique que $d(f,\ RM(1, n)) \leq 2^{n-1} - 2^{\frac{n}{2}-1}$. ■

D'après la proposition (100), la distance de la fonction Booléenne f à $RM(1, n)$ est au plus $2^{n-1} - 2^{\frac{n}{2}-1}$. Le code K_n est de distance minimale $2^{n-1} - 2^{\frac{n}{2}-1}$, alors distance de $f_1 + f_2$ à $RM(1, n)$ doit être égale à $2^{n-1} - 2^{\frac{n}{2}-1}$. Si la fonction $f_1 + f_2$ n'est pas affine alors elle est courbe.

4.2.3 Autres codes liés aux fonctions Booléennes et vectorielles

Il existe deux principales constructions des codes linéaires binaires à partir des fonctions Booléennes et vectorielles [66, 67] :

Codes à partir des fonctions Booléennes

— soit f une fonction Booléenne à n variables de support $supp(f)$ (définition (15)), alors en choisissant un ordre sur ce dernier et en assumant que cet ordre est de rang n, on définit le code linéaire $\mathcal{C}_{supp(f)}$. Les mots de code du code $\mathcal{C}_{supp(f)}$ sont la liste des valeurs de la restriction sur $supp(f)$ de la fonction linéaire $v \cdot x$, où $v \in \mathbb{F}_2^n$ et $\cdot$ est le produit scalaire dans $\mathbb{F}_2^n$. Autrement dit, le support de f est de rang n, et le code $\mathcal{C}_{supp(f)}$ est un code de paramètres $[wt(f), n, d]$, et est le code de toutes les fonctions ponctuées en toutes les positions qui ne sont pas dans $supp(f)$. Quand f est une fonction courbe de $n \geq 4$ variables (n pair), le code est alors un code de deux poids et d est leur minimum [170, 69]. Plus généralement, on peut considérer le code obtenu à partir de n'importe quel code Reed−Muller en le ponctuant à toutes les positions en dehors du $supp(f)$.

De nombreux liens existent entre les propriétés cryptographiques des fonctions Booléennes et la théorie des codes correcteurs d'erreurs. Par exemple :

— Dans [41], Carlet et al. ont introduit certains codes optimaux ou sous−optimaux, construits à partir de fonctions hautement non−linéaires.
— Les codes cycliques sont liés à l'immunité algébrique des fonctions Booléennes. Récemment, il a été démontré par Helleseth et Rønjom [95] l'existence d'un lien très étroit entre les annulateurs d'une fonction Booléenne et les codes cycliques 2^n−aires. Le lecteur intéressé pourra consulter par exemple [31, 37].

Codes à partir des fonctions vectorielles

Les fonctions vectorielles APN, AB et les codes correcteurs : Dans [38, 40, 165], les auteurs ont observé une relation entre les propriétés d'une (n, n)−fonction APN ou AB d'un côté, et les propriétés des codes associés de l'autre côté.

Proposition 101. *[38] Soit F une (n, n)−fonction tel que $F(0) = 0$. Soit H la matrice* $\begin{bmatrix} 1 & \alpha & \alpha^2 & \dots \alpha^{2^n-2} \\ F(1) & F(\alpha) & F(\alpha^2) & \dots & F(\alpha^{2^n-2}) \end{bmatrix}$*, où α est un élément primitif du corps fini $\mathbb{F}_{2^n}$, et chaque symbole représente la colonne de ces coordonnés en respectant une base de*

l'espace vectoriel $\mathbb{F}_2^n$.

Soit $\mathcal{C}_F$ *le code linéaire dont* H *est la matrice de contrôle, alors* F *est APN si et seulement si* $\mathcal{C}_F$ *est de distance minimale* 5*, et* F *est AB si et seulement si* $\mathcal{C}_F^{\perp}$ *est de poids* $0,\ 2^{n-1}-2^{\frac{n-1}{2}},\ 2^{n-1}$ *et* $2^{n-1}+2^{\frac{n-1}{2}}$.

Dans [23], pour exprimer le fait qu'une $(n,n)-$fonction est APN, une $(2n+1)\times(2^n-1)$ matrice est considérée : $H=\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 0 & 1 & \alpha & \alpha^2 & \dots \alpha^{2^n-2} & \\ F(0) & F(1) & F(\alpha) & F(\alpha^2) & \dots & F(\alpha^{2^n-2}) \end{bmatrix}$.
Alors F est APN si et seulement si le code $\hat{\mathcal{C}}_F$ qui a H comme matrice de contrôle, est de paramètres $[2^n, 2^n-1, 2n-6]$.

Autres codes a partir des fonctions vectorielles :

- Soit $\cdot$ le produit scalaire dans $\mathbb{F}_2^n$ et $\mathbb{F}_2^m$, les mots de code des sous$-$codes $\mathcal{C}_F'$ et $\mathcal{C}_F''$ de $RM(r,n)$, où $r\geq 2$ est le degré algébrique de F, sont les fonctions Booléennes $v\cdot F(x)+u\cdot x$ et $v\cdot F(x)+u\cdot x+\epsilon$ respectivement, où $u\in\mathbb{F}_2^n$, $v\in\mathbb{F}_2^n$ et $\epsilon\in\mathbb{F}_2$ peut être associé a une fonction vectorielle $F:\mathbb{F}_2^n\to\mathbb{F}_2^m$ qui n'a pas de fonctions composantes affines. Plus précisément, le code $\mathcal{C}_F''$ est l'union des $v\cdot F+RM(1,n)$, où $v\in\mathbb{F}_2^m$. les mots de code sont la liste des valeurs de ces fonctions. Le code $\mathcal{C}_F''$ est de paramètre $[2^n, n+m+1, d]$, où d est la non$-$linéarité de F. le lecteur peut consulter [12, 38, 41, 165, 174].
- Des codes (de poids constants) sont déduits dans [128] à partir des o$-$polynômes et en relation avec les fonctions vectorielles courbes (définition (49)).
- Dans [68, 71, 113, 160], nous trouvons des codes construits à partir des fonctions vectorielles.

Une construction hybride est proposée dans [160], en plus, d'autres constructions des codes cycliques à partir des fonctions vectorielles sont données dans [68, 71].

Chapitre 5

Codes linéaires à partir des fonctions vectorielles dans le contexte des attaques algébriques

Ce chapitre a fait l'objet d'une publication dans le journal international "Discrete Mathematics, Algorithms and Applications" [21].

Dans ce chapitre, nous présentons un nouveau lien entre les fonctions vectorielles et les codes cycliques LCD dans le contexte des attaques algébriques.

D'abord, nous donnons un lien direct entre les annulateurs des fonctions vectorielles sous forme polynomiale univariée, et certains 2^n-aires codes cycliques. Par conséquent, une borne inférieure sur l'immunité algébrique des fonctions vectorielles en terme de la distance minimale des codes LCD associés. Ensuite, nous montrons que ces codes sont des codes LCD et nous présentons quelques propriétés ainsi que les énumérateurs de poids de ces codes. Ce qui nous a mené à donner un lien entre l'immunité algébrique et l'immunité spectrale des fonctions vectorielles, ainsi qu'un lien entre ces deux critères et le poids de ces codes.
Enfin, nous proposons une généralisation de la notion du complément algébrique des fonctions Booléennes aux fonctions vectorielles. Cette généralisation nous a mené à donner une borne sur le degré algébrique le plus bas des annulateurs des fonctions vectorielles.

5.1 Codes linéaires à partir des fonctions vectorielles

Dans [95], Rønjom et Helleseth ont établi un lien entre les fonctions Booléennes et certains 2^n-aire codes cycliques, ensuite, ils ont donné une relation entre le problème de l'estimation de l'immunité algébrique des fonctions Booléennes et le haut$-$poids, noté wh, des mots de code des codes cycliques associer. Nous rappelons que la haut$-$poids d'un polynôme g est défini comme étant

$$wh(g) = \max\{wt(i);\, c_i \neq 0\},$$

où $wt(i)$ est le poids de Hamming du vecteur i, défini comme le nombre de ses coordonnées non nulles (voir [95]).

Le résultat suivant montre que tout annulateur d'une fonction Booléenne f appartient à certain code cyclique 2^n-aire de longueur 2^n-1, et donne une estimation de $AI(f)$.

Théorème 102. *[95] Soit f une fonction Booléenne sous forme polynomiale univariée. Alors tout annulateur de f, appartient a un code cyclique 2^n-aire généré par le polynôme G_f défini comme*

$$G_f = gcd(f(x)+1, x^{2^n-1}+1).$$

L'immunité algébrique de f est égale au haut$-$poids minimal des mots de code $g = \sum_{i=0}^{2^n-2} c_i x^i$ des deux codes cycliques 2^n-aires générés par les polynômes G_f et G_{1+f}.

Dans la suite, nous généralisons le théorème (102) aux $(n,m)-$fonctions, ensuite, nous donnons un lien direct entre les $(n,m)-$fonctions et les codes cycliques 2^n-aires.

Théorème 103. *Soit F une $(n,m)-$fonction sous forme polynomiale univariée. Tout annulateur de F, appartient a un 2^n-aire code cyclique généré par le polynôme g_F donné par :*

$$g_F = \begin{cases} x \prod\limits_{a\in\mathbb{F}_{2^n}^*} gcd(F(x)-a, x^{2^n-1}+1) \; si\, F(0) \neq 0. \\ \prod\limits_{a\in\mathbb{F}_{2^n}^*} gcd(F(x)-a, x^{2^n-1}+1) \; si\, F(0) = 0. \end{cases} \tag{5.1}$$

Preuve. Soit $G : \mathbb{F}_{2^n} \to \mathbb{F}_{2^m}$ un annulateur de F tel que $G(x)F(x) = 0$ pour tout $x \in \mathbb{F}_{2^n}$, ce qui est vrai pour tout $x \in \mathbb{F}_{2^n}^*$, alors on obtient :

$$G(x)F(x) \equiv 0 \, mod\ (x^{2^n}+x),$$

ce qui implique,

$$G(x) \equiv 0 \, mod\ (\frac{x^{2^n}+x}{gcd(F(x), x^{2^n}+x)}),$$

où

$$x^{2^n-1}+1=\prod_{a\in\mathbb{F}_{2^n}} gcd(F(x)-a,x^{2^n-1}+1).$$

Alors,

$$gcd(F(x),x^{2^n}+x)=\begin{cases} gcd(F(x),x^{2^n-1}+1)\, si\, F(0)\neq 0. \\ x\, gcd(F(x),x^{2^n-1}+1)\, si\, F(0)=0. \end{cases}$$

c'est−à−dire,

$$g_F(x)=\frac{x^{2^n}+x}{gcd(F(x),x^{2^n}+x)}=\begin{cases} x\prod\limits_{a\in\mathbb{F}_{2^n}^*} gcd(F(x)-a,x^{2^n-1}+1)\, si\, F(0)\neq 0. \\ \prod\limits_{a\in\mathbb{F}_{2^n}^*} gcd(F(x)-a,x^{2^n-1}+1)\, si\, F(0)=0. \end{cases}$$

Donc $G(x)$ est un mot de code du 2^n-aire code cyclique de polynôme générateur g_F.

■

Remarque 104. *Dans [95], les auteurs ont omis de séparer les cas où $F(0)=0$ et $F(0)\neq 0$, et donc les résultats de [95] sont valides seulement pour $F(0)=0$.*

On note par $\mathcal{C}(F^{-1}(z))$, où $z\in\mathbb{F}_{2^m}$, le code cyclique généré par g_F, et défini par le théorème (103). Le lemma suivant décrit l'ensemble des zéros du code cyclique $\mathcal{C}(F^{-1}(z))$.

Lemme 105. *Soit F une (n,m)-fonction sous forme polynomiale univariée. Alors,*

(1) *Soit $z=0$, alors*

(i) Si $F(0)=0$, alors les éléments de $F^{-1}(0)\cup\{0\}$ sont les zéros de $\mathcal{C}(F^{-1}(0))$.

(ii) Si $F(0)\neq 0$, alors les éléments de $F^{-1}(0)\cap\mathbb{F}_{2^n}^*$ sont les zéros de $\mathcal{C}(F^{-1}(0))$.

(2) *Soit $z\neq 0$, alors*

(i) Si $F(0)=0$, alors les éléments de $F^{-1}(z)\cap\mathbb{F}_{2^n}^*$ sont les zéros de $\mathcal{C}(F^{-1}(z))$.

(ii) Si $F(0)\neq 0$, alors les éléments de $F^{-1}(z)\cup\{0\}$ sont les zéros de $\mathcal{C}(F^{-1}(z))$.

Preuve. Supposons que $G(x):\mathbb{F}_{2^n}\to\mathbb{F}_{2^m}$ est un annulateur de F, associé à un mot de code du code généré par g_F (équation (5.1)), alors par la [proposition 4.13, [3]], $G(x)=0$ pour tout $x\in F^{-1}(z)$.

Soit $x\neq 0$, alors $\sum\limits_{i=0}^{2^n-1} a_i x^i=0$ veut dire que $a_0=0$. Puisque $\mathcal{C}(F^{-1}(z))$ est cyclique, alors $G(x)=x\sum\limits_{i=1}^{2^n-1} a_i x^{i-1}=\sum\limits_{j=0}^{2^n-2} a_{j+1}x^j=0$ pour tout $0\neq x\in F^{-1}(z)$, $z\in\mathbb{F}_{2^m}$.

Soit $x=0$, alors

(1) Soit $z=0$, alors

(i) Si $F(0) = 0$, alors $0 \in F^{-1}(0)$. Alors les zéros de $\mathcal{C}(F^{-1}(0))$ sont les éléments de $F^{-1}(0) \cup \{0\}$.
(ii) Si $F(0) \neq 0$, donc $0 \notin F^{-1}(0)$. Alors les zéros de $\mathcal{C}(F^{-1}(0))$ sont les éléments de $F^{-1}(0) \cap \mathbb{F}_{2^n}^*$.

(2) Soit $z \neq 0$, alors
(i) Si $F(0) = 0$, alors $0 \notin F^{-1}(z)$. Donc, les zéros de $\mathcal{C}(F^{-1}(z))$ sont les éléments de $F^{-1}(z) \cap \mathbb{F}_{2^n}^*$.
(ii) Si $F(0) \neq 0$, alors 0 peut être dans $F^{-1}(z)$. Donc les zéros de $\mathcal{C}(F^{-1}(z))$ sont les éléments de $F^{-1}(z) \cup \{0\}$. ■

L'ensemble de définition du code cyclique $\mathcal{C}(F^{-1}(z))$ est donné comme suit :

Lemme 106. *Soit $\mathcal{C}(F^{-1}(z))$ le code cyclique défini dans le théorème (103). L'ensemble de définition du code $\mathcal{C}(F^{-1}(z))$ est l'ensemble $D = \{i \in \mathbb{Z},\ F(\alpha^i) = z\}$.*

Preuve. Par définition, l'ensemble de définition du code $\mathcal{C}(F^{-1}(z))$ est l'ensemble $D = \{i \in \mathbb{Z},\ g_F(\alpha^i) = 0\}$, par le lemma (105), $D = \{i \in \mathbb{Z},\ F(\alpha^i) = z\}$. ■

Un résultat immédiat est le suivant :

Corollaire 107. *L'immunité algébrique d'une (n, m)-fonction F est égale au haut−poids minimal d'un mot de code du code $\mathcal{C}(F^{-1}(z))$, où $z \in \mathbb{F}_{2^m}$.*

Preuve. Supposons que G est un annulateur de F de degré algébrique égal à l'immunité algébrique de F. Puisque G est associé au mot de code $(a_1, \cdots, a_{2^n-1})$ du code $\mathcal{C}(F^{-1}(z))$ et le degré algébrique de G est égale à $\max\limits_{j=0,\cdots,2^n-1;\, a_j \neq 0} wt_2(j)$, alors pour tout $a_j \neq 0$, où $1 \leq j \leq 2^n - 1$, $wt_2(j)$ est égale à $wh(G)$. ■

5.2 Les propriétés du code cyclique $\mathcal{C}(F^{-1}(z))$ associé a la fonction vectorielle F

Dans cette section, nous prouvons tout d'abord que le code cyclique 2^n-aire $\mathcal{C}(F^{-1}(z))$, où $z \in \mathbb{F}_{2^m}$, est un code linéaire LCD.

Soit $\mathcal{C}^{\perp}(F^{-1}(z))$ le code dual du code $\mathcal{C}(F^{-1}(z))$, alors d'après le théorème (84) et le théorème (103), le polynôme générateur du code dual est donné par le théorème suivant :

Théorème 108. *Le polynôme générateur du code dual* $\mathcal{C}^{\perp}(F^{-1}(z))$ *du code* $\mathcal{C}(F^{-1}(z))$ *est donné par :*
(i) Si $F(0) \neq 0$*, alors*

$$g_F^{\perp} = x^{\deg(p)+1} p(x^{-1}) p_0^{-1}.$$

(ii) Si $F(0) = 0$*, alors*

$$g_F^{\perp} = x^{\deg(p)} p(x^{-1}) p_0^{-1}.$$

Où $p(x) = \gcd(F(x),\, x^{2^{2^n}-1} + 1)$.

Dans ce qui suit, nous présentons les paramètres du code LCD $\mathcal{C}(F^{-1}(z))$.

Théorème 109. *Soit* $\mathcal{C}(F^{-1}(z))$ *le code cyclique défini dans le théorème (103), alors*

(1) *Pour tout* $z \in \mathbb{F}_{2^m}^*$ *on a :*
(i) Si $F(0) = 0$*, alors* $\mathcal{C}(F^{-1}(z))$ *est un code LCD de paramètres* $[2^n - 1, \frac{1}{2}(2^n - 1 - (-1)^{\phi_z(0)} + W_{\phi_z}(0)), d]$.
(ii) Si $F(0) \neq 0$*, alors* $\mathcal{C}(F^{-1}(z))$ *est un code LCD de paramètres* $[2^n - 1, 2^{n-1} - 1 + \frac{1}{2}W_{\phi_z}(0), d]$.

(2) *Pour* $z = 0$*, alors* $\mathcal{C}(F^{-1}(0))$ *est un code LCD de paramètres* $[2^n - 1, 2^{n-1} - 1 + \frac{1}{2}W_{\phi_0}(0), d]$.

Preuve. Pour $F(0) \neq 0$, soit $q(x) = x\, gcd(F(x), x^{2^n-1}+1)$, c'est que $g_F^{\perp} = x^{deg(q)} q(x^{-1}) q_0^{-1}$. Soit β_i (β_i un élément primitif de $\mathbb{F}_{2^n}$) un zéro de $q(x)$, alors β_i^{-1} est un zéro de $g_F^{\perp}(x)$. Ce qui implique que $q(\beta_i) = g_F^{\perp}(\beta_i^{-1}) = 0$.
Par définition, les zéros de $\mathcal{C}^{\perp}$ sont les inverses des non zéros de $\mathcal{C}(F^{-1}(z))$, alors par le lemma (105), ils sont l'inverse des zéros de F. Ainsi $q(x) = g_F^{\perp}(x)$ pour tout $x \in \{x \in \mathbb{F}_{2^n},\, F(x) = 0\}$, cela implique que $\{\beta_1, \ldots, \beta_r\} = \{\beta_1^{-1}, \ldots, \beta_r^{-1}\}$. Alors par la proposition (88), $\mathcal{C}^{\perp}$ est un code LCD. Ainsi $\mathcal{C}(F^{-1}(z))$ est un code LCD.

Par définition, le degré du polynôme générateur de $\mathcal{C}(F^{-1}(z))$ est égal à $|D|$, où D est l'ensemble donné par le lemma (106).
Soit $z \in \mathbb{F}_{2^m}$ et $z \neq 0$. Alors,

(i) Soit $F(0) = 0$, par [*(i)*, (2), lemma (105)], les éléments de $F^{-1}(z) \cap \mathbb{F}_{2^n}^*$ sont les zéros de $\mathcal{C}(F^{-1}(z))$.

Soit ϕ_z la fonction indicatrice de $F^{-1}(z)$ définie sur $\mathbb{F}_{2^n}$ par $\phi_z(x) = 1$ si $F(x) = z$ et $\phi_z(x) = 0$ sinon. Alors $|D| = |F^{-1}(z)| = |\{x \in \mathbb{F}_{2^n},\, \phi_z(x) = 1\}|$. D'où,

$$\begin{aligned}
|D| &= |\{x \in \mathbb{F}_{2^n}^*,\, \phi_z(x) = 1\}|. \\
&= 2^n - 1 - |\{x \in \mathbb{F}_{2^n}^*,\, \phi_z(x) = 0\}|. \\
&= 2^n - 1 - \frac{1}{2}\sum_{y \in \mathbb{F}_2} \sum_{x \in \mathbb{F}_{2^n}^*} (-1)^{y\phi_z(x)}. \\
&= 2^n - 1 - \frac{1}{2}\sum_{x \in \mathbb{F}_{2^n}^*} (1 + (-1)^{\phi_z(x)}). \\
&= \frac{1}{2}(2^n - 1 + (-1)^{\phi_z(0)} - W_{\phi_z}(0)).
\end{aligned}$$

Par définition, la dimension du code $\mathcal{C}(F^{-1}(z))$ est égale à $2^n - 1 - |D|$. Donc, $\mathcal{C}(F^{-1}(z))$ est un code LCD de paramètres $[2^n - 1,\, \frac{1}{2}(2^n - 1 - (-1)^{\phi_z(0)} + W_{\phi_z}(0)),\, d]$.

Soit $z = 0$, alors $\mathcal{C}(F^{-1}(0))$ est un code LCD de paramètres $[2^n - 1,\, 2^{n-1} - 1 + \frac{1}{2}W_{\phi_0}(0),\, d]$.

(ii) Soit $F(0) \neq 0$, par $[(ii),\, (2),$ lemma (105), on a,

$$\begin{aligned}
|D| &= |\{x \in \mathbb{F}_{2^n},\, \phi_z(x) = 1\}|. \\
&= 2^n - |\{x \in \mathbb{F}_{2^n},\, \phi_z(x) = 0\}|. \\
&= 2^n - \frac{1}{2}\sum_{y \in \mathbb{F}_2} \sum_{x \in \mathbb{F}_{2^n}} (-1)^{y\phi_z(x)}. \\
&= 2^n - \frac{1}{2}\sum_{x \in \mathbb{F}_{2^n}} (1 + (-1)^{\phi_z(x)}). \\
&= 2^{n-1} - \frac{1}{2}W_{\phi_z}(0).
\end{aligned}$$

Donc, la dimension de $\mathcal{C}(F^{-1}(z))$ est égale à $2^{n-1} - 1 + \frac{1}{2}W_{\phi_z}(0)$.

Soit $z = 0$ et $F(0) = 0$, alors $(-1)^{\phi_0(0)} = 1$, ainsi $\mathcal{C}(F^{-1}(0))$ est un code LCD de paramètres $[2^n - 1,\, 2^{n-1} - 1 + \frac{1}{2}W_{\phi_0}(0),\, d]$. ■

5.2.1 Le poids des mots de code du code $\mathcal{C}(F^{-1}(z))$

Dans [5], les auteurs ont donné une borne supérieure sur l'immunité algébrique des $(n, m)-$fonctions.

Lemme 110. *Soit F une (n,m)-fonction telle que $AI(F) \leq d$, où d est le plus petit entier tel que $\sum_{i=0}^{d} \binom{n}{i} > 2^{n-m}$.*

En considérant le cas où $n = m$, la proposition suivante est valable.

Proposition 111. *Soit F une (n,m)-fonction, alors $AI(F) \leq 1$.*

Preuve. à partir du lemma (110), on a $AI(F) \leq d$, où d est le plus petit entier tel que $\sum_{i=0}^{d} \binom{n}{i} > 2^{n-m}$. Soit $n = m$, alors d est le plus petit entier tel que $\sum_{i=0}^{d} \binom{n}{i} > 2^{n-n} = 1$, c'est−à−dire $d = 1$ et $AI(F) \leq 1$. ■

Le théorème suivant donne une caractérisation des poids non nuls du code LCD $\mathcal{C}(F^{-1}(z))$.

Théorème 112. *Soit F une (n,m)-fonction, et $\mathcal{C}(F^{-1}(z))$ le code de distance minimale δ. Soit e le plus petit entier tel que $\sum_{i=0}^{e} \binom{n}{i} \geq \delta$, alors*

$$e \leq AI(F) \leq 1,$$

Nous avons,
(i) Si $e = 1$, alors $\mathcal{C}(F^{-1}(z))$ contient des mots de code non nuls de poids $\geq 1 + n$.
(ii) Si $e = 0$, alors $\mathcal{C}(F^{-1}(z))$ est le code LCD trivial.

Preuve. Supposons que G est un annulateur de F de degré algébrique $LDA(F)$, G est associé à un mot de code c du code $\mathcal{C}(F^{-1}(z))$, alors $wt(c)$ est égal à $\sum_{i=0}^{LDA(F)} \binom{n}{i}$.
à partir du théorème (115), $LDA(F) \geq e$, où e est le plus petit entier tel que $\sum_{i=0}^{e} \binom{n}{i} \geq \delta$.
Cependant, par la proposition (111), $e \leq AI(F) \leq 1$. Donc $e \in \{0, 1\}$.
Si $e = 1$, alors $wt(c) \geq \sum_{i=0}^{1} \binom{n}{i} = 1 + n$.
Si $e = 0$, le résultat est évident. ■

On note par $Deg(F)$ le plus haut degré algébrique d'un annulateur non nul de F. Soit $m < n$, la proposition suivante tient.

Proposition 113. *Soit F une $(n,m)-$fonction et $\mathcal{C}(F^{-1}(z))$ le code LCD défini par le théorème (103), alors*

(i) Si $AI(F) < Deg(F)$*, alors les poids des mots de code du code* $\mathcal{C}(F^{-1}(z))$ *sont tels que*

$$\sum_{i=0}^{AI(F)} \binom{n}{i} \leq wt(c) \leq \sum_{i=0}^{Deg(F)} \binom{n}{i}.$$

(ii) Si $AI(F) = Deg(F)$*, alors* $\mathcal{C}(F^{-1}(z))$ *est un* $1-$*poids code et* $wt(c) = \sum_{i=0}^{AI(F)} \binom{n}{i}$.

Preuve. Soient G et G' deux annulateurs de la fonction vectorielle F, associés à deux mots de code du code $\mathcal{C}(F^{-1}(z))$. Supposons que G et G' sont de degrés algébriques $AI(F)$ et $Deg(F)$ respectivement.
Il est évident que $AI(F) \leq Deg(F)$, alors le poids de tout mot de code c du code $\mathcal{C}(F^{-1}(z))$ est tel que

$$\sum_{i=0}^{AI(F)} \binom{n}{i} \leq wt(c) \leq \sum_{i=0}^{Deg(F)} \binom{n}{i}.$$

Si $AI(F) = Deg(F)$, alors $wt(c) = \sum_{i=0}^{AI(F)} \binom{n}{i}$. ■

5.2.2 Borne sur l'immunité algébrique des fonctions vectorielles

Notant par $LDA(f)$ le degré le plus bas des annulateurs d'une fonction Booléenne f, dans [130], les auteurs ont donné une borne inférieure sur $LDA(f)$ en étudiant les propriétés des codes cycliques 2^n-aires $\mathcal{C}(supp(f))$. Plus précisément, les auteurs ont prouvé le résultat suivant :

Théorème 114. *([130]) Soit* $f : \mathbb{F}_{2^n} \to \mathbb{F}_2$*, et* δ *la distance minimale de* $\mathcal{C}(supp(f))$*. Alors si* e *est le plus petit entier positif tel que* $\sum_{i=0}^{e} \binom{n}{i} \geq \delta$*, donc* $LDA(f) \geq e$*.*

Le résultat suivant est une extension du théorème (114) fournissant une borne inférieure sur l'immunité algébrique des (n, m)-fonctions.

Théorème 115. *Soit* F *une* (n, m)*-fonction sous forme polynomiale univariée,* $b \in \mathbb{F}_{2^m}$ *et* δ *la distance minimale de* $\mathcal{C}(F^{-1}(z))$*. Alors,*
(i) Si e *est le plus petit entier positif qui vérifie* $\sum_{i=0}^{e} \binom{n}{i} < \delta$*, alors* F *n'admet pas d'annulateurs non nuls de degré algébrique inférieur ou égal à* e*.*
(ii) Si e *est le plus petit entier positif qui vérifie* $\sum_{i=1}^{e} \binom{n}{i} \geq \delta$*, alors* $LDA(F) \geq e$*.*

Preuve. Soit $G(x) = \sum_{i=0}^{2^n-1} a_i x^i$ un annulateur de F de degré algébrique au plus e, c'est−à−dire, $a_i = 0$ pour tout i de $wt_2(i) > e$. Soit c un mot de code du code $\mathcal{C}(F^{-1}(z))$. Ainsi, le nombre de composantes non nulles de c est au plus $\sum_{i=0}^{e} \binom{n}{i}$, c'est−à−dire le poids de c est inférieur à δ. D'où c est nul, ce qui prouve que tout annulateur de F est de degré algébrique au moins $e+1$. Pour la partie (ii), supposons que e est le plus petit entier tel que $\sum_{i=0}^{e} \binom{n}{i} \geq \delta$. Alors $\sum_{i=0}^{e-1} \binom{n}{i} < \delta$ et alors $LDA(F) \geq e$. ■

5.2.3 Immunité spectrale des fonctions vectorielles

Dans ce qui suit, nous donnons une relation entre l'immunité spectrale d'une $(n,m)-$fonction F, son immunité algébrique et le poids de Hamming du code $\mathcal{C}(F^{-1}(z))$.
Soit $x \in \mathbb{F}_{2^n}$ l'état initial d'un générateur de filtrage, alors ce dernier produit la clef z_t, qui est une séquence sur $\mathbb{F}_2^n$, telle que $z_t = F(x\alpha^t)$ pour $t = 0, 1, \cdots$.

Soit G une $(n,m)-$fonction sous forme polynomiale univariée, telle que $G(x\alpha^t) = u_t$ est un annulateur de z_t. Selon la définition (7), l'immunité spectrale de z_t est la complexité linéaire la plus faible de u_t qui est le nombre de coefficients non nuls de

$$G(x) = \sum_{i=0}^{2^n-2} \delta_i x^i \in \mathbb{F}_{2^n}[x].$$

Théorème 116. *Soit F une $(n,m)-$fonction sous forme polynomiale univariée. Alors, sous l'hypothèse du théorème (103), l'hypothèse suivante se vérifie*
(i) L'immunité spectrale de F est égale au poids minimal de $\mathcal{C}(F^{-1}(z))$, où $z \in \mathbb{F}_{2^m}$.
(ii) La relation entre l'immunité spectrale et l'immunité algébrique de F est donnée par l'inégalité suivante :

$$SI(F) \leq \sum_{i=0}^{AI(F)} \binom{n}{i}.$$

Preuve. Soit G un annulateur de F de degré algébrique égal au moins à $AI(F)$, associé à un mot de code $c = (a_1, \cdots, a_{2^n-1})$ du code $\mathcal{C}(F^{-1}(z))$ tel que $a_i = 0$ pour tout i avec $wt_2(i) \leq AI(F)$. C'est pourquoi, G a au moins $\sum_{i=0}^{AI(F)} \binom{n}{i}$ coefficients non nuls dans sa représentation univariée, alors $SI(F) \leq \sum_{i=0}^{AI(F)} \binom{n}{i}$. ■

5.3 Le complément algébrique des fonctions vectorielles

Dans cette section, nous présentons une généralisation de la notion de complément algébrique des fonctions Booléennes aux $(n, m)-$fonctions. Ensuite, nous présentons une borne sur le degré algébrique le plus bas des annulateurs de F^c.

5.3.1 Le complément algébrique des fonctions Booléennes

Dans [172], les auteurs ont étendu le concept de l'immunité algébrique des fonctions Booléennes et alors ont introduit la notion de complément algébrique f^c comme indiqué dans la définition suivante :

Définition 117. *[172] Soit f une fonction Booléenne à n variables, le complément algébrique de f, désigné par f^c, est défini comme la fonction qui contient tous les monômes $x_1^{a_1}, \cdots, x_n^{a_n}$, où chaque $a_j \in \mathbb{F}_2$, qui ne sont pas dans l'ANF de la fonction Booléenne f, c'est−à−dire, les fonctions f et f^c n'ont pas de monômes en commun.*

Dans [166] et [172] les auteurs ont donné quelques propriétés du complément algébrique des fonctions Booléennes. Soit $\Delta(x) = \prod_{i=1}^{n}(1 + x_i)$, où $x = (x_1, \cdots, x_n) \in \mathbb{F}_2^n$. Alors, les auteurs ont montré que $f(x)\Delta(x) = 0$ pour $f(0) = 0$ et $f(x)\Delta(x) = \Delta(x)$ pour $f(0) = 1$. En outre, ils ont prouvé que la fonction f^c satisfait $f^c = f + \Delta$ (notons que $f^c(x) = f(x)$ pour tout élément non nul $x \in \mathbb{F}_2^n$).

La proposition suivante donne une relation directe entre l'ensemble des annulateurs de f et l'ensemble des annulateurs de f^c.

Proposition 118. *[166] Soit f une fonction Booléenne telle que $f(0) = 1$ alors $AN(f^c) = AN(f) \cup AN(f)^c$.*

5.3.2 Le complément algébrique des fonctions vectorielles

La notion du complément algébrique des fonctions Booléennes peut être étendue aux $(n, m)-$fonctions comme suit.

Définition 119. *Soit $F = (f_1, \cdots, f_m)$ une $(n, m)-$fonction. Le complément algébrique de F, désigné par F^c, est défini comme $F^c = (f_1^c, \cdots, f_m^c)$, où f_i^c est le complément algébrique de la fonction Booléenne f_i, pour $1 \leq i \leq m$.*

Dans ce qui suit, on donne quelques propriétés de la fonction vectorielle F^c, ainsi qu'une relation entre $AN(F)$ et $AN(F^c)$.

Lemme 120. *Soit F une (n,m)-fonction, alors*
1) $F^c(x) = F(x)$ pour tout élément non nul $x \in \mathbb{F}_2^n$.
2) $F^c(x) = F(x) + \Delta(x)(1, \cdots, 1)$ pour tout $x \in \mathbb{F}_2^n$.

Preuve. De [172, lemma 2], on a $f_i(x) = f_i^c(x)$ pour tout élément non nul $x \in \mathbb{F}_2^n$, $1 \leq i \leq m$, alors pour tout élément non nul $x \in \mathbb{F}_2^n$ on a,

$$F^c(x) = (f_1^c(x), \cdots, f_m^c(x)) = (f_1(x), \cdots, f_m(x)) = F(x).$$

De [172, Lemma 2], pour tout $x \in \mathbb{F}_2^n$, $f^c(x) = f(x) + \Delta(x)$, alors

$$\begin{aligned} F(x)^c &= (f_1^c(x), \cdots, f_m^c(x)) \\ &= (f_1(x) + \Delta(x), \cdots, f_m(x) + \Delta(x)) = F(x) + \Delta(x)(1, \cdots, 1). \end{aligned}$$

■

Nous rappelons que, selon la définition (64), une fonction Booléenne g est un annulateur de $F^{-1}(z)$ si $g(x) = 0$ pour tout $x \in F^{-1}(z) = \{a \in \mathbb{F}_2^n;\ f_i(a) = z_i,\ 1 \leq i \leq m\}$. Dans ce qui suit, nous donnons la relation entre $F^{-1}(z)$ et $(F^c)^{-1}(z)$, où $z \in \mathbb{F}_{2^m}$.

Lemme 121. *Soit F une (n,m)-fonction et F^c son complément algébrique, alors nous avons ce qui suit :*
(i) Si $0 \in F^{-1}(z)$, alors $(F^c)^{-1}(z) = F^{-1}(z) \cap \mathbb{F}_2^n \setminus \{0\}$.
(ii) Si $0 \notin F^{-1}(z)$, alors $(F^c)^{-1}(z) = F^{-1}(z) \cup \{0\}$.
(iii) Si $0 \notin F^{-1}(z)$ et $0 \notin (F^c)^{-1}(z)$, alors $F^{-1}(z) = (F^c)^{-1}(z)$.

Preuve. Soit $g \in AN(F^{-1}(z))$, $g(x) = 0$ pour tout $x \in F^{-1}(z)$. Par [172, Lemma 2, item 1], on a $f_i(x) = f_i^c(x)$ pour $x \in \mathbb{F}_2^n$, alors $(F^c)^{-1}(z) = F^{-1}(z)$ pour tout $z \in \mathbb{F}_2^n$.
Soit $0 \in F^{-1}(z)$, alors $0 \notin (F^c)^{-1}(z)$, donc $(F^c)^{-1}(z) = F^{-1}(z) \cap \mathbb{F}_2^n \setminus \{0\}$.
Soit $0 \notin F^{-1}(z)$, alors $0 \in (F^c)^{-1}(z)$, donc $(F^c)^{-1}(z) = F^{-1}(z) \cup \{0\}$. ■

Lemme 122. *Soit $g \in AN(F^{-1}(z))$, avec les notations précédentes on fixe $AN(F^{-1}(z))^c = \{g^c,\ g \in AN(F^{-1}(z))\}$, alors*
(i) $g \in AN((F^c)^{-1}(z))$ pour $F^{-1}(z) = (F^c)^{-1}(z)$.
(ii) $AN(F^{-1}(z)) \subset AN((F^c)^{-1}(z))$ avec $0 \in F^{-1}(z)$.
(iii) $AN(F^{-1}(z))^c \subset AN((F^c)^{-1}(z))$ avec $0 \notin F^{-1}(z)$.

Soit $g \in AN(F^c)$, alors
(iv) $AN((F^c)^{-1}(z)) \subset AN(F^{-1}(z))$ *pour* $0 \notin F^{-1}(z)$.
(v) $AN((F^c)^{-1}(z))^c \subset AN(F^{-1}(z))$ *pour* $0 \in F^{-1}(z)$.

Preuve. Soit g un annulateur de $F^{-1}(z)$, pour un élément $z \in \mathbb{F}_2^m$, $g(x) = 0$ pour tout $x \in F^{-1}(z)$.
Supposons d'abord que $0 \notin F^{-1}(z)$ et $0 \notin (F^c)^{-1}(z)$, alors par *(iii)* du Lemma (121), $F^{-1}(z) = (F^c)^{-1}(z)$, ce qui donne $g(x) = 0$ pour tout $x \in F^{-1}(z) = (F^c)^{-1}(z)$, alors g est un annulateur de $(F^c)^{-1}(z)$. Si $0 \in F^{-1}(z)$, alors $0 \notin (F^c)^{-1}(z)$ et par *(ii)* du Lemma (121), $F^{-1}(z) = (F^c)^{-1}(z) \cup \{0\}$. Alors $g(x) = 0$ pour tout $x \in (F^c)^{-1}(z) \cup \{0\}$, ainsi, $g(x) = 0$ pour tout $x \in (F^c)^{-1}(z)$. Donc $g \in AN((F^c)^{-1}(z))$ ce qui donne $AN(F^{-1}(z)) \subset AN((F^c)^{-1}(z))$.
Depuis que $0 \notin F^{-1}(z)$, alors par *(i)* du Lemma (121), $F^{-1}(z) = (F^c)^{-1}(z) \cap \mathbb{F}_2^n - \{0\}$, ce qui donne $g(x) = 0$ pour tout élément non nul $x \in (F^c)^{-1}(z) \cap \mathbb{F}_2^n$. Donc $g^c \in AN((F^c)^{-1}(z))$ puisque g^c prend la même valeur de g pour tout élément non nul $x \in \mathbb{F}_2^n$, et la valeur 0 pour $x = 0$. Donc $AN(F^{-1}(z))^c \subset AN((F^c)^{-1}(z))$. ■

La relation entre les deux ensembles $AN(F^{-1}(z))$ et $AN((F^c)^{-1}(z))$ est donnée par le théorème suivant :

Théorème 123. *Soit F une (n, m)-fonction et F^c son complément algébrique, on a ce qui suit :*
(i) $AN((F^c)^{-1}(z)) = AN(F^{-1}(z)) \cup AN(F^{-1}(z))^c$ *pour* $0 \in F^{-1}(z)$.
(ii) $AN(F^{-1}(z)) = AN((F^c)^{-1}(z)) \cup AN((F^c)^{-1}(z))^c$ *pour* $0 \notin F^{-1}(z)$.
(iii) $AN(F^{-1}(z)) = AN((F^c)^{-1}(z))$ *pour* $F^{-1}(z) = (F^c)^{-1}(z)$.

Preuve. Pour la partie (i), soit $A = \{g \in AN((F^c)^{-1}(z)), 0 \in (F^c)^{-1}(z)\} \subset AN((F^c)^{-1}(z))$.
Par (i) du lemma (122), $g \in A$ pour tout $g \in AN(F^{-1}(z))$ ce qui donne $A \subset AN(F^{-1}(z))$.
Par (iii) du lemma (122), $g \in AN(F^{-1}(z))$ pour tout $g \in A$, alors $AN(F^{-1}(z)) \subset A$, cela implique que $AN(F^{-1}(z)) = A$.
Soit $B = \{g \in AN((F^c)^{-1}(z)),\ 0 \notin (F^c)^{-1}(z)\}$, alors $|AN((F^c)^{-1}(z))| = |A| + |B|$. Par (ii) du lemma (122), $g \in AN(F^{-1}(z))$ et alors $g^c \in B$. Mais $AN(F^{-1}(z)) = A$, alors $g \in B$ pour tous $g \in A$. Ainsi $|A| = |B|$.
Soit φ une application de A dans B, tel que pour tout $g \in A$, $\varphi(g) = g^c \in B$. Pour prouver que φ est injective, on prend g^c, h^c tel que $g^c(x) = h^c(x)$, alors pour tout $x \in \mathbb{F}_2^n$, $g(x) + \Delta(x) = h(x) + \Delta(x)$, cela implique que $g(x) = h(x)$.
Puisque $|A| = |B|$, alors φ est surjective, c'est−à−dire que $AN((F^c)^{-1}(z)) = A \cup B = AN(F^{-1}(z)) \cup AN((F^c)^{-1}(z))$. Ce qui prouve le premier point du théorème. Quand $0 \in F^{-1}(z)$, $0 \notin (F^c)^{-1}(z)$. Et on prouve partie (ii) du théorème de la même manière que

la partie (i).
Pour la partie (iii), soit $F^{-1}(z) = (F^c)^{-1}(z)$. Par (v) du lemma (122), si $g \in AN(F^{-1}(z))$ alors $g \in AN((F^c)^{-1}(z))$, ainsi $AN((F^c)^{-1}(z)) \subset AN(F^{-1}(z))$, et si $g \in AN((F^c)^{-1}(z))$ alors $g \in AN(F^{-1}(z))$, ainsi $AN((F^c)^{-1}(z)) \subset AN((F^c)^{-1}(z))$. Donc $AN(F^{-1}(z)) = AN((F^c)^{-1}(z))$, ce qui complète la preuve. ■

Un lien entre l'immunité algébrique d'une $(n, m)-$fonction F et l'immunité algébrique de son complément algébrique est alors donnée par la proposition suivante :

Proposition 124. *Soit F une (n, m)-fonction et F^c son complément algébrique. Alors,*

$$AI(F) - 1 \leq AI(F^c) \leq AI(F) + 1.$$

Preuve. Soit g un annulateur de F de degré algébrique e. Puisque $F(x) = F^c(x)$ pour tout élément non nul $x \in \mathbb{F}_2^n$, alors $AI(F) = AI(F^c)$. Soit $0 \in F^{-1}(z)$ et soit g un annulateur de $F^{-1}(z)$, alors par (i) du théorème (123), g et g^c sont des annulateurs de $(F^c)^{-1}(z)$, c'est−à−dire que $|deg(g) - deg(g^c)| = 1$, alors $AI(F) - 1 \leq AI(F^c)$. Soit $0 \notin F^{-1}(z)$, alors par (ii) du théorème (123), si g est un annulateur de $(F^c)^{-1}(z)$ alors g et g^c sont des annulateurs de $F^{-1}(z)$, donc $AI(F^c) \leq AI(F)+1$. Ainsi $AI(F)-1 \leq AI(F^c) \leq AI(F)+1$.
■

5.3.3 Sur le plus bas degré algébrique d'annulateurs non nuls des fonctions Booléennes et vectorielles

Dans ce qui suit, nous nous intéressons tout d'abord au plus bas degré algébrique des annulateurs non nuls des fonctions Booléennes, ce qui est une réponse à un problème ouvert mentionner dans [127] par Mesnager et Cohen.

Soit α un élément primitif de $\mathbb{F}_{2^n}$, on définit l'ensemble des $t-$zéros consécutifs du code $\mathcal{C}(supp(f))$ par $V(\alpha, l, t) = \{\alpha^l, \alpha^{l+1}, \ldots, \alpha^{l+t-1}\}$.

Théorème 125. *[[117], page 206] Soit α un élément primitif de $\mathbb{F}_{2^n}$ et soit r, k et t des entiers non négatifs, m un entier positif relativement premier à n.*
Soit $\mathcal{C} \subset \mathbb{F}_{2^n}$ un code cyclique avec $\alpha^r, \alpha^{r+1}, \ldots, \alpha^{r+t-1}, \alpha^{r+m}, \alpha^{r+m+1}, \ldots, \alpha^{r+m+t-1}, \ldots, \alpha^{r+km}$ $\alpha^{r+km+1}, \ldots, \alpha^{r+km+t-1}$. Alors la distance minimale δ de $\mathcal{C}$ est supérieure à $t + k$.

Dans [127], le résultat suivant a été prouvé.

Théorème 126. *Soit $f : \mathbb{F}_{2^n} \to \mathbb{F}_2$ une fonction Booléenne, l, k et δ des entiers positifs, soit m un entier positif relativement premier à n. Supposons que S_f contient $V(\alpha, l, \delta -$*

$1) \cup V(\alpha, l+m, \delta-1) \cup \ldots \cup V(\alpha, l+km, \delta-1)$. *Alors* $LDA(f) \geq e$, *où* e *est le plus petit entier positif tel que* $\sum_{i=1}^{e} \binom{n}{i} \geq \delta + k$.

On donne ci-après une borne inférieure sur $LDA(1+f)$, où la fonction Booléenne f est définie par le théorème (126).

Théorème 127. *Soit* $f : \mathbb{F}_{2^n} \to \mathbb{F}_2$ *une fonction Booléenne et* l, k *et* δ *des entiers positifs, soit* m *un entier positif relativement premier à* n. *Alors*
(i) $Supp(1+f)$ *contient* $V(\alpha, l+\delta-1, 2^n-\delta) \cup V(\alpha, l+m+\delta-1, 2^n-\delta) \cup \ldots \cup V(\alpha, l+km+\delta-1, 2^n-\delta)$.
(ii) $LDA(1+f) \geq e-1$, *où* e *est le plus petit entier positif possible tel que* $\sum_{i=1}^{e} \binom{n}{i} \geq 2^n - \delta + k$.

Preuve. Soit f une fonction Booléenne et $\mathcal{C}(supp(f))$ le code LCD généré par le polynôme G_f (théorème (103)).
$supp(1+f) = zeros(f) \cap \mathbb{F}_{2^n}^*$, c'est−à−dire

$$\begin{aligned} S_{1+f} &= \{1, \alpha, \ldots, \alpha^{l-1}, \alpha^{l+\delta-1}, \ldots, \alpha^{2^n-2}\} \cup \{1, \alpha, \ldots, \alpha^{l+m-1}, \alpha^{l+m+\delta-1}, \ldots, \alpha^{2^n-2}\} \\ &\quad \cup \ldots \cup \{1, \alpha, \ldots, \alpha^{l+km-1}, \alpha^{l+km+\delta-1}, \ldots, \alpha^{2^n-2}\}. \\ &= \{\alpha^{l+\delta-1}, \ldots, \alpha^{l+\delta-1+(2^n-\delta)-1}\} \cup \{\alpha^{l+m+\delta-1}, \ldots, \alpha^{l+m+\delta-1+(2^n-\delta)-1}\} \cup \ldots \\ &\quad \cup \{\alpha^{l+km+\delta-1}, \ldots, \alpha^{l+km+\delta-1+(2^n-\delta)-1}\}. \\ &= V(\alpha, l+\delta-1, 2^n-\delta) \cup V(\alpha, l+m+\delta-1, 2^n-\delta) \cup \ldots \cup V(\alpha, l+km+ \\ &\quad \delta-1, 2^n-\delta). \end{aligned}$$

Due à [130, Corollary 02] et au théorème (125), $LDA(1+f) \geq e-1$, où e est le plus petit entier positif tel que $\sum_{i=1}^{e} \binom{n}{i} \geq 2^n - \delta + k$. ■

Par conséquent, on peut immédiatement déduire une borne inférieure sur l'immunité algébrique des fonctions Booléennes.

Corollaire 128. *Soit* f *une fonction Booléenne définie par le théorème (126). Alors* $AI(f) \geq e-1$, *où* e *est le plus petit entier tel que* $\sum_{i=1}^{e} \binom{n}{i} \geq min(\delta + k, 2^n - \delta + k)$.

Le lemma (105) nous donne une description de l'ensemble des zéros du code LCD $\mathcal{C}(F^{-1}(z))$. Dans la suite on donne une caractérisation de l'ensemble $F^{-1}(z)$, ainsi qu'une borne sur l'immunité algébrique de la (n,m)−fonction F.

Théorème 129. *Soit α un élément primitif de $\mathbb{F}_{2^n}$, alors si $F^{-1}(z) = \{1, \alpha, \ldots, \alpha^{2^{n-m}-2}\}$, donc*

$$AI(F) = \lceil \frac{n-m+1}{2} \rceil.$$

Preuve. Soit $\mathcal{C}(F^{-1}(z))$ le code défini par le théorème (103), et $\{1, \alpha, \ldots, \alpha^{2^{n-m}-2}\}$ son ensemble de zéros. Alors, à partir du théorème (115), $LDA(F) \geq e$, où e est le plus petit entier positif tel que $\sum_{i=0}^{e} \binom{n}{i} \geq 2^{n-m} - 2$. On a

$$\sum_{i=0}^{\lceil \frac{n-m+1}{2} \rceil} \binom{n}{i} < 2^{n-m} - 2 \leq \sum_{i=1}^{\lceil \frac{n-m+1}{2} \rceil} \binom{n}{i}.$$

Alors, due au théorème (125), $LDA(F) \geq \lceil \frac{n-m+1}{2} \rceil$.

Par définition, on a $AI(F) \leq d \leq \lceil \frac{n-m+1}{2} \rceil$, où d est le plus petit entier positif tel que $\sum_{i=0}^{d} \binom{n}{i} > 2^{n-m}$. Alors, $AI(F) = \lceil \frac{n-m+1}{2} \rceil$. ■

5.3.4 Lien entre $\mathcal{C}(F^{-1}(z))$ et $\mathcal{C}((F^c)^{-1}(b))$

Soit F une (n, m)-fonction sous forme polynomiale univariée et F^c son complément algébrique. Dans ce qui suit, on donne la relation entre $\mathcal{C}(F^{-1}(z))$ et $\mathcal{C}((F^c)^{-1}(b))$.

Lemme 130. *Soit $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^m}$ et F^c son complément algébrique, $z, b \in \mathbb{F}_{2^m}^*$. Soit $\mathcal{C}(F^{-1}(z))$ et $\mathcal{C}((F^c)^{-1}(b))$ deux codes cycliques 2^n-aires générés par g_F et g_{F^c} respectivement, alors*

(1) *Soit $F(0) = 0$, alors*
(i) $\mathcal{C}((F^c)^{-1}(b)) \subset \mathcal{C}(F^{-1}(z))$.
(ii) $\exists\, h_1 \in \mathbb{F}_{2^n}[x]$ tel que $g_{F^c} = g_F.h_1$.

(2) *Soit $F(0) \neq 0$, alors*
(i) $\mathcal{C}(F^{-1}(z)) \subset \mathcal{C}((F^c)^{-1}(b))$.
(ii) $\exists\, h_2 \in \mathbb{F}_{2^n}[x]$ tel que $g_F = g_{F^c}.h_2$.

Preuve. Soit $F(0) = 0$, c'est−à−dire $0 \notin F^{-1}(z)$, ce qui signifie que $0 \in (F^c)^{-1}(b)$. Puisque $F^{-1}(z) = (F^c)^{-1}(b) \cup \{0\}$, alors si $G \in AN(F^c)$ est associé à un mot de code $(a_1, \ldots, a_{2^n-1})$ du code $\mathcal{C}((F^c)^{-1}(b))$, alors par [(*i*), (2), lemma (105)], $\sum_{i=1}^{2^n-1} a_i x^i = 0$

pour tout $x \in (F^c)^{-1}(b)$, ainsi $\sum_{i=1}^{2^n-1} a_i x^i = 0$ pour tout $x \in (F^c)^{-1}(b) \cup \{0\}$. Donc $(a_1, \ldots, a_{2^n-1}) \in \mathcal{C}(F^{-1}(z))$.
Par [60, théorème 2.25], g_F divise g_{F^c}, alors il existe un polynôme $h_1(x) \in \mathbb{F}_{2^n}[x]$ tel que $g_{F^c} = g_F.h_1(x)$.

Soit $F(0) \neq 0$, c'est−à−dire $0 \in F^{-1}(z)$, ce qui signifie que $0 \notin (F^c)^{-1}(b)$. Puisque $F^c(x) = F(x)$ pour tout $x \in \mathbb{F}_{2^n}^*$, alors $F^{-1}(z) = (F^c)^{-1}(b) \cap \mathbb{F}_{2^n}^*$.
Supposons que $G \in AN(F)$ est associé à un mot de code $(a_1, \ldots, a_{2^n-1})$ du code $\mathcal{C}(F^{-1}(z))$, by [(ii), (2), lemma (105)],on a $\sum_{i=1}^{2^n-1} a_i x^i = 0$ pour tout $x \in F^{-1}(z)$. Donc $\sum_{i=1}^{2^n-1} a_i x^i = 0$ pour tout $x \in F^{-1}(z) \cap \mathbb{F}_{2^n}^*$, alors $(a_1, \ldots, a_{2^n-1}) \in \mathcal{C}((F^c)^{-1}(b))$.
Par [60, théorème 2.25], g_{F^c} divise g_F, alors il existe un polynôme $h_2(x) \in \mathbb{F}_{2^n}[x]$ tel que $g_F = g_{F^c}.h_2(x)$. ■

Soit $\mathcal{C}^\perp$ le code dual du code $\mathcal{C}(F^{-1}(z))$, alors d'après (84) et le théorème (103) on peut facilement déduire le polynôme générateur de $\mathcal{C}^\perp$.

Théorème 131. *Le polynôme générateur du code dual $\mathcal{C}^\perp$ du code cyclique $\mathcal{C}(F^{-1}(z))$ est donné comme suit :*
(i) Pour $F(0) \neq 0$,

$$g_F^\perp = x^{deg(p)+1} p(x^{-1}) p_0^{-1}.$$

(ii) Pour $F(0) = 0$,

$$g_F^\perp = x^{deg(p)} p(x^{-1}) p_0^{-1},$$

où $p(x) = gcd(F(x), x^{2^n-1} + 1)$.

Autrement dit, le polynôme générateur de $\mathcal{C}^\perp$ est le polynôme réciproque de $p(x)$.

5.4 Conclusion

Dans ce chapitre, nous avons établi un nouveau lien entre les annulateurs des fonctions vectorielles et les codes LCD, et alors une borne inférieure sur l'immunité algébrique des fonctions vectorielles et Booléennes est donnée en fonction de la distance minimale des codes LCD associés. Ensuite, nous avons présenté un lien entre l'immunité algébrique et l'immunité spectrale des fonctions vectorielles, ainsi qu'un lien entre ces deux critères et le poids des codes LCD associés à ces fonctions.

Enfin, nous avons proposé une généralisation de la notion du complément algébrique des fonctions Booléennes aux fonctions vectorielles, et alors nous avons obtenu une borne sur le degré algébrique le plus bas des annulateurs des fonctions vectorielles.

Chapitre 6

Propriétés cryptographiques du produit de Dirichlet des fonctions Booléennes et vectorielles

Ce chapitre a fait l'objet d'une publication à la conférence internationale "DIMACOS 2019" [20].

Le produit de Dirichlet a été d'abord introduit pour les fonctions arithmétiques dans [2], où une fonction arithmétique est une application définie sur l'ensemble des entiers strictement positifs et à valeurs dans l'ensemble des nombres complexes. Nous avons :

Définition 132. *[2](Produit de Dirichlet) Soient F et G deux fonctions arithmétiques, alors le produit de Dirichlet de F et G est la fonction arithmétique $F * G$ définie, pour tout $n \in \mathbb{N}^*$, par*

$$(F * G) = \sum_{d|n} F(d)G(\frac{n}{d}) = \sum_{xy=n} F(x)G(y),$$

où $\sum_{d|n}$ désigne la somme sur les diviseurs positifs d de n, et la somme $\sum_{xy=n}$ est prise sur les couples $(x, y) \in \mathbb{N}^2$ vérifiant $xy = n$.

L'opération $*$ définie sur l'ensemble des fonctions arithmétiques est commutative $F * G = G * F$, associative $F * (G * H) = (F * G) * H$, et possède l'élément neutre I défini comme suit :

$$I(n) = \begin{cases} 1 \ si \ n = 0 \\ 0 \ si \ n \neq 0, \end{cases} \tag{6.1}$$

où $F * I = I * F = F$. Alors, l'ensemble des fonctions arithmétiques $\mathbb{N} \to \mathbb{R}$ muni du produit de Dirichlet forme un monoïde Abélien. Cependant, si de plus $F(1) \neq 0$, alors F

admet un inverse, ainsi, le sous−ensemble de toutes les fonctions arithmétiques tel que $F(1) \neq 0$ est un groupe Abélien par rapport à la multiplication de Dirichlet.

Dans [137], les auteurs ont introduit le produit de Dirichlet pour les fonctions Booléennes, c'est−à−dire que pour deux fonctions Booléennes f et g, et pour tout $x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$,

$$(f * g)(x) = \sum_{u \preceq x} f(u)g(x-u).$$

Si on note par $\mathbb{B}_n^+$ l'ensemble des fonctions Booléennes $n-$variables telles que $f(0) = 1$. L'ensemble $(\mathbb{B}_n^+, *)$ est un groupe Abélien, et la fonction d'identité I satisfaisant

$$I(x) = \begin{cases} 1 \ si \ x = 0 \\ 0 \ si \ x \neq 0 \end{cases} \tag{6.2}$$

Dans ce qui suit, nous allons présenter une caractérisation des propriétés cryptographiques du produit de Dirichlet pour les fonctions Booléennes, telle que la transformée de Fourier discrète, la transformée de Walsh et le poids de Hamming. Ensuite, Nous présentons une généralisation de cette notion aux fonctions vectorielles.

6.1 Propriétés du produit de Dirichlet pour les fonctions Booléennes

Dans cette section, nous commencerons d'abord par donner la transformée de Walsh de $f * g$, en terme de la transformée de Walsh de f et la transformée de Walsh de g.

Soit g une fonction Booléenne, alors à partir de la formule de sommation de Poisson donnée par le théorème (25) nous avons ce qui suit :

$$\sum_{y \preceq \acute{x}} W_g(y) = 2^{n-wt(x)} \sum_{y \preceq x} (-1)^{g(y)}. \tag{6.3}$$

6.1.1 La transformée de Walsh de $f * g$

Soit $\hat{f}(x) = f_x = \sum_{u \preceq x} f(u)$. La transformée de Fourier discrète de $f * g$ est donnée par le lemma suivant :

Lemme 133. *La transformée de Fourier discrète de $f * g$, pour tout $\alpha \in \mathbb{F}_2^n$, est donnée*

par

$$F_{f*g}(a) = 2^{-n} \sum_{u \in \mathbb{F}_2^n} (-1)^{a \cdot u} \sum_{v \preceq \overline{u}} F_{\hat{f}}(v) \sum_{a+y \preceq u} F_g(y).$$

Preuve. À partir de la définition (22) de la transformée de Fourier discrète, nous avons,

$$\begin{aligned} F_{f*g}(a) &= \sum_{x \in \mathbb{F}_2^n} (f * g)(x)(-1)^{a \cdot x}. \\ &= \sum_{x \in \mathbb{F}_2^n} \sum_{u \preceq x} f(u) g(x-u)(-1)^{a \cdot x}. \\ &= \sum_{u \in \mathbb{F}_2^n} f(u) \sum_{x \in \mathbb{F}_2^n, u \preceq x} g(x-u)(-1)^{a \cdot x}. \\ &= \sum_{u \in \mathbb{F}_2^n} f(u)(-1)^{a \cdot u} \sum_{x \in \mathbb{F}_2^n, u \preceq x} g(x-u)(-1)^{a \cdot (x-u)}. \end{aligned}$$

D'après la relation (26),

$$\begin{aligned} \sum_{x \in \mathbb{F}_2^n, u \preceq x} g(x-u)(-1)^{a \cdot (x-u)} &= \sum_{x \in \mathbb{F}_2^n, x \preceq \overline{u}} g(x)(-1)^{a \cdot x}. \\ &= 2^{wt(\overline{u})-n} \sum_{a+y \preceq u} F_g(y). \\ &= 2^{-wt(u)} \sum_{a+y \preceq u} F_g(y). \end{aligned}$$

Par la définition (18) de la transformée de Möbius et par l'équation (6.3), nous avons,

$$f(u) = 2^{wt(u)-n} \sum_{v \preceq \overline{u}} F_{\hat{f}}(v).$$

C'est−à−dire,

$$\begin{aligned} F_{f*g}(a) &= \sum_{u \in \mathbb{F}_2^n} 2^{n-wt(u)} (-1)^{a \cdot u} \sum_{v \preceq \overline{u}} F_{\hat{f}}(v) 2^{wt(u)} \sum_{a+y \preceq u} F_g(y). \\ &= 2^{-n} \sum_{u \in \mathbb{F}_2^n} (-1)^{a \cdot u} \sum_{v \preceq \overline{u}} F_{\hat{f}}(v) \sum_{a+y \preceq u} F_g(y). \end{aligned}$$

Alors,

$$F_{f*g}(a) = 2^{-n} \sum_{u \in \mathbb{F}_2^n} (-1)^{a \cdot u} \sum_{v \preceq \overline{u}} F_{\hat{f}}(v) \sum_{a+y \preceq u} F_g(y).$$

■

En utilisant la relation entre la transformée de Walsh (définition (23)) et la transformée de Fourier discrète (définition (22)) des fonctions Booléennes donnée dans [31]. La

proposition suivante donne la transformée de Walsh de $f * g$.

Théorème 134. *Soit f et g deux fonctions Booléennes à n variables. Alors, pour tout $a \in \mathbb{F}_2^n$, la transformée de Walsh de $f * g$ est donnée comme suit :*

i) Pour $a \in \mathbb{F}_2^n$ et $a \neq 0$,

$$W_{f*g}(a) = -2^{n-1} \sum_{u,\, a \preceq u} (-1)^{a \cdot u} - \sum_{u \in \mathbb{F}_2^n} (2^{-1-n} \sum_{v \preceq \overline{u}} W_{\hat{f}}(v) - \frac{1}{2})(-1)^{a \cdot u} \sum_{a+y \preceq u} W_g(y) + \frac{1}{2} \sum_{u,\, a \preceq u} (-1)^{a \cdot u} \sum_{v \preceq \overline{u}} W_{\hat{f}}(v).$$

ii) Pour $a = 0$,

$$W_{f*g}(0) = 2^n - 2^{2n-1} + \frac{1}{2} \sum_{u \in \mathbb{F}_2^n} 2^{-wt(u)} W_{\hat{f}}(u)(2^n - \sum_{y \preceq \overline{u}} W_g(y)) + 2^{n-1} \sum_{u \in \mathbb{F}_2^n} 2^{-wt(u)} W_g(u).$$

Preuve. Par définition, nous avons $W_f(a) = 2^n \delta(a) - 2F_f(a)$, où

$$\delta(a) = \begin{cases} 1 \ si \ a = 0 \\ 0 \ si \ a \neq 0 \end{cases}$$

C'est−à−dire,

$$\begin{aligned} W_{f*g}(a) &= 2^n \delta(a) - 2F_{f*g}(a). \\ &= 2^n \delta(a) - 2^{1-n} \sum_{u \in \mathbb{F}_2^n} (-1)^{a \cdot u} \sum_{v \preceq \overline{u}} F_{\hat{f}}(v) \sum_{a+y \preceq u} F_g(y). \\ &= 2^n \delta(a) - 2^{1-n} \sum_{u \in \mathbb{F}_2^n} (-1)^{a \cdot u} \sum_{v \preceq \overline{u}} (2^{n-1}\delta(v) - 2^{-1} W_{\hat{f}}(v)) \sum_{a+y \preceq u} (2^n \delta(y) - 2^{-1} W_g(y)). \\ &= 2^n \delta(a) - 2^{1-n} \sum_{u \in \mathbb{F}_2^n} (-1)^{a \cdot u} (\sum_{v \preceq \overline{u}} 2^{n-1}\delta(v) - \frac{1}{2} \sum_{v \preceq \overline{u}} W_f(y))(\sum_{a+y \preceq u} 2^n \delta(y) - \frac{1}{2} \sum_{a+y \preceq u} W_g(y))). \\ &= 2^n \delta(a) - 2^{1-n} \sum_{u \in \mathbb{F}_2^n} (-1)^{a \cdot u} (2^{n-1} - \frac{1}{2} \sum_{v \preceq \overline{u}} W_{\hat{f}}(v))(2^{n-1} \sum_{a+y \preceq u} \delta(y) - \frac{1}{2} \sum_{a+y \preceq u} W_g(y)). \\ &= 2^n \delta(a) - 2^{n-1} \sum_{u,\, a \preceq u} (-1)^{a \cdot u} + \frac{1}{2} \sum_{u,\, a \preceq u} (-1)^{a \cdot u} \sum_{v \preceq \overline{u}} W_{\hat{f}}(v) + \frac{1}{2} \sum_{u \in \mathbb{F}_2^n} (-1)^{a \cdot u} \sum_{a+y \preceq u} W_g(y) - \\ & 2^{-1-n} \sum_{u \in \mathbb{F}_2^n} (-1)^{a \cdot u} \sum_{v \preceq \overline{u}} W_{\hat{f}}(v) \sum_{a+y \preceq u} W_g(y). \end{aligned}$$

i) Soit $a \neq 0$, alors

$$W_{f*g}(a) = -2^{n-1}\sum_{u,\, a\preceq u}(-1)^{a\cdot u} - \sum_{u\in\mathbb{F}_2^n}(2^{-1-n}\sum_{v\preceq\overline{u}}W_{\hat{f}}(v) - \frac{1}{2})(-1)^{a\cdot u}\sum_{a+y\preceq u}W_g(y) + \frac{1}{2}\sum_{u,\, a\preceq u}(-1)^{a\cdot u}\sum_{v\preceq\overline{u}}W_{\hat{f}}(v).$$

ii) Soit $a = 0$, c'est−à−dire $\delta(0) = 1$, alors

$$W_{f*g}(0) = 2^n - 2^{2n-1} + \frac{1}{2}\sum_{u\in\mathbb{F}_2^n}\sum_{v\preceq\overline{u}}W_{\hat{f}}(v) + \sum_{y\preceq u}W_g(y) - 2^{-1-n}\sum_{u\in\mathbb{F}_2^n}\sum_{v\preceq\overline{u}}W_{\hat{f}}(v)\sum_{y\preceq u}W_g(y).$$

Par le corollaire (27), nous avons :

$$\sum_{u\in\mathbb{F}_2^n}\sum_{v\preceq\overline{u}}W_{\hat{f}}(v) = \sum_{u\in\mathbb{F}_2^n}2^{n-wt(u)}W_{\hat{f}}(u),$$

alors

$$\sum_{u\in\mathbb{F}_2^n}\sum_{v\preceq\overline{u}}W_{\hat{f}}(v)\sum_{y\preceq u}W_g(y) = \sum_{u\in\mathbb{F}_2^n}2^{n-wt(u)}W_{\hat{f}}(u)\sum_{y\preceq\overline{u}}W_g(y).$$

C'est−à−dire,

$$W_{f*g}(0) = 2^n - 2^{2n-1} + \frac{1}{2}\sum_{u\in\mathbb{F}_2^n}2^{-wt(u)}W_{\hat{f}}(u)(2^n - \sum_{y\preceq\overline{u}}W_g(y)) + 2^{n-1}\sum_{u\in\mathbb{F}_2^n}2^{-wt(u)}W_g(u).$$

■

6.1.2 Le poids de Hamming $f * g$

Une autre propriété importante des fonctions Booléennes est le poids de Hamming de ces derniers. Alors, dans ce qui suit, nous donnons le poids de Hamming de la fonction Booléenne $f * g$ en terme du poids de Hamming de f et le poids de Hamming g.

Lemme 135. *Soit $f*g$ le produit de Dirichlet de deux fonctions Booléennes f et g. Alors le poids de Hamming $wt(f * g)$ est,*

$$wt(f * g) = -2^{2n-1} + \sum_{u\in\mathbb{F}_2^n}wt(\hat{f}_u)wt(g_{\overline{u}}).$$

Preuve. La relation entre le poids de Hamming de $f * g$ et sa transformée de Walsh est donnée par $wt(f * g) = 2^{n-1} - \frac{1}{2}W_{f*g}(0)$. à partir de (ii) du théorème (134) et de la relation (26) nous avons ce qui suit :

$$wt(f*g) = 2^{n-1} - \frac{1}{2}(2^n - 2^{2n-1} + \frac{1}{2}\sum_{u\in\mathbb{F}_2^n}(2^n - 2^{wt(\overline{u})+1}wt(\hat{f}_u) + 2^n - 2^{wt(u)+1}wt(g_{\overline{u}}))-$$
$$\frac{1}{2^{n+1}}\sum_{u\in\mathbb{F}_2^n}(2^n - 2^{wt(\overline{u})+1}wt(\hat{f}_u))(2^n - 2^{wt(u)+1}wt(g_{\overline{u}}))).$$
$$= -2^{2n-1} + \sum_{u\in\mathbb{F}_2^n} wt(\hat{f}_u)wt(g_{\overline{u}}).$$

■

6.1.3 Complément algébrique de $f * g$

Dans [172], les auteurs ont étendu le concept de l'immunité algébrique des fonctions Booléennes, ensuite, ils ont introduit la notion de complément algébrique f^c comme indiqué dans (5.3.1). Dans [167] et [172] les auteurs ont caractérisé certaines propriétés du complément algébrique des fonctions Booléennes.

Dans ce qui suit, nous utilisons les résultats de [167, 172] pour donner le complément algébrique de la fonction Booléenne $f * g$.

Lemme 136. *Soient f et g deux fonctions Booléennes telle que $f(0) = g(0) = 1$. Soient f^c et g^c leurs compléments algébriques, alors*

$$f^c * g^c = (f * g) + f + g + 1.$$

Preuve. Par la définition (123), nous avons ce qui suit :

$$\begin{aligned}(f^c * g^c)(x) &= \sum_{u\preceq x} f^c(u)g^c(x-u).\\ &= \sum_{u\preceq x}(f(u)+\Delta(u))(g(x-u)+\Delta(x-u)).\\ &= \sum_{u\preceq x} f(u)g(x-u) + f(u)\Delta(x-u) + \Delta(u)g(x-u) + \Delta(u)\Delta(x-u).\\ &= (f*g)(x) + (f*\Delta)(x) + (g*\Delta)(x) + (\Delta*\Delta)(x).\\ &= (f*g)(x) + (f*I)(x) + (g*I)(x) + 1(x).\\ &= (f*g)(x) + f(x) + g(x) + 1(x).\end{aligned}$$

■

Le degré algébrique de $f^c * g^c$ est donné par le lemma suivant :

Lemme 137. *Soit f et g deux fonctions Booléennes et f^c, g^c leurs compléments algébriques, alors*

$$deg(f^c * g^c) \leq 1 + 2deg(f * g).$$

Preuve. Due à [137], on a $deg(f)+deg(g) \geq deg(f*g*1)$. Puisque $f^c*g^c = (f*g)+f+g+1$, alors

$$\begin{aligned} deg(f^c * g^c) &= deg((f * g) + f + g + 1). \\ deg(f^c * g^c) &\leq deg(f * g) + deg(f + g + 1). \\ deg(f^c * g^c) &\leq deg(f) + deg(g) + deg(f + g + 1). \\ deg(f^c * g^c) &\leq 1 + 2(deg(f) + deg(g)). \end{aligned}$$

C'est−à−dire,

$$deg(f^c * g^c) \leq 1 + 2deg(f * g).$$

■

6.2 Le produit de Dirichlet des fonctions vectorielles

Dans cette section nous donnons une généralisation de la définition du produit de Dirichlet des fonctions Booléennes aux $(n, m)-$fonctions.

Dans ce qui suit, nous utilisons le produit de deux $(n, m)-$fonctions défini dans [177] comme suit :

Soient F et G deux $(n, m)-$fonctions, alors

$$\langle F.G \rangle = \sum_{i=1}^{m} f_i g_i.$$

Le lemma suivant donne le produit de Dirichlet de deux $(n, m)-$fonctions.

Lemme 138. *Soit F et G deux $(n, m)-$fonctions sur $\mathbb{F}_2^m$, alors pour $x \in \mathbb{F}_2^n$ le produit de Dirichlet $F * G$ est :*

$$(F * G)(x) = \sum_{i=1}^{m} (f_i * g_i)(x).$$

Preuve. Comme le produit de F et G est $(F.G)(x) = \sum_{i=1}^{m} f_i(x)g_i(x)$, nous avons,

$$\begin{aligned}(F * G)(x) &= \sum_{u \preceq x} F(u)G(x-u). \\ &= \sum_{u \preceq x} (f_1(u), \ldots, f_m(u))(g_1(x-u), \ldots, g_m(x-u)). \\ &= \sum_{u \preceq x} \sum_{i=1}^{m} f_i(u)g_i(x-u). \\ &= \sum_{i=1}^{m} \sum_{u \preceq x} f_i(u)g_i(x-u).\end{aligned}$$

C'est−à−dire, $(F * G)(x) = \sum_{i=1}^{m} (f_i * g_i)(x)$. ■

Nous désignons par B_n^m l'ensemble de toutes les fonctions vectorielles de $\mathbb{F}_2^n$ dans $\mathbb{F}_2^m$. Alors le résultat suivant montre que $(B_n^m, *)$ est un monoïde Abélien.

Lemme 139. *$(B_n^m, *)$ est un monoïde Abélien.*

Preuve. D'abord nous avons $F * G = G * F$

$$\begin{aligned}(F * G)(x) &= \sum_{i=1}^{m} (f_i * g_i)(x) \\ &= \sum_{i=1}^{m} (g_i * f_i)(x) = (G * F)(x).\end{aligned}$$

C'est−à−dire que le produit de Dirichlet des (n, m)−fonctions est commutatif. Soit H une (n, m)−fonction, pour $x \in \mathbb{F}_2^n$, nous avons,

$$\begin{aligned}((F * G) * H)(x) &= \sum_{u \leq x} (F * G)(u)H(x-u). \\ &= \sum_{u,v \leq x,\, u+v=x} (F * G)(u)H(v). \\ &= \sum_{u,v \leq x,\, u+v=x} \sum_{w \leq u} F(w)G(u-w)H(v). \\ &= \sum_{y,v,w \leq x,\, y+v+w=x} F(v)G(y)H(v). \\ &= (F * (G * H))(x).\end{aligned}$$

Alors le produit Dirichlet est associatif.

Enfin, soit $I = (I_1, \ldots, I_m)$ tel que pour tout $1 \leq i \leq m$,

$$I(x) = \begin{cases} 1 \ si \ x = 0 \\ 0 \ si \ x \neq 0 \end{cases}$$

Nous avons,

$$\begin{aligned} (F * I)(x) &= \sum_{u \leq x} F(u) I(x-u). \\ &= \sum_{u,v \leq x,\, u+v=x} F(u) I(v). \\ &= F(x). \end{aligned}$$

Alors, $(B_n^m, *)$ monoïd Abélien. ■

6.3 Conclusion

Dans ce chapitre, nous avons présenté une caractérisation des propriétés cryptographiques du produit de Dirichlet des fonctions Booléennes. Ainsi qu'une généralisation de la notion du produit de Dirichlet pour les fonctions Booléennes aux fonctions vectorielles.

Conclusion générale

Dans le domaine des codes correcteurs et de la cryptographie, la théorie des fonctions Booléennes et vectorielles est reliée à de nombreuses contraintes très différentes et certaines questions importantes sur ces objets restent toujours difficiles à atteindre, comme la question de la construction d'un grand nombre de fonctions satisfaisant au mieux tous les critères cryptographiques.

Les fonctions Booléennes et les fonctions vectorielles sont fondamentales dans les systèmes de chiffrement symétriques, notamment dans la conception des algorithmes de chiffrement à flot et par blocs. Afin de résister au mieux aux attaques connues (statistiques et algébriques), et aux attaques futures, les fonctions Booléennes ou vectorielles utilisées doivent vérifier un certain nombre de propriétés. Ainsi, elles doivent être équilibrées, avoir un haut degré algébrique et avoir une bonne non−linéarité. Après la découverte des attaques algébriques en 2003, qui sont des attaques réalisables contre les schémas à flot et plus difficilement contre les schémas par blocs. Cette famille d'attaques a donné lieu au critère d'immunité algébrique.

Dans le domaine des codes correcteurs, les fonctions Booléennes et vectorielles jouent un rôle très important, toute fonction Booléenne à n variables peut être spécifiée par un vecteur binaire de longueur 2^n représentant la liste des valeurs des fonctions Booléennes à n variables, ainsi, chaque code de longueur 2^n peut être défini comme un sous−ensemble de fonctions Booléennes, telles que les codes de Reed−Muller.

Cette étude a mis en avant les liens entre la cryptographie et la théorie des codes, à travers la correspondance entre les fonctions Booléennes et vectorielles cryptographiques d'un côté, et les codes correcteurs d'erreurs de l'autre côté.

Au cours de cette thèse, nous avons étudié plusieurs problèmes qui peuvent se formuler aussi bien en termes de cryptographie que de codes correcteurs. Cela n'est pas si étonnant étant donné que ces deux disciplines utilisent le même genre d'outils mathématiques. Nous

avons donné un lien direct entre les annulateurs des fonctions vectorielles et une classe des codes LCD.
Ensuite, nous avons utilisé les propriétés des codes LCD associés a ces fonctions et nous avons établi un lien entre l'immunité algébrique et l'immunité spectrale des fonctions vectorielles, ainsi qu'un lien entre ces deux critères et le poids des codes associés. Nous avons proposé une généralisation de la notion du complément algébrique des fonctions Booléennes aux fonctions vectorielles. Cette généralisation nous a mené à donner une borne sur le degré algébrique le plus bas des annulateurs des fonctions vectorielles. Enfin, nous avons étudié les propriétés cryptographiques du produit de Dirichlet pour les fonctions Booléennes. Par conséquent, nous avons proposons une généralisation de cette notion aux fonctions vectorielles.

À la fin, on peut dire qu'il y a beaucoup de choses qui restent à faire, comme trouver d'autres classes intéressantes de codes à partir des fonctions Booléennes ou des fonctions vectorielles avec de bons critères cryptographiques.

Bibliographie

[1] R. Anderson, Searching for the optimum correlation attack. Springer, Berlin Heidelberg, pp. 137−143, 1995.

[2] T.M. Apostol, Introduction to Analytic Number Theory. Springer, 1976.

[3] F. Armknacht, Algebraic attacks on certain stream ciphers. University of Mannheim, Germany, pp. 1-217, 2006.

[4] F. Armknecht, Improving Fast Algebraic Attacks. In FSE 2004, Lecture Notes in Computer Science, Vol. 3017, Springer, pp. 65-82, 2004.

[5] F. Armknecht and M. Krause, Constructing single-and multi-output Boolean functions with maximal immunity. ICALP (2), pp. 180-191, 2006.

[6] G. Ars, Applications des bases de Gröbner à la cryptographie. PhD thesis, University of Rennes I, 2005.

[7] G. Ars and J. C. Faugère, Algebraic immunities of functions over finite fields. Proceedings of the Conference BFCA 2005, Publications des universites de Rouen et du Havre, pp. 21-38, 2005.

[8] G. Ars, J. C. Faugère, H. Imai, M. Kawazoe, and M. Sugita, Comparison between XL and Gröbner basis algorithms. Advances in Cryptology, ASIACRYPT 2004,Lecture Notes in Computer Science, Vol. 3329 ,Springer, p. 338-353, 2004.

[9] D. Augot, Newton's identities for minimum codewords of a family of alternant codes. Short Abstract in the Proceeding of IEEE ISIT 95.

[10] M. Bardet, J. C. Faugère and B. Salvy, On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. Proceedings of the International Conference on Polynomial System Solving, pp. 71-74, 2004.

[11] M. Bardet, J. C. Faugère, B. Salvy, and B.Y. Yang, Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In MEGA 2005, 2005.

[12] P. Beelen and G. Leander, A new construction of highly nonlinear S-boxes. Cryptography and Communications 4(1), pp. 65-77, 2012.

[13] T. Beth and C. Ding, On almost perfect nonlinear permutations. Proceedings of EUROCRYPT 93, Lecture Notes in Computer Science, Vol. 765, pp. 65-76, 1994.

[14] E. Berlekamp, Algebraic Coding Theory. McGraw-Hill, 1968.

[15] A. Betten , M. Braun, H. Friperting, A. Kerber and A. Wassermann, Error corrcting codes. Algorithms and Computation in Mathematics, Vol. 18, Springer, 2006.

[16] E. Biham and A. Shamir, Differential Cryptanalysis of DES-like Cryptosystems. Journal of Cryptology, Vol. 4, (1), pp. 3-72, 1991.

[17] A. Biryukov and A. Shamir, Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers. Asiacrypt 2000, Lecture Notes in Computer Science, Vol. 2248, Springer, pp. 1-13, 2000.

[18] R.E. Blahut, Theory and practice of error control codes. Addison-Wesley, 1983.

[19] Y. Borissov, A. Braeken, S. Nikova and B. Preneel, On the covering radii of binary Reed-Muller codes in the set of resilient boolean functions. IEEE Transactions on Information Theory, 51(3), pp. 1182-1189, 2005.

[20] M. Boumezbeur and K. Guenda, Cryptographic properties of Dirichlet product for vectorial Boolean functions. DIMACOS, Tunisia, 2019.

[21] M. Boumezbeur, S. Mesnager and K. Guenda, Linear codes from vectorial Boolean functions in the context of algebraic attacks. Discrete Mathematics, Algorithms and Applications, Vol. 13 , 2020.

[22] E. Brier and P. Langevin, Classification of boolean cubic forms in nine variables. IEEE Information Theory Workshop, pp. 179-182, 2003.

[23] K. Browning, J. F. Dillon, R. E. Kibler and M. McQuistan, APN polynomials and related codes. Special volume of Journal of Combinatorics, Information and System Sciences, honoring the 75-th birthday of Prof. D.K.Ray-Chaudhuri, vol. 34, Issue 1-4, pp. 135-159, 2009.

[24] A. Canteaut, On the correlations between a combining function and functions of fewer variables. Proceedings of the Information Theory Workshop'02, Bangalore, 2002.

[25] A. Canteaut, C. Carlet, P. Charpin, and C. Fontaine, On cryptographic properties of the cosets of $R(1, m)$. IEEE Transactions on Information Theory 47 (4), pp. 1494-1513, 2001.

[26] A. Canteaut, C. Carlet, P. Charpin, and C. Fontaine, Propagation characteristics and correlation immunity of highly nonlinear Boolean functions. Proceedings of EUROCRYPT 2000, Lecture Notes in Computer Science, Vol. 187, pp. 507-522, 2000.

[27] A. Canteaut and M. Trabbia, Improved Fast Correlation Attacks Using ParityCheck Equations of Weight 4 and 5. Advanced in Cryptology- EUROCRYPT 2000. Lecture notes in computer science, Vol. 1807, pp. 573- 588, 2000.

[28] A. Canteaut and M. Videau, Degree of composition of highly nonlinear functions and applications to hight order differential cryptanalysis. Proceedings of EUROCRYPT 2002, Lecture Notes in Computer Science, Vol. 2332, pp. 518-533, 2002.

[29] C. Carlet, A Larger Class of Cryptographic Boolean Functions via a Study of the Maiorana-McFarland Construction. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 549-564 2002.

[30] C. Carlet, A lower bound on the higher order nonlinearity of algebraic immune functions. Cryptology ePrint Archive, Report 2005/469, 2005.

[31] C. Carlet, Boolean functions for cryptography and coding theory. Cambridge University Press, 2021.

[32] C. Carlet, On the higher order nonlinearities of algebraic immune functions. Lecture Notes in Computer Science, Vol. 4117, Springer, pp. 584-601, 2006.

[33] C. Carlet, On the higher order nonlinearities of Boolean functions and S-boxes, and their generalizations. Proceedings of International Conference on Sequences and Their Applications SETA 2008, Lecture Notes in Computer Science, Vol. 5203, pp. 345-367, 2008.

[34] C. Carlet, On the algebraic immunities and higher order nonlinearities of vectorial Boolean functions. NATO Science for Peace and Security Series, D : Information and Communication Security, Vol. 23, Enhancing Cryptographic Primitives with Techniques from Error Correcting Codes, pp. 104-116, 2009.

[35] C. Carlet, Recursive lower bounds on the nonlinearity profile of boolean functions and their applications. Cryptology ePrint Archive, Report 2006/459, 2006.

[36] C. Carlet, Two new classes of bent functions. Proceedings of EUROCRYPT 1993, Lecture Notes in Computer Science, Vol. 765, pp. 77-101, 1994.

[37] C. Carlet, Vectorial Boolean functions for cryptography, Chapter of the monography Boolean Models and Methods in Mathematics. Computer Science, and Engineering, Y. Crama and P. Hammer, eds., Cambridge University Press, pp. 398-469, 2010.

[38] C. Carlet, P. Charpin, and V. Zinoviev, Codes, bent functions and permutations suitable for DESlike cryptosystems. Designs, Codes and Cryptography, 15 (2), pp. 125-156, 1998.

[39] C. Carlet, D. Dalai, K. Gupta and S. Maitra, Algebraic Immunity for Cryptographically Significant Boolean Functions : Analysis and Construction. IEEE Transactions on Information Theory, vol. 52, no. 7, pp. 3105-3121, 2006.

[40] C. Carlet and C. Ding, Nonlinearities of S-boxes. Finite Fields and its Applications, Vol. 13 Issue 1, pp. 121-135, 2007.

[41] C. Carlet, C. Ding and J. Yuan, Linear codes from perfect nonlinear mappings and their secret sharing schemes. IEEE Transactions on Information Theory 51 (6), pp. 2089-2102, 2005.

[42] C. Carlet and K. Feng, An infinite class of balanced functions with optimal algebraic immunity, good immunity to fast algebraic attacks and good nonlinearity. In ASIACRYPT, Lecture Notes in Computer Science, Vol. 5350, Springer-Verlag, pp. 425-440, 2008.

[43] C. Carlet and A. Gouget, An upper bound on the number of m-resilient Boolean functions. Proceedings of ASIACRYPT 2002, Lecture Notes in Computer Science, Vol. 2501, pp. 484-496, 2002.

[44] C. Carlet and S. Guilley, Complementary dual codes for counter-measures to side-channel attacks. In : E. R. Pinto et al. (eds.), Coding Theory and Applications, CIM Series in Mathematical Sciences, vol. 3, pp. 97-105, Springer Verlag, 2014 and Journal Adv. in Math. of Comm. 10(1), pp. 131-150, 2016.

[45] C. Carlet and E. Prouff, On a new notion of nonlinearity relevant to multi-output pseudo-random generators. Proceedings of Selected Areas in Cryptography 2003, Lecture Notes in Computer Science, Vol. 3006, pp. 291-305, 2004.

[46] C. Carlet and P. Sarkar, Spectral Domain Analysis of Correlation Immune and Resilient Boolean Functions. Finite fields and Application 8, pp. 120-130, 2002.

[47] C. Carlet, D. Tang, Enhanced Boolean functions suitable for the filter model of pseudo-random generator. Designs, Codes and Cryptography 76 (3), pp. 571-587, 2015.

[48] F. Chabaud et S. Vaudenay, Links between differential and linear cryptanalysis. In Advances in Cryptology-EUROCRYPT'94, Lecture Notes in Computer Science, Vol. 950, Springer-Verlag, pp. 356-365, 1995.

[49] J. H. Cheon and D. H. Lee, Resistance of S-boxes against algebraic attacks. Proceedings of Fast Software Encryption FSE 2004, Lecture Notes in Computer Science, Vol. 3017, pp. 83-94, 2004.

[50] V. Chepyzhov and B. Smeets, On a Fast Correlation Attack on Certain Stream Ciphers. Proceedings of EUROCRYPT'91, Lecture Notes in Computer Science, Vol. 547, pp. 176-185, 1992.

[51] B. Chor, O. Goldreich, J. Hastad, J. Freidmann, S. Rudich, and R. Smolensky, The bit extraction problem or t-resilient functions. Proceedings of the 26'th IEEE Symposium on Foundations of Computer Science, pp. 396-407, 1985.

[52] C. Cid and G. Leurent, An analysis of the XSL algorithm. In Advances in Cryptology - ASIACRYPT 2005, Lecture Notes in Computer Science, Vol. 3788, Springer, pp. 333-352, 2005.

[53] C. Cid, S. Murphy, and M. Robshaw, Small Scale Variants of the AES. In Fast Software Encryption-FSE 2005, Lecture Notes in Computer Science, Vol. 3557, Springer-Verlag, pp. 145-162, 2005.

[54] N. Courtois, Cryptanalysis of SFINKS. ICISC 2005, Lecture Notes in Computer Science, Vol. 3935, Springer, pp. 261-269, 2006.

[55] N. Courtois, Fast Algebraic Attacks on Stream Ciphers with Linear Feedback. In Advances in Cryptology-CRYPTO 2003, Lecture Notes in Computer Science, Vol. 2729, Springer-Verlag, pp. 176-194, 2003.

[56] N. Courtois, A. Klimov, J. Patarin and A. Shamir, Efficient algorithmms for solving overdefined systems of multivariate polynomial equations. In Bart Preneel, Advances in Cryptology-EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Belgium, Proceeding, Lecture Notes in Computer Science, Vol. 1807, Springer, pp. 392-407, 2000.

[57] N. Courtois and W. Meier, Algebraic attacks on stream ciphers with linear feedback. In Advances cryptology-Eurocrypt 2003. Lecture Notes in computer science, Vol. 2656, Springer-Verlag, pp. 345-359, 2003.

[58] N. Courtois and J. Pieprzyk, Cryptanalysis of block ciphers with over-defined systems of equations. Proceedings of ASIACRYPT 2002, Lecture Notes in Computer Science, Vol. 2501, pp. 267-287, 2003.

[59] Y. Crama and PL. Hammer, Boolean models and methods in mathematics, computer science. and engineering, Encyclopedia of Mathematics and its Applications, Combridge university press, 2010.

[60] D. Cunsheng, Codes from Difference sets. World Scientific Publishing Company, 2004.

[61] J. Daemen and V. Rijmen, Probability distributions of correlation and differentials in block ciphers. Journal of Mathematical Cryptology (JMC) 1 (3), pp. 221-242, 2007.

[62] J. Daemen and V. Rijmen, The Design of Rijndael : AES. The Advanced Encryption Standard. Information Security and Cryptography, Springer, 2002.

[63] P. Delsarte, Bounds for unrestricted codes, by linear programming. Philips Research Reports 27, pp. 272-289, 1972.

[64] C. Diem, The XL algorithm and a conjecture from commutative algebra. Advances in Cryptology, ASIACRYPT 2004, Lecture Notes in Computer Science, Vol. 3329, Springer-Verlag, pp. 323-337, 2004.

[65] J. F. Dillon, Elementary Hadamard difference sets. Ph. D. Thesis, University of Maryland, 1974.

[66] C. Ding, A construction of binary linear codes from Boolean functions. Discrete Mathematics 339 (9), pp. 2288-2303, 2016.

[67] C. Ding, A sequence construction of cyclic codes over finite fields. Cryptography and Communications 10 (2), pp. 319-341, 2018.

[68] C. Ding, Cyclic codes from some monomials and trinomials. SIAM Journal on Discrete Mathematics 27 (4), pp. 1977-1994, 2013.

[69] C. Ding, Linear codes from some 2-designs. IEEE Transactions on Information Theory 60 (6), pp. 3265-3275, 2015.

[70] C. Ding, G.Z. Xiao and W. Shan, The Stability Theory of Stream Ciphers. Lecture Notes in Computer Science, Springer-Verlag, Vol. 561, 1991.

[71] C. Ding and Z. Zhou, Binary cyclic codes from explicit polynomials over GF(2m). Discrete Mathematics 321, pp. 76-89, 2014.

[72] Y. Du, B. Wei, F. Zhang and H. Zhang, On the (Fast) Algebraic Immunity of Boolean Power Functions. Cryptology ePrint Archive, Report 2015/435, 2015.

[73] Y. Du, F. Zhang, M. Liu, On the Resistance of Boolean Functions against Fast Algebraic Attacks. In ICISC 2011, Lecture Notes in Computer Science, Vol. 7259, Springer, pp. 261-274, 2012.

[74] P. Duvall and J. Mortick, Some symptoms of Boolean functions. 1970.

[75] P. Elias, Coding for two noisy channels. Information Theory, 3(6), pp. 61-76, 1955.

[76] H. Englund and T. Johansson, A new simple technique to attack filter generators and related ciphers. Selected Areas in Cryptography SAC, Lecture Notes in Computer Science, Vol. 3357, Springer-Verlag, pp. 39-53, 2004.

[77] J. C. Faugère, A new efficient algorithm for computing Gröbner bases without reduction to zero ($F5$). International Symposium on Symbolic and Algebraic Computation, ACM Press, pp. 75-83, 2002.

[78] J. C. Faugère, A new efficient algorithm for computing Gröbner bases ($F4$). Journal of Pure and Applied Algebra 139, pp. 61-88, 1999.

[79] J .C. Faugère and G. Ars, An algebraic cryptanalysis of nonlinear filter generators using Gröbner bases. Rapport de Recherche INRIA 4739, 2003.

[80] J. C. Faugère, P. Gianni, D. Lazard, and T. Mora, Efficient computation of zero-dimensional Gröbner bases by change of ordering. J. Symb. Comput., 16(4), pp. 329-344, 1993.

[81] K. Feng, Q. Liao and J. Yang, Maximal values of generalized algebraic immunity. Designs, Codes and Cryptography 50(2), pp. 243-252, 2009.

[82] "FIPS 46-3. Data Encryption Standard". Federal Information Processing Standards Publication 46-3, 1999. U.S. Department of Commerce/N.I.S.T.

[83] "FIPS 197. Advanced Encryption Standard". Federal Information Processing Standards Publication 197, 2001. U.S. Department of Commerce/N.I.S.T.

[84] S. Fischer and W. Meier, Algebraic immunity of S-boxes and augmented functions. FSE, Lecture Notes in Computer Science, Vol. 4593, pp. 366-381, 2007.

[85] D. G. Fon-Der-Flaass, A bound on correlation immunity. Sib. Elektron. Mat. Izv. 4, pp. 133-135, 2007.

[86] R. Forré, The strict avalanche criterion : Spectral properties of Boolean functions and an extended definition Lecture Notes in Computer Science, Vol. 263, Springer-Verlag, pp. 450-468, 1988.

[87] R. Fourquet and C. Tavernier. List decoding of second order Reed–Muller and its covering radius implications. Proceedings of Workshop on Coding and Cryptography WCC 2007, pp. 147-156, 2007.

[88] S. W. Golomb and G. Gong, Signal Design for Good Correlation. For Wireless Communication, Cryptography, and Radar. Cambridge University Press, 2004.

[89] G. Gong, A closer look at selective DFT attacks. CACR report 2011-35, University of Waterloo, 2011.

[90] G. Gong, Sequences, DFT and resistance against fast algebraic attacks. Proceedings of International Conference on Sequences and Their Applications SETA 2008, Lecture Notes in Computer Science, Vol. 5203, pp. 197-218, 2008.

[91] G. Gong, S. Rønjom, T. Helleseth and H. Hu, Fast Discrete Fourier Spectra Attacks on Stream Ciphers. IEEE Transactions on Information Theory, 57(8), pp. 5555-5565, 2011.

[92] M. H. Jr, Combinatorial Theory. Ginn-Blaisdell, Waltham, 1967.

[93] P. Hawkes and L. O'Connor, XOR and Non-XOR differential probabilities. Proceedings of EUROCRYPT 1999, Lecture Notes in Computer Science, Vol. 1592, pp. 272-285, 1999.

[94] P. Hawkes and G.G. Rose, Rewriting Variables : The Complexity of Fast Algebraic Attacks on Stream Ciphers. In Advances in Cryptology-CRYPTO 2004, Lecture Notes in Computer Science, Vol. 3152, Springer-Verlag, pp. 390-406, 2004.

[95] T. Helleseth and S. Rønjom, Simplifying Algebraic Attacks with Univariate Analysis. In IEEE, Information Theory and Applications (ITA), pp. 153-159, 2011.

[96] T. Herlestam, On linear shift registers with permuted feedback. In Ingemar Ingemarsson, EUROCRYPT'86, Lecture Notes in Computer Science, Springer, Heidelberg, pp. 38-39, 1986.

[97] N. Jacobson, Basic algebra I. Freeman, W.H. and Company, 1985.

[98] D. Kahn, The codebreakers. New York Macmillan, 1967.

[99] A. M. Kerdock, A class of low-rate non linear codes. Information and Control, Vol. 20, pp. 182-187, 1972.

[100] E.L. Key, An analysis of the structure and complexity of nonlinear binary sequence generators. IEEE Trans. Information Theory, Vol. 22, pp. 732-736, 1976.

[101] L. Knudsen, Truncated and Higher Order Differentials. In B. Preneel, editor, Fast Software Encryption, FSE'94, Lecture Notes in Computer Science, vol. 1008, Springer, pp. 196-211, 1994.

[102] L. R. Knudsen and M. P. J. Robshaw, Non-linear approximations in linear cryptanalysis. Proceedings of EUROCRYPT 1996, Lecture Notes in Computer Science, Vol. 1070, pp. 224-236, 1996.

[103] L. R. Knudsen and D. Wagner, Integral cryptanalysis. Proceedings of Fast Software Encryption FSE 2002, Lecture Notes in Computer Science, Vol. 2365, pp. 112-127, 2002.

[104] K. Kurosawa, T. Iwata and T. Yoshiwara, New covering radius of Reed-Muller codes for t-resilient functions. IEEE Transactions on Information Theory, 20(3), pp. 468-475, 2004.

[105] X. Lai, Higher order derivatives and differential cryptanalysis. Proceedings of the "Symposium on Communication, Coding and Cryptography", in honor of J. L. Massey on the Occasion of his 60'th birthday, pp. 227-233, 1994.

[106] P. Langevin, Covering radius of RM(1,9) in RM(3,9). Lecture Notes in Computer Science, Vol. 514, Springer, pp. 51-59, 1991.

[107] P. Langevin, P. Rabizonni, P. Véron and J. P. Zanotti, On the number of bent functions with 8 variables. In Boolean Functions : Cryptography and Applications, BFCA'06, Rouen, 2006.

[108] RJ. Lechner, Harmonic analysis of switching functions, Recent developments in switching theory, New York Academic Press, 1971.

[109] S. Leveiller, G. Zémor, P. Guillot, and J. Boutros, A new cryptanalytic attack for PN-generators filtered by a Boolean function. Selected Areas in Cryptography SAC 2020, Lecture Notes in Computer Science, Vol. 2595, Springer-Verlag, pp. 232-249, 2003.

[110] R. Lidl and H. Niederreiter, Finite Fields, Encyclopedia of Mathematics and its Applications. vol. 20, Addison-Wesley, Reading, Massachus-setts, 1983.

[111] P. Lisonek, L. Trummer, Algorithms for the minimum weight of linear codes. Adv. in Math. of Comm. 10, pp. 195-207, 2016.

[112] M. Liu, D. Lin, and D. Pei, Fast algebraic attacks and decomposition of symmetric Boolean functions. IEEE Transactions on Information Theory, 57(7), pp. 4817-4821, 2011.

[113] J. Liu, S. Mesnager, L. Chen, On the nonlinearity of S-boxes and linear codes. Journal Cryptography and Communications 9(3), pp. 345-361, 2017.

[114] S. Lloyd, Properties of binary functions. Proceedings of EUROCRYPT 1990, Lecture Notes in Computer Science, Vol. 473, pp. 124-139, 1991.

[115] M. Lobanov, Exact relation between nonlinearity and algebraic immunity. Discrete Mathematics and Applications 16(5), pp. 453-460, 2006.

[116] M. Lobanov, Tight bound between nonlinearity and algebraic immunity. IACR Cryptology ePrint Archive (http ://eprint.iacr.org/) 2005/441, 2005.

[117] F.J. MacWilliams and N.J. Sloane, The theory of error correcting codes. Amsterdam, North Holland 1977.

[118] J.L. Massey, Linear codes with complementary duals. Discrete Math., vol. 106-107, pp. 337-342, 1992.

[119] J.L. Massey, Reversible codes. Information&Control, Vol. 7. pp. 369-380, Sept 1964.

[120] J.L. Massey, Shift-register analysis and BCH decoding. IEEE Trans. Inf. Theory, vol. 15, pp. 122-127, 1969.

[121] J.L. Massey, The ubiquity of Reed-Muller codes. In AAECC-14 : Proceedings of the 14th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Springer-Verlag, pp. 1-12, 2001.

[122] J.L. Massey and S. Serconek, A Fourier transform approach to the linear complexity of nonlinearly filtered sequences. In Yvo Desmedt, CRYPTO'94, Lecture Notes in Computer Science, Vol. 839, Springer, Heidelberg, pp. 332-340, 1994.

[123] M. Matsui, Linear cryptanalysis method for DES cipher. In Proc. Advances in Cryptology-EUROCRYPT'93, Lecture Notes in Computer Science, Vol. 765 Springer-Verlag, pp. 386-397, 1994.

[124] W. Meier, E. Pasalic and C. Carlet, Algebraic attacks and decomposition of Boolean functions. in Eurocrypt 2004. Lecture notes in computer science, Vol. 3027, 2004.

[125] W. Meier and O. Staffelbach, Fast correlation attacks on certain stream ciphers. Journal of Cryptology, pp. 159-176, 1989.

[126] W. Meier and O. Staffelbach, Fast correlation attacks on stream ciphers. Advances in Cryptology, EUROCRYPT'88, Lecture Notes in Computer Science, Vol. 330, pp. 301-314, 1988.

[127] S. Mesnager, A note on linear codes and algebraic immunity of Boolean functions. Proceedings of the 21-st International Symposium on Mathematical Theory of Networks and Systems (MTNS), pp. 923-927, 2014.

[128] S. Mesnager, Bent vectorial functions and linear codes from o-polynomials. Designs, Codes and Cryptography. 77 (1), pp. 99-116, 2015.

[129] S. Mesnager, Improving the lower bound on the higher order non- linearity of Boolean functions with prescribed algebraic immunity. IEEE Transactions on Information Theory, Vol. 54, pp. 3656-3662, 2008.

[130] S. Mesnager and G.D. Cohen, Cyclic codes and algebraic immunity of Boolean functions. Proceedings of the IEEE Information Theory Workshop (ITW), Jerusalem, Israel, pp. 1-5, 2015.

[131] S. Mesnager and G.D. Cohen, Fast algebraic immunity of Boolean functions. Advances in Mathematics of Communications 11 (2), pp. 373-377, 2017.

[132] S. Mesnager and C. Tang, Fast algebraic immunity of Boolean functions and LCD codes. IACR Cryptol. ePrint Arch., Vol. 2020, 720, 2020.

[133] H. Molland, Improved linear consistency attack on irregular clocked keystream generators. FSE 2004 (B.K. Roy and W. Meier, eds.), Lecture Notes in Computer Science, Vol. 3017, Springer-Verlag, pp. 109-126, 2004.

[134] H. Molland and T. Helleseth, An improved correlation attack. Crypto 2004 (M. Franklin, ed.), Lecture Notes in Computer Science, Vol. 3152, Springer-Verlag, pp. 373-389, 2004.

[135] D.E. Muller, Application of Boolean algebra to switching circuit design and to error detecting. IRE Trans. Electronic Computers, EC-3, 1954.

[136] S. K. Muttoo and S. Lal, A reversible code over GF(q). Kybernetika, vol. 22, pp. 85-91, 1986.

[137] A. Nitaj, W. Susilo, J. Tonien, Dirichlet Product for Boolean Functions. IACR Cryptology ePrint Archive, 673, 2016.

[138] K. Nyberg, Differentially uniform mappings for cryptography. Proceedings of EUROCRYPT 1993, Lecture Notes in Computer Science, Vol 765, pp. 55-64, 1994.

[139] K. Nyberg, On the construction of highly nonlinear permutations. Proceedings of EUROCRYPT 1992, Lecture Notes in Computer Science, Vol. 658, pp. 92-98, 1993.

[140] K. Nyberg, Perfect non-linear S-boxes. Proceedings of EUROCRYPT 1991, Lecture Notes in Computer Science, Vol. 547, pp. 378-386, 1992.

[141] K. Nyberg and L. R. Knudsen, Provable security against differential cryptanalysis. Journal of Cryptology 8 (1), pp. 27-37, 1995, (extended version of the Proceedings of CRYPTO 1992, Lecture Notes in Computer Science, Vol. 740, pp. 566-574, 1993).

[142] L. O'Connor, On the distribution of characteristics in bijective mappings. Proceedings of EUROCRYPT 1993, Lecture Notes in Computer Science, Vol. 765, pp. 360-370, 1993.

[143] E. Pasalic, Almost Fully Optimized Infinite Classes of Boolean Functions Resistant to (Fast) Algebraic Cryptanalysis. In ICISC 2008, Lecture Notes in Computer Science, Vol. 5461, Springer, pp. 399-414, 2009.

[144] J. Pieprzyk and G. Finkelstein, Towards effective nonlinear cryptosystem design. IEE Proceedings Part E, 35(6), pp. 325-335, 1988.

[145] E. Prange, Cyclic error-correcting codes in two symbols. Electronics Research Directorate, Air Force Cambridge Research Center, No. AFCRC-TN-57-103, 1957.

[146] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts and J. Vandevalle, Propagation characteristics of Boolean functions, Proceedings of EUROCRYPT'90, Lecture Notes in Computer Sciences, Vol. 473, pp. 161-173, 1991.

[147] I.S. Reed, A class of multiple-error-correcting codes and the decoding scheme. IRE Trans. Inform. Th., IT-4, 1954.

[148] R.L. Rivest, A. Shamir, and L.M. Adleman, A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21(2), pp. 120-126, 1978.

[149] S. Rønjom et T. Helleseth, A new attack on the filter generator. IEEE Information Theory, 53(5), pp. 1752-1758, 2007.

[150] S. Rønjom, G. Gong and T. Helleseth, On attacks on filtering generators using linear subspace structures. In Sequences, Subsequences, and Consequences, SSC 2007, Lecture Notes in Computer Science, Vol. 4893, Springer, pp. 204-217, 2007.

[151] O.S. Rothaus, On bent functions. Journal of Combinatorial Theory, Series A, 20(3), 300-305, 1976.

[152] R.A. Rueppel, Analysis and Design of Stream Ciphers. Springer-Verlag, 1986.

[153] R.A. Rueppel and O. Staffelbach, Products of linear recurring sequences with maximum complexity. Information Theory, IEEE Transactions on, 33(1), pp.124-131, 1987.

[154] N. V. Semakov and V. A. Zinoviev, Balanced codes and tactical configurations. Problems of Info. Trans. 5(3), pp. 22-28, 1969.

[155] N. Sendrier, Linear codes with complementary duals meet the GilbertVarshamov bound. Discrete Math., 304, pp. 345-347, 2004.

[156] C. E. Shannon, A mathematical theory of communication. Bell System Technical Journal, 27, pp. 379-423, 1948.

[157] C.E. Shannon, Communication theory of secrecy systems. Bell system technical journal, 28, pp. 656-715, 1949.

[158] T. Siegenthaler, Correlation-immunity of nonlinear combining functions for cryptographic applications. IEEE Transactions on Information Theory, 30(5) 776, 1984.

[159] T. Siegenthaler, Decrypting a class of stream ciphers using ciphertexts only. IEEE Transactions on Computers, Vol. C34 (1), pp. 81-85, 1985.

[160] D. Tang, C. Carlet, and Z. Zhou, Binary linear codes from vectorial Boolean functions and their weight distribution. Discrete Mathematics 340 (12), pp. 3055-3072, 2017.

[161] Y. Todo, Structural evaluation by generalized integral property. Proceedings of EUROCRYPT 2015, Part I. Lecture Notes in Computer Science, Vol. 9056, pp. 287-314, 2015.

[162] J. H. Van Lint, Introduction to Coding Theory. Springer, 1982.

[163] A. Vardy, The intractability of computing minimum distance of a code. IEEE Transaction on Information Theory 43(6), pp. 1757-1766, 1997.

[164] J. F. Voloch, Symmetric cryptography and algebraic curves. Proceedings of "The first Symposium on Algebraic Geometry and its Applications" SAGA'07, Tahiti, 2007, published by World Scientific, Series on Number Theory and its Applications, Vol. 5, pp. 135-141, 2008.

[165] T. Wadayama, T. Hada, K. Wagasugi, and M. Kasahara, Upper and lower bounds on the maximum nonlinearity of n-input m-output Boolean functions. Designs, Codes and Cryptography 23, pp. 23-33, 2001.

[166] C.P. Wang and X.S. Chen, On extended algebraic immunity, Des. Codes Cryptography 57(3), pp. 271-281, 2010.

[167] J. Wang, K. Chen and S. Zhu, Annihilators of Fast Discrete Fourier Spectra Attacks, Advances in Information and Computer security. 7-th international workshop on security, IWSEC 2012 Fokuoka, Japan, pp. 182-196, 2012.

[168] Q. Wang and T. Johansson, A note on fast algebraic attacks and higher order nonlinearities. Proceedings of INSCRYPT 2010, Lecture Notes in Computer Science, Vol. 6584, pp. 84-98, 2010.

[169] A. F. Webster and S. E. Tavares, On the design of S-boxes. In Proceedings of CRYPTO'85, Lecture Notes in Computer Science, Vol. 219, pp. 523-534, 1985.

[170] J. Wolfmann, Bent functions and coding theory. Difference Sets, Sequences and their Correlation Properties, A. Pott, P. V. Kumar, T. Helleseth, and D. Jungnickel, eds., Kluwer, pp. 393-417. 1999.

[171] G. Z. Xiao and J. L. Massey, A spectral characterization of correlation-immune combining functions. IEEE Transactions on Information Theory, 34(3), pp.569-571, 1988.

[172] Xian-Mo Zhang, J. Pieprzyk and Y. Zheng, On algebraic Immunity and Annihilators. Lecture note in computer Science, Information Security and Cryptology-ICISC, pp. 65-77, 2006.

[173] X. Yang, J.L. Massey, The necessary and sufficient condition for a cyclic code to have a complementary dual. Discrete Math., vol. 126, nos.1-3, pp. 391-393, 1994.

[174] J. Yuan, C. Carlet, and C. Ding, The weight distribution of a class of linear codes from perfect nonlinear functions. IEEE Transactions on Information Theory 52 (2), pp. 712-717, 2006.

[175] M. Zhang and A. Chan, Maximum correlation analysis of nonlinear Sboxes in stream ciphers. Advances in Cryptology-CRYPTO 2000, Lecture Notes in Computer Science vol. 1880, Springer, pp. 501-514, 2000.

[176] J. Zhang, S. Song, J. Du and Q. Wen, On the construction of multi-output Boolean functions with optimal algebraic immunity. Science in China Information Sciences 55(7), pp. 1617-1623, 2012.

[177] X. Zhong and M. Wang, Characterization of algebraic immunity for Multi-output Boolean functions.