N^o d'ordre: 01/2015 - D/MT

RÉPUBLIQUE ALGERIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE
FACULTÉ DE MATHÉMATIQUES

THÈSE

Présentée pour l'obtention du grade de

DOCTORAT EN SCIENCES

En : **Mathématiques**

SPÉCIALITÉ : **ALGÈBRE ET THÉORIE DES NOMBRES**
- Cryptographie -

Par : **M. MERABET Brahim**

Thème

Critères de sécurité des fonctions Booléennes vis à vis des attaques algébriques sur les schémas par flot et par blocs

Soutenue publiquement le 13 janvier 2015, devant le jury composé de :

M. BENZAGHOU Benali	Professeur	à l'USTHB	Président
M. CARLET Claude	Professeur	à l'Université de Paris 8	Directeur de thèse
M. BETINA Kamel	Professeur	à l'USTHB	Co-Directeur de thèse
Mme.BELKREDIM F. Zohra	Maître de conférences A	à l'UHBB de Chlef	Examinatrice
Mme. GUENDA Kenza	Maître de conférences A	à l'USTHB	Examinatrice
Mme. LAOUDI Aini	Maître de conférences A	à l'USTHB	Examinatrice
Mme. MESNAGER Sihem	Maître de conférences	à l'Université de Paris 13	Examinatrice

Remerciements

Je tiens à remercier et à témoigner ma profonde gratitude à mon directeur de thèse le Professeur Claude CARLET pour m'avoir proposé le sujet de recherche de doctorat, pour son suivi, pour sa disponibilité et pour ses réponses rapides à tous mes courriers électroniques durant toute la période de mes travaux de doctorat. Je remercie aussi mon co-directeur de thèse le Professeur Kamel BETINA pour son suivi et pour son soutient.

Mes remerciements aussi aux membres de l'équipe "Mathématiques pour le Traitement de l'Information et de l'Image, (ex. MAATICAH)" du LAGA Paris 8, d'avoir m'accueillir dans leur université durant mon séjour à Paris pour finaliser mes travaux de doctorat.

Je tiens également à remercier chaleureusement Mmes BELKREDIM Fatma Zohra de l'université de Chlef, GUENDA Kenza et LAOUDI Aini de l'USTHB ainsi que MESNAGER Sihem de l'université Paris 13, qui ont eu la patience de lire et rapporter ma thèse.

Enfin, j'adresse mes remerciements aussi aux doctorants de l'équipe MTII du LAGA Paris 8 que j'ai échangé avec eux des idées dans notre domaine de recherche : Codage et Cryptographie.

Table des matières

Liste des tableaux

Table des figures

Résumé

Nous nous intéressons dans cette thèse à l'étude de l'immunité algébrique d'une fonction vectorielle à utilisation en Cryptographie. Il existe trois notions pour ce critère pour les fonctions booléennes à plusieurs sorties, chacune de ces notions correspond à un endroit donné. Une haute immunité algébrique est nécessaire contre les attaques algébriques. Notre intervention dans cette thèse est l'étude du comportement asymptotique de l'immunité algébrique standard d'une fonction booléenne à une et à plusieurs sorties [65].

La Cryptologie est la science du secret, elle regroupe deux disciplines : la Cryptographie (qui assure la sécurité de la transmission des messages) et la Cryptanalyse (qui évalue le niveau de sécurité des algorithmes utilisés). Le chiffrement est l'un des services assuré par la cryptographie pour lequel on utilise souvent les fonctions Booléennes à une ou à plusieurs sorties pour accroitre le niveau de sécurité de l'algorithme utilisé. La fonction Booléenne utilisée doit satisfaire des critères nécessaires, elles doivent être par exemple de haut degré algébrique, de haute non-linérité. Un autre critère nécessaire est d'avoir une immunité algébrique élevée pour mieux résister aux attaques algébriques. On s'intéresse notamment dans notre thèse à l'étude de ce critère. Dans le cas d'une fonction Booléenne à plusieurs sorties, on étudie la stabilité des différentes notions d'immunité algébrique sous transformations affines, on verra les liens entre ces derniers et la caractérisation des cas d'égalité. Dans le dernier chapitre, on étend le résultat de F. Didier qui concerne une borne inférieure asymptotique sur l'immunité algébrique d'une fonction Booléenne au cas de l'immunité algébrique standard d'une fonction à plusieurs sorties.

Introduction générale

La question de la sécurité dans l'échange de messages est depuis longtemps un problème sensible, en particulier au plus haut sommet des Etats. La notion de cryptographie est née à partir du moment où l'on a voulu communiquer, à l'abri de personnes non autorisées à savoir le contenu des messages échangés. Déjà dans l'antiquité, les Grecs avaient inventé des méthodes pour chiffrer les messages. L'une d'entre elles, datant du VIème siècle avant J.C., consistait à enrouler une bande de papier autour d'un cylindre, puis à écrire le message sur la bande. Une fois déroulé, le papier était envoyé au destinataire qui, dès lors qu'il possédait le diamètre du cylindre, pouvait déchiffrer le message. La période contemporaine a connu des améliorations dans les méthodes de chiffrement et a vu apparaître de nouveaux besoins cryptographiques ce qui a donné naissance à "la cryptographie moderne". Les communications militaires secrètes, ainsi que les transactions bancaires sont en particulier des questions sensibles. Il faut assurer outre la confidentialité, l'authenticité et l'intégrité des communications sur des réseaux publiques, pour que les pirates des temps modernes ne puissent s'infiltrer dans ces réseaux.

La **cryptologie** est la science du secret. Elle regroupe deux disciplines :

- La **cryptographie** qui est l'étude des algorithmes permettant de protéger (sécuriser) l'information. Ces algorithmes sont appelés cryptosystèmes. La figure 1 décrit un service assuré par la cryptographie qui est le chiffrement.
- la **cryptanalyse** qui est l'étude du niveau de sécurité des cryptosystèmes fournis par les cryptographes.

La présente thèse est organisée de la manière suivante :

Nous commençons dans le chapitre 1 par rappeler quelques notions de base de la cryptographie notamment le chiffrement (à flot et par bloc), nous décrivons quelques attaques connues (statistiques ou algébriques) contre les systèmes de chiffrement. Le lecteur intéressé pourra consulter par exemple Menezes [47] ou Schneier [57].

Nous présentons ensuite dans le chapitre 2 un élément central dans certains systèmes

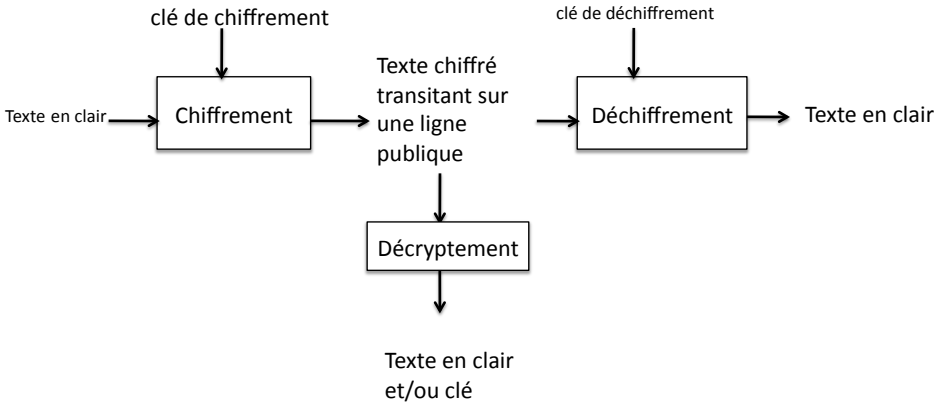

FIGURE 1 – Opérations de chiffrement et de déchiffrement

cryptographiques, que sont les fonctions Booléennes à une ou à plusieurs sorties. Plusieurs systèmes de chiffrement utilisent ces fonctions de façon centrale.

Un état de l'Art sur les attaques algébriques est traité dans le chapitre 3. Un critère nécessaire pour résister à ces attaques est d'avoir une immunité algébrique élevée ; cette notion est présentée dans le chapitre 4 dans le cas des fonctions Booléennes à une sortie, et dans le chapitre 5 dans le cas des fonctions à plusieurs sorties.
Il existe trois notions d'immunité algébrique d'une fonction Booléenne à plusieurs sorties ; nous prouvons des bornes concernant ces notions dans le chapitre 6.

Enfin, nous reprenons au chapitre 7 (le chapitre qui concerne essentiellement notre intervention dans le sujet de cette thèse [65]) les travaux de F. Didier [28] qui présentent une borne inférieure asymptotique sur l'immunité algébrique d'une fonction Booléenne aléatoire. Nous utilisons ses idées pour en déduire une borne inférieure sur l'immunité algébrique d'un ensemble, et comme conséquence une borne sur l'immunité algébrique des fonctions à plusieurs sorties.

Chapitre 1

Introduction à la cryptologie

Pendant plusieurs siècles, la cryptographie était réservée exclusivement au domaine militaire et diplomatique. La première publication fondamentale dans le domaine de la cryptographie moderne a été l'article de Claude Shannon [56] daté de 1949, dans lequel il introduisait les bases de la théorie de l'information. La cryptologie a évolué énormément ces dernières années avec les transactions et le commerce électroniques.

Si le but traditionnel de la cryptographie est de découvrir des méthodes permettant d'échanger des données de manière confidentielle, l'objectif de la cryptographie moderne est de chercher des solutions à tous les problèmes de sécurité des communications. Le but est donc d'offrir un certain nombre de services de sécurité comme la confidentialité, l'intégrité, l'authentification des données transmises et l'authentification d'un tiers. En général, un cryptosystème doit résoudre plusieurs problèmes de sécurité des communications que l'on peut rencontrer :

- *Confidentialité* : Protection de l'information contre les lectures non autorisées,
- *Intégrité* : Protection contre la modification non autorisée de l'information,
- *Authentification* :
    1. Authentification d'entités : (entity authentication) procédé permettant à une entité d'être sûre de l'identité d'une seconde entité (par exemple : présence physique, biométrique, etc.). Le terme "Identification" est parfois utilisé pour désigner également ce service.
    2. Authentification de l'origine des données : (data origin authentication) procédé permettant à une entité d'identifier la source originale d'un ensemble de données.
- *Non-répudiation* : Offre la garantie qu'une entité ne pourra pas nier être l'origine d'une transmission de données,

- *Non-Duplication* : Protection contre les copie illicites,
- *Anonymat* (d'entité ou d'origine de données) : Permet de garder secrète l'identité d'une entité ou de la source d'une information ou d'une transaction.
- *Preuve à apport nul de connaissance* ou "*zero knowledge*" : Vérification de la possession d'une information sans donner la moindre information sur cette information.
- *Certification* : Détention d'information par une autorité reconnue, un "tiers de confiance ".
- *Contrôle d'accès* : Moyen de limiter l'accès à certaines ressources à certaines entités privilégiées .

Chaque problème ou primitive possède une ou plusieurs solutions, plus ou moins satisfaisantes. Une solution peut en général se présenter sous la forme d'un protocole, à savoir une suite d'actions. Plus il y a d'entités ou de participants, plus le protocole est compliqué.
On peut regrouper les primitives en trois grandes classes :

- les primitives sans clé (fonctions de hachage, suites aléatoires),
- les primitives symétriques, ou à clé privée (chiffrement symétrique, fonction de hachage avec clé, suite pseudo-aléatoire, Identification),
- les primitives asymétriques, ou à clé publique (chiffrement, signature, Identification).

Nous nous intéresserons dans toute la suite à la **Confidentialité**.

La confidentialité est historiquement le premier problème posé à la cryptographie. On le résout par le chiffrement des données.
Un algorithme de chiffrement transforme un message, appelé texte clair, en un texte chiffré qui ne sera lisible que par son destinataire légitime. Cette transformation est effectuée par une fonction de chiffrement paramétrée par une clé de chiffrement. Un interlocuteur privilégié peut alors déchiffrer le message en utilisant la fonction de déchiffrement s'il détient la clé de déchiffrement correspondante.
Un tel système n'est sûr que s'il est impossible à un intrus de déduire le texte clair du message chiffré, et a fortiori de retrouver la clé de déchiffrement.
Cette formalisation a duré pendant plus d'un siècle. A cette époque, les cryptographes ont pris conscience qu'il n'était pas réaliste de faire reposer la sécurité d'un système de chiffrement sur la confidentialité de l'algorithme utilisé mais sur la confidentialité de la clé. L'intérêt d'un procédé à clé réside dans le fait que si la clé est compromise, il suffit de la changer, sans avoir à implanter un nouveau procédé, et qu'il est plus facile de protéger une clé de taille environ 128 bits (dans le cas du chiffrement symétrique) qu'un chiffrement complet.
Il existe deux grandes familles d'algorithmes avec clé : les algorithmes à clé secrète ou

FIGURE 1.1 – Chiffrement symétrique

algorithmes symétriques et les algorithmes à clé publique ou algorithmes asymétriques.

On désigne usuellement par $\mathcal{A}$ un alphabet, à savoir un ensemble de symboles qui sera dans la pratique $\{0, 1\}$ car tout message sera codé en binaire pour pouvoir être traité par ordinateur.
$\mathcal{M}$ désigne l'espace des messages clairs, il constitue un ensemble de mots dans un alphabet $\mathcal{A}$. Un élément m de $\mathcal{M}$ est appelé un texte clair.
$\mathcal{C}$ désigne l'espace des messages chiffrés. Un élément c de $\mathcal{C}$ est un texte chiffré.
On note $\mathcal{K}$ l'espace des clés. Pour chaque clé k de $\mathcal{K}$ on peut définir de manière unique une bijection de $\mathcal{M}$ vers $\mathcal{C}$, notée E_k. On note $D_{k'}$ une bijection de $\mathcal{C}$ vers $\mathcal{M}$ de clé k'. Un schéma de chiffrement est tel que $D_{k'}(E_k(m)) = m$ pour tout m de $\mathcal{M}$. On a parfois $k' = k$ et $D_{k'} \equiv E_k$ (cas du chiffrement symétrique).

1.1 Chiffrement symétrique

Les algorithmes à clé symétrique ou secrète sont des algorithmes dans lesquels la clé de chiffrement est secrète (alors le plus souvent la clé de chiffrement et la clé de déchiffrement sont identiques). Pour de tels algorithmes, l'émetteur **Alice** et le destinataire **Bob** doivent se mettre d'accord sur une clé à utiliser avant d'échanger des messages chiffrés. L'utilisation d'un algorithme à clé secrète lors d'une communication nécessite donc l'échange préalable d'un secret entre les tiers communicants à travers un canal sécurisé ou au moyen d'autres techniques cryptographiques.

Les algorithmes symétriques sont de deux types :
– Les algorithmes de *chiffrement en continu* ou *chiffrement à flot* appelés en anglais *stream ciphers* ;

– Les algorithmes de chiffrement par blocs appelés en anglais *block ciphers*.

1.1.1 Chiffrement à flot

Le principe du chiffrement à flot est de chiffrer une suite de caractères bit par bit à l'aide d'une transformation qui varie au fur et à mesure du texte. Les procédés de chiffrement à flot sont des techniques qui permettent d'assurer la confidentialité d'une communication, dans des contextes où il est nécessaire de pouvoir chiffrer et déchiffrer très rapidement au moyen de ressources (notamment d'une capacité de stockage) très limitées. La plupart des systèmes embarqués entrent dans ce cadre. Chaque bit transmis peut être chiffré ou déchiffré indépendamment des autres, en particulier sans qu'il soit nécessaire d'attendre les bits suivants. Un autre avantage de ces techniques est que, contrairement aux algorithmes par blocs, le processus de déchiffrement ne propage pas les erreurs de transmission.

Les systèmes de chiffrement par flot reposent sur le célèbre chiffrement à usage unique, appelé aussi technique du masque jetable, et également connu sous son appellation anglo-saxonne "one-time-pad". Cette technique, inventée par Vernam pour protéger les communications télégraphiques pendant la première guerre mondiale, consiste simplement à effectuer un **XOR** (qu'on appelle "ou exclusif") bit à bit entre le message clair et une suite de bits *aléatoire* (dans le cas du "one-time-pad" et *pseudo-aléatoire* dans le cas du chiffrement par flot) de même longueur qui constitue la clé secrète du système ; cette suite est engendrée par un générateur aléatoire (resp. pseudo-aléatoire).

Définition 1.1.1. *Un générateur aléatoire de bits est un procédé produisant une suite de bits à la fois statistiquement indépendants et non biaisés (*$\Pr(0) = \Pr(1) = \frac{1}{2}$*).*

Rappelons que l'opération XOR 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 :

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

Si $m = m_1 m_2 \cdots$ est le texte clair (*plaintext* en anglais), $k = k_1 k_2 \cdots$ est la clé et $c = c_1 c_2 \cdots$ est le texte chiffré (*ciphertext*), l'opération de chiffrement se fait en calculant pour tout i : $c_i = m_i + k_i$; le déchiffrement est défini par $m_i = c_i + k_i$.
On peut démontrer que le chiffrement à usage unique est incassable dans la mesure où la connaissance du message chiffré n'apporte aucune information sur le message clair si on ignore la clé. Mais cette propriété n'est garantie que si :

– La clé secrète (*suite chiffrante*) est une suite vraiment aléatoire aussi longue que le message clair, il est donc difficile dans ce cas-là de respecter la condition de

Kerckhoffs : *la clé doit pouvoir être communiquée et retenue sans le secours de notes écrites et être changée et modifiée au gré des correspondants* ;
- Elle n'est utilisée que pour transmettre un seul message, d'où le nom de *masque jetable*.

L'utilisation d'une clé unique pour chiffrer plusieurs messages peut être exploitée pour retrouver cette clé. En effet, si on chiffre deux messages m_1 et m_2 avec la même clé K et que les messages chiffrés correspondants c_1 et c_2 sont interceptés, l'attaquant peut alors calculer $c_1 + c_2 = (m_1 + K) + (m_2 + K) = m_1 + m_2$. Les propriétés statistiques des messages en clair permettent alors une cryptanalyse aisée des données interceptées. C'est ce qui est arrivé pendant la guerre froide où des opérateurs du KGB avaient utilisé la clé la deuxième fois en sens inverse et ça rendait plus difficile le déchiffrement (à partir de $m_1 + m_2^i$, où m_2^i est m_2 en sens inverse). Cette observation a permis à la National Security Agency (NSA) de déchiffrer des télégrammes soviétiques de première importance. L'opération secrète était connue sous le nom de projet VENONA et son existence n'a été révélée par la NSA qu'à la fin des années 90.

L'utilisation d'une clé secrète aléatoire à usage unique et de même longueur que le message à transmettre est malheureusement nécessaire pour obtenir un chiffrement inconditionnellement sûr, c'est-à-dire pour lequel on peut prouver qu'il est impossible de retrouver le message clair à partir du message chiffré sans connaître la clé secrète. Cette condition rend généralement tous les chiffrements parfaits, comme le chiffrement à usage unique, inutilisables puisqu'il n'est pas pratique de s'échanger préalablement un secret aussi long que la totalité du message à transmettre. L'usage de ces systèmes est donc réservé aux communications exigeant un niveau de sécurité extrêmement élevé, comme les communications diplomatiques, (pour transmettre les suites aléatoires secrète, le canal sécurisé utilisé n'est autre que la valise diplomatique). C'était par exemple le cas du célèbre "téléphone rouge" entre Washington et Mosco pendant la guerre froide.

On résout le problème consistant à échanger des clés longues par l'utilisation des suites chiffrantes pseudo-aléatoires produites par des générateurs pseudo-aléatoires. A partir d'une clé de petite taille (appelée *graine*) qui est l'entré d'un générateur pseudo-aléatoire, on aura en sortie une suite déterministe ayant des propriétés proche d'une suite aléatoire, qui peut-être ainsi utilisée pour chiffrer des messages aussi longs.

On distingue deux classes de chiffrement à flot :

1. *chiffrement à flot synchrone* : La suite chiffrante est produite par un générateur paramétré uniquement par la clé secrète, indépendamment du texte clair, et du texte chiffré (on s'intéresse que par ce type de systèmes) ;
2. *chiffrement à flot auto-synchronisant* : La suite chiffrante est produite par la clé secrète et un nombre donné de bits du clair (ou de manière équivalente du chiffré).

Dans un chiffrement à flot synchrone, la suite chiffrante est générée à partir de la clé secrète par un générateur pseudo-aléatoire. Le destinataire du message partageant cette clé, peut ainsi produire la même suite chiffrante et retrouver le message en clair en la combinant au message chiffré.

Définition 1.1.2. *Un générateur pseudo-aléatoire de bits est un automate à états finis qui génère une suite en produisant à chaque instant un ou plusieurs bits (qui semblent aléatoires), calculés à partir de son état interne.*

Il existe plusieurs générateurs pseudo-aléatoires, par exemple : générateur X.9.17, RSA-Rondom, Micali-Schnorr, BBS... . Les générateurs couramment utilisés et les plus connus sont les générateurs basés sur les registres à décalage à rétroaction linéaire (LFSR pour Linear Feedback Shift Register).

1.1.2 Chiffrements à flot à base de LFSRs

Le registre à décalage à rétroaction linéaire est un composant élémentaire bien adapté aux implémentations matérielles. En outre, les suites binaires produites par de tels composants possèdent (sous certaines conditions) une période élevée et de bonnes propriétés statistiques. La famille de chiffrements à flot conçus à partir de LFSRs est effectivement celle qui a fait l'objet des études les plus nombreuses.
Comme son nom l'indique, le registre à rétroaction linéaire produit une suite binaire dans laquelle chaque bit est la sortie d'une fonction linéaire de son vecteur initial. La suite produite aisément à partir de ce composant ne peut être utilisée telle quelle dans un but cryptographique, puisque dans ce cas, l'algorithme de *Berlekamp-Massey* (voir Section 1.3.1) permet facilement de retrouver le vecteur initial. Dans le but d'améliorer les propriétés cryptographique de telle suite, on introduit une non-linéarité pour rendre impossible une recherche aisée de la valeur du vecteur initial à partir de l'observation de la suite chiffrante et ce sans modifier les bonnes propriétés statistiques de la suite produite par un LFSR. Ceci se fait à l'aide d'une *fonction Booléenne* utilisée soit pour combiner les sorties de plusieurs registres, soit pour filtrer l'état interne d'un registre unique.

Registre à décalage à rétroaction linéaire

Un registre à décalage à rétroaction linéaire de longueur L est constitué de L bascules reliées par une fonction de rétroaction linéaire (polynôme de rétroaction). La figure 1.2

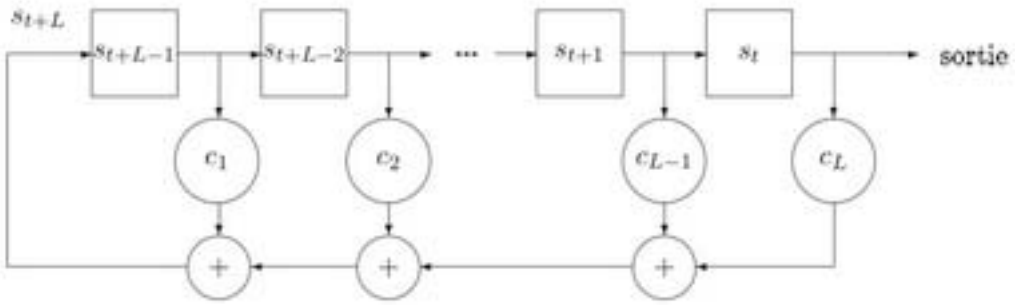

FIGURE 1.2 – Fonctionnement d'un LFSR

illustre le principe d'un tel composant.

A chaque cycle d'horloge, les L bits du registre sont décalés vers la sortie produisant ainsi le bit le plus ancien du registre. La bascule libérée reçoit alors un nouveau bit calculé grâce à la relation de rétroaction :

$$s_{t+L} = c_1 s_{t+L-1} \oplus c_2 s_{t+L-2} \oplus \cdots \oplus c_L s_t$$

La suite s produite par une telle récurrence linéaire est périodique, c'est-à-dire qu'il existe n_0 telle que la suite $(s_n)_{n \geq n_0}$ est périodique.

Remarque 1.1.1. *Il existe aussi des registres à décalage non linéaire. Par exemple la rétroaction définie par :* $s_j = 1 \oplus s_{j-2} \oplus s_{j-3} \oplus s_{j-1}s_{j-2}$*, qui produit la suite de Bruijn.*

On représente classiquement la suite par sa série génératrice :

$$s(X) = \bigoplus_{i \geq 0} s_i X^i$$

De même, on définit le polynôme de rétroaction du registre (voir [47]), $p_s \in \mathbb{F}_2[X]$:

$$p_s(X) = 1 + c_1 X + c_2 X^2 + \cdots + c_L X^L$$

La même suite peut être générée à partir de différents polynômes de rétroaction, c'est pourquoi on définit le *polynôme de rétroaction minimal* de la suite comme le polynôme de plus bas degré permettant d'engendrer cette suite. Le degré du polynôme de rétroaction minimal de la suite s correspond à la complexité linéaire de s, On le note $L(s)$. Lorsque le polynôme de rétroaction minimal est *primitif* et que l'état initial $(s_0, s_1, \cdots, s_{L-1})$ est non-nul, la période de la suite s est maximale et égale à $2^{L(s)} - 1$. Les suites périodiques engendrées par des polynômes de rétroaction primitifs sont appelées m-séquences ou suites *ML* (de longueur maximale).

Définition 1.1.3. *La complexité linéaire d'une suite infinie s de bits, notée $L(s)$, est :*
- *$L(s) = 0$ si $s_i = 0$ pour tout i ;*
- *$L(s) = \infty$ si aucun registre à décalage linéaire ne produit s ;*
- *$L(s) = n$ si le plus petit registre à décalage linéaire produisant s a pour longueur n.*

Il existe une méthode efficace pour retrouver le polynôme de rétroaction minimal permettant de générer la suite observée. En effet, étant donné une suite binaire s, l'algorithme de Berlekamp-Massey [4, 43] détermine le LFSR équivalent, c'est-à-dire le polynôme de rétroaction minimal d'une suite de complexité linéaire $L(s)$, grâce à l'observation de $2L(s)$ éléments consécutifs de la suite et ce, sans connaissance préalable de $L(s)$. Ainsi, pour produire une suite chiffrante de complexité linéaire élevée (c'est-à-dire supérieure à $\frac{k}{2}$, où k est la taille de la clé (la valeur initiale du registre), pour se mettre à l'abri d'une attaque par l'algorithme de Berlekamp-Massey) et de grande période en utilisant des LFSRs de taille raisonnable, pour leur simplicité d'implémentation, il est impératif d'introduire une composante non-linéaire au système. Cette composante peut être obtenue par plusieurs principes de conception ; par exemple :

Générateur de combinaison : n LFSRs combinés par une fonction Booléenne à n variables.

Générateur de filtrage : Un seul LFSR dont on choisit n bascules avec des espacements soigneusement déterminés, que l'on filtre par une fonction Booléenne à n variables.

Générateur de combinaison contrôlé : n LFSRs combinés, dont l'un contrôle l'horloge, par exemple : A5/1 utilisé dans GSM.

On peut aussi combiner n LFSRs par une fonction vectorielle ayant n bits d'entrées (les sorties des LFSRs) et m bits de sorties, ceci étant dans le but d'accélérer l'opération de chiffrement et de déchiffrement.
Pour que la suite obtenue en sortie de la fonction Booléenne f ait une complexité linéaire significativement plus grande que celle d'une des composantes d'entrée de f, il est nécessaire que le degré algébrique de f soit élevé [54]. Les générateurs de combinaison et de filtrage sont représentés par les figures 1.3 et 1.4.
Si on note z_t le bit de la suite chiffrante à l'instant t, on a :

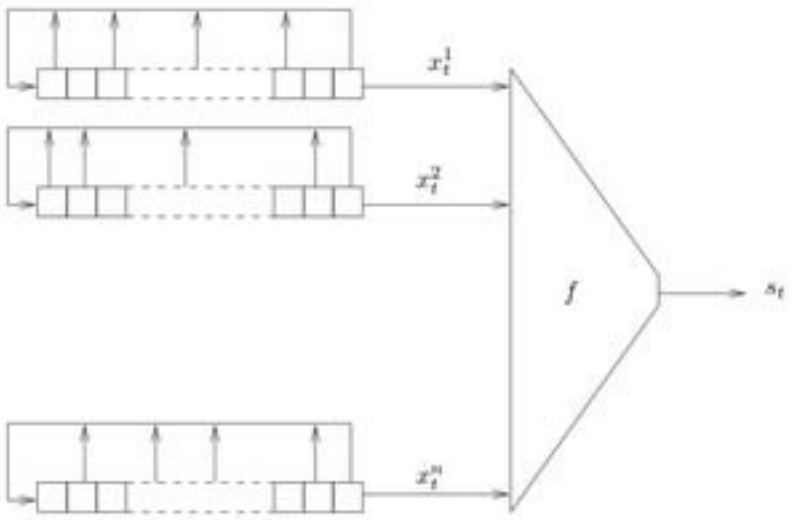

FIGURE 1.3 – Générateur de combinaison

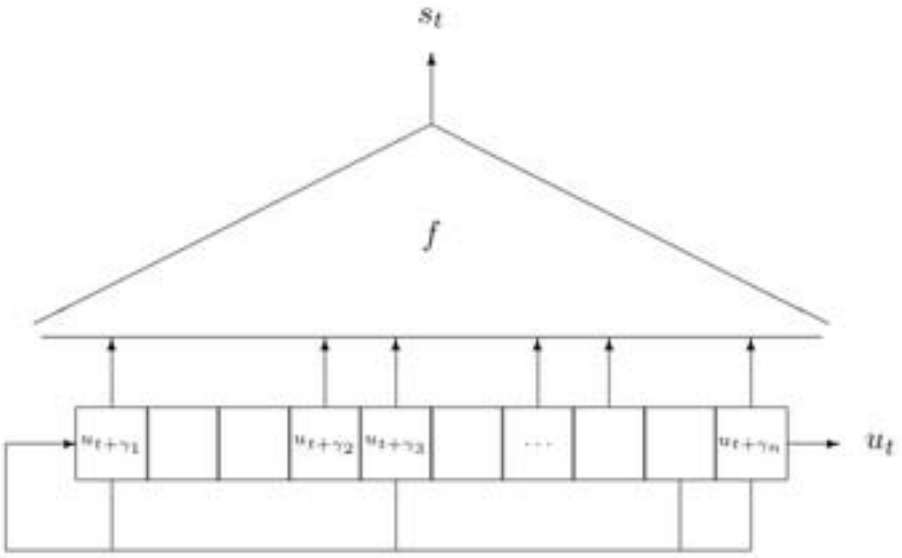

FIGURE 1.4 – Générateur de filtrage

$$z_t = f(x_1^t, x_2^t, \cdots, x_n^t)$$

où $x_1^t, x_2^t, \cdots, x_n^t$ correspondent soit aux sorties de n LFSRs à l'instant t pour le schéma par combinaison, soit aux états de n bascules d'un LFSR à l'instant t. Dans les deux cas, un ensemble de propriétés de la fonction Booléenne permet d'établir des critères nécessaires pour la sécurité des systèmes conçus suivant ces principes, ces critères nécessaires sont décrit au Chapitre 2.

1.1.3 Chiffrement par bloc

Un système de chiffrement est dit par blocs s'il divise le texte clair en blocs de n bits et traite chaque bloc indépendamment (ou presque). Un tel système chiffre ainsi un bloc de n bits à la fois. La taille des blocs est généralement de 64 ou de 128 bits. Utiliser un chiffrement par blocs revient à effectuer une permutation sur l'ensemble des mots de n bits qui va transformer un message de n bits en un mot chiffré de n bits. Cette permutation est paramètrée par une clé secrète (appelée *sous-clé* de longueur k qui change pour chaque tour). Un chiffrement par blocs idéal associe à chaque clé une permutation aléatoire.

Définition 1.1.4. *Un procédé de chiffrement par bloc sur n bits consiste à itérer r fois une fonction E de $\{0,1\}^n \times \mathcal{K}$ dans $\{0,1\}^n$ telle que pour toute clé de tour $k_i \in \mathcal{K}$, $E(., k_i)$ est une bijection de $\{0,1\}^n$ dans lui-même, notée E_k. La fonction inverse est notée D_k. Les r clés $(k_1, \cdots, k_r)$ sont en général dérivées d'une unique clé-maîtresse par un algorithme d'expansion (ou de cadencement) de la clé.*

Pour un texte clair m de n bits, on note $c = E_k(m)$ le message chiffré correspondant, et donc $m = D_k(c)$.

Les exemples classiques de chiffrement par bloc sont DES (Data Encryption Standard), FEAL (Fast Data Encipherment Algorithm, $k = n = 64$), IDEA (International Data Encryption Algorithm, $k = 128$, $n = 64$), RC5 (RC5-32 pour des mots de 32 bits, clé de 16 octets soit 128 bits), AES (Advanced Encryption Standard), les algorithmes les plus connus sont le DES et l'AES.

Algorithme de chiffrement DES

Jusqu'au début de l'an 2000, le système de chiffrement à clé secrète le plus célèbre et le plus utilisé était le DES. Il a été adopté comme standard américain en 1977 (standard FIPS 46 [1]) pour les communications commerciales, puis par l'ANSI en 1991.

Définition 1.1.5. *(chiffrement de Feistel)*
Un chiffrement de Feistel est un chiffrement itératif par blocs opérant sur des blocs de $2n$ *bits. La fonction itérée* E_{k_i} *est définie par :*

$$\begin{aligned} E_{k_i} : \quad \mathbb{F}_2^n \times \mathbb{F}_2^n \quad &\to \mathbb{F}_2^n \times \mathbb{F}_2^n \\ (L_{i-1}, R_{i-1}) \quad &\mapsto (L_i, R_i) \end{aligned}$$

où : $L_i = R_{i-1}$ *et* $R_i = L_{i-1} + f(R_{i-1}, k_i)$.
Quelle que soit la fonction f *utilisée, un chiffrement de Feistel est inversible. Pour déchiffrer, il suffit d'utiliser le même processus à* r *tours en inversant l'ordre des clés* $k_1, \cdots, k_r$.

- Le DES a un schéma de *Feistel*, il opère par blocs de 64 bits avec une clé de 56 bits (complétés de 8 bits de redondance).
- Le clair est permuté par une permutation initiale IP puis 16 itérations sont effectuées. Les sous-clés sont de 48 bits. - Après les 16 tours, on applique IP^{-1}. - La fonction de chiffrement f qui opère à chaque tour est le composé (voir figure 1.5) : d'une fonction d'expansion E qui est une application linéaire de $\mathbb{F}_2^{32}$ dans $\mathbb{F}_2^{48}$ ne faisant pas intervenir la clé, de l'ajout bit à bit de la clé, d'une fonction de substitution (boite S) qui est une application non linéaire de $\mathbb{F}_2^{48}$ dans $\mathbb{F}_2^{32}$ et d'une permutation P des bits (qui est une fonction linéaire de $\mathbb{F}_2^{32}$ dans $\mathbb{F}_2^{32}$) ne faisant pas intervenir la clé. La boite S, qui est la plus complexe à spécifier et qui assure la fonction de *confusion*, essentielle à la sécurité du DES, est en fait la parallélisée de 8 fonctions de substitution de $\mathbb{F}_2^6$ dans $\mathbb{F}_2^4$ données par leurs tables de valeurs (look up tables). Les applications E et P assurent la *diffusion*.

Remarque 1.1.2. *La confusion a pour but de rendre inextricables les liens entre le message en clair, la clé et le message chiffré. La diffusion permet de réduire les possibilités d'utilisation des données statistiques présentes dans le texte en clair en diluant ses données fréquentielles tout au long du texte chiffré.*

La clé de DES étant de 56 bits, il est donc désormais vulnérable aux attaques exhaustives. Une telle attaque, demandant en moyenne 2^{55} chiffrements DES, a été

1. http ://csrc.nist.gov/cryptval/des.htm

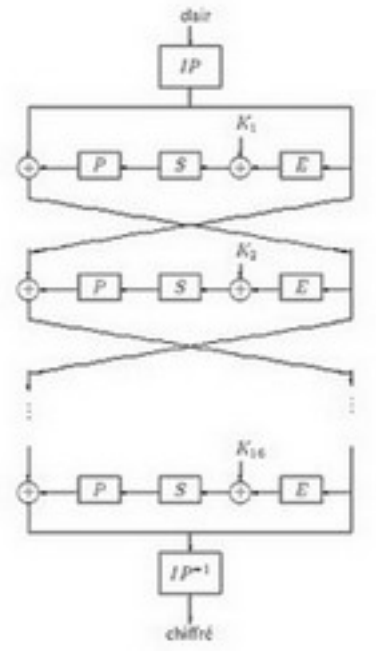

FIGURE 1.5 – Fonctionnement de DES

réalisée en janvier 1998 en 39 jours sur 10 000 Pentium en parallèle, puis en 56 heures en juillet 1998 à l'aide d'une machine dédiée (EFF DES Cracker) comportant 1500 composants DES [2]. Le coût d'une telle machine était alors estimé à 210 000 dollars. C'est pourquoi la plupart des applications utilisent maintenant cet algorithme sous la forme d'un triple DES à deux clés, constitué de trois chiffrements DES successifs avec deux clés secrètes. Cette technique permet de doubler la taille de la clé secrète (112 bits). Le triple DES à deux clés a notamment été adopté dans les standards ANSI X9.17 et ISO 8732. Il est très utilisé pour les applications bancaires.

Algorithme de chiffrement AES

L'AES est le nouveau standard de chiffrement à clé secrète. Il a été choisi en octobre 2000 parmi les 15 systèmes proposés en réponse à l'appel d'offre lancé par le NIST (National Institute of Standards and Technology). Cet algorithme, initialement appelé RIJNDAEL, a été conçu par deux cryptographes belges, V. Rijmen et J. Daemen. Il est disponible pour trois tailles de blocs différentes : 128, 192 et 256 bits avec les mêmes tailles de clés secrètes correspondantes. Les spécifications de l'AES ainsi que diverses implémentations sont disponibles sur la page Web du NIST ;

2. http ://www.eff.org/Privacy/Crypto/Crypto-misc/DESCracker/

http ://csrc.nist.gov/encryption/aes/rijndael.

Définition 1.1.6. *(réseau de substitution-permutation)*
Un réseau de substitution-permutation itératif opère sur des blocs de n bits. La fonction itérée E_{k_i} est composée de permutations linéaires, de substitution (composante non-linéaire) et d'une fonction d'insertion de sous-clé. Pour que le chiffrement soit inversible E_{k_i} doit être une bijection.

L'AES est un chiffrement ayant une structure de réseau de substitution-permutation (en anglais *substitution permutation network*), pour lequel à chaque tour, le chiffré produit par le tour précédent subit une fonction non-linéaire qui assure la confusion (on verra dans le chapitre 2 des critères nécessaires pour la conception de telles fonctions), puis une permutation linéaire qui assure la diffusion, puis la clé du tour est ajoutée bit à bit pour donner le prochain bloc chiffré. Les différentes sous-clés de l'AES sont dérivées de la clé secrète par un algorithme de cadencement de clé. Le nombre de tours est 10 pour une clé de 128 bits et de 14 pour une clé de 256 bits. La première itération est précédée d'un ou exclusif bit-à-bit entre le message clair et la sous-clé numéro 0 ; de même, la dernière itération est légèrement différente des itérations précédentes.

Fonction de substitution (boite S). Chaque octet est considéré comme un élément du corps $\mathbb{F}_{256}$. La boite S est constituée de 16 boites identiques consécutives agissant chacune sur un octet.
Chaque boite S_i consiste en l'application $F : x \in \mathbb{F}_{256} \mapsto x^{254} \in \mathbb{F}_{256}$. Notons que si $x = 0$ alors $x^{254} = 0$ et si $x \neq 0$ alors $x^{254} = \frac{1}{x}$.

Le résultat de cette transformation (la sortie de la boite S) subit ensuite une transformation affine et est ajouté avec la clé du tour.

Cette boite S permet au système de résister à *l'attaque différentielle* et à *l'attaque linéaire*. La raison de ce choix est que la fonction F choisie est telle que l'équation $F(x) + F(x + a) = b$ admet au plus 4 solutions (cela évite l'existence de différentielles de probabilités élevées) et que l'équation $a \cdot X + b \cdot F(X) = 0$ admet un nombre de solutions proche de 2^{255} pour tout $a \neq 0$ et tout $b \neq 0$ (ce qui rend coûteuse *l'attaque linéaire*).

Mentionnons quand même qu'une fois un chiffrement par blocs construit, il ne suffit pas de l'appliquer directement à chaque bloc pour obtenir un chiffrement sûr. Il y a en effet un grand nombre d'applications où l'on peut être amené à chiffrer des blocs identiques, et l'on ne veut certainement pas que leurs chiffrés le soient également.

Un exemple est le chiffrement d'une image avec des zones complètement uniformes qu'un tel mode de chiffrement ne cacherait pas. Il existe donc des modes opératoires qui, étant donné une fonction de chiffrement par blocs, spécifient comment l'appliquer sur un message réel. On distingue 4 modes opératoires : ECB (electronic codebook), CBC (cipher-block chaining), CFB (cipher feedback), et OFB (output feedback). Par exemple, le mode ECB fonctionne comme suit :

- Chiffrement : $c_j = E_k(m_j)$;
- Déchiffrement : $m_j = E_k^{-1}(c_j)$.

Deux blocs identiques (avec la même clé k) produisent le même chiffré. Les blocs sont chiffrés indépendamment les uns des autres ; une permutation des blocs chiffrés produit donc la même permutation des blocs clairs. Une erreur de transmission sur un bloc chiffré n'affecte que ce bloc. Quant à lui, le mode CBC fait dépendre c_j du chiffré précédent :

- Chiffrement : $c_j = E_k(c_{j-1} \oplus m_j)$;
- Déchiffrement : $m_j = c_{j-1} \oplus E_k^{-1}(cj)$.

Ici, deux blocs identiques (même chiffrés avec la même clé k) produisent des chiffrés différents. Une permutation des blocs chiffrés empêche un déchiffrement correct. Une erreur sur c_j affecte le déchiffrement de tous les blocs $c_i, i \geq j$.

1.2 Chiffrement asymétrique

La cryptographie à clé publique a été introduite par Diffie et Hellman en 1976, elle a créé une révolution pour l'époque. Il a fallu cependant attendre deux ans, soit 1978, pour que Rivest, Shamir et Adleman mettent au point la première réalisation pratique de chiffrement à clé publique, avec le système maintenant bien connu sous le nom de RSA [52]. Ces algorithmes permettent d'échanger de l'information chiffrée sans s'être mis d'accord au préalable sur un secret.
Deux clés "différentes sont nécessaires, une clé publique p (clé de chiffrement) et une clé secrète s (clé de déchiffrement) de telle sorte que la connaissance de p ne permette pas de retrouver facilement s. Plus précisément, il ne doit pas exister d'algorithme efficace permettant de calculer s à partir de p.

Un algorithme asymétrique est généralement basé sur un problème complexe, c'est-à-dire difficile à résoudre, de sorte qu'il n'existe pas d'algorithme efficace permettant de trouver la solution. L'efficacité d'un algorithme est susceptible d'évoluer en fonction des progrès des ordinateurs (gain en puissance et en rapidité), mais aussi en fonction des découvertes et des améliorations d'algorithmes permettant de résoudre les problèmes difficiles sous-jacents plus efficacement.

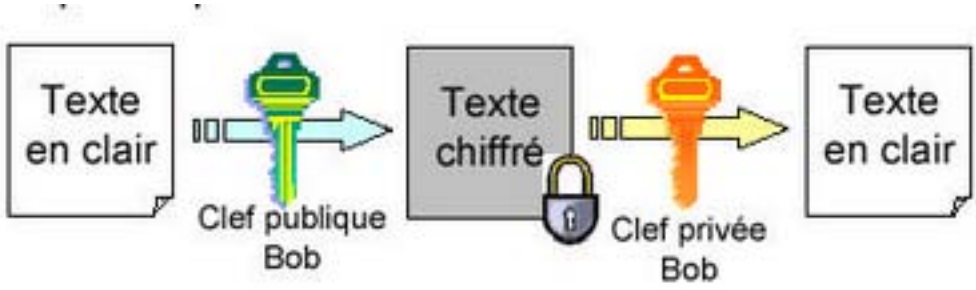

FIGURE 1.6 – Chiffrement asymétrique

Dans le but d'envoyer un message confidentiel à Bob, Alice utilise la clé publique de Bob pour chiffrer ce message ; Bob étant le seul possesseur de la clé secrète (seule clé autorisant le déchiffrement), il est le seul à même de déchiffrer le message envoyé par Alice. C'est grâce à ces techniques à clé publique que la plupart des applications de la cryptographie autres que le simple chiffrement sont devenues possibles.

Si n personnes veulent communiquer 2 à 2, il leur faut en tout $2n$ clés (chacune détient une clé secrète et diffuse une clé publique), alors que si elles utilisent un chiffrement conventionnel, il leur faut une clé secrète pour chaque paire de correspondants, c'est à dire en tout $\binom{n}{2} = \frac{n(n-1)}{2}$ clés.

Mais, si ces algorithmes à clé publique offrent de nombreux avantages, ils sont en général beaucoup plus lents que les algorithmes basés sur l'utilisation d'une clé secrète qui a été échangée au préalable entre les participants. Ces contraintes de performance imposent l'usage d'algorithmes de chiffrement à clé secrète, puisque les algorithmes à clé publique connus actuellement n'offrent pas un débit suffisant pour permettre le chiffrement ou le déchiffrement en ligne. En pratique dans les applications modernes, on utilise des algorithmes à clé publique pour initialiser une session chiffrée (échange de clé de session symétrique) plutôt que pour chiffrer toute une session.

1.2.1 Technique de Diffie-Hellman pour l'échange de clés

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 $\alpha = g^a$; Bob choisit un entier $b \in [1, n-1]$ et envoie à Alice $\beta = g^b$. Alice calcule β^a, et Bob calcule α^b. Alice et Bob partagent un secret commun $\beta^a = \alpha^b = g^{ab}$.
Pour trouver ce secret commun à partir de g^a et g^b, l'adversaire Charlie doit résoudre le problème du logarithme discret, i.e. retrouver a et b.

1.2.2 Le système RSA.

RSA est utilisé pour le chiffrement mais aussi pour la signature. Il est le premier standard de signature à clé publique, adopté en 1991 (ISO/IEC 9796) :

- *Génération de clé :* Alice choisit deux nombres premiers p et q, ainsi qu'un entier e aléatoire dans [0, (p-1)(q-1)], premier avec $(p-1)(q-1)$. Elle calcule $N = pq$, et $d = e^{-1} \bmod (p-1)(q-1)$, par pgcd étendu. La clé publique est (N, e), la clé secrète est (p, q, d).
- *Chiffrement :* Pour envoyer un message à Bob, Alice le coupe en blocs qu'elle peut encoder via des entiers inférieurs à N. Pour chaque bloc m, elle calcule $c = m^e \bmod N$, qu'elle envoie à Bob.
- *Déchiffrement :* Pour déchiffrer c, Bob calcule $c^d \bmod N$. On a en effet, $c = m^e \bmod N$, donc $c^d = m^{ed} \bmod N$. On a aussi $c^d = m^{ed} \bmod p$ et $c^d = m^{ed} \bmod q$. Considérons $c^d \bmod p$: comme $ed = 1 \bmod (p-1)(q-1)$, on a $ed = 1 + \lambda(p-1)$ pour un certain entier λ. Donc $m^{ed} = m^{1+\lambda(p-1)}$. Or pour tout $m \neq 0$, $m^{p-1} = 1 \bmod p$ (petit théorème de Fermat), donc $m^{ed} = m \bmod p$, qui est vrai aussi pour $m = 0$. Le théorème des restes de chinois permet de conclure que $m^{ed} = m \bmod N$.

En 1985, El Gamal invente un autre procédé à clé publique, basé sur le problème du logarithme discret, qui avait déjà été utilisé par Diffie et Hellman. Le système d'El Gamal a été adopté pour un standard de chiffrement en 1994, son désavantage par rapport à RSA est que le message chiffré est deux fois plus gros que le clair.

1.3 Cryptanalyse des systèmes de chiffrement

Si le but de la cryptographie est d'élaborer des méthodes de protection de l'information, le but de la cryptanalyse est au contraire de casser ces protections. Une tentative de cryptanalyse d'un système est appelée une attaque, et elle peut conduire à différents résultats :

- Cassage complet : l'attaquant retrouve la clé de déchiffrement.
- Obtention globale : l'attaquant trouve un algorithme équivalent à l'algorithme de déchiffrement, mais qui ne nécessite pas la connaissance de la clé de déchiffrement.
- Obtention locale : l'attaquant retrouve le texte en clair correspondant à un message chiffré.
- Obtention d'information : l'attaquant obtient quelques indications sur le texte en clair ou la clé (certains bits de la clé, un renseignement sur la forme du texte en clair,...)

D'une manière générale, on suppose toujours que le cryptanalyste connaît le détail des algorithmes, fonctions mathématiques ou protocoles employés. Même si ce n'est pas toujours le cas en pratique, il serait risqué de se baser sur le secret des mécanismes utilisés pour assurer la sécurité d'un système, puisque si ces derniers sont interceptés, il faut tous les changer, ce qui est coûteux.
On distingue plusieurs types d'attaques suivant les informations que peut obtenir le cryptanalyste.
On a les types d'attaques possibles suivantes, par ordre décroissant de difficulté pour l'attaquant :

Attaque à texte chiffré seulement : Le cryptanalyste ne connaît qu'un ensemble de textes chiffrés ; il essaye de retrouver les textes en clair et si possible de retrouver la clé.

Attaque à texte en clair connu : Le cryptanalyste connaît non seulement les textes chiffrés, mais aussi les textes en clair correspondants ; son but est alors de retrouver la clé. Du fait de la présence, dans la plupart des textes chiffrés, de parties connues (en-têtes de paquets, champs communs à tous les fichiers d'un type donné,...), ce type d'attaques est très courant.

Attaque à texte en clair choisi : Le cryptanalyste peut, de plus, choisir des textes en clair à chiffrer et donc utiliser des textes lui permettant d'obtenir plus d'informations sur la clé. Si le cryptanalyste peut de plus adapter ses choix en fonction des textes chiffrés précédents, on parle d'attaque adaptative.

Attaque à texte chiffré choisi : qui est l'inverse de la précédente : le cryptanalyste peut choisir des textes chiffrés pour lesquels il connaîtra le texte en clair correspondant ; sa tâche est alors de retrouver la clé. Ce type d'attaques est principalement utilisé contre les systèmes à clé publique, pour retrouver la clé privée.

On considère qu'une attaque est efficace si elle a une probabilité non négligeable de réussir en un temps inférieur ou égal à quelques années (voire plus !) sur une ou plusieurs machines puissantes. Cela fixe de nos jours à 2^{80} le nombre minimal d'opérations élémentaires nécessaires à une attaque. Le niveau normal de sécurité est la résistance à l'attaque à clair choisi. Tout nouveau chiffrement qui veut être considéré comme une proposition sérieuse, s'accompagne de garanties sur le fait que les attaques connues qui lui sont appliquées, présentent une complexité proche de celle de la recherche exhaustive.

l'attaque exhaustive . Il est toujours possible de mener sur un algorithme de chiffrement une attaque dite *exhaustive* pour retrouver la clé. Cette attaque consiste simplement à énumérer toutes les clés possibles du système et à essayer chacune d'entre elles pour décrypter un message chiffré. Une telle attaque devient donc hors de portée si l'espace des clés est suffisamment grand. Le temps moyen de cette attaque est égal au temps d'un déchiffrement multiplié par la moitié de la

taille de l'espace des clés.

La parade bien sûr consiste à choisir un système cryptographique dont l'espace des clés a une taille suffisante. Le DES a été écarté précisément parce que la taille de la clé utilisée de 56 bits est insuffisante.

Contrairement aux chiffrements asymétriques dont on mesure la sécurité par réduction à des problèmes mathématiques connus, la mesure de la sécurité des chiffrements symétriques couramment utilisés, repose soit sur des arguments de théorie de l'information (ce qui conduit à des preuves de sécurité dans différents modèles), soit sur l'étude de la complexité des meilleures attaques connues à ce jour. Dans ce dernier cas, on en distingue naturellement plusieurs types :

- *la complexité en espace* : c'est le nombre de données nécessaires, par exemple de paires clair-chiffré. Pour des messages sur n bits, on peut toujours monter une attaque en espace 2^n : en effet, il suffit de tabler les 2^n paires possibles (m_i, c_i), puis pour c donné de rechercher quel m correspond.
- *la complexité en mémoire* mesure la quantité de mémoire nécessaire au cours de l'attaque ;
- *la complexité en temps* : c'est le temps de calcul nécessaire. Pour une clé de k bits, une attaque en temps 2^k est toujours possible. En effet, pour une paire (m, c) donnée, il suffit d'essayer les 2^k clés possibles, jusqu'à trouver $E_k(m) = c$.

On considère qu'un algorithme de chiffrement est sûr lorsqu'on ne connaît pas d'attaque de complexité significativement inférieure à la recherche exhaustive sur la clé ou à la recherche exhaustive sur les textes.

1.3.1 Attaques sur les chiffrements par flot

Nous nous intéressons ici aux attaques permettant de définir des critères de sécurité pour la fonction de filtrage. Nous remarquons qu'on peut montrer que le générateur filtré est équivalent au générateur de combinaisons mais les attaques contre ces deux générateurs n'ont pas la même complexité. La sécurité d'un tel système contre une attaque à clair connu est équivalente à la sécurité du générateur pseudo-aléatoire. Toutes les analyses présentées ici sur le registre filtré considèrent la suite chiffrante partiellement connue : il s'agit donc d'attaques à clair connu.

Les attaques sur les chiffrements à flot tirent partie :

- soit de la structure algébrique du système, c'est le cas de l'algorithme de Berlekamp-Massey et des attaques dites algébriques (c'est l'objet du chapitre 3), - soit de données statistiques, c'est le cas des attaques par distingueur et par corrélation.

Algorithme de Berlekamp-Massey

En 1968, E. Berlekamp a proposé dans son livre [4] un algorithme pour le décodage des codes BCH. En 1969, J.L. Massey [43] a montré que ce dernier permet également de trouver le plus petit LFSR générant la suite $(s_n)_n$ à partir uniquement d'une suite de $2L(s)$ bits consécutifs de $(s_n)_n$.

Proposition 1.3.1. *Soit $(s_n)_{n\geq 0}$ une suite binaire à récurrence linéaire de complexité linéaire $L(s)$. L'algorithme de Berlekamp-Massey détermine l'unique LFSR de longueur $L(s)$ qui génère $(s_n)_{n\geq 0}$ à partir de n'importe quelle suite de $2L(s)$ bits consécutifs de $(s_n)_{n\geq 0}$.*

La complexité linéaire d'un LFSR est donc un paramètre déterminant pour sa sécurité cryptographique : Si $L(s)$ est petit, l'observation d'un petit nombre de bits seulement de la suite permet en effet de la reconstituer entièrement . C'est ce qui a amené à l'étude de la combinaison ou du filtrage d'un ou plusieurs LFSRs, à l'aide de fonctions Booléennes. Bien évidemment, ces fonctions doivent être choisies avec beaucoup de soin.

Exemple 1.3.1. *: Si on applique l'algorithme de Berlekamp-Massey à la suite 0110010101 de longueur 10, on trouve qu'un LFSR de taille minimale générant cette suite a pour longueur 5 et polynôme de rétroaction $1 + X^4 + X^5$.*

Attaque par corrélation

Les attaques par corrélation forment une famille très importante d'attaques sur les chiffrements à flot. Elles ont été découvertes par Siegenthaler [60] et s'appliquent dans leur forme initiale uniquement dans le cas de plusieurs registres combinés (ou filtrés) par une fonction Booléenne. Cette classe d'attaques entre dans la catégorie plus générale des attaques du type *diviser pour régner* qui s'appliquent à chaque fois qu'on peut décomposer le système en composantes élémentaires cryptographiquement faibles.
Si la fonction de combinaison (ou de filtrage) des registres est mal choisie, il peut exister une corrélation entre la sortie d'un des registres internes (ou une combinaison de ceux-ci) et la suite chiffrante produite. On peut alors faire une recherche exhaustive sur ces registres uniquement et retrouver l'état qui conduit à une corrélation maximale. La propriété de la fonction Booléenne qui intervient ici est la *résilience* qui mesure l'équilibre de la fonction lorsque certaines de ses variables d'entrées sont fixées (voir le chapitre 2).

Exemple 1.3.2. *: le générateur de Geffe*
On prend trois LFSRs de longueurs maximales deux à deux premiers entre eux L_1, L_2, L_3, et comme fonction Booléenne de combinaison : $f(x_1, x_2, x_3) = x_1x_2 + x_2x_3 + x_3$. Alors la période de la suite engendrée est égale à $(2^{L_1} - 1)(2^{L_2} - 1)(2^{L_3} - 1)$ et sa complexité linéaire vaut $L = L_1L_2 + L_2L_3 + L_3$. On voit facilement que, si $z(t)$ est la sortie de ce générateur à l'instant t, c'est-à-dire $z(t) = f(x_1(t), x_2(t), x_3(t))$, alors $z(t) = x_1(t)$ avec probabilité 3/4 (et, de même, $z(t) = x_3(t)$ avec probabilité 3/4). En effet, si $x_2 = 1$, alors $z = x_1$, et si $x_2 = 0$ alors $z = x_3$, mais $x_3 = x_1$ avec probabilité 1/2 !. On peut alors monter l'attaque suivante : parcourir les valeurs possibles de la clé du LFSR1 jusqu'à ce que sa sortie et la sortie de z coïncident dans 3/4 des cas.

Attaque par distingueur

Un distingueur est une machine de Turing probabiliste avec oracle qui fournit une réponse binaire après un nombre de requêtes. Dans le cas d'un chiffrement par bloc, 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, c'est à dire de déterminer le biais statistique lié au chiffrement d'un nombre réduit de tours, indépendant de la clé et non uniformément distribuée.

Attaque algébrique

Le principe de base des attaques algébriques remonte aux travaux de Shannon qui avait eu l'idée de traduire l'opération de chiffrement par un système d'équations à plusieurs variables de grande taille, qui peut-être résolu pour retrouver la clé secrète. La difficulté de résolution d'un tel système réside dans le degré algébrique élevé de ces équations.
Etant donné un générateur de filtrage de fonction de transition linéaire L et de fonction de filtrage f, si $x^0 \in \mathbb{F}_2^n$ est l'état initial du registre alors, le bit de sortie s_t à l'instant t est exprimé par

$$s_t = f(L^t(x^0)) = f(\underbrace{L \circ L \circ \ldots \circ L}_{t fois}(x^0)).$$

L'attaque algébrique est réalisable s'il existe des relations de bas degré entre les bits d'entrée et les bits de sortie du registre, ces relations correspondent aux multiples de f de bas degré, c'est à dire aux relations de la forme $g(x)f(x) = h(x)$, où f et h sont des fonctions Booléennes à n variables, h est de bas degré. On verra au Chapitre 3 que la recherche des fonctions g, h vérifiant $gf = h$ est équivalent à la recherche d'une

fonction h vérifiant $h(1+f)=0$. Si on dispose d'un nombre suffisant de ces équations (les fonctions h sont de degré au plus d), on aura un système d'équations de degré d à n inconnues. On utilisera ensuite des techniques pour résoudre ce système d'équation et obtenir la clé secrète.
Remarquons que les attaques algébriques sont aussi faisable sur les algorithmes de chiffrement par bloc avec un degré de difficulté beaucoup plus élevé, car l'idée de trouver deux fonctions g et h ne s'applique pas, nous expliquerons comment au Chapitre 3.

1.3.2 Attaques sur les chiffrements par bloc

On ne présente que les deux attaques les plus connues, l'attaque différentielle et l'attaque linéaire. L'attaque algébrique sur les schémas par bloc est parfois faisable : elle sera présentée au chapitre 3.

Cryptanalyse différentielle

La cryptanalyse différentielle est une méthode générique de cryptanalyse à clair choisi, elle est applicable aux algorithmes de chiffrement itératif par blocs.
Dans son sens le plus large, elle consiste en l'étude de la manière dont les différences entre les données en entrée affectent les différences de leurs sorties. Le terme se rapporte à l'ensemble des techniques permettant de retracer les différences à travers le réseau des transformations, découvrant ainsi où l'algorithme montre un comportement prédictible et exploitant ainsi ces propriétés afin de retrouver la clé secrète.

Cryptanalyse linéaire

La cryptanalyse linéaire est une attaque à texte clair connu. Elle a été proposée par Matsui dans [44] dans une version contre le DES, et a été raffinée et généralisée par la suite. Le principe général de l'attaque linéaire repose sur l'approximation de la fonction de chiffrement (par bloc) E_k par une fonction affine. Son but est de trouver des relations linéaires non équilibrée entre les bits du texte clair, ceux du texte chiffré et ceux de la clé qui sont vérifiées avec une probabilité $p \neq 1/2$. Ceci veut dire qu'on cherche des équations probabilistes de la forme :

$$a \cdot x \oplus b \cdot E_k(x) = 0$$

où a, b sont des vecteurs à coordonnées dans $\mathbb{F}_2$, x et $E_k(x)$ sont considérés comme vecteurs sur $\mathbb{F}_2$ et "$\cdot$" désigne le produit scalaire.

1.3.3 Attaques sur les algorithmes asymétriques

On peut citer les différentes attaques suivantes pour un système à clé publique : attaque sur la clé, attaque à texte en clair deviné et problème de la faible entropie, attaque à texte chiffré choisi et attaque temporelle.

Chapitre 2

Fonctions Booléennes et vectorielles

2.1 Introduction

Une fonction Booléenne est une fonction de l'espace vectoriel $\mathbb{F}_2^n$ des vecteurs binaires de longueur n, dans l'espace vectoriel à deux éléments $\mathbb{F}_2$. L'ensemble des fonctions Booléennes à n variables est un espace vectoriel de dimension 2^n qu'on notera $\mathbb{B}_n$. Les fonctions Booléennes jouent un rôle central dans la théorie des codes correcteurs d'erreurs et en cryptographie :
- Tout code de longueur 2^n, avec n un entier positif, peut être vu comme une liste de valeurs des fonctions Booléennes à n variables. Toute fonction Booléenne à n variables peut être représentée par sa table de vérité, l'ensemble de ses valeurs peut être associé à un mot binaire de longueur 2^n et vice versa.
- Le rôle des fonctions Booléennes en cryptographie conventionnelle est important, elles sont une composante clé des chiffrements par flot et par blocs (générateurs pseudo-aléatoires dans les chiffrements par flot, boites-S dans les chiffrements par blocs).
Nous présenterons dans ce chapitre des notions essentielles sur les fonctions Booléennes et vectorielles utiles à des fins cryptographiques. Nous verrons au cours de ce chapitre des liens avec les codes de Reed-Muller et plus généralement la théorie des codes correcteurs d'erreurs. Une bonne référence de la théorie des fonctions Booléennes dont est inspiré ce chapitre est [13], celle des fonctions vectorielles est [14].
Le nombre de variables n d'une fonction Booléenne est généralement petit, il était au plus 10 variables dans le cas des chiffrements à flot avant la découverte des attaques algébriques et il est maintenant au plus de 20 variables en général (voir les détails de ces attaques dans le Chapitre 3). Même avec un nombre de variables petit, l'étude des fonctions Booléennes satisfaisant les conditions requises n'est pas possible par recherche exhaustive ; en effet, le nombre de fonctions Booléennes à n variables est égale à 2^{2^n},

ce nombre est significativement grand quand $n \geq 6$, comme l'indique le tableau 2.1 ci-après.

Supposons que la visite d'une fonction Booléenne à n variable et la vérification qu'elle

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

satisfait une certaine condition nécessite un nano seconde (10^{-9} secondes), alors on aura besoin d'un million d'heures pour visiter toutes les fonctions à 6 variables et environ un billiard de fois l'âge de l'univers pour visiter celles de 7 variables. Le nombre de fonctions Booléennes à 8 variables est approximativement égal au nombre d'atomes dans l'univers.

2.2 Fonctions Booléennes

Avant de présenter une brève théorie des fonctions Booléennes, nous avons besoin de quelques définitions et notations concernant les vecteurs binaires de n bits.

2.2.1 Définitions et notations concernant les vecteurs de $\mathbb{F}_2^n$

L'ensemble $\{0, 1\}$ peut être muni d'une structure de corps et est noté $\mathbb{F}_2$, et l'ensemble $\mathbb{F}_2^n = \{0, 1\}^n$ de vecteurs binaires de longueur n peut être vu comme un espace vectoriel sur $\mathbb{F}_2$. Si x est un vecteur de $\mathbb{F}_2^n$, on l'écrit sous la forme $x = (x_1, x_2, \ldots, x_n)$. On notera simplement 0 le vecteur nul.

Nous utiliserons deux notions importantes sur les vecteurs binaires qui sont le *support* et le *poids de Hamming*. Ils sont ainsi définis pour des vecteurs de longueur n :

Définition 2.2.1 (Support)**.** *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, c'est à dire*

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

Définition 2.2.2 (Poids de Hamming)**.** *Soit un vecteur $x = (x_1, x_2, \ldots, x_n)$ de $\mathbb{F}_2^n$. Le poids de Hamming de x est égal à son nombre de composantes non nulles, on le notera*

$w_H(x)$. *C'est à dire*

$$w_H(x) = |supp(x)| = \text{ cardinal du support de } x \text{ .}$$

Nous aurons besoin de définir une relation d'ordre total entre les vecteurs de $\mathbb{F}_2^n$. L'ordre que nous utiliserons est l'ordre lexicographique inverse, nous l'appelons aussi ordre usuel. Il est défini par

Définition 2.2.3. *pour* $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n)$ *et* $x \neq y$ *:*

$$x \prec y \Leftrightarrow \exists i_0, 1 \leq i_0 \leq n : x_{i_0} < y_{i_0} \text{ et } \forall i > i_0, x_i = y_i$$

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

Définition 2.2.4. *On associe à un élément* $x^{(i)}$ *de* $\mathbb{F}_2^n$ *un entier* n_i *de l'ensemble* $\{0, 1, \ldots, 2^n - 1\}$ *par la relation :*

$$n_i = \sum_{j=1}^{n} x_j^{(i)} 2^{j-1}$$

On a donc $x^{(i)} \prec x^{(j)}$ *si et seulement si* $n_i < n_j$.

Exemple 2.2.1. *Pour* n *égal à* 5*, le monôme* $X_1X_3X_4$ *est par exemple désigné par le vecteur* $(1, 0, 1, 1, 0)$ *et correspond au nombre* 13.

2.2.2 Définition et représentation des fonctions Booléennes

On définira d'une manière analogue à celle d'un vecteur de $\mathbb{F}_2^n$, le support et le poids de Hamming d'une fonction Booléenne.

Définition 2.2.5. *Soit* f *une fonction Booléenne à* n *variables :*
- Le support de f *que nous noterons* $supp(f)$ *est l'ensemble de vecteurs de* $\mathbb{F}_2^n$ *dont la valeur par* f *est non nulle, c'est à dire*

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

- Le poids de Hamming de f *qu'on notera* $w_H(f)$ *est le cardinal de son support, c'est à dire*

$$w_H(f) = |supp(f)|$$

- La distance de Hamming entre deux fonctions Booléennes à n variables f et g qu'on notera d_H est le cardinal de l'ensemble $\{x \in \mathbb{F}_2^n, f(x) \neq g(x)\}$, elle est égale alors à $w_H(f \oplus g)$, où $\oplus$ est l'addition dans $\mathbb{F}_2$.

Notations - L'addition des bits (de $\mathbb{F}_2$) sera notée "$\oplus$" et s'il n'y a pas risque de confusion, on le notera encore "+".
- L'addition bit à bit des vecteurs de $\mathbb{F}_2^n$ sera noté "+", il en est de même pour l'addition des éléments du corps $\mathbb{F}_{2^n}$ puisqu'on peut identifier l'espace vectoriel $\mathbb{F}_2^n$ au corps fini $\mathbb{F}_{2^n}$ de la manière suivante :
$\mathbb{F}_{2^n}$ est un espace vectoriel de dimension n sur $\mathbb{F}_2$. Si on choisit une base $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ de cet espace vectoriel, alors tout élément x de $\mathbb{F}_2^n$ peut être identifié à l'élément $x_1\alpha_1 + x_2\alpha_2 + \ldots + x_n\alpha_n$ de $\mathbb{F}_{2^n}$, on notera encore x cette élément du corps.

Définition 2.2.6 (Dérivée)**.** *Soit f une fonction Booléenne à n variables et soit a un vecteur de $\mathbb{F}_2^n$. On appelle dérivée de f dans la direction de a la fonction Booléenne $D_a f(x) = f(x) \oplus f(x + a)$.*

On peut naturellement étendre récursivement la définition de la dérivée au cas des dérivées d'ordre supérieur.

2.2.3 Représentation des fonctions Booléennes

Les fonctions coordonnées $x = (x_1, x_2, \ldots, x_n) \to x_i$ (que l'on notera simplement x_i) sont les fonctions Booléennes les plus élémentaires, après les fonctions constantes 0 et 1. Le produit de fonctions coordonnées sont aussi des fonctions Booléennes sur $\mathbb{F}_2^n$, elles sont appelées les *monômes*. Le degré du monôme $\prod_{i \in I} x_i$ est le cardinal $|I|$ de I. L'unique monôme de degré 0 est la fonction constante 1. A chaque monôme correspond un mot $u \in \mathbb{F}_2^n$ tel que ce monôme s'écrive sous la forme $\prod_{i=1}^n x_i^{u_i}$, qu'on notera simplement x^u.
Le degré de ce monôme est le poids de Hamming de u. L'ensemble des monômes constitue une base de l'espace vectoriel des fonctions Booléennes.

Représentation sous forme normale algébrique "ANF"

La représentation des fonctions Booléennes largement utilisée en cryptographie et en codage est la représentation polynomiale à n variables sur $\mathbb{F}_2$. Si f est une fonction

Booléenne à n variables, alors f peut être représentée sous la forme :

$$f(x) = \bigoplus_{I \in \mathcal{P}(N)} a_I \left(\prod_{i \in I} x_i \right) = \bigoplus_{I \in \mathcal{P}(N)} a_I x^I, \tag{2.1}$$

où $\mathcal{P}(N)$ est l'ensemble des parties de $N = \{1, 2, \ldots, n\}$.
Une autre représentation de ce même polynôme utilise une indexation par les vecteurs de $\mathbb{F}_2^n$; si, pour un tel vecteur u, on note a_u ce qui a été noté $a_{supp(u)}$ dans la Relation (2.1), (où $supp(u)$ est le support de u), on aura la représentation équivalente :

$$f(x) = \bigoplus_{u \in \mathbb{F}_2^n} a_u \left(\prod_{i=1}^{n} x_i^{u_i} \right) = \bigoplus_{u \in \mathbb{F}_2^n} a_u x^u \tag{2.2}$$

Proposition 2.2.1. *[13] Toute fonction Booléenne f s'écrit d'une façon unique sous l'une des formes équivalentes (2.1) ou (2.2).*
Ces écriture s'appellent la forme normale algébrique*, ou avec une notation abrégée en anglais ANF.*

Il est commode de voir en particulier le comportement d'une fonction monôme.

Proposition 2.2.2. *[13] Une fonction monôme x^u où x, u sont des vecteurs de $\mathbb{F}_2^n$ ne vaut* 1 *qu'aux points x qui vérifient $u \subseteq x$ où l'inclusion est définie par*

$$u \subseteq x \iff \{i | u_i = 1\} \subseteq \{i | x_i = 1\}$$

En particulier, elle est nulle sur tous les points x strictement plus petits que u pour l'ordre usuel et vaut 1 en u.

Il existe une transformation involutive (i.e qui est sa propre inverse) connue sous le nom de *transformation de Möbius* binaire, très utile dans la pratique et qui relie les deux principales représentations d'une fonction Booléenne.

Théorème 2.2.1 (Transformée de Möbius binaire)**.** *[13]. Soit f une fonction Booléenne à n variables. Ses valeurs $f(x)$ et les coefficients de son ANF a_u sont reliés par*

$$f(x) = \sum_{u \subseteq x} a_u \text{ et } a_u = \sum_{x \subseteq u} f(x)$$

Degré algébrique d'une fonction Booléenne Le degré algébrique d'une fonction Booléenne f est le degré algébrique de sa forme normale algébrique, noté $d^\circ f$ (ou $\deg(f)$). Si f est écrite sous la forme (2.1), alors

$$d^\circ f = \max\{|I| / a_I \neq 0\},$$

où $|I|$ désigne le cardinal de I.

Proposition 2.2.3. *Le degré algébrique d'une fonction Booléenne est un* affine invariant*, c'est à dire est stable sous transformation affine :*
Pour toute fonction Booléenne f à n variables et pour tout automorphisme affine A sur $\mathbb{F}_2^n$, on a $d^\circ(f \circ A) = d^\circ f$.

Les fonctions les plus simples du point de vue de leurs ANF sont les fonctions Booléennes de degré au plus 1, appelées *fonctions affines*, elles peuvent s'écrire sous la forme :

$$a_1x_1 \oplus a_2x_2 \oplus \ldots \oplus a_nx_n \oplus a_0 = a \cdot x \oplus a_0 \tag{2.3}$$

où "$\cdot$" est le produit scalaire usuel dans $\mathbb{F}_2^n$.
L'écriture d'une fonction affine sous la forme (2.3) n'est pas la seule, on peut trouver d'autre formes en définissant d'autre produits scalaires.

2.2.4 Représentations trace

Dans cette représentation, le sous-espace vectoriel $\mathbb{F}_2^n$ est muni d'une structure de corps $\mathbb{F}_{2^n}$.
1. Il a été montré dans le livre de Carlet [14] que toute fonction f de $\mathbb{F}_{2^n}$ dans $\mathbb{F}_{2^n}$ admet une unique représentation comme polynôme à une variable sur $\mathbb{F}_{2^n}$ et de degré (univarié) au plus $2^n - 1$, cette représentation a la forme

$$f(x) = \sum_{i=0}^{2^n-1} \delta_i x^i \tag{2.4}$$

Toute fonction Booléenne sur $\mathbb{F}_{2^n}$ est un cas particulier d'une fonction vectorielle de $\mathbb{F}_{2^n}$ dans $\mathbb{F}_{2^n}$ (puisque $\mathbb{F}_2$ est un sous-corps de $\mathbb{F}_{2^n}$), elle admet alors la représentation unique (2.4) que l'on appellera la *représentation univariée* de f.
Un polynôme univarié $\sum_{i=0}^{2^n-1} \delta_i x^i, \delta_i \in \mathbb{F}_{2^n}$ est donc la représentation univariée d'une fonction Booléenne si et seulement si $\left(\sum_{i=0}^{2^n-1} \delta_i x^i\right)^2 = \sum_{i=0}^{2^n-1} \delta_i^2 x^{2i} = \sum_{i=0}^{2^n-1} \delta_i x^i [$ mod $x^{2^n} + x]$, c'est à dire que $\delta_0, \delta_{2^n-1} \in \mathbb{F}_2$ et, pour tout $i = 1, \ldots, 2^n - 2, \delta_{2i} = \delta_i^2$ où l'indice $2i$ est calculé mod $(2^n - 1)$.
2. La fonction définie sur $\mathbb{F}_{2^n}$ par $tr_n(x) = x + x^2 + x^{2^2} + \ldots + x^{2^{n-1}}$ est linéaire sur $\mathbb{F}_2$ et satisfait $(tr_n(x))^2 = tr_n(x^2) = tr_n(x)$, elle est donc à valeurs dans $\mathbb{F}_2$. Cette fonction est appelée la *fonction trace* de $\mathbb{F}_{2^n}$ sur son corps premier $\mathbb{F}_2$. La fonction $(u, v) \mapsto tr_n(uv)$ définit un produit scalaire sur $\mathbb{F}_{2^n}$. On en déduit que toute fonction Booléenne f peut être écrite sous la forme

$$f(x) = tr_n\left(\sum_{i=0}^{2^n-1} \beta_i x^i\right) \tag{2.5}$$

où $\beta_i \in \mathbb{F}_{2^n}$. Cette représentation n'est pas unique.
3. La relation $\delta_{2i} = \delta_i^2$ permet de regrouper les termes correspondant à une même classe cyclotomique. On obtient une autre forme d'écrire une fonction Booléenne

$$f(x) = \sum_{j \in \Gamma(n)} tr_{n_j}(A_j x^j) + \mu(1 + x^{2^n-1}) \tag{2.6}$$

où $\mu = w_H(f)[\mod 2]$, n_j est le cardinal de la classe cyclotomique de 2 modulo $2^n - 1$ contenant j, $\Gamma(n)$ est l'ensemble obtenu en choisissant un élément de chaque classe cyclotomique de 2 modulo $2^n - 1$ et $A_j = \sum_{k=0}^{2^n-2} f(\alpha^k)\alpha^{kj}$, avec α un élément primitif de $\mathbb{F}_{2^n}$.
On appelle globalement "représentations trace" les trois expressions (2.4), (2.5) et (2.6).
La représentation sous la forme normale algébrique peut être déduite de la forme univariée . On peut écrire x sous la forme $x = \sum_{i=1}^{n} x_i\alpha_i$, où $(\alpha_1, \ldots, \alpha_n)$ est une base de $\mathbb{F}_{2^n}$, d'autre part, tout entier naturel j admet une représentation binaire de la forme $j = \sum_{s=0}^{n-1} j_s 2^s, j_s \in \{0, 1\}$. On a donc

$$\begin{aligned} f(x) &= \sum_{j=0}^{2^n-1} \delta_j \left(\sum_{i=1}^{n} x_i\alpha_i \right)^j \\ &= \sum_{j=0}^{2^n-1} \delta_j \left(\sum_{i=1}^{n} x_i\alpha_i \right)^{\sum_{s=0}^{n-1} j_s 2^s} \\ &= \sum_{j=0}^{2^n-1} \delta_j \prod_{s=0}^{n-1} \left(\sum_{i=1}^{n} x_i\alpha_i^{2^s} \right)^{j_s}. \end{aligned}$$

En simplifiant la dernière expression, on trouvera la ANF de f.

Proposition 2.2.4. *Le degré algébrique de la fonction f écrite en représentation univariée ($f(x) = \sum_{j=0}^{2^n-1} \delta_j \left(\sum_{i=1}^{n} x_i\alpha_i\right)^j$) est*

$$d^\circ f = \max_{j=0,\ldots,2^n-1, \delta_j \neq 0} w_2(j) = \max_{j=0,\ldots,2^n-1, \delta_j \neq 0} |\{s, s \neq 0 / j = \sum_{s=0}^{n-1} j_s 2^s\}| \tag{2.7}$$

Le nombre $w_2(j)$ est appelé le 2-poids de j.

2.2.5 Transformée de Walsh

La transformation de Walsh est une notion très utile dans l'étude de quelques critères de sécurité des fonctions Booléennes.

Définition 2.2.7. *La transformée de Fourier discrète d'une fonction Booléenne (ou plus généralement d'une fonction numérique) f définie sur $\mathbb{F}_2^n$ (aussi appelé transformée de Hadamard) est définie par*

$$\widehat{f}(u) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{x \cdot u} \tag{2.8}$$

où $x \cdot u$ est un produit scalaire sur $\mathbb{F}_2^n$.

Définition 2.2.8. *La transformée de Walsh d'une fonction Booléenne f est la transformée de Fourier discrète de sa fonction signe $f_\chi = (-1)^f$,*

$$\widehat{f}_\chi(u) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus x \cdot u} \tag{2.9}$$

De plus, il existe une transformée inverse

$$f(x) = \frac{1}{2^n} \sum_{u \in \mathbb{F}_2^n} \widehat{f}_\chi(x)(-1)^{x \cdot u} \tag{2.10}$$

Définition 2.2.9 (Spectre de Walsh)**.** *L'ensemble des valeurs de la transformée de Walsh de f est connu sous le nom de* spectre de Walsh *de f.*

Le spectre de Walsh nous donne une troisième représentation d'une fonction Booléenne par la transformée de Walsh inverse.
Il existe plusieurs propriétés importantes de la transformée de Fourier et de Walsh, on rappelle une relation très utile dans la pratique : la *relation de Parseval* :

Proposition 2.2.5. *Pour toute fonction numérique φ à n variables , on a :*

$$\sum_{u \in \mathbb{F}_2^n} \widehat{\varphi}^2(u) = 2^n \sum_{x \in \mathbb{F}_2^n} \varphi^2(x)$$

Si φ est à valeurs dans $\{-1, 1\}$, la relation de Parseval donne

$$\sum_{u \in \mathbb{F}_2^n} \widehat{\varphi}^2(u) = 2^{2n} \tag{2.11}$$

2.3 Lien avec les codes de Reed-Muller

Les fonctions Booléennes sont fortement liées aux codes correcteurs d'erreurs, nous présentons ici quelques notions utiles pour la suite de cette thèse qui concernent les codes de Reed-Muller.

Soit r un entier positif. Le code de Reed-Muller d'ordre r est l'ensemble des vecteurs de longueur 2^n représentant la liste des valeurs des fonctions Booléennes à n variables (nous appelerons ces listes de valeurs les vecteurs de valeurs) de degrés algébriques au plus r, il est noté $RM(r,n)$. Ce code linéaire est de dimension $k = \sum_{i=0}^{r} \binom{n}{i}$ et de distance minimale [1] $d = 2^{n-r}$
Pour $r = 1$, $RM(1,n)$ est une classe particulièrement importante celle des *fonctions affines*. Le poids d'une fonction affine non constante est le nombre de points d'un hyperplan affine.

2.4 Critères cryptographiques des fonctions Booléennes

Le recherche d'une fonction Booléenne ayant de bonnes propriétés cryptographique est un problème difficile ; en pratique les cryptographes n'ont qu'un choix restreint pour la conception d'un système de chiffrement. Il est en effet nécessaire d'utiliser des fonctions avec de bonnes propriétés pour résister aux nombreuses attaques. Bien entendu, les propriétés cryptographiques d'une fonction Booléennes ne peuvent être toutes optimum simultanément, il existe toujours un compromis entre elles. Certaines propriétés des fonctions Booléennes sont invariantes sous transformations affines. Deux fonctions f et g sur $\mathbb{F}_2^n$ seront dites affinement équivalentes s'il existe un automorphisme affine L telle que $f(x) = g \circ L(x)$ pour tout $x \in \mathbb{F}_2^n$.

2.4.1 Degré algébrique

Les fonctions cryptographiques doivent être de degré algébrique élevé. Tous système cryptographique (à flot ou par bloc) utilisant une fonction Booléenne de bas degré est vulnérable aux attaques. Dans le cas du chiffrement par flot, le degré de la fonction Booléenne à n variables utilisée doit être très proche de n à cause de l'attaque algébrique rapide, alors que pour les chiffrements par blocs, il suffit que le degré soit sensiblement plus grand que 2.. Par exemple, dans le cas dans le générateur de combinaison de n LFSRs de longueurs $L_1, \ldots, L_n$ combinés par la fonction

$$f(x) = \bigoplus_{u \in \mathbb{F}_2^n} f_u \left(\prod_{i=1}^{n} x_i^{u_i} \right)$$

1. La distance minimale d'un code est la plus petite distance entre deux mots du code

alors, (voir [55]) la suite produite par f a une complexité linéaire égale à

$$L \leq \bigoplus_{u \in \mathbb{F}_2^n} f_u \left(\prod_{i=1}^{n} L_i^{u_i} \right)$$

De plus, on a l'égalité si les séquences produites par les LSFRs sont de périodes maximales et les L_i sont deux à deux premiers entre eux.
Le degré algébrique d'une fonction Booléenne est invariant sous transformation affine, puisque $d^\circ(f \circ L) = d^\circ f$ pour tout automorphisme affine L car le degré de L est égal à 1.

2.4.2 Fonction équilibrée, fonction résiliente

Les fonctions cryptographiques doivent être *équilibrées* pour éviter une dépendance statistique entre le clair et le chiffré. Une fonction Booléenne est dite équilibrée si elle produit un nombre égal de 0 et de 1 (c'est à dire que la suite de sa sortie est uniformément distribuée sur $\{0, 1\}$). On peut déduire de la définition de la transformée de Walsh d'une fonction f que cette fonction est équilibrée si et seulement si $\widehat{f_\chi}(0) = 0$.
Le critère d'équilibre est un invariant affine puisqu'il est clair que si f est équilibrée alors $f \circ L$ est aussi équilibrée pour tout automorphisme affine L.
Une fonction Booléenne non équilibrée utilisée dans le générateur de filtrage rend possible une attaque par distingueur. Ces attaques ont pour but de distinguer une suite pseudo-alétoire d'une suite aléatoire.
Il existe une condition supplémentaire (outre l'équilibre) dans le cas d'un générateur de combinaison : toute fonction de combinaison f doit rester équilibrée si on fixe un certain nombre de ses entrées.

Définition 2.4.1. *Soient $0 \leq m < n$ deux entiers. Une fonction Booléenne à n variables f est dite m-résiliente si chacune de ses restrictions obtenue en fixant au plus m entrées est équilibrée.*

Cette définition a été introduite par Siegenthaler dans [59] suite à une attaque menée sur le générateur de combinaison, appelée *attaque par corrélation* : si f n'est pas m-résiliente, alors il existe une corrélation entre la sortie de f et au plus m coordonnées de ses entrées, cette attaque a été faite contre le générateur de Geffe (voir exemple 1.3.2).
L'*attaque par corrélation rapide* est une attaque plus efficace (même si l'attaque par corrélation classique n'est pas réalisable) si la fonction Booléenne de combinaison n'est pas hautement non linéaire [45]. Plus précisément, Canteaut et Trabbia dans [10] et Canteaut dans [9] ont montré que l'attaque par corrélation sur le générateur de combinaison (par une fonction m-résiliente) est inefficace que possible si le coefficient de

Walsh $\widehat{f_\chi}(u)$ 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. On voit ainsi que la non-linéarité est liée à plusieurs attaques.
La résilience a été caractérisée par Xiao et Massey à l'aide des transformées de Fourier et de Walsh.

Théorème 2.4.1. *[32] Une fonction Booléenne à n variables f est m-résiliente si et seulement si $\widehat{f_\chi}(u) = 0$ pour tout $u \in \mathbb{F}_2^n$ tel que $w_H(u) \leq m$. Ceci est équivalent à dire que f est m-résiliente si et seulement si elle équilibrée et $\widehat{f_\chi}(u) = 0$ pour tout $u \in \mathbb{F}_2^n$ tel que $0 < w_H(u) \leq m$.*

Il existe une borne supérieure sur l'ordre de résilience en fonction du degré algébrique (et vice versa) d'une fonction Booléenne.

Théorème 2.4.2. *Soit f une fonction Booléenne à n variables. Si f est m-résiliente et si $0 \leq m < n-1$, alors $m + \deg(f) \leq n-1$.*

Contrairement au degré algébrique et à l'équilibre, l'ordre de résilience d'une fonction Booléenne n'est pas un invariant affine, sauf pour l'ordre nul (c'est l'équilibre) et l'ordre n.

2.4.3 Non-linéarité

Dans un système de chiffrement, une *confusion* introduite par une fonction cryptographique est sa distance de Hamming aux fonctions affines. La distance entre la fonction cryptographique et toutes les fonctions affines doit être proche de 2^{n-1}. En effet, l'existence d'approximations affines des fonctions Booléennes (ou vectorielles) utilisées dans les systèmes de chiffrements est exploitable pour monter des attaques sur les chiffrements à flots et par blocs (voir [64, 44]).
Dans le cas des chiffrements à flots comme nous l'avons déjà indiqué sont les *attaques par corrélations rapides*, (voir [10, 22] par exemple pour les détails de ces attaques). Si f est la fonction de combinaison, on essaye de construire le même générateur à la différence près que la fonction de combinaison est une fonction g de distance à f inférieure à 2^{n-1} ; si s est la suite produite par f et t celle produite par g alors, la suite s correspond à la transmission avec erreurs de la suite t. Attaquer le générateur se fait en essayant de corriger les erreurs de la suite t comme si elle était transmise sur un canal bruité.

Définition 2.4.2. *La non-linéarité d'une fonction f est la distance minimale entre f et toutes les fonctions affines.*

Une haute non-linéarité est une condition suffisante pour que l'attaque par corrélation rapide soit non efficace.

La non-linéarité d'une fonction Booléenne est un invariant affine, puisque, $d_H(f \circ L, \ell \circ L) = d_H(f, \ell)$ pour toute fonction f et toute fonction affine ℓ, et pour tout automorphisme affine, et puisque $\ell \circ L$ décrit tout l'ensemble des fonctions affines quand ℓ décrit cet ensemble.

Proposition 2.4.1. *La non-linéarité d'une fonction Booléenne f est égale à*

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

Alors une fonction est de haute non-linéarité si toutes les valeurs du spectre de Walsh sont de petite valeur absolue.

Proposition 2.4.2 (Borne du rayon de recouvrement)**.** *La non-linéarité d'une fonction Booléenne à n variables est bornée comme suit :*

$$nl(f) \leq 2^{n-1} - 2^{\frac{n}{2}-1} \tag{2.12}$$

De plus, une fonction f atteint cette borne si et seulement si $a \mapsto \widehat{f_\chi}^2(a)$ est une fonction constante.

Démonstration. La relation de Parseval (2.11) appliquée à la fonction f_χ donne $\sum_{a \in \mathbb{F}_2^n} \widehat{f_\chi}^2(a) = 2^{2n}$, ce qui implique que la moyenne de $\widehat{f_\chi}^2(a)$ est égale à 2^n. Le maximum de $\widehat{f_\chi}^2(a)$ étant supérieur ou égal à sa moyenne (avec égalité si et seulement si $\widehat{f_\chi}^2(a)$ est constante), on en déduit que $\max_{a \in \mathbb{F}_2^n} |\widehat{f_\chi}(a)| \geq 2^{n/2}$, d'où le résultat. □

Définition 2.4.3 (Fonction courbe)**.** *Les fonctions pour lesquelles la borne (2.12) est atteinte sont appelées les fonctions courbes.*

Une fonction courbe f vérifie $|\widehat{f_\chi}(a)| = 2^{n/2}$ pour tout vecteur $a \in \mathbb{F}_2^n$. Les fonctions courbes existent seulement si n est pair. Pour n impair l'inégalité (2.12) n'est pas atteinte, ainsi, la non-linéarité maximale d'une fonction Booléenne à n variables est entre $2^{n-1} - 2^{\frac{n-1}{2}}$ (qui est atteinte par les fonctions quadratiques) et $2^{n-1} - 2^{\frac{n}{2}-1}$.
Les fonctions courbes ne sont pas équilibrées, elles ne sont donc pas utilisées telles qu'elles dans les systèmes cryptographiques. On peut équilibrer une fonction courbe mais notons que c'est la résistance aux attaques algébriques qui est la plus problématique.

La non-linéarité d'ordre r : Une notion plus précise que la non-linéarité pour une fonction Booléenne est son profil de non-linéarité ; cette notion joue un rôle important dans la sécurité des systèmes cryptographiques. Pour tout entier positif $r \leq n$, on note $nl_r(f)$ la distance de Hamming minimale entre f et toutes les fonctions Booléennes de degrés au plus r (le cas où $r = 1$ correspond à la notion de non-linéarité usuelle notée simplement $nl(f)$). En d'autre termes, $nl_r(f)$ est égal à la distance entre f et le code de Reed-Muller $RM(r, n)$ de longueur 2^n et d'ordre r ; cette distance est égale au nombre de bits à changer dans la table de vérité de f pour obtenir une fonction Booléenne de degré algébrique au plus r. Ce paramètre est appelé la *non-linéarité de f d'ordre r*, et est entre autre lié à la corrélation maximale de la fonction Booléenne par rapport à un sous ensemble de variables (qui est utilisable dans l'attaque par corrélation).
Le profil de non-linéarité d'une fonction Booléenne f est par définition la suite des valeurs $nl_r(f)$ pour r allant de 1 à $n - 1$.

Quelques propriétés de la non-linéarité d'ordre r

- La non-linéarité d'ordre r d'une fonction Booléenne f ne change pas si l'on ajoute à f une fonction de degré au plus r.
- $RM(r, n)$ étant invariant par automorphisme affine, composer une fonction Booléenne par un automorphisme affine ne change pas sa non-linéarité d'ordre r (i.e nl_r est un invariant affine).
- La distance minimale de $RM(r, n)$ étant égale à 2^{n-r} pour tout $r \leq n$, nous avons $nl_r(f) \geq 2^{n-r-1}$ pour toute fonction f de degré algébrique égal à $r + 1 \leq n$. En effet, si g, h sont deux fonctions distinctes de $RM(r, n)$ alors on a,

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

  où d_H est la distance de Hamming, ce qui implique que

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

  D'où $nl_r(f) \geq 2^{n-r-1}$.
- Iwata et Kurosawa [34] ont observé que si f_0 est la restriction de f à l'hyperplan linéaire H d'équation $x_n = 0$ et f_1 est la restriction de f à l'hyperplan affine H' d'équation $x_n = 1$ (ces deux fonctions peut être vues comme fonctions à $(n - 1)$ variables), alors on a $nl_r(f) \geq nl_r(f_0) + nl_r(f_1)$ puisque, pour toute fonction g de degré algébrique au plus r, les restrictions de g aux hyperplans H et H' sont de gré au plus r, on a $d_H(f, g) \geq nl_r(f_0) + nl_r(f_1)$. De plus, si $f_0 = f_1$ alors on a l'égalité puisque si g est la meilleure approximation de degré au plus r de $f_0 = f_1$, alors g est de distance $2nl_r(f_0)$ de f.

2.4.4 Critère d'Avalanche Stricte et Critère de Propagation

Le *Critère d'Avalanche Stricte* (*SAC*) a été introduit par Webster et Tavares [62], puis généralisé au *Critère de Propagation* (*PC*) par Preneel et al [51]. Le SAC et ses généralisations sont basés sur les propriétés des dérivées des fonctions Booléennes. Ces propriétés traduisent le comportement d'une fonction quand on change certaines des coordonnées de ses entrées. Elles sont liées par conséquent aux propriétés de diffusion des systèmes cryptographiques utilisant cette fonction. Soit f une fonction Booléenne sur $\mathbb{F}_2^n$ et $E \subset \mathbb{F}_2^n$. La fonction f satisfait le *critère de propagation PC* par rapport à E si, pour tout $a \in E$, la dérivée $D_a f(x) = f(x) \oplus f(x+a)$ est équilibrée. On dit que f satisfait *PC(l)* si elle satisfait *PC* par rapport à tous les vecteurs non nuls de poids au plus l. Le critère *SAC* correspond à *PC(1)*.
Les critères *SAC, PC* et *PC(l)* ne sont pas des invariants affines en général.

2.4.5 Non-existence de structure linéaire non nulle

On appelle *noyau linéaire* d'une fonction f (Booléenne ou vectorielle) l'ensemble de vecteurs e pour lesquels la fonction $D_e f$ est constante. Le noyau linéaire d'une fonction est un sous espace vectoriel de $\mathbb{F}_2^n$ sur $\mathbb{F}_2$. Un élément e du noyau linéaire de f s'appelle une *structure linéaire* de f. Si $D_e f = b$, on dit que f a une structure linéaire (e, b). Il faut éviter l'utilisation des fonctions cryptographiques ayant des structures linéaires non nulles.
L'existence/non-existence de structures linéaires non nulles est clairement un invariant affine.

2.4.6 Immunité algébrique

Nous nous intéressons plus particulièrement à cette notion dans cette thèse ; nous l'étudierons d'une façon plus détaillée dans le cas des fonctions vectorielles. L'immunité algébrique a été introduite suite à la découverte des attaques algébriques (dont on verra les détails au Chapitre 3) par Courtois et Meier [26] en 2003, ces attaques sont efficaces (sous certains conditions) contre les chiffrements par flot. Considérons le cas du générateur filtré ou le générateur de combinaison, de partie linéaire (les n LFSRs dans le cas du générateur de combinaison, un LFSR dans le cas du générateur filtré) 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 bas degré telles que $fg = h$ alors l'attaque algébrique est réalisable. L'hypothèse est équivalente à l'existence d'une fonction $g \neq 0$ de bas degré telle que $fg = 0$ ou $(f \oplus 1)g = 0$. En effet, $fg = h$ implique que $f^2g = fh$, c'est à dire (puisque $f^2 = f$), $f(g \oplus h) = 0$, ce qui donne l'égalité cherchée si $g \neq h$ en remplaçant $g \oplus h$ par g, et si $g = h$ alors $fg = h$ est équivalente à $(f \oplus 1)g = 0$. Remarquons que, plus le degré de g est petit, plus l'attaque est efficace. Un critère nécessaire (mais non suffisant) contre ces attaques est que la fonction f ait une immunité algébrique élevée ; pour plus de détails voir le chapitre 4.
Le degré minimal de toutes les fonctions non nulles g vérifiant $fg = 0$ ou $(f \oplus 1)g = 0$ est appelé l'*immunité algébrique* de la fonction Booléenne f et est noté $AI(f)$. On a aussi

$$AI(f) = \min\{\max(d^\circ g, d^\circ(fg)), g \neq 0\}.$$

Ces deux définitions sont équivalentes puisque les deux conditions décrites ci-dessus sont équivalentes.
Nous verrons au chapitre 4 que l'immunité algébrique d'une fonction Booléenne (ou en général d'un sous ensemble de $\mathbb{F}_2^n$) est un invariant affine.

2.5 Fonctions Booléennes vectorielles

On s'intéresse dans cette section aux fonctions Booléennes à plusieurs sorties d'un point de vue cryptographique. Ce sont les fonctions de l'espace vectoriel $\mathbb{F}_2^n$ des vecteurs binaires de longueur n, dans l'espace vectoriel $\mathbb{F}_2^m$, pour n, m deux entiers positifs. Le cas $m = 1$ correspond aux fonctions Booléennes (à une sortie). On se réfère aux définitions et notations vues dans la section précédente "Fonctions Booléennes" quand c'est nécessaire.

Soit n, m deux entiers positifs. Une fonction de $\mathbb{F}_2^n$ dans $\mathbb{F}_2^m$ est appelée (n, m)-fonction. Une telle fonction F étant donnée, les fonctions Booléennes $f_1, \ldots, f_m$ définies pour tout $x \in \mathbb{F}_2^n$ par $F(x) = (f_1(x), \ldots, f_m(x))$, sont appelées les fonctions coordonnées de F. Quand les entiers n, m ne sont pas spécifiés, les (n, m)-fonctions sont appelées *fonctions Booléennes à plusieurs sorties, fonctions Booléennes vectorielles* ou *boites-S*[2], ce dernier terme est le plus utilisé en cryptographie, il est réservé aux fonctions vectorielles dont le rôle est d'assurer un niveau suffisant de confusion au système. Les boites-S constituent la partie non linéaire des systèmes de chiffrement par bloc, elle est la partie qui garantit la sécurité de tels systèmes.

2. "S" pour substitution

Les attaques les plus connues sur les systèmes de chiffrement par bloc sont l'attaque différentielle [5], l'attaque linéaire [44], l'attaque différentielle d'ordre supérieur[39], l'attaque par interpolation[35] (une description de l'attaque différentielle et de l'attaque linéaire est dans le chapitre 1) et l'attaque algébrique (voir le chapitre 3). Cette dernière attaque n'est pas efficace (contre les schémas par blocs) mais représente une menace potentielle.

Les (n, m)-fonctions peuvent être utilisées comme générateurs pseudo-aléatoires dans un chiffrement à flot, chaque fonction coordonnée est utilisée pour combiner les sorties de n LFSRs, ou pour en filtrer un, dans le but de générer m bits à chaque cycle d'horloge, ce qui permet l'accélération de l'opération de chiffrement. Mais ces générateurs sont moins sûres. Les attaques décrites dans la section précédente sont aussi réalisables contre ce type de générateurs, elles sont en fait plus efficaces.
Convention : Dans toute cette thèse, on convient d'appeler *fonctions Booléennes* les fonctions Booléennes à une sortie, et *fonctions vectorielles* les fonctions Booléennes à sorties multiples.

2.6 Généralités sur les fonctions vectorielles

On conserve les notations utilisées pour les fonctions Booléennes. Plusieurs notions sur les fonctions Booléennes peuvent être généralisés aux fonctions vectorielles en considérant les fonctions composantes. Si F est une (n, m)-fonction alors, les fonctions composantes de $F = (f_1, \ldots, f_m)$ sont les fonctions $v \cdot F = v_1 f_1 \oplus \ldots \oplus v_m f_m$ où $v \in \mathbb{F}_2^m, v \neq 0$.

2.6.1 Représentation d'une fonction vectorielle

La Forme Normale Algébrique. La notion de forme normale algébrique d'une fonction Booléenne peut être généralisée aux fonctions vectorielles. Si F est une (n, m)-fonction alors, chacune de ses coordonnées est une fonction Booléenne à n variables, elle admet donc une représentation sous forme d'une ANF, la fonction F est alors représentée d'une façon unique sous la forme d'un polynôme de la même forme avec coefficients dans $\mathbb{F}_2^m$:

$$F(x) = \bigoplus_{I \in \mathcal{P}(N)} a_I \left(\prod_{i \in I} x_i \right) = \bigoplus_{I \in \mathcal{P}(N)} a_I x^I, \tag{2.13}$$

où $\mathcal{P}(N)$ est l'ensemble des parties de $N = \{1, 2, \ldots, n\}$, et $a_I \in \mathbb{F}_2^m$. Le polynôme (2.13) est encore appelé la forme normale algébrique (ANF) de F.
Le *degré algébrique* d'une (n, m)-fonction est le degré algébrique de son ANF, ce degré est égal à

$$d^\circ F = \max\{|I|, a_I \neq (0, \ldots, 0)\}.$$

Donc il est égal au degré algébrique maximal des fonctions coordonnées de F. Il est aussi égal au degré maximal de toutes les fonctions composantes de F.
Le degré algébrique d'une (n, m)-fonction est un invariant affine à gauche et à droite (le degré de F ne change pas si on la compose à gauche ou à droite par un automorphisme affine).

La représentation comme polynôme sur $\mathbb{F}_{2^n}$. Une seconde représentation des (n, m)-fonctions existe quand $n = m$. Comme nous avons expliqué dans la représentation trace des fonctions Booléennes, toute (n, n)-fonction F admet une unique *représentation polynomiale univariée* sur $\mathbb{F}_{2^n}$, de degré au plus $2^n - 1$:

$$F(x) = \sum_{i=0}^{2^n-1} \delta_i x^i, \delta_i \in \mathbb{F}_{2^n} \tag{2.14}$$

On peut obtenir la forme normale algébrique à partir de la représentation polynomiale de la même manière que celle des fonctions Booléennes. On en déduit le degré algébrique de F : si $F(x) = \sum_{j=0}^{2^n-1} \delta_j \left(\sum_{i=1}^{n} x_i \alpha_i\right)^j$, alors

$$d^\circ F = \max_{j=0,\ldots,2^n-1, \delta_j \neq 0} w_2(j) = \max_{j=0,\ldots,2^n-1, \delta_j \neq 0} |\{s, s \neq 0 / j = \sum_{s=0}^{n-1} j_s 2^s\}|. \tag{2.15}$$

Si m est un diviseur de n, alors toute (n, m)-fonction F peut être considérée comme 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 F admet une représentation polynômiale univariée de la forme (2.14) avec $\delta_i \in \mathbb{F}_{2^m}$.

Dérivée, structure linéaire d'une fonction vectorielle

Définition 2.6.1 (Dérivée)**.** *Soit F une (n, m)-fonction et soit a un vecteur de $\mathbb{F}_2^n$.*
- On appelle dérivée de F dans la direction a la fonction définie par $D_a F(x) = F(x) + F(x + a)$.
- La dérivée d'ordre k de F est la fonction $D_{a_1} D_{a_2} \cdots D_{a_k} F(x) = \sum_{b \in \mathbb{F}_2^k} F(x + \sum_{i=1}^{k} b_i a_i)$, où $a_1, a_2, \cdots, a_k$ sont des vecteurs de $\mathbb{F}_2^n$.

Le degré algébrique de $D_a F$ est au plus $d^\circ F - 1$. La fonction $D_{a_1} D_{a_2} \cdots D_{a_k} F$ est de degré algébrique au plus $d^\circ F - k$. Notons que $D_{a_1} D_{a_2} \cdots D_{a_k} F$ est nulle si les vecteurs $a_1, \cdots, a_k$ ne sont pas linéairement indépendants.

Définition 2.6.2 (structure linéaire)**.** *Une structure linéaire d'une (n, m)-fonction F est un couple (a, b) de $\mathbb{F}_2^n \times \mathbb{F}_2^m$ tel que D_aF est une fonction constante égale à b.*

2.6.2 Transformée de Walsh d'une fonction vectorielle

La transformée de Walsh d'une (n, m)-fonction F calculée en un point (u, v) et notée W_F est la transformée de Walsh de la fonction composante $v \cdot F$ au point u, plus précisément :

$$W_F(u, v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{v \cdot F(x) \oplus u \cdot x} \tag{2.16}$$

Elle est aussi égale à la transformée de Fourier de l'indicatrice du graphe de F, où le graphe de F noté $gr(F)$ est l'ensemble de points (x, y) de $\mathbb{F}_2^n \times \mathbb{F}_2^m$ tels que $y = F(x)$ et où l'indicatrice du graphe de F est la fonction Booléenne notée $1_{gr(F)}$ qui vaut 1 sur le graphe de F et 0 ailleurs.

2.6.3 Critères de sécurité des fonctions vectorielles

Les fonctions vectorielles utilisées en cryptographie doivent satisfaire des critères de sécurité pour résister à quelques attaques. Essentiellement, une fonction vectorielle doit être : de haute non-linéarité [49], de haute non-linéarité d'ordre supérieur [37], d'Immunité algébrique élevée et de préférence équilibrée et résiliente. Nous ne rappelons ici que quelques critères essentielles.

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

Proposition 2.6.1. *[40] Une (n, m)-fonction F est équilibrée si et seulement si toutes ses fonctions composantes sont équilibrées, c'est à dire que, pour tout élément non nul $v \in \mathbb{F}_2^m$, la fonction Booléenne $v \cdot F$ est équilibrée.*

Non-linéarité Les fonctions Booléennes en Cryptographie doivent être de grande distance de Hamming de toutes les fonctions affines, pour cela il faut que la fonction

Booléenne ait une haute non-linéarité. Plusieurs résultats sur la non-linéarité des fonctions vectorielles sont des généralisations de celles des fonctions Booléennes.

Définition 2.6.3. *[50, 21] La non-linéarité $nl(F)$ d'une (n,m)-fonction F est la non-linéarité minimale de toutes ses fonctions composantes $v \cdot F, v \neq 0$.*

La non-linéarité quantifie le degré de résistance d'une boite-S aux attaques linéaires [44].
Il est facile de voir que la non-linéarité est un invariant affine à gauche et à droite (elle ne change pas si on compose F à gauche ou à droite par un automorphisme affine).
La non-linéarité d'une fonction vectorielle est aussi exprimée (comme dans le cas des fonctions Booléennes) en fonction de la transformée de Walsh :

$$\begin{aligned} nl(F) &= 2^{n-1} - \frac{1}{2} \max_{v \in \mathbb{F}_2^{m*}; u \in \mathbb{F}_2^n} \left| \sum_{x \in \mathbb{F}_2^n} (-1)^{v \cdot F(x) \oplus u \cdot x} \right| \\ &= 2^{n-1} - \frac{1}{2} \max_{v \in \mathbb{F}_2^{m*}; u \in \mathbb{F}_2^n} \left| \widehat{1_{gr(F)}}(u,v) \right| \end{aligned}$$

La borne du rayon de recouvrement (2.12) sur la non-linéarité des fonctions Booléennes est encore valide pour les fonctions vectorielles puisque la non-linéarité d'une fonction vectorielle est le minimum des non-linéarités de ses fonctions composantes. Si F est une (n,m)-fonction, alors

$$nl(F) \leq 2^{n-1} - 2^{\frac{n}{2}-1} \tag{2.17}$$

Définition 2.6.4. *Une fonction courbe est une (n,m)-fonction pour laquelle la borne du rayon de recouvrement (2.17) est atteinte.*

La notion de fonction vectorielle courbe est un invariant sous composition à droite et à gauche par automorphismes affine.

Proposition 2.6.2. *Une (n,m)-fonction est courbe si et seulement si toutes ses dérivées $D_a F(x) = F(x) + F(x+a), a \in \mathbb{F}_2^{n*}$ sont équilibrées.*

Nous avons la condition nécessaire suivante pour l'existence de fonctions courbes.

Proposition 2.6.3. *Les (n,m)-fonctions courbes n'existent que si n est pair et $m \leq n/2$.*

La Proposition 2.6.3 nous amène à se demander s'il existe une borne supérieure de la non-linéarité, meilleure que la borne du rayon de recouvrement si $m > n/2$. Une telle borne a été (dans un sens) redécouverte[3] par Chabaud et Vaudenay dans [21] :

Théorème 2.6.1 (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 :*

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

Problème ouvert : Trouver une borne meilleure que la borne du rayon de recouvrement (2.17) quand :
- n est impair et $m < n$,
- n est pair et $n/2 < m < n$.

Non-linéarité d'ordre r

Définition 2.6.5. *La non-linéarité d'ordre r d'une fonction vectorielle F notée $nl_r(F)$ est la non-linéarité minimale d'ordre r de toutes ses fonctions composantes $v \cdot F, v \in \mathbb{F}_2^m \setminus \{0\}$, où " $\cdot$ " désigne le produit scalaire usuel dans $\mathbb{F}_2^m$.*

D'une façon équivalente, $nl_r(F)$ est la non-linéarité minimale d'ordre r de toutes les fonctions $\ell \circ F$, où ℓ appartient à l'ensemble des formes linéaires sur $\mathbb{F}_2^m$ (ainsi, les fonctions $\ell \circ F$ sont les combinaisons linéaires des fonctions coordonnées ayant les coefficients non tous nuls).

Immunité algébrique. L'existence d'équations multivariées de bas degré en les bits d'entrées et les bits de sorties d'une boite S peut être exploitée dans les attaques algébriques sur les schémas par bloc [25] (ce seront traitées au Chapitre 3). Le système d'équations qu'on obtient est généralement difficile à résoudre contrairement au cas d'un chiffrement à flots. Un critère nécessaire mais non suffisant contre ces attaques (comme le cas d'une fonction Booléenne) 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, l'Immunité algébrique par composantes et l'Immunité algébrique du graphe. Nous allons étudier ces trois notions en détails dans le Chapitre 5 et des bornes les concernant dans le Chapitre 6.

3. on écrit "redécouverte" puisque une borne sur les séquences due à Sidelnikov [58] est équivalente à la borne obtenue par Chabaud et Vaudenay pour les fonctions puissance et leur démonstration est en fait valable pour toutes les fonctions.

Chapitre 3

Attaques algébriques

Nous décrivons dans ce chapitre un nouveau type d'attaques découvertes ces dernières années sur les modèles du registre filtré et des registres combinés ; elles sont essentiellement de nature algébrique, contrairement aux attaques par corrélation par exemple, qui sont plus de nature statistique. Ces attaques sont appelées *Attaques Algébriques*, leur principe remonte aux travaux de Claude Shannon et son article [56]. La technique des attaques algébriques consiste à essayer d'exprimer l'algorithme de chiffrement sous la forme d'un gros système d'équations algébriques à plusieurs variables, la résolution d'un tel système donnant la clé secrète ou du moins l'initialisation du registre filtré (ou des registres combinés) dans un chiffrement par flot. Un paramètre majeur pour augmenter la complexité de l'attaque algébrique est alors un degré algébrique élevé des équations du système en question. Supposons qu'on a un générateur filtré (ou combiné) par une fonction f, si la fonction de transition entre deux états consécutifs du LFSR filtré est linéaire (resp. les fonctions de transition entre deux états consécutifs des LFSRs combinés sont linéaires) alors chaque bit de la suite chiffrante est exprimé comme une fonction de degré égal au degré de f en fonction du vecteur d'initialisation du LFSR filtré (resp. vecteurs d'initialisation des LFSRs combinés). Pour cette raison, il a été considéré (jusqu'à récemment) qu'un degré élevé de la fonction de filtrage (ou de combinaison) est le critère essentiel pour résister aux attaques algébriques.
Mais les attaques algébriques sont devenues plus efficaces grâce aux idées de Courtois et Meier [26] : elles sont réalisables parfois même si la fonction de filtrage (ou de combinaison) est de degré élevé grâce à la recherche de relations algébriques de bas degré, impliquant les bits de la clé (ou de l'initialisation du LFSR) et des bits de sortie.

3.1 Attaques algébriques sur les schémas par flot

Dans le cas des chiffrements par flots, on peut avoir un système d'équations surdéterminé (i.e un système avec un nombre d'équations indépendantes strictement plus grand que le nombre d'inconnues). Considérons le cas du générateur de filtrage, avec une partie linéaire (un LFSR) de taille N et avec une fonction Booléenne f à n variables comme fonction de filtrage ; il existe alors une permutation linéaire $L : \mathbb{F}_2^N \to \mathbb{F}_2^N$ et une application linéaire $L' : \mathbb{F}_2^N \to \mathbb{F}_2^n$ telles que, en notant $(u_1, \cdots, u_N)$ le vecteur d'initialisation du LFSR et $(s_i)_{i\geq 0}$ la suite pseudo-aléatoire fournie par le générateur, on ait pour chaque $i \geq 0$:

$$s_i = f(L' \circ L^i(u_1, \cdots, u_N))$$

Le nombre d'équations peut être plus grand que le nombre d'inconnues, ce qui réduit la complexité de la résolution du système en utilisant les bases de *Gröbner* (voir [29]), le calcul de ces bases peut se faire d'une manière efficace grâce aux algorithmes F4 et F5 de Jean-Charles Faugère, malheureusement il est difficile d'évaluer la complexité de ces algorithmes. On peut aussi résoudre le système d'équations en question par la technique plus élémentaire de la linéarisation du système (i.e l'obtention d'un système d'équations linéaires en remplaçant chaque monôme de degré plus grand que 1 par une nouvelle inconnue) ; le système d'équations obtenu a ainsi beaucoup plus d'inconnues et ne peut-être résolu. Toutefois, Courtois et Meier [26] ont eu une idée simple mais très efficace. Supposons qu'il existe deux fonctions $g \neq 0$ et h de bas degrés (disons de degrés au plus d) telles que $f \times g = h$ (où $f \times g$ désigne le produit de Hadamard, produit de f et g ayant un support égal à l'intersection des supports de f et de g, on omettra d'écrire $\times$ dans la suite). On a alors, pour chaque $i \geq 0$:

$$s_i g(L' \circ L^i(u_1, \cdots, u_N)) = h(L' \circ L^i(u_1, \cdots, u_N)).$$

Cette équation en $u_1, u_2, \cdots, u_N$ est de degré au plus d, puisque L, L' sont linéaires, et le système d'équations obtenu après linéarisation peut-être résolu par élimination de Gauss.

3.1.1 Attaque algébrique standard

Nous présentons dans cette section les détails de l'attaque algébrique standard (sur un schéma par flot) donnée par Courtois et Meier dans [26]. Supposons qu'on a une fonction de filtrage f (ou de combinaison) de haut degré. L'attaque algébrique contre le générateur filtré est réalisable (comme nous avons mentionné dans la section précédente) si

C1 Il existe une fonction $g \neq 0$ telle que $fg = h$, où h est une fonction non nulle de degré au plus d.

Cette condition est équivalente à

C2 Il existe une fonction $g \neq 0$ de degré au plus d telle que $fg = 0$ ou $(f \oplus 1)g = 0$.

Les conditions **C1** et **C2** sont en effet équivalentes,
Supposons que **C1** est satisfaite : $fg = h$ implique que $f^2g = fh$, c'est à dire (puisque $f^2 = f$) que $f(g \oplus h) = 0$, ce qui donne la condition **C2** si $g \neq h$ en remplaçant $g \oplus h$ par g, et si $g = h$ alors $fg = h$ est équivalente à $(f \oplus 1)g = 0$. L'implication inverse est évidente.

Ainsi, l'attaque algébrique standard repose sur la notion d'annulateur d'une fonction Booléenne.

Définition 3.1.1 (Annulateur)**.** *Soit f une fonction Booléenne à n variables. Toute autre fonction Booléenne à n variables et non nulle g telle que*

$$f(x)g(x) = 0, \forall x \in \mathbb{F}_2^n$$

est appelée annulateur de f.

On utilise généralement le scénario **C2** dans une attaque algébrique pour chercher des annulateurs de la fonction f. Supposons qu'on veut retrouver l'état initial du LFSR de longueur N du générateur filtré par la fonction f à n variables et qu'on dispose d'un annulateur g non nul de f. L'idée de l'attaque est alors basée sur le fait qu'à chaque temps t tel que l'on connaisse $s_t = f(L' \circ L^t(u_1, \cdots, u_N)) = 1$, on aura un système d'équations de la forme :

$$g(L' \circ L^t(u_1, \cdots, u_N)) = 0, t \geq 0 \tag{3.1}$$

Si g est de degré d et son ANF a la forme

$$a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{1 \leq i < j \leq n} a_{i,j} x_i x_j + \ldots + \sum_{1 \leq i_1 < i_2 < \ldots \leq n} a_{i_1 \ldots i_d} x_{i_1} \ldots x_{i_d}$$

Alors g est un annulateur de f si et seulement si $f(x) = 1$ implique que $g(x) = 0$. Ainsi, g est un annulateur de f si et seulement si les coefficients de son ANF satisfont un système de $w_H(f)$ équations linéaires homogènes à $\sum_{i=0}^{d} \binom{n}{i}$ inconnues (les coefficients des monômes de degrés au plus d).
Si on note $M_{f,d}$ la matrice à $w_H(f)$ lignes et $\sum_{i=0}^{d} \binom{n}{i}$ colonnes dont les entrées sont les valeurs prises par les monômes de degrés au plus d en les éléments du support de f, alors le rang de cette matrice $r_{f,d}$ satisfait $r_{f,d} \leq \min\{w_H(f), \sum_{i=0}^{d} \binom{n}{i}\}$, de plus :

1. Si $r_{f,d} = \sum_{i=0}^{d} \binom{n}{i}$, alors il n'existe pas d'annulateur non nul de f de degré au plus d,
2. Si $r_{f,d} < \sum_{i=0}^{d} \binom{n}{i}$, alors il existe des annulateurs non nuls de f de degrés au plus d. Il y a en fait $\sum_{i=0}^{d} \binom{n}{i} - r_{f,d}$ annulateurs linéairement indépendants de degrés au plus d.

Supposons que $r_{f,d-1} = \sum_{i=0}^{d-1} \binom{n}{i}$ (donc il n'existe pas d'annulateur non nul de f de degré au plus $d-1$), et $r_{f,d} < \sum_{i=0}^{d} \binom{n}{i}$, alors il existe des annulateurs non nuls de degré d.
Pour tout annulateur non nul de f de degré d, on a une équation de degré d en les variables $x_i, i = 1, \ldots, n$, cette équation est aussi de degré d en les variables $u_i, i = 1, \ldots, N$, puisque les applications L et L' (modélisants le registre) sont linéaires. Pour retrouver l'état initial du registre, il suffit de considérer un nombre suffisant d'annulateurs de degré d, et de résoudre le système d'équations en question en utilisant la méthode de linéarisation par exemple, (on linéarise un système en remplaçant chaque monôme de degré plus grand que 1 par une nouvelle variable).
La complexité de l'attaque algébrique standard est de l'ordre de D_N^3 où $D_N = \sum_{i=0}^{d} \binom{N}{i}$ et où N est la longueur du LFSR, remarquons que cette complexité est dominée par celle d'une élimination de Gauss, de plus, on aura besoin de connaitre $\mathcal{O}(D_N)$ bits de la suite chiffrante.

Les relations algébriques de bas degrés existent pour quelques chiffrements par flot qui étaient résistant à toutes les attaques connues précédemment, on peut citer parmi ces systèmes Toyocrypt et LILI-128, ces deux derniers ayant été attaqués par Courtois et Meier dans leurs article original [26].

Remarquons qu'aux points où f vaut 0, on utilise une méthode similaire pour chercher les annulateurs de la fonction $(1 + f)$ qui prend les valeurs opposées à celles de f. Willi Meier, Enes Pasalic et Claude Carlet ont introduit la notion d'immunité algébrique d'une fonction Booléenne dans leur article [46]. Nous étudierons plus en détail cette notion dans le chapitre 4.

Définition 3.1.2 (Immunité algébrique)**.** *L'immunité algébrique d'une fonction Booléenne f est le plus petit entier d tel que f ou $1 + f$ admette un annulateur non nul de degré d.*

Nous verrons au Chapitre 4 que l'immunité algébrique d'une fonction Booléenne à n variables est au plus égale à $\lceil n/2 \rceil$; de plus cette borne est atteinte.

Supposons qu'une fonction Booléenne f à n variables, d'immunité algébrique égale à $\lceil n/2 \rceil$ est utilisée comme fonction de filtrage d'un LFSR de longueur N avec $N \geq 2k$, où

k est la taille de la clé (cette condition est nécessaire sinon, le système serait vulnérable aux attaques dites " time-memory-data trade-off" [6]), alors (d'après [26]), la complexité d'une attaque algébrique utilisant une telle fonction f est d'ordre

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

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}. Si l'attaquant connaît plusieurs annulateurs indépendants de degré $\lceil n/2\rceil$, alors on doit accroître le nombre de variables. Dans la pratique, le nombre de variables doit être proche de 20 (mais cela peut poser alors un problème d'efficacité du système de chiffrement utilisée).

L'utilisation d'une fonction de filtrage f de haute Immunité algébrique ne garantie pas la résistance aux attaques algébriques, en effet, il existe une version plus évoluée appelée "Attaques algébriques rapides".

3.1.2 Attaques algébriques rapides

Ce type d'attaques a été présenté par Nicolas Courtois dans son article [23]. Cette appellation est liée au fait que les idées utilisées sont semblables et sa complexité ne peut pas être pire que la version standard de l'attaque. Cette attaque a été plus tard améliorée par Armknecht dans [1] et par Hawkes et Rose dans [33].
L'attaque algébrique rapide sur un générateur filtré par une fonction f à n variables est basée sur l'existence de deux fonctions Booléennes à n variables, g et h telles que l'on ait

$$f(x)g(x) = h(x) \qquad \forall x \in \mathbb{F}_2^n \tag{3.2}$$

Ceci nous amène à introduire une notion d'immunité aux attaques algébriques rapides :

Définition 3.1.3 (Immunité aux attaques 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 (3.2) telles que $d°g = e < d°h = r$ et $r + e < n$.*

La relation $fg = h$ implique que $fh = ffg = fg = h$, h est alors un annulateur de $1 + f$, si $h \neq 0$ alors son degré est au minimum égal à l'immunité algébrique de f. On en déduit qu'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.

L'idée de l'attaque est d'exploiter les relations de la forme $fg = h$ et d'autoriser un degré plus grand pour h que l'immunité algébrique de f en espérant qu'il existe des fonctions g de degré plus faible. Remarquons que le raisonnement avec $1 + f$ n'aboutit à rien puisqu'on a $(1 + f)g = h$ donc $fg = h + g$ et comme le degré de g est plus petit que celui de h, on retrouve le même type de relations avec différentes fonctions h.

La relation (3.2) donne pour chaque top d'horloge l'équation algébrique suivante :

$$h(L' \circ L^t(u_1, \cdots, u_N)) = s_t g(L' \circ L^t(u_1, \cdots, u_N)) \tag{3.3}$$

où $(u_1, \cdots, u_N)$ est la clé secrète. En faisant une combinaison linéaire de plusieurs relations de ce type, on peut éliminer (en utilisant l'algorithme de Berlekamp-Massey) les monômes de degré plus grand que e dans l'ANF de h. Remarquons qu'il n'est pas nécessaire de connaître les valeurs de s_t pour chercher de telles combinaisons. On cherche alors des coefficients p_i, pour i un entier allant de 0 à T, on choisit T le plus petit possible tel que tous les monômes de degré plus grand que e disparaissent dans la somme :

$$h^*(L' \circ L(u_1, \cdots, u_N)) = \sum_{i=0}^{T} p_i h(L' \circ L^{i+t}(u_1, \cdots, u_N)) \tag{3.4}$$

On peut voir les coefficients p_i comme ceux d'un polynôme $p(X)$ de degré T. La combinaison linéaire des relations (3.3) selon les coefficients p_i de $p(X)$ donne de nouvelles équations de degré e :

$$h^*(L' \circ L^t(u_1, \cdots, u_N)) = g^*(L' \circ L^t(u_1, \cdots, u_N)), \qquad \forall t \geq 0 \tag{3.5}$$

avec g^* la fonction définie par

$$g^*(L' \circ L^t(u_1, \cdots, u_N)) = \sum_{i=0}^{T} p_i s_{t+i} g(L' \circ L^{t+i}(u_1, \cdots, u_N))$$

La résolution d'un tel système se ramène à une résolution d'un système linéaire (à seulement $\sum_{i=0}^{e} \binom{N}{i}$ inconnues) par linéarisation.
Le polynôme $g^*(X)$ dépend de la suite chiffrante et il ne peut être calculé au préalable.

En résumé, l'attaque algébrique rapide se fait en quatre étapes, chacune de ces étapes nécessite en fait beaucoup de calcul. Une petite description des étapes avec les complexité correspondantes est dans ce qui suit :

-Recherche de relations. On cherche des fonctions g et h de bas degrés e et r respectivement ($e < r$), telles que $fg = h$. Si on note $L_r = \sum_{i=0}^{r} \binom{n}{i}$ et $L_e = \sum_{i=0}^{e} \binom{n}{i}$, alors la recherche de telles fonctions g et h (quand elles existent) nécessite la résolution d'un système à $L_r + L_e$ équations, avec une complexité de $\mathcal{O}((L_r + L_e)^3)$.

-Pré-calculs. On cherche dans cette étape des relations linéaires particulières qui permettent d'éliminer les monômes de degrés supérieurs à e dans les équations $fg = h$ (ce qui nécessite la connaissance de plus de bits de la suite chiffrante que d'une attaque algébrique standard), la recherche des coefficients de tels combinaisons linéaires se fait en utilisant l'algorithme de Berlekamp-Massey. Ceci nécessite la connaissance de $R_N + E_N = \sum_{i=0}^{r} \binom{N}{i} + \sum_{i=0}^{e} \binom{N}{i}$ bits de la suite chiffrante, où N est la taille du LFSR. La complexité de cette étape est la même que celle de l'algorithme de Berlecamp-Massey, elle est de l'ordre de $\mathcal{O}(R_N \log^2(R_N))$, voir [33].

- Substitution. Dans cette étape on élimine les monômes de degrés supérieurs à e. Ceci a une complexité naturelle de l'ordre de $\mathcal{O}(E_N^2 R_N)$, avec $E_N = \sum_{i=0}^{e} \binom{N}{i}$, mais en utilisant la transformation de Fourrier discrète, cette complexité peut être ramenée à $\mathcal{O}(E_N R_N \log(R_N))$, voir [33].

- Résolution. On résout le système obtenu avec E_N équations linéaires en $\mathcal{O}(E_N^3)$.

3.1.3 Attaques basées sur les fonctions augmentées.

Les générateurs pseudo-aléatoires peuvent être aussi vulnérables aux attaques sur les fonctions augmentées, ce type d'attaque a été introduit par Fischer et Meier dans [31]. On considère dans cette attaque plusieurs bits de sortie à la fois (m bits) au lieu d'un seul bit comme dans l'attaque algébrique standard, ce qui peut réduire - dans certain cas - le degré des équations cherchées. Il est important de signaler que parfois l'attaque basée sur les fonctions augmentées est plus efficace qu'une attaque algébrique standard, mais son efficacité dépend du choix de la partie linéaire du générateur pseudo aléatoire, contrairement aux attaques algébriques classiques (rapide ou non). On considère donc que la résistance à cette attaque ne fait pas partie des critères de conception de la fonction de filtrage ou de combinaison..

Définition 3.1.4. *Considérons un chiffrement par flot avec état intérieur de n bits, une fonction de transition L, et une fonction Booléenne f à n variables qui donne un bit dans chaque itération. Alors, la fonction augmentée S_m ($m \leq n$) est définie par*

$$\begin{aligned} S_m : \quad \mathbb{F}_2^n &\to \mathbb{F}_2^m \\ x &\mapsto (f(x), f(L(x)), \ldots, f(L^{m-1}(x))) \end{aligned}$$

La fonction de transition L de $\mathbb{F}_2^n$ dans $\mathbb{F}_2^n$ peut être linéaire (c'est le cas des modèles combiné et filtré), ou non linéaire (comme par exemple dans le cas d'un candidat à eSTREAM qui s'appelle Trivium). Les n bits d'entrée sont l'image par L des bits

correspondant à l'état intérieur du registre, et les m bits de sortie $(z_1, z_2, \ldots, z_m) = z$ correspondent aux bits connus de la suite chiffrante. Le but est de trouver l'état initial du registre.

Description de l'attaque

On cherche pour chaque vecteur de sortie z à m bits fixé des relations algébriques de bas degré d_z et de la forme $F_z(x) = 0$. Pour cela, on construit pour chaque z fixé, une matrice M_z (avec entrées dans $\mathbb{F}_2$) à $N_l = |S_m^{-1}(z)|$ lignes et $N_c = \sum_{i=0}^{d_z} \binom{n}{i}$ colonnes. Chaque ligne correspond à une entrée x telle que $S_m(x) = z$, et chaque colonne correspond aux évaluations d'un monôme de degré au plus d_z en les variables x_i (avec un ordre fixé) dans les entrées x correspondantes aux lignes, la matrice M_z a donc $N_l = |S_m^{-1}(z)| \approx 2^{n-m}$ lignes (si S_m est approximativement équilibrée) et $N_c = D_z = \sum_{i=0}^{d_z} \binom{n}{i}$ colonnes. Si le nombre de colonnes N_c est strictement plus grand que le nombre de lignes N_l, alors il existe des annulateurs de $S_m^{-1}(z)$ de degré d_z. Le rang de M_z détermine le nombre de solutions, voir [24, 25].

Proposition 3.1.1. *[31]. Considérons une boite-S, $S_m : \mathbb{F}_2^n \to \mathbb{F}_2^m$. Alors, pour tout z dans $\mathbb{F}_2^m$, le nombre d'équations (indépendantes) de degrés d_z est $N_c - rang(M_z)$. Par conséquent, une condition suffisante pour l'existence d'une équation (non triviale) est $N_c > 2^{n-m}$ si S_m est équilibrée.*

La condition suffisante de la Proposition 3.1.1 est équivalente à

$$\log_2(N_c) = \log_2\left(\sum_{i=0}^{d_z}\binom{n}{i}\right) > \log_2(2^{n-m}) = n - m, \text{ ce qui est équivalent à}$$

$$m > m_0 = n - \log_2\left(\sum_{i=0}^{d_z}\binom{n}{i}\right) \tag{3.6}$$

La taille du bloc de sortie m est ainsi un paramètre et $\lceil m_0 \rceil$ est une borne inférieure sur m pour qu'il existe des équations de degré d_z. Si la taille du bloc m croît alors le nombre d'images réciproques de z par S_m décroit, ce qui réduit suffisamment le nombre de lignes dans M_z pour avoir moins d'équations. Il est possible de trouver des équations de degrés d_z pour chaque z à m bits si la condition (3.6) est satisfaite. Une sortie z de m bits est dite *faible*, s'il existe des équations non triviales de degré d_z pour $S_m^{-1}(z) \gg \sum_{i=0}^{d_z} \binom{n}{i}$ (la sortie est fortement non équilibrée), ce qui correspond à $m \ll m_0$.

On peut se demander s'il existe une méthode dédiée pour calculer les images réciproques des sorties à m bits dans le contexte des fonctions augmentées. L'équation $S_m(x) = z$

peut être traduite par un système d'équations non linéaires. Si le nombre d'équations dans ce système est proche du nombre d'inconnues alors la résolution est en général NP-difficile. Néanmoins, il existe des chiffrements par flot ayant une structure simple, ce qui permet de résoudre le système d'équations en question. Le calcul efficace des images réciproques des blocs de sortie à m bits est appelé *Echantiannage*"*Sampling*" (en anglais). Il existe des chiffrements qui sont vulnérables aux *Echantillonnages*, voir [3, 6]. On peut trouver différent types d'*Echantillonnage*, qui dépendent de la structure de la fonction augmentée : "inversion filtrée, Echantillonnage linéaire, voir [31].

En résumé : On prend à chaque instant t un bloc de m bits de la suite chiffrante, on peut faire un calcul "on line" des images réciproques et trouver R équations de degré d_z. Dans le cas où la fonction de transition L est linéaire, chaque équation peut être transformée en une équation de degré d_z en l'état initial x. Dans le cas d'une fonction de transition non linéaire, le degré des équations croît avec le temps. Néanmoins, la partie non linéaire est parfois très simple, tel que des équations de différents instants peuvent être combinées efficacement.

Les paramètres de l'attaque sont le degré des équations d_z, la taille du bloc de la sortie m, et le nombre des images réciproques calculées. L'attaque est peut être efficace si :
1. Il existe suffisamment d'équations pour une petite taille m du bloc de sortie,
2. Il est possible de calculer efficacement les images réciproques pour ces bloc de taille m.

3.2 Attaques algébriques sur les schémas par blocs

L'idée de monter les attaques algébriques sur les chiffrements par bloc a été proposée par Courtois et Pieprzyk [25]. Il n'est pas toujours possible de monter une attaque sur un chiffrement par bloc concrètement puisque l'attaquant est obligé de résoudre un système d'équations algébriques qui est (contrairement au cas des chiffrements par flot) non sur-détérminé. L'attaque se fait en exploitant l'existence de relations de bas degré entre les bits d'entrée et les bits de sortie des boites-S utilisées dans le système de chiffrement, c'est à dire, pour une boite-S à n entrées et m sorties S, on cherche une fonction F de bas degré d telle que

$$F(x_1, x_2, \ldots, x_n, S_1(x_1, x_2, \ldots, x_n), \ldots, S_m(x_1, x_2, \ldots, x_n)) = 0$$

où $S_1, S_2, \ldots, S_m$ sont les fonctions Booléennes coordonnées de la boite-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_S (ou encore du graphe) de la fonction S. La fonction caractéristique de S est la fonction Booléenne à $(n+m)$ variables définie par

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

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

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

Considérons la matrice M construite pour chercher des relations de degrés d : la matrice a 2^n lignes (correspondant aux éléments x de $\mathbb{F}_2^n$) et $\sum_{i=0}^{d} \binom{n+m}{i}$ colonnes (correspondant aux monômes en x_i, y_j, pris dans un certain ordre, tels que $S_j(x) = y_j$, $1 \leq i \leq n, 1 \leq j \leq m$, et de degré au plus d), les entrées de la matrice sont les évaluations des monômes (colonnes) en les éléments x (lignes). On remarque que, comme dans le cas d'une attaque sur les schémas par flot, il est possible de trouver des relations de degré d si le nombre de lignes de la matrice M est strictement plus petit que le nombre de colonnes, c'est à dire

$$\sum_{i=0}^{d} \binom{n+m}{i} > 2^n.$$

On remarque ainsi qu'on peut trouver des relations de bas degré si le nombre de variables de la boite-S est petit, qui est généralement le cas. Par exemple pour toute boite-S à 8 entrées et 8 sorties (ce que la boite-S de AES), 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 boite-S. L'attaque contre l'AES a montré mieux, l'existence de 39 relations quadratique.

Dans les attaques algébriques sur les chiffrements par bloc, les sorties des boites-S sont usuellement inconnues puisqu'elles sont les variables intermédiaires correspondant aux bits de sortie obtenues après chaque tour. Le degré de ces relations en ces variables doit être alors bas. Mais, dans d'autre cas, les sorties de la boite-S sont connues. Cette situation est le cas par exemple si la boite-S est utilisée comme fonction de filtrage ou de combinaison pour produire plusieurs bits de sortie dans les chiffrements basés sur les LFSRs, cette technique est utilisée pour accélérer l'opération de chiffrement et de déchiffrement, mais elle augmente la vulnérabilité potentielle des fonctions Booléennes. L'attaque contre ce type de système se fait (pour une (n, m)-fonction F, de filtrage d'un LFSR de taille N ou de combinaison de n LFSRs) en cherchant pour tout $z = (z_1, z_2, \ldots, z_m)$ fixé dans $\mathbb{F}_2^m$, des relations des bas degrés (disons de degré au plus d) de la forme

$$F(x_1, x_2, \ldots, x_n) = (z_1, z_2, \ldots, z_m)$$

Pour cela, on construit pour chaque z fixé, une matrice M_z à $|F^{-1}(z)|$ lignes (correspondant aux images réciproques de z par F) et $\sum_{i=0}^{d} \binom{n}{i}$ colonnes (correspondant aux

monômes en x_i de degrés au plus d, pris dans un certain ordre), les entrées de la matrice M_z sont les évaluations des monômes (colonnes) en les éléments x (lignes). On essaye ensuite de résoudre le système $M_z G = 0$, avec G est le vecteur constitué des coefficients des annulateurs de $F^{-1}(z)$ cherché.
Pour une fonction F à n entrées et m sorties, il existe des relations de degré d (en les variables d'entré) si

$$\sum_{i=0}^{d} \binom{n}{i} > 2^{n-m}.$$

En effet, on sait qu'il existe z tel que $|F^{-1}(z)| \leq 2^{n-m}$, si la dernière condition est satisfaite alors $|F^{-1}(z)| \leq 2^{n-m} < \sum_{i=0}^{d} \binom{n}{i}$, c'est à dire le nombre de lignes de la matrice M_z est strictement plus petit que le nombre de ses colonnes ce qui implique l'existence d'un annulateur non nulle de degré d.
Si le degré d est petit, on peut résoudre le système d'équations obtenues et trouver les bits d'entrées de la fonction F.

3.2.1 Comparaison avec l'attaque sur les chiffrements à flot

On va montrer dans ce qui suit que dans le cas d'une attaque algébrique sur un chiffrement utilisant des fonctions Booléennes à plusieurs sorties (fonctions vectorielles), il n'y a plus équivalence entre les deux conditions **C1** et **C2** vues dans l'attaque algébrique standard sur les chiffrements à flot, c'est à dire qu'il existe une fonction à m sorties $F = (f_1, f_2, \cdots, f_m)$ telle que :

- Il existe une fonction $G = (g_1, g_2, \cdots, g_m)$ quelconque, pour laquelle on a : $F \cdot G = f_1 g_1 + \cdots + f_m g_m = h$, où h est une fonction non nulle de bas degré et,
- Il n'existe aucune fonction $G^b = (g_1^b, g_2^b, \cdots, g_m^b)$ de bas degré, pour laquelle on a : $F \cdot G^b = f_1 g_1^b + \cdots + f_m g_m^b = h^b$, où h^b est une fonction non nulle de bas degré.

Il suffit de montrer cette non-équivalence dans le cas d'une fonction $F = (f_1, f_2)$ à deux sorties. Supposons qu'il existe une fonction $G = (g_1, g_2)$ quelconque telle que :

$$f_1 g_1 + f_2 g_2 = h \tag{3.7}$$

où h est une fonction non nulle de bas degré égale à d.
Remarquons d'abord que :

1. Si $f_1 = f_2 = f$ alors (3.7) s'écrit $f(g_1 + g_2) = h$, on serait ainsi comme dans le cas Booléen (dans la condition **C1** si $d^\circ g_1 \neq d^\circ g_2$).
2. Si f_1 (resp. f_2) a un multiple nul, et f_2 (resp. f_1) a un multiple non nul de bas degré h, alors on a $f_2 g_2 = h$ (resp. $f_1 g_1 = h$) ; on serait aussi dans le cas Booléen.

Notons $f_1g_1 = h_1$ et $f_2g_2 = h_2$; puisque leurs somme est égale à h qui est de bas degré égal à d, on a deux cas possibles :

- h_1, h_2 sont de bas degrés (au plus égaux à d) ; dans ce cas, il existe $g_1^b \neq 0, g_2^b \neq 0$ et h_1^b, h_2^b de bas degrés tous les quatre telles que $f_1g_1^b + f_2g_2^b = h_1^b + h_2^b = h^b$; où h^b est une fonction de bas degré ; ou :
- h_1, h_2 sont de hauts degrés, et l'ensemble des monômes de degrés strictement supérieurs à d dans l'ANF de $f_1g_1 = h_1$ est égal à l'ensemble des monômes de degrés strictement supérieurs à d dans l'ANF de $f_2g_2 = h_2$.

Remarque 3.2.1. *Si h_1, h_2 sont des multiples de f_1, f_2 respectivement, telles qu'il existe une fonction g vérifiant : $f_1g = h_1, f_2g = h_2$, alors (3.7) s'écrit $(f_1 + f_2)g = h$, et on aura comme dans le cas Booléen le résultat.*

Il faut chercher donc f_1, f_2 pour lesquelles on serait dans le dernier cas.
Soit $F = (f_1, f_2)$ telle que $f_2 = f_1 + 1$ et f_1 n'a pas de multiple de bas degré. Pour une telle fonction :

- Il existe g_1, g_2 quelconques et $h \neq 0$ de bas degré telles que $f_1g_1 + f_2g_2 = h$.
  En effet, il suffit de prendre $g_1 = f_1, g_2 = f_2$; pour ce choix, on a : $f_1g_1 + f_2g_2 = f_1^2 + f_1^2 + 1 = f_1 + f_1 + 1 = 1$.

- Il n'existe pas de fonctions $(g_1^b, g_2^b) \neq (0, 0), g_1^b \neq g_2^b$ et h^b de bas degrés toutes les trois telles que $f_1g_1^b + f_2g_2^b = h^b$.
  En effet : Soit g_1^b, g_2^b deux fonctions Booléennes de bas degrés telles que $(g_1^b, g_2^b) \neq (0, 0), g_1^b \neq g_2^b$. On a $f_1g_1^b + f_2g_2^b = f_1(g_1^b + g_2^b) + g_2^b$ qui est toujours de haut degré puisque f_1 n'a pas de multiple de bas degré par hypothèse.

Plus généralement, Soit $F = (f_1, f_2)$, avec $f_1 + f_2 = f_0$ est de bas degré, $d^\circ f_0 = k$ et f_1 est une fonction n'admettant pas de multiple de bas degré.
Posons : $f_1 = f_1^{>k} + f_1^{\leq k}$, $f_2 = f_2^{>k} + f_2^{\leq k} = f_1^{>k} + f_2^{\leq k}$ donc $f_0 = f_1^{\leq k} + f_2^{\leq k}$

- Il existe g_1, g_2 quelconques et $h \neq 0$ de bas degrés telles que $f_1g_1 + f_2g_2 = h$.
  En effet, il suffit de prendre $g_1 = f_1, g_2 = f_2$; pour ce choix, on a :
  $f_1g_1 + f_2g_2 = f_1^2 + f_2^2 = f_1 + f_2 = f_0$, qui est de bas degré.

- Il n'existe pas de fonctions $(g_1^b, g_2^b) \neq (0, 0), g_1^b \neq g_2^b$ et h^b de bas degrés toutes les trois telles que $f_1g_1^b + f_2g_2^b = h^b$.
  En effet : Soit g_1^b, g_2^b deux fonctions Booléennes de bas degrés telles que $(g_1^b, g_2^b) \neq (0, 0), g_1^b \neq g_2^b$; On a :
  $f_1g_1^b + f_2g_2^b = (f_1^{>k} + f_1^{\leq k})g_1^b + (f_1^{>k} + f_2^{\leq k})g_2^b = f_1^{>k}(g_1^b + g_2^b) + f_1^{\leq k}g_1^b + f_2^{\leq k}g_2^b$ qui est toujours de haut degré puisque f_1 n'a pas de multiple de bas degré par hypothèse et les fonctions $f_i^{\leq k}$, $i = 1, 2$; sont de bas degrés.

On en déduit que l'attaque algébrique standard sur les chiffrements par flot n'est pas applicable aux chiffrements par bloc.

Chapitre 4

Immunité algébrique d'une fonction Booléenne

Les attaques algébriques, introduites par N. Courtois en 2003 [26] contre les schémas de chiffrement à flots ont fait l'objet d'études nombreuses ces dernières années. Un critère important et nécessaire pour avoir une résistance contre ces attaques est que la fonction (de combinaison ou de filtrage) ait une haute immunité algébrique. On s'intéresse dans ce chapitre à l'étude de ce critère. Pour commencer on rappelle quelque notions préliminaires.

Définition 4.0.1 (Immunité algébrique d'un sous-ensemble)**.** *Soit H un sous-ensemble de $\mathbb{F}_2^n$. Toute fonction $p \neq 0$, nulle sur H est appelée annulateur de H. L'immunité algébrique de H, noté $AI(H)$, est le degré minimal d de tous les annulateurs non nuls de H.*

$$AI(H) = \min\{\deg(p) : 0 \neq p \in \mathbb{B}_n, p(x) = 0; \forall x \in H\} \tag{4.1}$$

4.1 Immunité algébrique d'une fonction Booléenne

Les fonctions Booléennes utilisées dans les chiffrements à flot doivent être de bonne immunité algébrique pour permettre au chiffrement de résister aux attaques algébriques standard.

Soit f une fonction Booléenne à n variables, l'ensemble noté $AN(f)$ des fonctions non-nulles g vérifiant $fg = 0$ est l'ensemble des annulateurs de la fonction f.

Définition 4.1.1. *L'immunité algébrique de f est le plus petit entier d tel que f ou $1+f$ admet un annulateur de degré d.*

$$AI(f) \quad = \quad \min\{\deg(g) : 0 \neq g \in \mathbb{B}_n, fg = 0 \text{ ou } (f+1)g = 0\} \tag{4.2}$$

C'est donc le minimum de $AI(supp(f))$ et $AI(supp(f+1))$.

Remarquons que $AI(f) \leq \deg(f)$ puisque $1+f$ est un annulateur de f, en effet, $(1+f)f = f+f^2 = f+f = 0$. On verra dans la section suivante qu'il existe une borne meilleur.

4.1.1 Immunité algébrique et codes de Reed-Muller

Nous allons voir maintenant que la notion d'immunité algébrique est intimement liée aux codes de Reed-Muller et à leur comportement sur le canal à effacements.

Canal à effacements

Le canal à effacements est un canal tel qu'après transmission d'un mot de code, certaines positions sont "effacées". C'est-à-dire que, contrairement au canal binaire à erreurs, la position des erreurs (les effacements) est connue, on ne sait juste pas quel symbole il y avait à cette place.
Pour un code linéaire sur le canal à effacements, la probabilité d'erreur (la probabilité que le décodage ne soit pas unique en supposant que les $N-w$ positions effacées soient choisies de manière uniforme, N étant la longueur du code et w le nombre de positions connues) est indépendante du mot transmis et ne dépend que des positions d'effacements que l'on va désigner par motif d'effacements :

Définition 4.1.2. *(Motif d'effacements). 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 de $\mathbb{F}_2^N$.*

Décoder un code linéaire au maximum de vraisemblance sur le canal à effacements est beaucoup plus simple que dans le cas du canal binaire à erreurs. Il suffit en effet de résoudre un simple système linéaire. Ainsi, chaque position connue nous donne une équation linéaire que doivent satisfaire les k bits d'information qui définissent le mot

transmis. Cette équation est donnée, pour une position non effacée d'indice i, par la i-me colonne M_i de la matrice génératrice M du code. Plus formellement, si la position i du mot reçu n'est pas effacée et vaut y_i, on a l'équation suivante sur les k bits d'information $b_1, \ldots, b_k$ du mot émis :

$$(b_1, \ldots, b_k) \cdot M_i = y_i \tag{4.3}$$

On remarque en particulier que si l'on note le système à résoudre sous forme matricielle, ce n'est rien d'autre qu'une sous matrice de la matrice génératrice du code.
Le décodage est unique si et seulement s'il y a en dehors du support du motif d'effacements k positions qui donnent des équations indépendantes. De tels ensembles de k positions forment ce que l'on appelle un ensemble d'information :

Définition 4.1.3. *(Ensemble d'information). Un ensemble d'information $\mathcal{B}$ est la donnée de k positions dans un mot de code qui conduisent à des équations de la forme (4.3) indépendantes.*

Si w positions ne sont pas effacées, on ne dispose que de w équations linéaires, dans ce cas, un décodage unique n'est possible qu'à la condition nécessaire que la dimension k du code soit plus petite que w.

Lien avec l'immunité algébrique

Le résultat essentiel de cette section montre que l'immunité algébrique d'une fonction Booléenne à n variables ($N = 2^n$) est reliée au comportement sur le canal à effacements de $RM(r, n)$ en présence du motif d'effacements égal à f ou à $1 + f$:

Remarque 4.1.1. *(Lien avec l'immunité algébrique). Chercher les annulateurs de degré au plus r d'une fonction Booléenne f est la même chose que décoder $RM(r, n)$ en présence d'un motif d'effacements égal au vecteur des valeurs de $1 + f$. Pour ce motif, l'ensemble des annulateurs est alors exactement l'ensemble $\mathcal{I}$ défini dans la section précédente.*

En effet, supposons que l'on cherche des annulateurs de degrés au plus r d'une fonction Booléenne f. On recherche donc une fonction g qui doit prendre la valeur 0 en tous les points où f vaut 1. Ce problème peut se voir comme la recherche d'un mot de $RM(r, n)$ (associé à g) qui aurait pu être transmis lorsque l'on reçoit le mot tout à 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 $\mathcal{I}$ défini plus haut lorsque le motif d'effacements est le support de f.

Dans le cas des codes de Reed-Muller, l'équation linéaire (4.3) associée à une position x sur les k coefficients de l'ANF d'une fonction Booléenne f de degré au plus r découle directement de la transformée de Möbius binaire :

$$f(x) = \sum_{u \preceq x} f_u, |u| \leq r, \tag{4.4}$$

où : $u \preceq x$ signifie que $supp(u) \subseteq supp(x)$.

4.2 Calcul de l'immunité algébrique

On se donne une fonction booléenne f à n variables ; le problème est de trouver son immunité algébrique. Nous supposerons que la fonction f est donnée sous la forme de son vecteur des valeurs (la dernière colonne de sa table de vérité). L'approche la plus efficace actuellement repose sur de l'algèbre linéaire.

4.2.1 Utilisation de l'algèbre linéaire

Nous détaillons ici comment ramener le problème du calcul de l'immunité algébrique à un simple problème d'algèbre linéaire. En fait, nous nous intéresserons au problème de calculer les annulateurs de degré au plus r d'une fonction f donnée. En faisant varier r et en considérant aussi $1+f$, l'immunité algébrique peut s'en déduire facilement. Soit f de poids de Hamming w. On cherche des fonctions g de degré au plus r telles que pour tout x de $\mathbb{F}_2^n$, $f(x)g(x)$ est égal à 0. On a vu que de telles fonctions g sont entièrement représentées par $k = \sum_{i=0}^{r} \binom{n}{i}$ coefficients dans $\mathbb{F}_2$, ce sont les k coefficients $(g_u)_{u \in \mathbb{F}_2^n, |u| \leq r}$ de l'ANF de g. Pour tout point x tel que $f(x)$ vaut 1, $g(x)$ doit nécessairement valoir 0. Rappelons que g est un annulateur de f si et seulement si les coefficients de son ANF g_u, satisfont un système de $w_H(f) = w$ équations linéaires homogènes à $\sum_{i=0}^{r} \binom{n}{i}$ inconnues. En utilisant la transformée de Möbius binaire, pour un tel x on obtient une équation linéaire en les coefficients de g :

$$g(x) = \sum_{u \preceq x} g_u = 0, |u| \leq r. \tag{4.5}$$

En regroupant toutes les équations de la forme de l'équation (4.5) pour les différents éléments x dans le support de f, on obtient un système linéaire de w équations à k inconnues que l'on peut mettre sous la forme

$$M_1 \overline{g} = 0 \tag{4.6}$$

Le vecteur $\overline{g}$ contient les k coefficients de g qui sont ici des inconnues que l'on a arrangées de haut en bas en utilisant l'ordre usuel sur les éléments u de $\mathbb{F}_2^n$ de poids au plus r. Nous indexerons ces monômes de degrés au plus r par $u_1, \ldots, u_k, k = \sum_{i=0}^{r} \binom{n}{i}$. Chaque ligne de la matrice M_1 correspond à l'équation (4.5) associée à un élément x tel que $f(x)$ vaut 1. Les annulateurs de f sont alors les fonctions g de coefficients arrangés dans $\overline{g}$ qui vérifient (4.6).
Comme on l'a vu dans la section 4.1.1 c'est exactement le problème du décodage des codes de Reed-Muller d'ordre r et à n variables sur le canal à effacements. Le produit matrice-vecteur de M_1 par $\overline{g}$ correspond à une évaluation d'une fonction de degré au plus r de coefficients d'ANF donnés par $\overline{g}$ aux points tels que $f(x) = 1$.

Définition 4.2.1. *(Matrice d'évaluation). La matrice d'évaluation d'une fonction de degré r aux points $x^{(1)}, \ldots, x^{(L)}$ de $\mathbb{F}_2^n$ est la matrice $L \times k$ que nous noterons $V^r_{\{x^{(1)},\ldots,x^{(L)}\}}$ définie par*

$$V^r_{\{x^{(1)},\ldots,x^{(L)}\}} = (u_j(x^{(i)}))_{i\in[1,L],j\in[1,k]} = (\text{ Evaluations des monômes } u_j \text{ dans les éléments } x^i).$$

Cette matrice est telle que

$$V^r_{\{x^{(1)},\ldots,x^{(L)}\}} \begin{pmatrix} g_{u_1} \\ \vdots \\ g_{u_k} \end{pmatrix} = \begin{pmatrix} g(x^{(1)}) \\ \vdots \\ g(x^{(k)}) \end{pmatrix}$$

où g est la fonction Booléenne de degré r et de coefficients de son ANF $g_{u_1}, \ldots, g_{u_k}$.

Le nom "matrice d'évaluation" provient du fait qu'un produit matrice-vecteur est égal à l'évaluation d'une fonction Booléenne en certains points de $\mathbb{F}_2^n$. Cette propriété découle directement de la transformée de Möbius, et l'on voit que la ligne i de la matrice code est en fait l'équation (4.5) associée à $x^{(i)}$. Finalement, la matrice M_1 est la matrice

$$M_1 = V^r_{\{x\in\mathbb{F}_2^n, f(x)=1\}}.$$

Si la matrice $V^r_{\{x\in\mathbb{F}_2^n, f(x)=1\}}$ est de rang plein alors f n'admet pas d'annulateur non trivial de degré au plus r. Dans le cas contraire, les annulateurs sont directement donnés par les éléments non nuls du noyau de cette matrice.

Le calcul de l'immunité algébrique se ramène donc à la résolution d'un système linéaire. C'est l'approche retenue par tous les algorithmes efficaces de calcul de l'immunité algébrique qui exploitent la structure particulière de ce système pour le résoudre plus rapidement qu'avec une simple élimination Gaussienne en $O(wk^2)$.

4.2.2 Fonctions Booléennes d'immunité algébrique maximale

Rappelons que la recherche d'annulateurs de degré d d'une fonction Booléenne f à n variables se ramène à la résolution d'un système linéaire de la forme $M_1 G = 0_k$, où 0_k est le vecteur nul de $\mathbb{F}_2^k$, G est le vecteur constitué des coefficients de l'ANF des annulateurs cherchés, et M_1 est la matrice à $w_H(f)$ lignes et $k = \sum_{i=0}^{d} \binom{n}{i}$ colonnes décrite ci-dessous. On rappelle que k est égal au nombre de monômes de degrés au plus d en les n variables booléennes $x_1, x_2, \ldots, x_n$ de f. Chaque ligne de la matrice M_1 correspond à une variable $x^{(i)} = (x_1^{(i)}, \ldots, x_n^{(i)})$ appartenant au support de f et chaque colonne correspond à un monôme de degré au plus d, les entrées de la matrice M_1 sont les évaluations de ces monômes aux points $x^{(i)}$. Si le noyau de la matrice M_1 n'est pas réduit à zéro, alors il existe des annulateurs de degré d, il suffit pour cela que le nombre de lignes (d'équations) soit strictement inférieur au nombre de colonnes (d'inconnues), c'est à dire qu'il suffit que $w_H(f) < \sum_{i=0}^{d} \binom{n}{i}$.

Immunité algébrique maximale d'une fonction Booléenne

Le résultat suivant donne la valeur maximale de l'immunité algébrique d'une fonction Booléenne.

Proposition 4.2.1. *L'immunité algébrique maximale d'une fonction Booléenne à n variables est égal à $\lceil n/2 \rceil$.*

Ainsi, si f est une fonction Booléenne à n variables, alors,

$$AI(f) \leq \max\{\deg(f), \lceil n/2 \rceil\}$$

Démonstration. Nous avons vu qu'il existe des annulateurs de f de degré d si $w_H(f) < \sum_{i=0}^{d} \binom{n}{i}$. Comme l'immunité algébrique de f est le degré minimal de tous les annulateurs de f et de $1+f$, alors si on a à la fois $w_H(f) < \sum_{i=0}^{d} \binom{n}{i}$ et $w_H(1+f) = 2^n - w_H(f) < \sum_{i=0}^{d} \binom{n}{i}$ (c'est le cas si f est équilibrée, on a $w_H(f) = w_H(1+f) = 2^{n-1}$), 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 on a

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

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

4.3 Construction de fonctions d'immunité algébrique maximale

La fonction majorité est l'exemple le plus connu de fonction d'immunité algébrique maximale. La fonction majorité à n variables est une fonction symétrique qui vaut 1 si et seulement si son entrée est de poids de Hamming strictement supérieur à $n/2$.

Théorème 4.3.1. *La fonction majorité est d'immunité algébrique maximale.*

Démonstration. 1- Définissons deux fonctions f_1, f_2 par

$$f_1(x_1,\ldots,x_n) = \begin{cases} 0, \text{ si } w_H(x_1,\ldots,x_n) \geq \lceil n/2 \rceil \\ 1, \text{ si } w_H(x_1,\ldots,x_n) < \lceil n/2 \rceil \end{cases}$$

$$f_2(x_1,\ldots,x_n) = \begin{cases} 0, \text{ si } w_H(x_1,\ldots,x_n) \leq \lceil n/2 \rceil \\ 1, \text{ si } w_H(x_1,\ldots,x_n) > \lceil n/2 \rceil \end{cases}$$

Montrons que f_1 et f_2 n'ont pas d'annulateurs de degré $< \lceil n/2 \rceil$.
Supposons que g est un annulateur non nul de f_1 et de degré $< \lceil n/2 \rceil$ de la forme

$$a_0 + \sum_{i=0}^{n} a_i x_i + \cdots + \sum_{1 \leq i_1 < \ldots < i_{\lceil n/2 \rceil - 1} \leq n} a_{i_1,\ldots,i_{\lceil n/2 \rceil - 1}} x_{i_1} \cdots x_{i_{\lceil n/2 \rceil - 1}},$$

où les coefficients a_u sont dans $\{0,1\}$ non tous nuls. Comme g est un annulateur de f_1, $g(x) = 0$ quand $f_1(x) = 1$. Ainsi résolvons le système d'équations linéaires homogènes (en considérant a_u comme les inconnues) formé par les équations $g(x) = 0$ quand $f_1(x) = 1$, ce système a nécessairement une solution non triviale en les a_u.

On sait d'après la transformée de Möbius binaire (la relation reliant les valeurs d'une fonction Booléenne avec les coefficients de son ANF) que $a_u = \sum_{x \preceq u} g(x)$ (voir [13]), où $x \preceq u$ signifie que $supp(x) \subseteq supp(u)$. Soit alors u un vecteur de poids $w_H(u)$ strictement inférieur à $\lceil n/2 \rceil$, considérons une entrée x telle que $x \preceq u$, on a $g(x) = 0$ puisque pour ce vecteur x on a $f_1(x) = 1$. Alors on a

$$a_u = \sum_{x \preceq u} g(x) = 0,$$

Ainsi, g est une fonction nulle, ce qui est une contradiction puisque g n'est pas une fonction nulle par hypothèse. Alors f_1 n'a pas d'annulateur de degré strictement inférieur à $\lceil n/2 \rceil$.

Montrons maintenant que f_2 n'a pas d'annulateur de degré strictement inférieur à $\lceil n/2 \rceil$. Supposons que f_2 a un annulateur h de degré strictement inférieur à $\lceil n/2 \rceil$, c'est à dire que $h(x_1, \ldots, x_n) = 0$ si $f_2(x_1, \ldots, x_n) = 1$, i.e, $h(x_1, \ldots, x_n) = 0$ si $w_H(x_1, \ldots, x_n) > \lceil n/2 \rceil$. Notons que

$$f_1(1+x_1, \ldots, 1+x_n) = \begin{cases} 0, \text{ si } w_H(x_1, \ldots, x_n) \leq n - \lceil n/2 \rceil \\ 1, \text{ si } w_H(x_1, \ldots, x_n) > n - \lceil n/2 \rceil \end{cases}$$

$$= \begin{cases} f_2(x), \text{ si } n \text{ est pair,} \\ f_2(x), \text{ si } n \text{ est impair et } w_H(x) \neq \lceil n/2 \rceil, \\ f_2(x) + 1, \text{ si } n \text{ est impair et } w_H(x) = \lceil n/2 \rceil \end{cases}$$

Soit alors x tel que $w_H(x_1, \ldots, x_n) > \lceil n/2 \rceil$, on a alors :
$f_1(1+x_1, \ldots, 1+x_n) = f_2(x_1, \ldots, x_n) = 1$. Ainsi $f_1(1+x_1, \ldots, 1+x_n)h(x_1, \ldots, x_n) = 0$. Définissons une fonction h' par $h'(x_1, \ldots, x_n) = h(1+x_1, \ldots, 1+x_n)$, i.e., $h(x_1, \ldots, x_n) = h'(1 + x_1, \ldots, 1 + x_n)$. On a $\deg(h') = \deg(h) < \lceil n/2 \rceil$. Ainsi, on a $f_1(1 + x_1, \ldots, 1 + x_n)h'(1 + x_1, \ldots, 1 + x_n) = 0$, i.e. $f_1(x_1, \ldots, x_n)h'(x_1, \ldots, x_n) = 0$. Alors, f_1 a un annulateur de degré $< \lceil n/2 \rceil$, ce qui est une contradiction.

2- Montrons que la fonction majorité f définie par

$$f(x_1, \ldots, x_n) = \begin{cases} 0, \text{ si } w_H(x_1, \ldots, x_n) \leq n/2 \\ 1, \text{ si } w_H(x_1, \ldots, x_n) > n/2 \end{cases}$$

est d'immunité algébrique maximale. On a $supp(1 + f) \supseteq supp(f_1)$ et $supp(f) \supseteq supp(f_2)$, donc tout annulateur de $(1+f)$ est un annulateur de f_1 et tout annulateur de f est un annulateur de f_2. Comme f_1 et f_2 n'ont pas d'annulateurs de degré strictement inférieur à $\lceil n/2 \rceil$, donc il en est de même pour f et $1 + f$, alors $AI(f) = \lceil n/2 \rceil$. □

La fonction majorité étant symétrique, elle peut présenter des vulnérabilités, elle a une mauvaise non-linéarité par exemple.

Classes de fonctions de haute immunité algébrique

Une construction itérative d'une classe infinie de fonctions d'immunité algébrique optimale a été faite dans [27] et étudiée par la suite dans [17]. Ces fonctions ne sont ni équilibrées ni de haute non-linéarité, de plus, elles sont vulnérable aux attaques algébriques évoluées.
Il existe d'autres fonctions d'immunité algébrique optimale, données dans [11]. Quelques fonctions construites sont de meilleure non-linéarité.

Nous donnerons dans le théorème suivant une fonction d'immunité algébrique optimale, de bonne immunité aux attaques algébriques évoluées, de bonne non-linéarité. [18]

Théorème 4.3.2. *[18]. Soit n un entier positif tel que $n \geq 2$ et α un élément primitif du corps $\mathbb{F}_{2^n}$. Soit f la fonction Booléenne sur $\mathbb{F}_{2^n}$ de support $\{0, 1, \alpha, \cdots, \alpha^{2^{n-1}-2}\}$. Alors f a une immunité algébrique optimale $\lceil n/2 \rceil$.*

Démonstration. . Nous donnerons la preuve donné dans [18].
Soit g une fonction Booléenne de degré algébrique au plus $\lceil n/2 \rceil - 1$. Soit $g(x) = \sum_{i=0}^{2^n-1} g_i x^i$ sa représentation univariée dans $\mathbb{F}_{2^n}$, où $g_i \in \mathbb{F}_{2^n}$ est nul si le 2-poids $w_2(i)$ de i est au moins $\lceil n/2 \rceil$ (ce qui implique en particulier que $g_{2^n-1} = 0$).
Si g est un annulateur de f, alors on a $g(\alpha^i) = 0$ pour tout $i = 0, \cdots, 2^{n-1} - 2$, ce qui veut dire que le vecteur $(g_0, \cdots, g_{2^n-2})$ appartient au code de Reed-Solomon sur $\mathbb{F}_{2^n}$ de zéros $1, \alpha, \cdots, \alpha^{2^{n-1}-2}$ (le code de Reed-Solomon de zéros $\alpha^\ell, \cdots, \alpha^{\ell+r}$ est égal par définition l'ensemble de vecteurs $(g_0, \cdots, g_{2^n-2})$ de $\mathbb{F}_{2^n}^{2^n-1}$ tels que ces éléments sont des racines du polynôme $\sum_{i=0}^{2^n-2} g_i x^i$, voir [42]).
Compte tenu de la borne BCH, si g est non nulle, alors le vecteur $(g_0, \cdots, g_{2^n-2})$ est de poids de Hamming au moins 2^{n-1}. La preuve générale de cette borne inférieure est aussi dans [42]. En effet, par définition de g on a :

$$\begin{pmatrix} g(1) \\ g(\alpha) \\ g(\alpha^2) \\ \vdots \\ g(\alpha^{2^n-2}) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{2^n-2} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(2^n-2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \alpha^{2^n-2} & \alpha^{2(2^n-2)} & \cdots & \alpha^{(2^n-2)(2^n-2)} \end{pmatrix} \times \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{2^n-2} \end{pmatrix}$$

ce qui implique (puisque pour tout $0 \leq i, j \leq 2^n - 2$, la somme $\sum_{k=0}^{2^n-2} \alpha^{(i-j)k}$ égal 1 si $i = j$ et 0 sinon) que :

$$\begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{2^n-2} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha^{-1} & \alpha^{-2} & \cdots & \alpha^{-(2^n-2)} \\ 1 & \alpha^{-2} & \alpha^{-4} & \cdots & \alpha^{-2(2^n-2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \alpha^{-(2^n-2)} & \alpha^{-2(2^n-2)} & \cdots & \alpha^{-(2^n-2)(2^n-2)} \end{pmatrix} \times \begin{pmatrix} g(1) \\ g(\alpha) \\ g(\alpha^2) \\ \vdots \\ g(\alpha^{2^n-2}) \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{-(2^{n-1}-1)} & \alpha^{-2^{n-1}} & \cdots & \alpha^{-(2^n-2)} \\ \vdots & \vdots & \cdots & \vdots \\ \alpha^{-(2^{n-1}-1)(2^n-2)} & \alpha^{-2^{n-1}(2^n-2)} & \cdots & \alpha^{-(2^n-2)(2^n-2)} \end{pmatrix} \times \begin{pmatrix} g(\alpha^{2^{n-1}-1}) \\ g(\alpha^{2^{n-1}}) \\ \vdots \\ g(\alpha^{2^n-2}) \end{pmatrix}$$

Supposons que au moins 2^{n-1} des éléments g_i sont nuls. Alors $g(\alpha^{2^{n-1}-1}), \ldots, g(\alpha^{2^n-2})$ satisfont un système d'équations homogènes ayant une matrice $2^{n-1} \times 2^{n-1}$ de Vandermonde, son déterminant est donc non nul. Ce qui implique que $g(\alpha^{2^{n-1}-1}), \ldots, g(\alpha^{2^n-2})$

et par conséquent g sont nuls, ce qui est une contradiction. Ainsi le vecteur $(g_0, \ldots, g_{2^n-2})$ est de poids au moins 2^{n-1}.
Supposons que le vecteur $(g_0, \ldots, g_{2^n-2})$ est de poids 2^{n-1}. Alors $g(x) = \sum_{\substack{0 \leq i \leq 2^n-2 \\ w_2(i) \leq (n-1)/2}} x^i$ et n est impair (pour qu'il y ait 2^{n-1} termes dans l'écriture sous la forme polynomiale de $g(x)$) ; ce qui contredit le fait que $g(0) = 0$. On en déduit que le vecteur $(g_0, \ldots, g_{2^n-2})$ est de poids strictement plus grand que 2^{n-1}, ce qui nous amène à une contradiction avec le fait que g a un degré algébrique au plus $\lceil n/2 \rceil - 1$, puisque le nombre d'entiers de 2-poids au plus $\lceil n/2 \rceil - 1$ n'est pas strictement plus grand que 2^{n-1}.
Soit maintenant g un annulateur non nul de $f+1$. Le vecteur $(g_0, \ldots, g_{2^n-2})$ appartient au code de Reed-Solomon sur $\mathbb{F}_{2^n}$ de zéros $\alpha^{2^{n-1}-1}, \ldots, \alpha^{2^n-2}$. La borne BCH implique que ce vecteur a un poids de Hamming strictement plus grand que 2^{n-1}. On arrive a la même contradiction. Ainsi, il n'existe pas d'annulateur non nul de f ou de $f+1$ de degré algébrique au plus $\lceil n/2 \rceil - 1$, f est alors d'immunité algébrique optimale $\lceil n/2 \rceil$. □

4.4 Relation entre immunité algébrique, poids et non-linéarité

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

Théorème 4.4.1. *[17] Soit f une fonction Booléenne à n variables telle que $AI(f) > d$. Alors*

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

Ainsi, pour toute fonction Booléenne à n variables, on a

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

Démonstration. L'inégalité de gauche doit être satisfaite sinon, le nombre d'équations $w_H(f)$ (dans le système linéaire d'équations qu'on construit pour chercher les annulateurs de f de degré au plus d) est plus petit que le nombre d'inconnues (i.e le nombre de coefficients dans sa forme normale algébrique), ce qui implique l'existence d'un annulateur de degré $d < AI(f)$ ce qui impossible. L'inégalité de droite est obtenue à partir de l'autre en remplaçant f par $f + 1$. □

Corollaire 4.4.1. *Soit f une fonction Booléenne à n variables. S'il existe $d \in \mathbb{N}$, $1 \leq d \leq n$ tel que $w_H(f) < \sum_{i=0}^{d} \binom{n}{i}$ ou $w_H(f) > \sum_{i=0}^{n-d-1} \binom{n}{i}$, alors $AI(f) \leq d$.*

Notons que l'inverse du Théorème 4.4.1 n'est pas toujours vrai. Par exemple, les fonctions affines sont équilibrées, mais elles ont clairement des annulateurs linéaires. Une conséquence du Théorème 4.4.1 est qu'une fonction f telle que $AI(f) = \frac{n+1}{2}$, n impair doit être équilibrée.

Regardons maintenant la relation entre l'immunité algébrique et la non linéarité, on aura besoin du résultat suivant,

Lemme 4.4.1. *Pour toute fonction Booléenne f et toute fonction affine ℓ à n variables, on a*

$$AI(f) - 1 \leq AI(f + \ell) \leq AI(f) + 1$$

Plus généralement, pour toute fonction Booléenne f et toute fonction Booléenne h de degré r, on a

$$AI(f) - r \leq AI(f + h) \leq AI(f) + r$$

Démonstration. Pour toute fonction g telle que $fg = 0$, on a $(f+h)((h+1)g) = 0$. Pour toute g telle que $(1+f)g = 0$, on a $(1+f+h)((h+1)g) = 0$, ce qui donne les inégalités de droite. En les appliquant à $f + l$ et $f + h$ au lieu de f on trouve les inégalités de gauche. □

Une conséquence du lemme précédent est le résultat suivant,

Théorème 4.4.2. *[17] Si f est une fonction Booléenne à n variables avec $nl(f) < \sum_{i=0}^{d} \binom{n}{i}$, alors $AI(f) \leq d + 1$. Plus généralement, si la non-linéarité d'ordre r de f satisfait $nl_r(f) < \sum_{i=0}^{d} \binom{n}{i}$ alors $AI(f) \leq d + r$. En d'autre terme*

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

Démonstration. Soit h une fonction de degré au plus r telle que $nl_r(f) = d(f, h) = w_H(f + h)$. Si d est le plus petit entier vérifiant $nl_r(f) < \sum_{i=0}^{d} \binom{n}{i}$ alors le Théorème 4.4.1 implique que $AI(f + h) \leq d$. Le Lemme 4.4.1 implique alors que $AI(f) \leq d + r$. La dernière inégalité est obtenue en choisissant $d = AI(f) - r - 1$. □

On peut déduire du Lemme 4.4.1 et du Théorème 4.4.1 le résultat suivant qui est dans [61].

Corollaire 4.4.2. *Soit f une fonction Booléenne à n variables. Si $AI(f) > d+1$, alors*

$$\sum_{i=0}^{d} \binom{n}{i} \leq nl(f) \leq 2^n - \sum_{i=0}^{d} \binom{n}{i}.$$

La borne sur la non-linéarité d'ordre 1 a été amélioré par Lobanov[41] en la borne

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

et pour la la non-linéarité d'ordre r en la borne

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

En fait, une amélioration un peu plus forte mais plus compliquée est dans [12]. Enfin une borne meilleure dans [48] :

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

Chapitre 5

Immunité algébrique d'une fonction vectorielle

Les attaques algébriques, introduites par N. Courtois en 2003 [26] pour les schémas à flots et par N. Courtois et J. Pieprzyk dans le cas des schémas par blocs [25], font l'objet d'études nombreuses. Un critère important contre ces attaques est l'immunité algébrique. On s'intéresse dans ce chapitre à l'étude de ce critère pour les fonctions vectorielles.

5.1 Notions d'immunité algébrique d'une fonction vectorielle

Pour commencer la définition des notions d'immunité algébrique d'une fonction vectorielle, on rappelle d'abord la définition générale de l'immunité algébrique d'un sous-ensemble quelconque de $\{0,1\}^n$.

Définition 5.1.1. *Soit H un sous-ensemble de $\mathbb{F}_2^n$. Toute fonction $p \neq 0$ nulle sur H est appelée annulateur de H. L'immunité algébrique de H, noté $AI(H)$, est le degré minimal d de tous les annulateurs de H.*

$$AI(H) = \min\{d, \textit{il existe une fonction } p \neq 0, \deg p(x) = d, p(x) = 0; \forall x \in H\} \quad (5.1)$$

On peut définir l'immunité algébrique d'une fonction Booléenne d'une manière équivalente à celle de la Définition 4.1.1 :

Définition 5.1.2. *L'immunité algébrique d'une fonction Booléenne f à n variable est égal au minimum entre $AI(f^{-1}(0))$ et $AI(f^{-1}(1))$.*

Cette définition se généralise facilement au cas d'une fonction vectorielle.
Soit $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ une fonction Booléenne vectorielle.

Définition 5.1.3. *(voir[2]).*
L'immunité algébrique standard de F est le minimum des immunité algébriques de toutes les images réciproques $F^{-1}(z)$ des éléments z de $\mathbb{F}_2^m$ par F :

$$\begin{aligned} AI(F) &= \min\{AI(F^{-1}(z)), z \in \mathbb{F}_2^m\} && (5.2)\\ &= \min_{z\in\mathbb{F}_2^m} \min\{d, \deg p(x) = d, p(x) = 0; \forall x \in F^{-1}(z)\} && (5.3)\end{aligned}$$

Notons que $AI(F)$ est égal aussi au minimum des immunité algébriques de toutes les indicatrices des images réciproques $F^{-1}(z)$.
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 $m = n$ (en effet, dans ce cas, il existe au moins un élément z tel que $|F^{-1}(z)| \leq 1$ et chaque singleton a une immunité algébrique égal à 1 puisqu'il existe au moins une fonction affine nulle sur ce singleton) et si $m > n$ l'immunité algébrique est nulle (puisqu'il existe au moins un élément z tel que $F^{-1}(z) = \emptyset$).

Une autre notion joue aussi un rôle important dans le cadre des chiffrements par blocs : L'existence des relations de bas degré entre les bits d'entrée et les bits de sortie des boites S peut-être exploitée dans les attaques algébriques sur les schémas par bloc [25] (et aussi sur les schémas par flots à plusieurs sorties), ce qui correspond à l'immunité algébrique du graphe de F , $(gr(F) = \{(x, F(x)), x \in \mathbb{F}_2^n\} \subseteq \mathbb{F}_2^{n+m}$)

Définition 5.1.4. *L'immunité algébrique du graphe d'une fonction vectorielle F notée $AI_{gr}(F)$ est égale à l'immunité algébrique de son graphe.*

$$AI_{gr}(F) = \min\{d, \exists p(x, z) \neq 0, \deg(p) = d, p(x, F(x)) = 0; \forall x \in \mathbb{F}_2^n\} \quad (5.4)$$

Une troisième définition naturelle de l'immunité algébrique d'une fonction vectorielle s'appelle l'immunité algébrique par composantes.

Définition 5.1.5. *L'immunité algébrique par composantes d'une fonction vectorielle est égale au minimum des immunités algébriques de ses fonctions composantes. On la note AI_{comp}*

$$AI_{comp}(F) = \min_{v\in\mathbb{F}_2^m\setminus\{0\}} AI(v \cdot F) \quad (5.5)$$

5.1.1 Borne supérieure de l'AI et de l'AI_{gr}

Armknecht et Krause ont observé dans [2] que

Théorème 5.1.1. *L'immunité algébrique standard d'une (n,m)-fonction est au plus $d_{n,m}$, où $d_{n,m}$ est le plus petit entier d tel que $\sum_{i=0}^{d}\binom{n}{i} > 2^{n-m}$.*

Démonstration. En effet, 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 $d_{n,m}$ et de ANF

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

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 $\sum_{i=0}^{d_{n,m}}\binom{n}{i}$ inconnues (les coefficients des monômes de degrés au plus $d_{n,m}$). Comme le nombre d'équations est strictement inférieur au nombre d'inconnues, le système a nécessairement des solutions non triviales. □

Cette borne supérieure sur AI est atteinte [30]. Nous donnons dans le Tableau 5.1, pour n compris entre 5 et 20 et pour m compris entre 1 et 17 les valeurs de $d_{n,m}$.

Les auteurs de [2] ont remarqué aussi que :

Théorème 5.1.2. *L'immunité algébrique du graphe d'une (n,m)-fonction est au plus $D_{n,m}$, où $D_{n,m}$ est le plus petit entier D tel que $\sum_{i=0}^{D}\binom{n+m}{i} > 2^n$.*

Démonstration. La preuve est similaire (en considérant des annulateurs de $n+m$ variables). Soit donc G un annulateur du graphe d'une (n,m)-fonction F, de degré algébrique égal à $D_{n,m}$ et de ANF

$$G(x,y) = \sum_{\substack{(u,v)\in\mathbb{F}_2^n\times\mathbb{F}_2^m \\ |supp(u)|+|supp(v)|\leq D_{n,m}}} G_{u,v}x^u y^v$$

Alors G est un annulateur du graphe de F si et seulement si les coefficients de son ANF satisfont un système d'équations linéaires homogènes correspondantes au fait que $G(x,y) = 0$ pour tout $(x,y) \in \mathbb{F}_2^n \times \mathbb{F}_2^m$ tels que $y = F(x)$. Dans ce systèmes, on a 2^n équations linéaires à $\sum_{i=0}^{D_{n,m}}\binom{n}{i}$ inconnues (les coefficients des monômes en x et y de degrés au plus $D_{n,m}$). Comme le nombre d'équations est strictement inférieur au nombre d'inconnues, le système a nécessairement des solutions non triviales. □

$n\backslash m =$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
5	3	2	1	1	1	0											
6	3	2	2	1	1	1	0										
7	4	3	2	2	1	1	1	0									
8	4	3	2	2	1	1	1	1	0								
9	5	3	3	2	2	1	1	1	1	0							
10	5	4	3	3	2	2	1	1	1	1	0						
11	6	4	4	3	2	2	2	1	1	1	1	0					
12	6	5	4	3	3	2	2	2	1	1	1	1	0				
13	7	5	4	4	3	3	2	2	2	1	1	1	1	0			
14	7	6	5	4	4	3	3	2	2	2	1	1	1	1	0		
15	8	6	5	5	4	3	3	3	2	2	2	1	1	1	1	0	
16	8	7	6	5	4	4	3	3	2	2	2	1	1	1	1	1	0
17	9	7	6	5	5	4	4	3	3	2	2	2	1	1	1	1	1
18	9	8	7	6	5	5	4	4	3	3	2	2	2	1	1	1	1
19	10	8	7	6	5	5	4	4	3	3	3	2	2	2	1	1	1
20	10	8	7	7	6	5	5	4	4	3	3	3	2	2	2	1	1

TABLE 5.1 – Quelques valeurs de $d_{n,m}$

La borne supérieure $d_{n,m}$ sur l'AI est atteinte, mais on ne sait pas si la borne $D_{n,m}$ sur l'AI_{gr} est atteinte (il a été montré dans [2] qu'elle est atteinte pour $n \leq 14$ et partiellement pour $n = 15$).
Remarquons que $D_{n,m} = d_{n+m,m}$, donc montrer que la borne $AI_{gr}(F) \leq D_{n,m}$ est atteinte pour un certain n et m est équivalent à montrer que la borne $AI(\phi) \leq d_{n+m,m}$ est atteinte pour une $(n+m, m)$-fonction ϕ telle que $AI(\phi) = d_{n+m,m}$ et telle qu'il existe deux sous-espaces E et E' de $\mathbb{F}_2^{n+m}$, de dimensions n et m respectivement ayant une somme directe égale à $\mathbb{F}_2^{n+m}$, et tels que, pour tout $x \in E$ et tout $z \in \mathbb{F}_2^m$, il existe un unique $y \in \mathbb{F}_2^m$ tel que $\phi(x + y) = z$. Ceci est suffisant pour avoir $AI_{gr}(F) = D_{n,m}$ puisque $\phi^{-1}(0)$ par exemple est à un automorphisme linéaire près, le graphe d'une (n, m)-fonction. Il est nécessaire puisque si une (n, m)-fonction F' existe ayant un graphe d'immunité algébrique $D_{n,m}$ alors la $(n+m, m)$-fonction $\phi : (x, y) \in \mathbb{F}_2^n \times \mathbb{F}_2^m \to y+F'(x)$ satisfait $AI(\phi) = d_{n+m,m}$.

Nous donnons dans le Tableau 5.2, pour n compris entre 5 et 20 et pour m compris entre 1 et 17 les valeurs de $D_{n,m}$.

$n\backslash m =$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
5	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
6	4	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2
7	4	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2
8	5	4	4	3	3	3	3	3	3	3	3	3	3	3	2	2	2
9	5	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3
10	6	5	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3
11	6	5	5	5	4	4	4	4	4	4	4	4	3	3	3	3	3
12	7	6	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4
13	7	6	6	5	5	5	5	5	4	4	4	4	4	4	4	4	4
14	8	7	6	6	5	5	5	5	5	5	5	4	4	4	4	4	4
15	8	7	7	6	6	6	5	5	5	5	5	5	5	5	5	4	4
16	9	8	7	7	6	6	6	6	5	5	5	5	5	5	5	5	5
17	9	8	7	7	7	6	6	6	6	6	6	5	5	5	5	5	5
18	10	8	8	7	7	7	7	6	6	6	6	6	6	6	5	5	5
19	10	9	8	8	7	7	7	7	7	6	6	6	6	6	6	6	6
20	11	9	9	8	8	8	7	7	7	7	7	6	6	6	6	6	6

TABLE 5.2 – Quelques valeurs de $D_{n,m}$

5.1.2 Bornes sur les nombres $d_{n,m}$ et $D_{n,m}$

Nous avons $d_{n,m} \leq n - m$ (ce qui implique que $D_{n,m} \leq n$) puisque $\sum_{i=0}^{n-m} \binom{n}{i} > \sum_{i=0}^{n-m} \binom{n-m}{i} = 2^{n-m}$. En fait, pour $1 \leq m \leq n/2$ et m proche de $n/2$, cette borne est faible puisqu'il est facile de remarquer que $d_{n,m}$ croît quand m décroît ($d_{n,m} \leq d_{n,1} = \lceil \frac{n}{2} \rceil$). Nous donnerons dans ce qui suit plus de détails sur $d_{n,m}$.

Proposition 5.1.1. *[20] Soit $\lambda \leq 1/2$ un nombre réel positif. Pour tout couple d'entiers positifs (n, m) tel que :*

$$m > n(1 - H_2(\lambda)) + \frac{1}{2}(3 + \log_2 n + \log_2 \lambda + \log_2(1 - \lambda)), \tag{5.6}$$

où $H_2(x) = -x \log_2 x - (1 - x) \log_2(1 - x)$, on a :

$$d_{n,m} \leq \lceil \lambda n \rceil.$$

Démonstration. On sait (d'après [42], page 310) que, pour tout nombre positif $\lambda \leq 1/2$ et tout entier positif n, on a $\sum_{i=0}^{\lceil \lambda n \rceil} \binom{n}{i} \geq \frac{2^{nH_2(\lambda)}}{\sqrt{8\lambda n(1-\lambda)}}$. Cette borne implique, pour tout m, l'inégalité

$$d_{n,m} \leq \min \left\{ \lceil \lambda n \rceil \, ; \, nH_2(\lambda) - \frac{1}{2}(3 + \log_2 n + \log_2 \lambda + \log_2(1 - \lambda)) > n - m \right\}.$$

c'est à dire

$$d_{n,m} \leq \min\left\{\lceil \lambda n \rceil \, ; \, m > n(1 - H_2(\lambda)) + \frac{1}{2}(3 + \log_2 n + \log_2 \lambda + \log_2(1-\lambda))\right\}.$$

Ce qui implique le résultat. □

Nous donnerons dans le Tableau 5.3 ci-après, pour n compris entre 5 et 20 et pour λ appartenant à $\{0.1, 0.2, 0.3, 0.4\}$ la plus petite valeur de m pour laquelle la condition (5.6) sur m soit satisfaite.
Le terme dans $\frac{1}{2}(3 + \log_2 n + \log_2 \lambda + \log_2(1-\lambda))$ est asymptotiquement négligeable

$n\backslash\lambda$	0.1	0.2	0.3	0.4
5	4	3	3	2
6	5	4	3	2
7	5	4	3	3
8	6	4	3	3
9	7	5	4	3
10	7	5	4	3
11	8	5	4	3
12	8	6	4	3
13	9	6	4	3
14	10	6	4	3
15	10	7	5	3
16	11	7	5	3
17	11	7	5	4
18	12	8	5	4
19	12	8	5	4
20	13	8	5	4

TABLE 5.3 – Les valeurs minimales de m pour lesquelles Proposition 5.1.1 implique que $d_{n,m} \leq \lceil \lambda n \rceil$

par rapport à n, donc, la version asymptotique de la relation (5.6) est

$$m \geq (1 - H_2(\lambda))n$$

Ainsi, on a le résultat

Corollaire 5.1.1. *[20] Pour tout couple de nombres réels positifs (λ, μ) tel que $\lambda \leq 1/2$ et $\mu > 1 - H_2(\lambda)$, il existe un entier positif N tel que, pour tout $n \geq N$ et tout $m \geq \mu n$, on a $d_{n,m} \leq \lceil \lambda n \rceil$.*

Nous donnons dans le tableau 5.4 les valeurs de $1-H_2(\lambda)$ pour λ dans $\{0.1, 0.2, 0.3, 0.4\}$. Appliquons la Proposition 5.1.1 avec $n+m$ au lieu de n, et utilisons le fait que

λ	0.1	0.2	0.3	0.4
$1-H_2(\lambda)$	0.53	0.28	0.19	0.03

TABLE 5.4 – Les valeurs de $1-H_2(\lambda)$

$D_{n,m} = d_{n+m,m}$ pour obtenir :

Corollaire 5.1.2. *Soit $\lambda \leq 1/2$ un nombre réel positif. Pour tout couple d'entiers positifs (n,m) tel que*

$$mH_2(\lambda) > n(1-H_2(\lambda)) + \frac{1}{2}(3+\log_2(n+m)+\log_2\lambda+\log_2(1-\lambda)),$$

on a :

$$D_{n,m} \leq \lceil \lambda(n+m) \rceil.$$

Remarque 5.1.1. *Ces bornes générales sur $d_{n,m}$ et $D_{n,m}$ peuvent être améliorées pour des valeurs spécifiques de m (en fonction de n). Un exemple pour cela peut être trouvé dans la Proposition 6.1.5 , au Chapitre 6.*

Bornes plus explicites

On verra dans la proposition suivante une borne supérieure sur $d_{n,m}$ (donc sur l'immunité algébrique standard d'une (n,m)-fonction) en fonction de n et m plus simple à calculer.

Lemme 5.1.1. *Pour tout $n \geq m \geq 1$ on a*

$$\sum_{i=0}^{\lceil \frac{n-m+1}{2} \rceil} \binom{n}{i} - 2^{n-m} > 0$$

On utilise ce lemme pour montrer le résultat suivant

Proposition 5.1.2. *Soit F une fonction vectorielle de $\mathbb{F}_2^n$ dans $\mathbb{F}_2^m$, alors :*

$$AI(F) \leq d_{n,m} \leq \left\lceil \frac{n-m+1}{2} \right\rceil \tag{5.7}$$

Pour $m = 1$ ($F = f$ est une fonction Booléenne) la borne (5.7) devient $AI(f) \leq \lceil \frac{n}{2} \rceil$ qui est la borne de Courtois-Meier donnée dans [26]. Cette borne est atteinte par exemple par la fonction majorité.

Démonstration. (du Lemme 5.1.1). Posons $k = n - m \geq 0$, il revient à montrer que pour tout $n > k \geq 0$ on a :

$$\sum_{i=0}^{\lceil \frac{k+1}{2} \rceil} \binom{n}{i} - 2^k > 0$$

La fonction qui à n associe $\binom{n}{i}$ étant croissante pour tout i, il suffit donc de montrer l'inégalité précédente pour la plus petite valeur de n qui) est $k+1$ (puisque $m \geq 1$. L'inégalité :

$$\sum_{i=0}^{\lceil \frac{k+1}{2} \rceil} \binom{k+1}{i} - 2^k > 0$$

est vraie car on a :

$$\sum_{i=0}^{\lceil \frac{k+1}{2} \rceil} \binom{k+1}{i} > \frac{2^{k+1}}{2} = 2^k$$

□

La borne de la proposition précédente peut-être améliorée pour des valeurs de m plus grandes.

Proposition 5.1.3. *Pour tout couple d'entiers (n, m) tel que $m \geq 2$ et $n \geq m+1$ on a*

$$\sum_{i=0}^{\lfloor \frac{n-m+1}{2} \rfloor} \binom{n}{i} - 2^{n-m} > 0$$

Ce qui implique que si F est une fonction vectorielle de $\mathbb{F}_2^n$ dans $\mathbb{F}_2^m$ avec $n \geq m+1 \geq 3$, alors :

$$AI(F) \leq \left\lfloor \frac{n-m+1}{2} \right\rfloor \tag{5.8}$$

Démonstration. Elle est similaire à celle de la proposition précédente. Posons toujours $k = n - m \geq 1$, montrer cette inégalité revient à montrer l'inégalité pour la plus petite valeur de n qui est maintenant $k+2$. On montre donc que

$$\forall k \geq 1 : \sum_{i=0}^{\lfloor \frac{k+1}{2} \rfloor} \binom{k+2}{i} - 2^k > 0 \tag{5.9}$$

On étudie les deux cas de parité de k :

– Si $k+1$ est pair alors

$$\sum_{i=0}^{\lfloor \frac{k+1}{2} \rfloor} \binom{k+2}{i} - 2^k = \frac{2^{k+2}}{2} - 2^k = 2^k > 0$$

– Si $k+1$ est impair alors $\lfloor \frac{k+1}{2} \rfloor = \frac{k}{2}$ et

$$\sum_{i=0}^{\lfloor \frac{k+1}{2} \rfloor} \binom{k+2}{i} - 2^k = \frac{2^{k+2}}{2} - \frac{1}{2}\binom{k+2}{\frac{k}{2}+1} - 2^k =$$

$$2^k - \frac{1}{2}\binom{k+1}{\frac{k}{2}+1} - \frac{1}{2}\binom{k+1}{\frac{k}{2}} = 2^k - \binom{k+1}{\frac{k}{2}} = 2^k - \binom{k}{\frac{k}{2}} - \binom{k}{\frac{k}{2}-1} > 0$$

(car $\binom{k}{\frac{k}{2}}$ et $\binom{k}{\frac{k}{2}-1}$ sont deux coefficients binomiaux dans le développement de 2^k).Alors l'inégalité est vraie dans les deux cas. □

Pour tout couple d'entiers (n, m) tel que, $n, m \geq 1$, on note $D_{n,m}$ le nombre :

$$D_{n,m} = \min\{D \mid \sum_{i=0}^{D} \binom{n+m}{i} - 2^n > 0\} \tag{5.10}$$

On déduit de ce qui précède que :

$$D_{n,m} \leq \lfloor \tfrac{n+1}{2} \rfloor, \quad \text{pour tout } n \geq m+1 \geq 3 \tag{5.11}$$

Ce qui donne les bornes supérieures suivantes sur l'immunité algébrique du graphe d'une (n, m)-fonction F [2] :

$$AI_{gr}(F) \leq \left\lceil \frac{n+1}{2} \right\rceil, \text{pour tout } n \geq 1, m \geq 1 \tag{5.12}$$

$$AI_{gr}(F) \leq \left\lfloor \frac{n+1}{2} \right\rfloor, \text{pour tout } n \geq m+1 \geq 3 \tag{5.13}$$

Notons que la dernière borne est atteinte par cinq boites-S de DES (data encryption standard).

5.2 Construction de fonctions vectorielles d'une immunité algébrique standard maximale

Les auteurs de [30] ont construit une (n, m)-fonction dont l'AI est maximale.

Théorème 5.2.1. *Pour tout couple d'entiers positifs (n, m), il existe une (n, m)-fonction F telle que $AI(F) = d_{n,m}$, où $d_{n,m}$ est le plus petit entier d tel que $\sum_{i=0}^{d} \binom{n}{i} > 2^{n-m}$.*

Nous utiliserons pour la démonstration du Théorème 5.2.1 le fait que $AI(F) \leq d_{n,m}$ et le lemme suivant qu'on peut trouver dans [30].

Lemme 5.2.1. *Soient d, n, s trois entiers positifs. Supposons que $0 < d < n$, $k = \sum_{i=0}^{d} \binom{n}{i}$ (la dimension de $RM(d, n)$), $s \geq 1$ tel que $sk \leq 2^n$. Alors, il existe s sous-ensembles disjoints $A_1, \ldots, A_s$ de $\mathbb{F}_2^n$ satisfaisant les conditions suivantes :*

1. *$|A_j| = k$, $(1 \leq j \leq s)$,*
2. *Pour toute fonction Booléenne non nulle à n variables g telle que $\deg(g) \leq d$ et tout j, $(1 \leq j \leq s)$, il existe $v \in A_j$ tel que $g(v) \neq 0$.*

Démonstration. Voir [30]. □

Démonstration. (du Théorème 5.2.1) Par la définition du $d_{n,m}$ on a $\sum_{i=0}^{d_{n,m}-1} \binom{n}{i} \leq 2^{n-m}$. Sachant que $\sum_{i=0}^{d_{n,m}-1} \binom{n}{i}$ est la dimension de $RM(d_{n,m} - 1, n)$, par le Lemme 5.2.1, il existe 2^m sous ensembles disjoints A_j, $(0 \leq j \leq 2^m - 1)$ de $\mathbb{F}_2^n$ satisfaisant $|A_j| = \sum_{i=0}^{d_{n,m}-1} \binom{n}{i}$, $(0 \leq j \leq 2^m - 1)$ et pour toute fonction Booléenne à n variables non nulle g, telle que $\deg(g) \leq d_{n,m} - 1$ et tout j, $(0 \leq j \leq 2^m - 1)$ il existe $v \in A_j$ tel que $g(v) \neq 0$.

Définissons les fonctions Booléennes f_i, $(0 \leq i \leq 2^m - 1)$ par

$$f_i(v) = \begin{cases} 1, & \text{si } v \in A_j \text{ et } j_i = 1, 0 \leq j \leq 2^m - 1 \\ 0, & \text{si } v \in A_j \text{ et } j_i = 0, 0 \leq j \leq 2^m - 1 \\ c, & \text{si } v \notin A_0 \cup A_1 \ldots \cup A_{2^m-1} \end{cases}$$

où j_i est le coefficient dans l'expansion 2-adic de j, $j = j_0 + j_1 2 + \ldots + j_{m-1} 2^{m-1}$, et c peut être 0 ou 1.

Montrons que pour $F = (f_0, \ldots, f_{m-1})$, nous avons $AI(F) = d_{n,m}$. Pour tout $b = (b_0, b_1, \ldots, b_{m-1}) \in \mathbb{F}_2^m$, et $a \in A_0 \cup A_1 \ldots \cup A_{2^m-1}$, on a

$$\begin{aligned} a \in F^{-1}(b) &\iff f_i(a) = b_i, (0 \leq i \leq m-1) \\ &\iff \text{pour tout } i, (0 \leq i \leq m-1), a \in \bigcup \{A_j | 0 \leq j \leq 2^m - 1, j_i = b_i\} \\ &\iff a \in A_j \text{ où } j = b_0 + b_1 2 + \ldots + b_{m-1} 2^{m-1}. \end{aligned}$$

Ainsi $F^{-1}(b) \supseteq A_j$ pour $j = b_0 + b_1 2 + \ldots + b_{m-1} 2^{m-1}$. Soit g une fonction Booléenne à n variables telle que $\deg(g) \leq d_{n,m} - 1$. S'il existe $b = (b_0, b_1, \ldots, b_{m-1}) \in \mathbb{F}_2^m$ tel que $g|_{F^{-1}(b)} = 0$ alors $g|_{A_j} = 0$ pour $j = b_0 + b_1 2 + \ldots + b_{m-1} 2^{m-1}$. En utilisant la contraposée du point 2 du Lemme 5.2.1 on a $g \equiv 0$. Ainsi $AI(F) \geq d_{n,m}$ alors $AI(F) = d_{n,m}$. □

5.3 Caractérisation des fonctions ayant une petite valeur d'immunité algébrique

Pour mieux comprendre les notions d'immunité algébrique des fonctions vectorielles, il est intéressant de caractériser les cas où ces fonctions ont de petites valeurs d'immunité algébrique, à savoir les cas où ces valeurs valent 0, 1 ou même ont des valeurs strictement plus grandes, pour les trois notions.

5.3.1 Fonctions vectorielles d'une immunité algébrique nulle

On trouve dans le lemme suivant le cas des valeurs nulles :

Lemme 5.3.1. *Soit F une (n,m)- fonction, alors :*

1. $AI(F) = 0$ *si et seulement si F n'est pas surjective ;*
2. $AI_{comp}(F) = 0$ *si et seulement si l'image de F est contenue dans un hyperplan affine de $\mathbb{F}_2^m$, ou encore le rang de F est strictement inférieur à m ; autrement dit :*
   $AI_{comp}(F) = 0 \Leftrightarrow (ImF \subseteq H, H$ *hyperplan de* $\mathbb{F}_2^m) \Leftrightarrow rg(F) < m$
3. $AI_{gr}(F)$ *ne s'annule jamais.*

Démonstration. Si F est une (n,m)- fonction alors :
1- $AI(F) = 0$ si et seulement s'il existe $z \in \mathbb{F}_2^m$ tel que la fonction 1 s'annule sur $F^{-1}(z)$, i.e, $F^{-1}(z) = \emptyset$ ce qui équivaut au fait que F n'est pas surjective.
2- $AI_{comp}(F) = 0$ si et seulement s'il existe v non nul dans $\mathbb{F}_2^m$ tel que $v \cdot F$ est une fonction constante égale à 0 ou à 1, c'est à dire que pour tout x dans $\mathbb{F}_2^n$, $F(x)$ appartient à l'hyperplan d'équation $v \cdot y = 0$ ou à son complémentaire, autrement dit, l'image de F est contenue dans un hyperplan affine de $\mathbb{F}_2^m$, ou encore le rang de F est strictement inférieur à m.
3- $AI_{gr}(F)$ n'est jamais nul puisqu'il existe au moins un élément (x, y) dans le graphe de F sinon la fonction ne serait pas définie. □

5.3.2 Fonctions vectorielles d'une immunité algébrique égale à 1

On peut aussi caractériser les cas d'égalité à 1 des valeurs de l'immunité algébrique pour les trois notions :

Lemme 5.3.2. *Soit F une (n,m)- fonction, alors :*

1. *$AI(F) = 1$ si et seulement si F est surjective et il existe z de $\mathbb{F}_2^m$ tel que $F^{-1}(z)$ soit contenu dans un hyperplan affine de $\mathbb{F}_2^n$, c'est à dire que le rang affine de $F^{-1}(z)$ est strictement inférieur à n ;*
2. *$AI_{comp}(F) = 1$ si et seulement si aucune fonction composante de F, $v \cdot F$, $v \in \mathbb{F}_2^{m^*}$ n'est constante, et il existe un hyperplan affine H_m de $\mathbb{F}_2^m$ et un hyperplan affine H_n de $\mathbb{F}_2^n$ tels que $F^{-1}(H_m) \subseteq H_n$;*
3. *$AI_{gr}(F) = 1$ si F s'écrit sous la forme $F = \frac{G+G^2}{b} + L$, où G est une (n,m)-fonction quelconque, $b \neq 0$ et L est une fonction affine.*

Démonstration. 1-Evident, on utilise le fait que l'immunité algébrique d'un hyperplan est égale à 1, et la croissance de la fonction AI.
2- $AI_{comp}(F) = 1$ si et seulement aucune fonction composante de F, $v \cdot F$, $v \in \mathbb{F}_2^{m^*}$ n'est constante, et il existe $v \neq 0$, tel que l'une des images réciproques $(v \cdot F)^{-1}(\epsilon)$, $\epsilon = 0$ ou $\epsilon = 1$ soit contenue dans un hyperplan affine de $\mathbb{F}_2^n$:
$(v \cdot F)^{-1}(\epsilon) = \{x \in \mathbb{F}_2^n, v \cdot F(x) = \epsilon\} = \cup_{v \cdot z = \epsilon} F^{-1}(z) \subseteq H_n$, où H_n est un hyperplan affine de $\mathbb{F}_2^n$, ce qui veut dire que pour tout z dans H_m-l'hyperplan d'équation $v \cdot z = \epsilon$- on a $F^{-1}(z) \subseteq H_n$, i.e. $F^{-1}(H_m) \subseteq H_n$.
3- $AI_{gr}(F) = 1$ si et seulement si le graphe de F est contenu dans un hyperplan affine de $\mathbb{F}_2^{n+m}$. Or $(x, F(x))$ appartient à un hyperplan affine de $\mathbb{F}_2^{n+m}$ si $tr(ax + bF(x)) = c$, c constante, c'est à dire que $F(x) = \frac{a}{b}x + \frac{F_1}{b}, b \neq 0$, avec F_1 une fonction de trace constante, i.e. $F(x) = \frac{a}{b}x + \frac{G(x)+G^2(x)}{b} + c'$. □

L'immunité algébrique standard des fonctions courbes est au moins 1 car :

Proposition 5.3.1. *Une (n,m)-fonction courbe F est nécessairement surjective.*

Démonstration. Soit z un élément de $\mathbb{F}_2^m$, on montre que son image réciproque par F n'est pas vide i.e. $|F^{-1}(z)| \neq 0$:
Soit v un élément de $\mathbb{F}_2^m$, la fonction $v \mapsto v \cdot (F(x) + z)$ étant linéaire, elle est donc nulle ou équilibrée, alors :

$$\sum_{v \in \mathbb{F}_2^m} (-1)^{v \cdot (F(x)+z)} = \begin{cases} 2^m, & \text{si } F(x) = z \\ 0, & \text{sinon.} \end{cases} \tag{5.14}$$

d'où :

$$\sum_{x\in\mathbb{F}_2^n, v\in\mathbb{F}_2^m} (-1)^{v\cdot(F(x)+z)} = 2^m|F^{-1}(z)|$$

On a donc $|F^{-1}(z)| = 2^{-m}\sum_{x\in\mathbb{F}_2^n, v\in\mathbb{F}_2^m}(-1)^{v\cdot(F(x)+z)}$. La fonction F étant courbe alors il en est de même pour $v\cdot F$. On note $\widetilde{v\cdot F}$, $v\in\mathbb{F}_2^{m*}$ sa fonction duale ; on a par définition $\widehat{(v\cdot F)}_\chi(u) = \sum_{x\in\mathbb{F}_2^n}(-1)^{v\cdot F(x)\oplus x\cdot u} = 2^{n/2}(-1)^{\widetilde{v\cdot F}(u)}$, donc $\sum_{x\in\mathbb{F}_2^n}(-1)^{v\cdot F(x)} = 2^{n/2}(-1)^{\widetilde{v\cdot F}(0)}$. Le cardinal de $F^{-1}(z)$ égale alors $2^{n-m} + 2^{n/2-m}\sum_{v\in\mathbb{F}_2^{m*}}(-1)^{\widetilde{v\cdot F}(0)+v\cdot z}$, cette quantité est strictement positive puisque la dernière somme ne peut être égale à $-2^{n/2}$ car elle est calculée pour les valeurs de v dans $\mathbb{F}_2^{m*}$ et sachant que $m \le n/2$. □

La Proposition 5.3.1 implique qu'une fonction courbe est nécessairement d'immunité algébrique standard au moins 1. En fait, elle peut atteindre 1 pour les $(4,2)$-fonctions courbes par exemple, puisque leurs AI est au plus $d_{4,2}$ qui est égale à 1.

5.3.3 Fonctions vectorielles d'une immunité algébrique supérieure à 1

Soit F une (n,m)-fonction, alors il existe z dans $\mathbb{F}_2^m$ tel que $|F^{-1}(z)| \le 2^{n-m}$; si $2^{n-m} \le n-1$, i.e. $n-m \le \log_2(n-1)$, alors $F^{-1}(z)$ est contenu dans un hyperplan affine, par conséquent $AI(F) \le 1$ (égale à 1 si F est surjective). On en déduit une condition nécessaire sur m en fonction de n pour que l'immunité algébrique standard d'une (n,m)- fonction soit supérieure à 1.

Proposition 5.3.2. *Soit F une fonction vectorielle à n entrées et m sorties ; si $AI(F) > 1$ alors $m < n - \log_2(n-1)$.*

Cette proposition confirme ce qu'on a mentionné au début de ce chapitre, les valeurs de l'immunité algébrique standard $AI(F)$ d'une fonction vectorielle F ne sont intéressantes que si m est assez petit.
Soit F une (n,m)-fonction telle que $AI(F) > k$, alors on a nécessairement $\sum_{i=0}^{k}\binom{n}{i} \le 2^{n-m}$ (car $d_{n,m}$ est le plus petit entier d vérifiant $\sum_{i=0}^{d}\binom{n}{i} > 2^{n-m}$), donc $AI(F) > k$ nécessite que :

$$m \le n - \log_2\left(\sum_{i=0}^{k}\binom{n}{i}\right) \tag{5.15}$$

Une autre condition nécessaire sur n, m pour que $AI(F) > k$ est dans le résultat suivant

Proposition 5.3.3. *$AI(F) > k$, (k un entier positif) implique que*

$$m \leq n(1 - H_2(k/n)) + \frac{1}{2}(3 + \log_2(k(1 - k/n))) \tag{5.16}$$

Le graphe de la fonction f_3, définie par

$$\begin{aligned} f_3(n,k) &= n(1 + k/n \log_2(k/n) + (1 - k/n)\log_2(1 - k/n)) \\ &+ (1/2)(3 + \log_2(k(1 - k/n)) - \left(n - \log_2\left(\sum_{i=0}^{k}\binom{n}{i}\right)\right) \end{aligned}$$

est situé en dessus du plan nOk, c'est à dire que

$$n(1 - H_2(k/n)) + \frac{1}{2}(3 + \log_2(k(1 - k/n))) > n - \log_2\left(\sum_{i=0}^{k}\binom{n}{i}\right)$$

Alors, la borne (5.16) sur m est moins forte que (5.15).

On peut caractériser le cas où $AI(F) > 1, AI_{comp} > 1$ et $AI_{gr} > 1$ en considérant la contraposée des points 1, 2 et 3 du Lemme 5.3.2 :

Corollaire 5.3.1. *Soit F une (n,m)- fonction, alors :*

1. *$AI(F) > 1$ si et seulement si F est surjective et pour tout z dans $\mathbb{F}_2^m$, le rang affine de $F^{-1}(z)$ est égal à n ;*
2. *$AI_{comp}(F) > 1$ si et seulement si aucune fonction composante de F, $v \cdot F$, $v \in \mathbb{F}_2^{m^*}$ n'est constante, et pour tout hyperplan affine H_m de $\mathbb{F}_2^m$, le rang affine de $F^{-1}(H_m)$ est égal à n ;*
3. *Si $AI_{gr}(F) > 1$ alors F ne peut s'écrire sous la forme $F = \frac{G+G^2}{b} + L$, où G est une (n,m)-fonction quelconque, $b \neq 0$ et L est une fonction affine.*

Remarquons pour le point 3, on a seulement une condition nécessaire.

5.4 Stabilité de l'immunité algébrique sous transformations affines

L'introduction de différentes notions d'immunité algébrique d'une fonction vectorielle, nous amène à poser des questions sur la stabilité des valeurs d'immunité algébrique par rapport à chaque notion d'équivalence. La première équivalence classique est l'équivalence affine.

5.4.1 Immunité algébrique et équivalence affine

On étudie dans cette section la stabilité de l'immunité algébrique d'une fonction vectorielle sous transformations affines.

Définition 5.4.1 (A-Equivalence)**.** *Deux fonctions F et G sont dites affinement équivalentes (A- équivalentes), si l'une est égale à l'autre composée à gauche et à droite par des permutations affines, i.e.*

$$F \sim_{A.eq} G \Leftrightarrow \exists A, A' \text{ permutations affines, } G = A \circ F \circ A'$$

Proposition 5.4.1. *Les trois différentes notions d'immunité algébrique d'une fonction vectorielle sont stables sous équivalence affine.*

Démonstration. Soit F une (n, m)-fonction. On montre que pour tout couple de permutations affines (A, A') on a :

$$\begin{aligned} AI(F) &= AI(A \circ F \circ A') \\ AI_{comp}(F) &= AI_{comp}(A \circ F \circ A') \\ AI_{gr}(F) &= AI_{gr}(A \circ F \circ A') \end{aligned}$$

1- L'immunité algébrique standard AI est un invariant affine, remarquons pour montrer ceci que pour tout x dans $\mathbb{F}_2^n$ et tout z dans $\mathbb{F}_2^m$, on a :

$$\begin{aligned} F(x) = z &\Leftrightarrow A \circ F(x) = A(z) \\ &\Leftrightarrow A \circ F \circ A'(x') = z', (x = A'(x'), z' = A(z)) \\ &\Leftrightarrow x' \in (A \circ F \circ A')^{-1}(z') \end{aligned}$$

Alors p est un annulateur de $F^{-1}(z)$ si et seulement si $p \circ A'^{-1}$ est un annulateur de $(A \circ F \circ A')^{-1}(z')$ avec $z' = A(z)$. En prenant un annulateur de degré minimum, et comme les permutations A et A' sont de degré 1, le degré de p égale au degré de $p \circ A'^{-1}$.

2- On peut montrer de même l'invariance affine de l'immunité algébrique par composantes AI_{comp} :
On a $AI_{comp}(F) = \min_{v \neq 0} AI(v \cdot F) = \min_{v \neq 0, \epsilon \in \mathbb{F}_2} AI(v \cdot F + \epsilon)$, et pour tout $v \neq 0, \epsilon \in \mathbb{F}_2$, on a :
$\{v \cdot F(x) + \epsilon, x \in \mathbb{F}_2^n\} = \{v \cdot A \circ F(x) + \epsilon, x \in \mathbb{F}_2^n\}$, pour toute permutation affine d'éléments de $\mathbb{F}_2^m$, ce qui implique que $\min_{v \neq 0, \epsilon \in \mathbb{F}_2} AI(v \cdot F + \epsilon) = \min_{v \neq 0, \epsilon \in \mathbb{F}_2} AI(v \cdot A \circ F \circ A' + \epsilon)$ pour toute permutation affine d'éléments de $\mathbb{F}_2^n$.

3- Pour voir l'invariance affine de l'immunité algébrique du graphe, remarquons que (x, y) est dans le graphe de F si et seulement si $(x', y') = (A'^{-1}(x), A(y))$ est dans le graphe de $A \circ F \circ A'$. En fait l'immunité algébrique du graphe est stable par une équivalence plus générale, l'équivalence CCZ (voir plus bas). □

Remarque 5.4.1. *Remarquons que dans la démonstration de* $AI(F) = AI(A \circ F \circ A')$, *on n'a pas utilisé le fait que* A *soit affine. Alors on a en général* $AI(F) = AI(\pi \circ F \circ A')$, *pour toute permutation* π *d'éléments de* $\mathbb{F}_2^m$.

5.4.2 Immunité algébrique et équivalence affine étendue

On introduit une deuxième notion d'équivalence, l'équivalence affine étendue (EA-Equivalence).

Définition 5.4.2 (EA-Equivalence)**.** *Deux* (n, m)*-fonctions* F *et* G *sont dites affinement équivalentes de façon étendue (EA- équivalentes) si l'une est la somme d'une* (n, m)*-fonction affine et d'une fonction affinement équivalente à l'autre. i.e*

$$F \sim_{EA.eq} G$$
$$\Leftrightarrow \quad \exists A_1, A_2 \text{ permutations affines de } \mathbb{F}_2^n, \text{ et une } (n, m)\text{-fonction affine } A_3 \text{ tels que}$$
$$G = A_1 \circ F \circ A_2 + A_3 \tag{5.17}$$

Cette notion d'équivalence est plus générale que celle de la Définition 5.4.1, en effet, si la fonction affine A_3 dans la relation (5.17) est nulle on retrouve la définition d'équivalence affine.

Proposition 5.4.2. *L'immunité algébrique standard (AI) et l'immunité algébrique par composantes* (AI_{comp}) *ne sont pas stables sous transformations affines étendues. Plus précisément :*
Si F *et* G *sont deux* (n, m)*-fonctions affinement équivalentes de façon étendue, alors*

$$AI_{comp}(F) - 1 \leq AI_{comp}(G) \leq AI_{comp}(F) + 1 \tag{5.18}$$

Démonstration. Les deux contre exemples ci-après montrent l'instabilité de AI et AI_{comp} sous transformations affine étendues.
Montrons maintenant la Relation (8.1). Soit F, G deux (n, m)-fonctions EA-équivalentes, si $G = A_1 \circ F \circ A_2 + L$, avec L affine. On utilisera le résultat sur les fonctions Booléennes

à une sortie de [17] et la stabilité de l'immunité algébrique par composantes par transformation affine. On a

$$\begin{aligned} AI_{comp}(G) &= \min_{v\in\mathbb{F}_2^{m*}} AI(v\cdot(A_1\circ F\circ A_2)+v\cdot L) \\ &\leq \min_{v\in\mathbb{F}_2^{m*}} [AI(v\cdot(A_1\circ F\circ A_2))+1], \text{ d'après [17]} \\ &= \min_{v\in\mathbb{F}_2^{m*}} [AI(v\cdot(A_1\circ F\circ A_2))]+1 \\ &= AI_{comp}(A_1\circ F\circ A_2)+1 \\ &= AI_{comp}(F)+1 \end{aligned}$$

On peut montrer de la même façon l'autre inégalité. □

Exemple 5.4.1. *De deux fonctions AE-équivalentes n'ayant pas la même AI : Soient F, L deux (n,m)-fonctions telles que F est surjective et L est affine, alors on a $(F+L) \sim_{EA.eq} F$.*

- *Si la i^{ieme} coordonnée de F est une fonction affine égale à celle de la fonction L alors $F+L$ n'est pas surjective, donc $AI(F+L)=0$ bien que $AI(F)=k\geq 1$.*
- *Si F n'est pas surjective et $F+L$ est surjective, alors $AI(F)=0$ et $AI(F+L)=k\geq 1$.*

Exemple 5.4.2. *De deux fonctions AE-équivalentes n'ayant pas la même AI_{comp} : Soient F, L deux (n,m)-fonctions telles que L est affine surjective. On a $(F+L) \sim_{EA.eq} F$ et pour tout $v\neq 0$ dans $\mathbb{F}_2^m$, on a $v\cdot L\neq 0$ (puisque L est surjective).*

- *S'il existe $v_0\neq 0$ dans $\mathbb{F}_2^m$ tel que $v_0\cdot F$ est constante alors $AI_{comp}(F)=0$ et $AI_{comp}(F+L)=\min_v AI(v\cdot F+v\cdot L)=1$.*
- *Supposons maintenant que pour tout $v\neq 0$ dans $\mathbb{F}_2^m$, on a que $v\cdot F$ est non constante. S'il existe $v_1\neq 0$ tel que $v_1\cdot F$ est affine et $v_1\cdot F+v_1\cdot L$ est constante alors $AI_{comp}(F)=1$ et $AI_{comp}(F+L)=0$.*

5.4.3 Immunité algébrique et équivalence CCZ

Une troisième notion d'équivalence existe, qui est plus générale que celles définies plus haut.

Définition 5.4.3 (CCZ-Equivalence)**.** *On dit que deux (n,m)-fonctions F et G sont équivalentes au sens de Carlet-Charpin-Zinoviev (CCZ équivalentes)*[1] *si leurs graphes $gr(F)=\{(x,y)\in\mathbb{F}_2^n\times\mathbb{F}_2^m | y=F(x)\}$ et $gr(G)=\{(x,y)\in\mathbb{F}_2^n\times\mathbb{F}_2^m | y=G(x)\}$ sont affinement équivalents.*

1. Cette notion a été introduite dans [15] et a été nommée CCZ-équivalence dans [7, 8], elle peut être aussi nommée équivalence des graphes

Cette notion d'équivalence est encore plus générale que celles de la définition 5.4.1 et de la définition 5.4.2 comme on le verra dans la proposition 5.4.3.
On peut écrire la définition 5.4.3 comme suit

$$\begin{aligned} F \sim_{CCZ} G &\Leftrightarrow gr(F) \sim_{A.eq} gr(G) \\ &\Leftrightarrow \begin{cases} \exists L = (L_1, L_2) : \text{ automorphisme affine de } \mathbb{F}_2^n \times \mathbb{F}_2^m | \\ \qquad\qquad y = F(x) \Leftrightarrow L_2(x,y) = G(L_1(x,y)) \end{cases} \\ &\Leftrightarrow \begin{cases} \exists L = (L_1, L_2) : \text{ automorphisme affine de } \mathbb{F}_2^n \times \mathbb{F}_2^m | \\ F_1(x) = L_1(x, F(x)) \text{ est une permutation de } \mathbb{F}_2^n \text{ et} \\ \qquad\qquad G = F_2 \circ F_1^{-1} \text{ avec } F_2(x) = L_2(x, F(x)) \end{cases} \end{aligned}$$

Proposition 5.4.3. *L'équivalence affine étendue implique l'équivalence CCZ.*

Démonstration. Soient F et G deux (n,m)-fonctions telles que $F \sim_{EA.eq} G$, $G = A_1 \circ F \circ A_2 + A_3$. On définit la fonction $L = (L_1, L_2)$ de $\mathbb{F}_2^n \times \mathbb{F}_2^m$ dans lui même par :

$$\begin{cases} L_1(x,y) &= A_2^{-1}(x) \\ L_2(x,y) &= A_3 \circ A_2^{-1}(x) + A_1(y) \end{cases}$$

L est un automorphisme affine puisque les fonctions A_1, A_2 sont des permutations affines et A_3 est une fonction affine.
La fonction $F_1(x) = L_1(x, F(x)) = A_2^{-1}(x)$ est une permutation d'éléments de $\mathbb{F}_2^n$ car A_2 l'est, de plus on a

$$\begin{aligned} G \circ F_1(x) &= A_1 \circ F \circ A_2(F_1(x)) + A_3(F_1(x)) \\ &= A_1 \circ F \circ A_2 \circ A_2^{-1}(x) + A_3 \circ A_2^{-1}(x) \\ &= A_1 \circ F(x) + A_3 \circ A_2^{-1}(x) \\ &= A_1(y) + A_3 \circ A_2^{-1}(x), y = F(x) \\ &= L_2(x, F(x)) \end{aligned}$$

Alors F est CCZ-équivalente à G. □

Proposition 5.4.4. *Deux fonctions CCZ équivalentes ont la même immunité algébrique du graphe.*

Démonstration. Soient F et G deux fonctions CCZ équivalentes et $L = (L_1, L_2)$ un automorphisme affine qui transforme le graphe de la fonction F en le graphe de la fonction G. Si $L(x,y) = (x', y')$, on a

$$\begin{aligned} p \text{ est un annulateur du graphe de } F &\Leftrightarrow p(x,y) = 0, \forall (x,y) \text{ vérifiant } y = F(x) \\ &\Leftrightarrow p \circ L^{-1}(x', y') = 0, \forall (x', y') \text{ vérifiant } y' = G(x') \\ &\Leftrightarrow p \circ L^{-1} \text{ est un annulateur du graphe de } G. \end{aligned}$$

L^{-1} étant de degré 1 puisque L est affine, alors le degré de p est égal au degré de $p \circ L^{-1}$, en considérant un annulateur de minimum degré on aura le résultat. □

Une conséquence directe de la Proposition 5.4.3 et de la Proposition 5.4.4 est le résultat suivant

Corollaire 5.4.1. *Deux fonctions affinement équivalentes de façon étendue ont la même immunité algébrique du graphe.*

Chapitre 6

Bornes concernant les différentes notions d'immunité algébrique

6.1 Relations entre les différentes notions d'immunité algébrique d'une fonction vectorielle

Il peut exister des fonctions vectorielles ayant une immunité algébrique élevée d'un certain type et en ayant une petite pour un autre type. Dans ce chapitre on cherche des inégalités entre les différentes définitions de l'immunité algébrique d'une fonction vectorielle et on caractérise les cas d'égalité.
Remarquons que la définition 5.1.3 (de l'AI) n'est intéressant que si $m \ll n$ (c'est le cas par exemple pour les boites S utilisées dans les schémas à flots), puisque si $m \approx n$ alors $|F^{-1}(z)| \approx 1$ pour tout z dans $\mathbb{F}_2^m$ ce qui donne $AI(F) \leq 1$. Tandis que la définition 5.1.4 (de l'AI_{gr}) est plus significative quand m est proche de n (qui est le cas des boites S dans les schémas par blocs).

6.1.1 Bornes concernant AI et AI_{comp}

Par définition, le degré algébrique (resp. la nonlinéarité, l'immunité aux corrélations) d'une fonction vectorielle est égal au degré algébrique (resp. la nonlinéarité, l'immunité aux corrélations) minimum de ses fonctions composantes. Cette propriété n'est pas toujours valable pour l'immunité algébrique comme on le verra dans ce qui suit. On

commence par le résultat suivant :

Proposition 6.1.1. *Pour toute fonction vectorielle F de $\mathbb{F}_2^n$ dans $\mathbb{F}_2^m$, on a :*

$$AI(F) \leq AI_{comp}(F)$$

On utilise pour la preuve de ce résultat le lemme suivant :

Lemme 6.1.1. *Si A et B sont deux ensembles non vide de $\mathbb{F}_2^n$, alors :*

$$A \subseteq B \Rightarrow AI(A) \leq AI(B)$$

Démonstration. (de la Proposition 6.1.1)
Soit $x \in \mathbb{F}_2^n$ et soit $z, v \in \mathbb{F}_2^m$ tels que $v \neq 0$ et $AI_{comp}(F) = AI(v \cdot F) = \min\{AI(v \cdot F)^{-1}(0), AI(v \cdot F)^{-1}(1)\}$. On a alors, $v \cdot F(x) = v \cdot z$ ou $v \cdot F(x) = v \cdot z + 1$; ce qui signifie que $F^{-1}(z) \subseteq (v \cdot F)^{-1}(v \cdot z)$ ou $F^{-1}(z) \subseteq (v \cdot F)^{-1}(v \cdot z + 1)$, le Lemme 6.1.1 implique que $AI(F^{-1}(z)) \leq AI((v \cdot F)^{-1}(v \cdot z))$ ou $AI(F^{-1}(z)) \leq AI((v \cdot F)^{-1}(v \cdot z + 1))$. On en déduit que $AI(F) \leq \min\{AI(v \cdot F)^{-1}(0), AI(v \cdot F)^{-1}(1)\} = AI_{comp}(F)$. □

On verra dans la remarque suivante une condition suffisante pour l'égalité.

Remarque 6.1.1. *Soit z_0 un élément vérifiant $AI(F^{-1}(z_0) = \min_{z \in \mathbb{F}_2^m} AI(F^{-1}(z)) = AI(F)$. S'il existe un élément $\tilde{v}$ tel que $F^{-1}(z_0) = (\tilde{v} \cdot F)^{-1}(0)$ ou $F^{-1}(z_0) = (\tilde{v} \cdot F)^{-1}(1)$ alors :*

$$AI_{comp}(F) = AI(F)$$

Démonstration.

$$\begin{aligned} AI_{comp}(F) = \min_{v \in \mathbb{F}_2^m} AI(v \cdot F) &\leq AI(\tilde{v} \cdot F) \\ &\leq AI(\tilde{v} \cdot F)^{-1}(i), (i = 0 \vee i = 1) \\ &= AI(F^{-1}(z_0)) \\ &= AI(F) \end{aligned}$$

L'autre inégalité est déduite de la Proposition 6.1.1. □

6.1.2 Bornes concernant AI_{comp} et AI_{gr}

Carlet a remarqué dans son exposé à SETA 2008 [16] que

$$AI_{gr}(F) \leq AI_{comp}(F) + 1$$

En effet :
- Si g est un annulateur non nul de $v \cdot F$, où v est un vecteur non nul de $\mathbb{F}_2^m$ alors $h(x, y) = g(x)(v \cdot y)$ est un annulateur non nul du graphe de F ;
- 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.

6.1.3 Bornes concernant AI et AI_{gr}

On rappelle le résultat suivant qui est dans [2] :

Proposition 6.1.2. *Pour toute fonction F de $\mathbb{F}_2^n$ dans $\mathbb{F}_2^m$, on a :*

$$AI(F) \leq AI_{gr}(F) \leq AI(F) + m. \tag{6.1}$$

Démonstration. L'inégalité de gauche est immédiate, il suffit de considérer la restriction d'un annulateur du graphe $g(x, y)$ obtenue en fixant y à une valeur y_0 telle que $g(., y_0)$ n'est pas identiquement nul. Pour l'inégalité de droite, on sait qu'il existe z et un annulateur non nul $g(x)$ de $F^{-1}(z)$ de degré algébrique égal à $AI(F)$; la fonction $h(x, y) = g(x) \prod_{i=1}^{m}(y_i + z_i + 1)$ est un annulateur non nul du graphe de F de degré $AI(F) + m$. □

Une question naturelle se pose ; existe-t-il des fonctions F pour lesquelles on a des cas d'égalité ?
La réponse est oui, comme le montrent les résultats suivants. Considérons d'abord le cas d'une fonction Booléenne :

Proposition 6.1.3. *Soit f une fonction Booléenne à n variables alors, si f a une structure linéaire $(v, 1)$, v étant un vecteur de $\mathbb{F}_2^n$; alors on a $AI(f) = AI_{gr}(f)$.*

Démonstration. Si f a une structure linéaire $(v, 1)$ i.e $f(x + v) + f(x) = 1$ pour tout x alors $f^{-1}(0) = f^{-1}(1) + v$. Soit $p(x)$ un annulateur non nul de $f^{-1}(1)$ de degré d,

alors $p(x+v)$ est un annulateur non nul de $f^{-1}(0)$ de même degré. Alors la fonction Booléenne g définie sur $\mathbb{F}_2^n \times \mathbb{F}_2^m$ par $g(x,y) = yp(x) + (1+y)p(x+v)$ est un annulateur non nul du graphe de f de degré d. En effet : si $f(x) = 0$ alors $g(x,0) = p(x+v) = 0$ et si $f(x) = 1$ alors $g(x,1) = p(x) = 0$. Par conséquent, on a $AI_{gr}(f) \leq d = AI(f)$. Compte tenu de la première inégalité de (6.1) on a le résultat.

□

Exemple 6.1.1. *Soit f la fonction majorité, l'immunité algébrique de f est égale à $\lceil \frac{n}{2} \rceil$. Cette fonction est définie par :*

$$f(x) = \begin{cases} 1, & si\ w_H(x) \geq \frac{n}{2} \\ 0, & si\ w_H(x) < \frac{n}{2} \end{cases}$$

Alors on peut vérifier que cette fonction a une $((1,1,...,1),1)$ structure linéaire.

Remarque 6.1.2. *Si f a une structure linéaire $(v,0)$ i.e $f(x+v)+f(x) = 0$ alors ce raisonnement ne marche pas puisque si on considère la fonction $g(x,y) = (1+y)(p(x)+p(x+v))$ avec $p(x)$ un annulateur non nul de $f^{-1}(0)$, alors g pourrait être identiquement nulle (si $p(x+v)+p(x) = 0$).*

On peut généraliser ce résultat aux fonctions vectorielles :

Proposition 6.1.4. *Soit F une fonction de $\mathbb{F}_2^n$ dans $\mathbb{F}_2^m$. Si pour tout $b \in \mathbb{F}_2^m$, il existe $a \in \mathbb{F}_2^n$ tels que (a,b) soit une structure linéaire de F. Alors on a $AI(F) = AI_{gr}(F)$.*

Démonstration. Pour tout $i = 1, \cdots, m$, on note e_i le i-ième vecteur de la base canonique de $\mathbb{F}_2^m$ et v_i un vecteur de $\mathbb{F}_2^n$ tel que (v_i, e_i) soit une structure linéaire de F. Soit z_0 un élément de $\mathbb{F}_2^m$ tel que $AI(F)$ soit égal à l'immunité algébrique de l'ensemble $F^{-1}(z_0)$. Quitte à remplacer $F(x)$ par $F(x)+z_0$, on peut supposer sans perte de généralité que $z_0 = 0$. Soit $p(x)$ un annulateur non nul de degré $d = AI(F)$ de $F^{-1}(0)$. Alors la fonction booléenne à $(n+m)$ variables définie par

$$h(x,y) = \sum_{b \in \mathbb{F}_2^m} \prod_{i=1}^{m} (y_i + b_i + 1) p(x + \sum_{i=1}^{m} b_i v_i)$$

est un annulateur non nul du graphe de F. En effet, $\prod_{i=1}^{m}(y_i + b_i + 1)$ si et seulement si $y = b$; ainsi, pour tout $x \in \mathbb{F}_2^n$, si on note I le support du vecteur $F(x)$, on a

$$h(x, F(x)) = p(x + \sum_{i \in I} v_i).$$

On a $F(x+\sum_{i\in I} v_i) = F(x) + \sum_{i\in I} e_i = 0$, alors, $x + p(x+\sum_{i\in I} v_i \in F^{-1}(0)$, par conséquent $h(x, F(x)) = 0$.
La fonction h est de même degré que p. En effet, on peut écrire h sous la forme :

$$h(x,y) = \sum_{J\subseteq\{1,\cdots,m\}} \left(\prod_{i\in J} y_i\right) \phi_J(x).$$

Pour tout vecteur $b \in \mathbb{F}_2^m$, on note I le support de b, on a

$$\prod_{i=1}^{m}(y_i + b_i + 1) = \sum_{\substack{J\subseteq\{1,\cdots,m\}/\\ I\subseteq J}} \left(\prod_{i\in J} y_i\right).$$

Ainsi, $\phi_J(x) = \sum_{b\in\mathbb{F}_2^m/supp(b)\subseteq J} p(x+\sum_{i=1}^m b_i v_i)$ est une dérivée de p d'ordre $|J|$ et a un degré algébrique au plus $d^\circ p - |J|$. Alors, on a $d^\circ h \leq d^\circ p$ et en fait $d^\circ h = d^\circ p$, puisque la partie de $\phi_\emptyset$ indépendante de y dans $h(x,y)$ égale à $p(x)$. Par conséquent $AI_{gr}(F) \leq d = AI(F)$. Sachant que $AI(F) \leq AI_{gr}(F)$ on a $AI(F) = AI_{gr}(F)$. □

La condition sur F dans la Proposition 6.1.4 est équivalent à dire que : $\mathbb{F}_2^n$ est la somme directe de deux sous espaces E et E' et il existe une fonction linéaire surjective L de E dans $\mathbb{F}_2^m$ telle que, pour tout $x \in E$ et tout $x' \in E'$ nous avons $F(x+x') = L(x) + F(x')$. Cette condition est clairement suffisante puisque pour tout $v \in E$, on a $D_v F(x+x') = L(v)$. Elle est aussi nécessaire puisqu'on sait que l'ensemble des éléments v tels que il existe a tels que (v,a) est une structure linéaire de F est un sous espace vectoriel de $\mathbb{F}_2^n$ (appelé le noyau linéaire de F). On peut prendre E ce sous espace ; donc E' peut être n'importe quel sous espace vectoriel de $\mathbb{F}_2^n$, supplémentaire à E.

Cherchons maintenant des cas d'égalité pour l'inégalité de droite de (6.1), on va montrer d'abord que ces cas d'égalité sont impossibles pour certaines valeurs de m, en montrant une borne qui est souvent plus fine que l'inégalité droite de (6.1).

6.1.4 Amélioration de la borne $AI_{gr} \leq AI + m$

D'après [2] on sait que pour toute (n,m)-fonction vectorielle F on a $AI_{gr}(F) \leq \min\{D, \sum_{i=0}^{D}\binom{n+m}{i} > 2^n\} = D_{n,m}$. On verra que pour certaines valeurs de n et m, on a $D_{n,m} \leq AI(F) + m$ même si $AI(F) = 0$. Pour voir ceci, on cherche une condition suffisante pour que $AI_{gr}(F) \leq m$.

Proposition 6.1.5. *Soient (n,m) un couple d'entiers positifs. Si $1 \leq n \leq 2m$ ou si*

$25 \leq n \leq 3m$, *alors on a*

$$\sum_{i=0}^{m} \binom{n+m}{i} > 2^n, \text{ c'est à dire que } D_{n,m} \leq m. \tag{6.2}$$

Démonstration. Pour tout $m \geq 1$, on a $\sum_{i=0}^{m+1} \binom{3m+3}{i} > 4 \sum_{i=0}^{m} \binom{3m}{i}$. En effet, en itérant trois fois la relation $\binom{N}{i} = \binom{N-1}{i} + \binom{N-1}{i-1}$, on a

$$\sum_{i=0}^{m+1} \binom{3m+3}{i} =$$

$$\sum_{i=0}^{m+1} \binom{3m}{i} + 3\sum_{i=0}^{m} \binom{3m}{i} + 3\sum_{i=0}^{m-1} \binom{3m}{i} + \sum_{i=0}^{m-2} \binom{3m}{i} > 4\sum_{i=0}^{m} \binom{3m}{i}, \ \forall m \geq 1.$$

Comme on peut vérifier directement que pour $m = 1$ on a $\sum_{i=0}^{m} \binom{3m}{i} = 2^{2m}$, cela implique par récurrence que $\sum_{i=0}^{m} \binom{3m}{i} > 2^{2m}$ pour tout $m \geq 2$ et donc que, pour tout $n \geq 4$ pair et tout $m \geq n/2$, on a $\sum_{i=0}^{m} \binom{n+m}{i} > 2^n$ (puisque $\sum_{i=0}^{m} \binom{n+m}{i} \geq \sum_{i=0}^{n/2} \binom{3n/2}{i} > 2^n$). Et on en déduit que pour tout $n \geq 3$ impair et tout $m \geq n/2$ on a $\sum_{i=0}^{m} \binom{n+m}{i} > \frac{1}{2} \sum_{i=0}^{m} \binom{n+1+m}{i} > 2^n$. Donc pour tout $n \geq 3$ et tout $m \geq n/2$ on a $\sum_{i=0}^{m} \binom{n+m}{i} > 2^n$.

Pour tout $m \geq 9$, on a $\sum_{i=0}^{m+1} \binom{4m+4}{i} \geq 8 \sum_{i=0}^{m} \binom{4m}{i}$. En effet, en itérant quatre fois la relation $\binom{N}{i} = \binom{N-1}{i} + \binom{N-1}{i-1}$, on a

$$\sum_{i=0}^{m+1} \binom{4m+4}{i} =$$

$$\sum_{i=0}^{m+1} \binom{4m}{i} + 4\sum_{i=0}^{m} \binom{4m}{i} + 6\sum_{i=0}^{m-1} \binom{4m}{i} + 4\sum_{i=0}^{m-2} \binom{4m}{i} + \sum_{i=0}^{m-3} \binom{4m}{i} =$$

$$8\sum_{i=0}^{m} \binom{4m}{i} + \binom{4m}{m+1} - 3\binom{4m}{m} + 3\sum_{i=0}^{m-1} \binom{4m}{i} + 4\sum_{i=0}^{m-2} \binom{4m}{i} + \sum_{i=0}^{m-3} \binom{4m}{i} \geq$$

$$8\sum_{i=0}^{m} \binom{4m}{i} + \binom{4m}{m+1} - 3\binom{4m}{m} + 3\binom{4m}{m-1}.$$

Or, on a

$$\binom{4m}{m+1} - 3\binom{4m}{m} + 3\binom{4m}{m-1} = \left(\frac{3m}{m+1} - 3 + \frac{m}{3m+1}\right)\binom{4m}{m} =$$

$$\frac{3m(3m+1) - 3(m+1)(3m+1) + m(m+1)}{(m+1)(3m+1)}\binom{4m}{m} =$$

$$\frac{m^2+m-3}{(m+1)(3m+1)}\binom{4m}{m} \geq 0,\ \forall m \geq 9.$$

Comme on peut vérifier directement que pour $m = 9$ on a $\sum_{i=0}^{m}\binom{4m}{i} = 135142796 > 2^{3m} = 134217728$, cela implique par récurrence que $\sum_{i=0}^{m}\binom{4m}{i} > 2^{3m}$ pour tout $m \geq 9$ et donc que, pour tout $n \geq 27$ divisible par 3 et tout $m \geq n/3$, on a $\sum_{i=0}^{m}\binom{n+m}{i} > 2^n$. Et on en déduit que pour tout $n \geq 26$ congru à -1 modulo 3 et tout $m \geq n/3$, on a $\sum_{i=0}^{m}\binom{n+m}{i} > \frac{1}{2}\sum_{i=0}^{m}\binom{n+1+m}{i} > 2^n$ et que pour tout $n \geq 25$ congru à -2 modulo 3 et tout $m \geq n/3$, on a $\sum_{i=0}^{m}\binom{n+m}{i} > \frac{1}{4}\sum_{i=0}^{m}\binom{n+2+m}{i} > 2^n$. Donc, pour tout $n \geq 25$ et tout $m \geq n/3$, on a $\sum_{i=0}^{m}\binom{n+m}{i} > 2^n$. □

En appliquant la Proposition 6.1.5 à $n - m$ au lieu de n, on a

Corollaire 6.1.1. *Soient (n, m) un couple d'entiers positifs. Si $2 \leq n \leq 3m$ ou si $20 \leq n \leq 4m$, alors on a $d_{n,m} \leq m$.*

6.2 Relations entre AI et autres critères cryptographiques des fonctions vectorielles

6.2.1 *AI* et poids de Hamming des fonctions composantes

Théorème 6.2.1. *Soit F une (n, m)-fonction. S'il existe $v \in \mathbb{F}_2^m$ non nul, $d \in \mathbb{N}$, $1 \leq d \leq n$ tels que $w_H(v \cdot F) < \sum_{i=0}^{d}\binom{n}{i}$ ou $w_H(v \cdot F) > \sum_{i=0}^{n-d-1}\binom{n}{i}$, alors $AI(F) \leq d$.*

Démonstration. Appliquons le Corollaire 4.4.1 à la fonction Booléenne $v \cdot F$, nous avons $AI(v \cdot F) \leq d$, donc $AI_{comp}(F) = \min\{AI(v \cdot F), v \in \mathbb{F}_2^m\} \leq d$. Par conséquent $AI(F) \leq AI_{comp}(F) \leq d$ □

Corollaire 6.2.1. *Soit F une (n, m)-fonction, si $AI(F) > d$, alors*

$$\sum_{i=0}^{d}\binom{n}{i} \leq w_H(v \cdot F) \leq 2^n - \sum_{i=0}^{d}\binom{n}{i}, \forall v \in \mathbb{F}_2^m.$$

6.2.2 *AI* et spectre de Walsh des fonctions vectorielles

Théorème 6.2.2. *Soit F une (n,m)-fonction. S'il existe $v \in \mathbb{F}_2^m$ non nul, $\lambda \in \mathbb{F}_2^n$ et $d \in \mathbb{N}$, $2 \leq d \leq n$ tels que la transformée de Walsh de F vérifie*

$$|W_F(\lambda, v)| > 2^n - 2\sum_{i=0}^{d-1} \binom{n}{i}$$

alors $AI(F) \leq d$.

Démonstration. On a $W_F(\lambda, v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{v \cdot F(x) + \lambda \cdot x} = 2^n - 2w_H(v \cdot F(x) + \lambda \cdot x)$.
Or $|W_F(\lambda, v)| > 2^n - 2\sum_{i=0}^{d-1} \binom{n}{i}$, nous avons donc :

$$w_H(v \cdot F(x) + \lambda \cdot x) < \sum_{i=0}^{d-1} \binom{n}{i} \text{ ou } w_H(v \cdot F(x) + \lambda \cdot x) > 2^n - \sum_{i=0}^{d-1} \binom{n}{i}.$$

Appliquons le Corollaire 4.4.1 à la fonction $v \cdot F + \lambda \cdot x$, nous avons $AI(v \cdot F + \lambda \cdot x) \leq d-1$.
Compte tenu du Lemme 4.4.1, nous avons $AI(v \cdot F) - 1 \leq AI(v \cdot F + \lambda \cdot x) \leq d - 1$.
Ainsi,
$AI(v \cdot F) \leq d$. Puisque $AI(F) \leq AI_{comp}(F)$, nous avons $AI(F) \leq d$. □

Corollaire 6.2.2. *Soit F une (n,m)-fonction et soit d un entier positif. Si $AI(F) > d$, alors la transformée de Walsh de F satisfait*

$$|W_F(\lambda, v)| \leq 2^n - 2\sum_{i=0}^{d-1} \binom{n}{i}, \forall \lambda \in \mathbb{F}_2^n.$$

6.2.3 *AI* et non-linéarité des fonctions vectorielles

Théorème 6.2.3. *Soit F une (n,m)-fonction et soit d un entier positif. Si $AI(F) > d+1$, alors*

$$\sum_{i=0}^{d} \binom{n}{i} \leq nl(F) \leq 2^n - \sum_{i=0}^{d} \binom{n}{i}.$$

Démonstration. On a $AI_{comp}(F) \geq AI(F) > d+1$, ce qui veut dire que $\min\{AI(v \cdot F) : v \in \mathbb{F}_2^m\} > d+1$. Alors pour tout $v \in \mathbb{F}_2^m$, on a $AI(v \cdot F) > d+1$. Appliquons le Corollaire 4.4.2 à la fonction Booléenne $v \cdot F$, on a

$$\sum_{i=0}^{d} \binom{n}{i} \leq nl(v \cdot F) \leq 2^n - \sum_{i=0}^{d} \binom{n}{i}.$$

v étant arbitraire, on a alors

$$\sum_{i=0}^{d}\binom{n}{i} \leq \min_{v\in\mathbb{F}_2^m}\{nl(v\cdot F)\} = nl(F) \leq 2^n - \sum_{i=0}^{d}\binom{n}{i}.$$

□

Corollaire 6.2.3. *Soit F une (n,m)-fonction. S'il existe $d \in \mathbb{N}$, $1 \leq d \leq n$ tel que $nl(F) < \sum_{i=0}^{d}\binom{n}{i}$ ou $nl(F) > 2^n - \sum_{i=0}^{d}\binom{n}{i}$, alors $AI(F) \leq d+1$.*

C. Carlet a montré dans [16] que la borne inférieure de Lobanov sur la non-linéarité des fonctions Booléennes peut être généralisée aux (n,m)-fonctions comme suit :

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

En appliquant la borne de Lobanov aux fonctions composantes de F, on trouve,

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

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

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

6.2.4 *AI* et non-linéarité d'ordre r des fonctions vectorielles

C. Carlet a montré aussi dans [16] que la borne inférieure sur la non-linéarité d'ordre r des fonctions Booléennes peut être généralisée aux (n,m)-fonctions comme suit :

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

et

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

(c'est à dire que la première borne peut être améliorée comme pour les fonctions Booléennes). Notons que, en appliquant les bornes valables pour les fonctions Booléennes aux fonctions composantes de F, on obtient

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

et

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

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

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

et

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

6.2.5 *AI* et résilience des fonctions vectorielles

Théorème 6.2.4. *Soit F une (n,m)-fonction. Si F est t-résiliente, $0 \leq t < n-1$, alors $AI(F) \leq n-t-1$.*

Démonstration. F est t-résiliente, alors $v \cdot F$ est t-résiliente pour tout $v \neq 0$, $v \in \mathbb{F}_2^m$. Compte tenu du Corollaire 2.4.2, on a $\deg(v \cdot F) + t \leq n-1$.
On a $AI(F) \leq AI_{comp}(F) \leq AI(v \cdot F) \leq \deg(F)$, ce qui implique que $AI(F) + t \leq n-1$. □

Chapitre 7

Borne inférieure asymptotique sur AI des fonctions vectorielles

Les auteurs de l'article [46] ont trouvé une borne supérieure asymptotique sur la probabilité d'existence d'annulateurs d'une fonction Booléenne aléatoire équilibrée, il en résulte une borne inférieure asymptotique sur l'immunité algébrique. Un peu plus tard, F. Didier [28] a montré une nouvelle borne sur la probabilité d'erreur de correction d'un code dans le cas d'un canal à effacements, et traduit son résultat en termes d'existence d'annulateurs d'une fonction Booléenne lorsque le code utilisé est de Reed-Muller. La borne obtenue ainsi est meilleure que la borne trouvée dans [46]. Une conséquence de cette recherche qui s'inscrit dans notre sujet est une borne inférieure asymptotique de l'immunité algébrique d'une fonction Booléenne [65]. Dans ce chapitre, nous reprenons le travail de F.Didier [28] sur les fonctions Booléennes, nous complétons et généralisons les démonstrations ; nous prouvons tous les résultats intermédiaires admis dans [28], et nous étendons les résultats obtenus aux fonctions vectorielles.

7.1 Notions préliminaires

Dans cette section, nous rappelons quelques notions de base sur les codes de Reed Muller, les fonctions Booléennes et vectorielles que nous avons pas abordé au Chapitre 2 et que nous aurons besoin dans les sections suivantes.

On rappelle que l'ensemble des annulateurs de degrés au plus r d'un sous-ensemble E de $\mathbb{F}_2^n$, qu'on notera $Ann_r(E)$ est un sous espace vectoriel de $RM(r,n)$ [46], et on

note $AI(E)$ le nombre $\min\{\deg g, g \neq 0; g \in Ann(E)\} = \min\{r/Ann_r(E) \neq \{0\}\}$; (0 désigne la fonction nulle).

7.1.1 Distances de Hamming généralisées

La définition des distances de Hamming généralisées nécessite d'introduire la définition du support d'un code.

Définition 7.1.1. *Soit C un code linéaire de longueur N. Le support de C, noté $supp(C)$, est l'ensemble des positions non toujours nulles pour C, i.e.*

$$supp(C) = \{j = 1, \ldots, N/\exists c = (c_1, \ldots, c_N) \in C, c_j \neq 0\} = \bigcup_{c \in C} supp(c)$$

Définition 7.1.2. *Soit C un code linéaire de longueur N et de dimension k. Les distances de Hamming généralisées de C sont définies pour $i = 1, \ldots, k$ par :*

$$d_i = \min_{V \in \mathcal{V}_i}(|supp(V)|)$$

où $\mathcal{V}_i$ est l'ensemble des sous codes de C de dimension i.

On convient de poser $d_0 = 0$.
Nous donnerons ici les résultats sur les distances de Hamming généralisées nécessaires pour la suite.

Théorème 7.1.1. *[63] Les distances de Hamming généralisées d'un code linéaire C sont strictement croissantes.*

Démonstration. Les relations $d_{i-1} \leq d_i$, pour tout $i = 1, \ldots, k$ découlent directement de la définition ; il reste à prouver que les inégalités sont strictes. Soit V_i un sous-code de C de dimension i tel que $|supp(V_i)| = d_i$. Soit j un élément de $supp(V_i)$, l'espace $V_i^j = \{x \in V_i : x_j = 0\}$ est un sous-code de V_i de dimension $i - 1$, donc on a $d_{i-1} \leq |supp(V_i^j)| \leq |supp(V_i)| - 1 = d_i - 1$. □

Distances généralisées des codes de Reed-Muller

Considérons le code $RM(r, n)$ de dimension k. Ordonnons les éléments de l'ensemble $V_n^{\leq r} = \{x \in \mathbb{F}_2^n, |x| \leq r\}$ par l'ordre usuel (l'ordre lexicographique inverse).

$V_n^{\leq r}$ est de cardinal $k = \sum_{i=0}^{r} \binom{n}{i}$; ainsi, les éléments de $V_n^{\leq r}$ sont ordonnés comme $x^{(0)} \prec x^{(1)} \prec \ldots \prec x^{(k-1)}$. On associe à un élément $x^{(i)}$ de $V_n^{\leq r}$ un entier n_i de l'ensemble $\{0, 1, \ldots, 2^n - 1\}$ par la relation :

$$n_i = \sum_{j=1}^{n} x_j^{(i)} 2^{j-1}$$

On a donc $x^{(i)} \prec x^{(j)}$ si et seulement si $n_i < n_j$.

Proposition 7.1.1. *[63] Les distances de Hamming généralisées de $RM(r, n)$ sont données par*

$$d_i = 2^n - n_{k-i},$$

où $k = \sum_{i=0}^{r} \binom{n}{i}$ est la dimension de $RM(r, n)$.

Exemple 7.1.1. *Pour le code $RM(r, n)$ on a : $d_1 = 2^{n-r}$ et $d_k = 2^n$, en effet :*

- *$d_k = 2^n - n_0 = 2^n$,*
- *$d_1 = 2^n - n_{k-1}$, l'élément associé à n_{k-1} est $x^{(k-1)} = (0, 0, \ldots, 0, \overbrace{1, 1, \ldots, 1}^{r})$, donc $n_{k-1} = 2^{n-r} + 2^{n-r+1} + \cdots + 2^{n-1} = 2^n - 2^{n-r}$, d'où $d_1 = 2^{n-r}$.*

7.1.2 Codes à rendement cohérent

La notion de code à rendement cohérent est utile dans la démonstration du Théorème 7.2.2 ci-dessous.

On sait que le rendement d'un code linéaire $[N, k, d]$ est égal au rapport de sa dimension sur sa longueur : $R = k/N$. D'une façon analogue, on définit le rendement d'un sous-code comme le rapport de sa dimension sur la taille de son support. La fraction i/d_i correspond au rendement maximal d'un sous-code de dimension i.

Définition 7.1.3. *(Codes à rendement cohérent)*
Soient C un code de longueur N et de dimension k et soient $d_i, i = 1, \ldots, k$ ses poids de Hamming généralisés, alors C est à rendement cohérent si

$$\forall i = 1, \ldots, k, \frac{i}{d_i} \leq \frac{k}{N} \tag{7.1}$$

Les codes à rendement cohérent sont donc des codes pour lesquels le rendement de n'importe quel sous-code est plus petit ou égal au rendement du code tout entier.

Lemme 7.1.1. *[28] Les codes de Reed-Muller sont à rendement cohérent.*

7.2 Probabilité d'existence d'annulateurs de degré borné d'un sous-ensemble donné

Dans cette section, nous expliquerons comment les résultats de Didier [28] sur la probabilité d'erreurs de correction des codes de Reed-Muller $RM(r,n)$ sur un canal à effacements peuvent être traduits en termes d'existence d'annulateurs de degré borné d'un sous-ensemble de $\mathbb{F}_2^n$ choisi selon une loi de probabilité uniforme. Nous expliquerons aussi comment peut-on en déduire une borne supérieure sur une telle probabilité impliquant une borne inférieure asymptotique (presque sûre) sur l'immunité algébrique des fonctions Booléennes, meilleure que la borne donnée dans [46].

7.2.1 Borne sur la probabilité d'existence d'annulateurs

Nous pouvons considérer un motif d'effacement de longueur $N = 2^n$ comme la liste des valeurs d'une fonction Booléenne à n variables f . Si nous avons une autre fonction Booléenne à n variables g telles que $supp(g) \subset supp(f)$ alors nous avons $fg = g$. En d'autre termes, le sous espace des mots de code de $RM(r,n)$ inclus dans le motif d'effacement peuvent être vu comme le sous espace des fonctions de degrés au plus r qui sont invariantes par la multiplication par f, c'est à dire, ces fonctions sont des annulateurs du complément du support de f. Alors, comme nous avons vu au Chapitre 4 :
Chercher les annulateurs de degré au plus r d'une fonction Booléenne f est la même chose que décoder $RM(r,n)$ en présence d'un motif d'effacements égal au vecteur des valeurs de $1+f$. Donc, trouver une borne sur la probabilité de correction est la même chose que de trouver une borne sur la probabilité d'existence d'annulaeurs.

7.2.2 Borne préliminaire sur la probabilité d'existence d'annulateurs en fonction des distances généralisées

Soit E un sous-ensemble non vide de $\mathbb{F}_2^n$. On note $|E|$ le cardinal de E et E^c son complémentaire.

Théorème 7.2.1. *Soit w un entier positif fixé et soit $d_1, d_2, \ldots, d_k$ les poids de Hamming généralisés de $RM(r,n)$, avec $k = \sum_{i=0}^{r} \binom{n}{i}$ la dimension de $RM(r,n)$. Supposons que E est choisi selon une loi de probabilité uniforme parmi les sous-ensembles de $\mathbb{F}_2^n$ de cardinal w, alors*

1. *Si* $w > 2^n - 2^{n-r}$ *alors* E *n'a pas d'annulateurs de degré au plus* r,
2. *Si* $w \leq 2^n - 2^{n-r}$ *alors la probabilité que* E *ait au moins un annulateur non nul à* n *variables de degré inférieur ou égal à* r *satisfait :*

$$\Pr\{Ann_r(E) \neq \{0\}\} \leq \prod_{i=2^n-w+1}^{2^n} \left(\frac{i-d_1}{i}\right) \prod_{j=2}^{k} \left(\frac{d_j}{d_j - d_1}\right). \tag{7.2}$$

La deuxième partie du Théorème 7.2.1 et les résultats auxilliaires nécessaires (Lemme 7.2.1, ..., Lemme 7.2.4) n'ont pas de différences majeures avec l'analyse donnée dans [28] ; nous les donnerons pour deux raisons : pour rendre la lecture autonome et parce que quelques détails ont été omis dans [28].

La démonstration du Théorème 7.2.1 se fait en plusieurs étapes.
Considérons les w éléments $x^1, \ldots, x^w$ de E pris dans un ordre choisi uniformément.

Définition 7.2.1. *Soit* $Ann_r^j(E)$*, le sous-espace des fonctions Booléennes à* n *variables, de degrés au plus* r*, qui sont nulles sur les* j *premiers éléments de* E*. Pour tout* $0 \leq i \leq k$ *et* $1 \leq j \leq w$*, soit* $E_{i,j}$ *l'événement : "*$\dim(Ann_r^j(E)) = i$*". On note* $p_{i,j}$ *la probabilité de l'événement* $E_{i,j}$*. On définit* $p_{i,0} = 0$ *pour tout* $i = 0, 1, \ldots, k-1$*,* $p_{k,0} = 1$ *et* $p_{i,j} = 0$ *pour tout* $i \geq k$ *et* $j = 1, \ldots, w$*.*

Par définition nous avons $Ann_r(E) = Ann_r^w(E)$ et E n'a pas d'annulateur non nul de degré au plus r si et seulement si $\dim(Ann_r^w(E)) = 0$. Donc la probabilité que E n'admette pas d'annulateur non trivial de degré au plus r est égale à $p_{0,w}$.

Supposons que $\dim(Ann_r^j(E)) = i$. Considérons l'espace $Ann_r^{j+1}(E)$. Il y a deux possibilités pour sa dimension :

- S'il existe une fonction Booléenne de $Ann_r^j(E)$ qui est non nulle en x^{j+1}, alors $\dim(Ann_r^{j+1}(E)) = \dim(Ann_r^j(E)) - 1 = i - 1$,
- Si toutes les fonctions Booléennes de $Ann_r^j(E)$ sont nulles en x^{j+1}, alors $\dim(Ann_r^{j+1}(E)) = \dim(Ann_r^j(E)) = i$.

Cela nous conduit à la définition des probabilités de transition entre les événements.

Définition 7.2.2. *Pour tout* $1 \leq i \leq k$ *et* $1 \leq j \leq w-1$*, on note* $t_{i,j}$ *la probabilité de l'événement* $E_{i-1,j+1}$ *sachant que l'événement* $E_{i,j}$ *est vrai, i.e :*

$$t_{i,j} = P(E_{i-1,j+1} | E_{i,j}).$$

Donc

$$1 - t_{i,j} = P(E_{i,j+1} | E_{i,j}).$$

On définit $t_{i,j}$ *pour tout* $i = 0$ *et* $j = 0$ *comme suit*

– $t_{0,j} = 0$ *pour tout* $j = 1, \ldots, w-1$,
– $t_{i,0} = 0$ *pour tout* $i = 0, \ldots, k-1$, $t_{k,0} = 1$,
et on définit $t_{i,j}$ *pour tout* $i \geq k+1$ *par* $t_{i,j} = 0$.

La borne inférieure ci-dessous est utile pour la suite.

Lemme 7.2.1 ([28])**.** *Pour tout* $0 \leq i \leq k$ *et* $1 \leq j \leq w-1$, *soit*

$$t'_{i,j} = \frac{d_i}{2^n - j}, t'_{k,0} = 1, \text{ et } t'_{i,0} = 0 \text{ pour tout } i = 0, \ldots, k-1. \tag{7.3}$$

Alors, on a $t_{i,j} \geq t'_{i,j}$. *De plus, l'événement* $E_{i,j}$ *est impossible si* $d_i > 2^n - j$.

Démonstration. Pour être autonomes, nous donnons une démonstration avec plus de détails que celle dans [28]. Supposons que $E_{i,j}$ est vrai. On considère la dimension de l'espace $Ann_r^{j+1}(E)$, il y a deux cas ; comme nous l'avons vu plus haut :

– S'il existe une fonction Booléenne de $Ann_r^j(E)$ non nulle en x^{j+1}, alors $j+1$ est dans le support de $Ann_r^j(E)$ et $\dim(Ann_r^{j+1}(E)) = \dim(Ann_r^j(E)) - 1$;
– Si toutes les fonctions Booléennes de $Ann_r^j(E)$ sont nulles en x^{j+1}, alors $j+1$ est en dehors du support de $Ann_r^j(E)$ et le sous espace $Ann_r^{j+1}(E)$ reste égal à $Ann_r^j(E)$.

Ainsi, $t_{i,j}$ est la probabilité que $j+1$ soit dans le support de $Ann_r^j(E)$. Cette probabilité est égale au rapport du cardinal du support de $Ann_r^j(E)$ sur le nombre d'éléments de $\mathbb{F}_2^n$ parmi lesquels on peut choisir x^{j+1}, c'est à dire $|\mathbb{F}_2^n \setminus \{x^1, \ldots, x^j\}| = 2^n - j$, en effet, le choix de x^{j+1} dans $|\mathbb{F}_2^n \setminus \{x^1, \ldots, x^j\}|$ est uniforme puisque celui de E est uniforme. Comme d_i est inférieur ou égal à $|supp(Ann_r^j(E))|$ puisque $Ann_r^j(E)$ est de dimension i, on a

$$t_{i,j} = \frac{|supp(Ann_r^j(E))|}{2^n - j} \geq \frac{d_i}{2^n - j} = t'_{i,j}.$$

Donc, $E_{i,j}$ est impossible lorsque $d_i > 2^n - j$, puisque $t_{i,j} > 1$ étant impossible, $E_{i,j}$ est donc faux. □

Les probabilités $p_{i,j}$ et $t_{i,j}$ sont liées par une relation de récurrence, donnée par le lemme suivant.

Lemme 7.2.2. *On a pour tout* $i = 0, 1, \ldots, k-1$ *et tout* $j = 0, 1, \ldots, w-1$

$$p_{i,j+1} = p_{i,j}(1 - t_{i,j}) + p_{i+1,j} t_{i+1,j}. \tag{7.4}$$

Démonstration. Pour $j = 0$ la Relation (7.4) est vérifiée quelque soit $i = 0, 1, \ldots, k-1$ d'après notre convention, car $p_{i,1} = 0$ pour $i < k-1$; $p_{k-1,1} = t_{k,0} = 1$ et $p_{k,1} =$

$1 - t_{k,0} = 0$. En utilisant la théorie des probabilités et la définition des $t_{i,j}$, on a pour tout $i = 0, 1, \ldots, k-1$ et tout $j = 1, \ldots, w-1$

$$\begin{aligned} p_{i,j+1} = P(E_{i,j+1}) &= P(E_{i,j+1} \cap E_{i,j}) + P(E_{i,j+1} \cap E_{i+1,j}) \\ &= P(E_{i,j+1}|E_{i,j})P(E_{i,j}) + P(E_{i,j+1}|E_{i+1,j})P(E_{i+1,j}) \\ &= (1 - t_{i,j})p_{i,j} + t_{i+1,j}p_{i+1,j}. \end{aligned}$$

D'où le lemme. □

En vue d'exploiter la Relation (7.4) et le fait que $t_{i,j} \geq t'_{i,j}$, nous définissons d'une façon analogue une suite double $(p'_{i,j})_{i,j}$, par $p'_{k,0} = 1$, $p'_{i,0} = 0$ pour tout $i = 0, 1, \ldots, k-1$, $p'_{i,j} = 0$ pour tout $i \geq k$ et tout $j = 1, \ldots, w$ et par la relation de récurrence

$$p'_{i,j+1} = p'_{i,j}(1 - t'_{i,j}) + p'_{i+1,j}t'_{i+1,j} \tag{7.5}$$

pour tout $i = 0, 1, \ldots, k-1$ et tout $j = 0, 1, \ldots, w-1$.

Les Relations $t_{i,j} \geq t'_{i,j}$, (7.4) et (7.5) permettent de montrer que $p_{0,w} \geq p'_{0,w}$. La preuve est plus simple si on considère les sommes partielles

$$S_{i,j} = \sum_{h=0}^{i} p_{h,j} \quad \text{et} \quad S'_{i,j} = \sum_{h=0}^{i} p'_{h,j}$$

on a

Lemme 7.2.3 ([28]). *Pour tout $j, 0 \leq j \leq w$ et pour tout $i, 0 \leq i \leq k$, on a :*

$$S'_{i,j} \leq S_{i,j} \tag{7.6}$$

De plus, pour $i = k$, on a pour tout $j = 0, \ldots, w$

$$S'_{k,j} = S_{k,j} = 1$$

Démonstration. Nous donnons une preuve originale de ce résultat à partir de celle donnée dans [28] qui est incomplète. La Relation (7.4) et la définition des $p_{i,j}$ et $t_{i,j}$ impliquent que pour tout $0 \leq i \leq k-1$ et tout $0 \leq j \leq w-1$, on a :

$$S_{i,j+1} = S_{i,j} - p_{0,j}t_{0,j} + p_{i+1,j}t_{i+1,j} = S_{i,j} + t_{i+1,j}(S_{i+1,j} - S_{i,j}) \tag{7.7}$$

car $t_{0,j} = 0$. La suite $S_{i,j}$ est stationnaire pour $i \geq k$ et tout $j, 0 \leq j \leq w$ fixé car pour ces valeurs de i on a $p_{i,j} = 0$.

Remarquons que nous avons la même relation entre les $S'_{i,j}$ et les $t'_{i,j}$,

$$S'_{i,j+1} = S'_{i,j} + t'_{i+1,j}(S'_{i+1,j} - S'_{i,j}) \tag{7.8}$$

On montre le lemme par récurrence sur j. Pour tout $i = 0, 1, \ldots, k-1$ fixé, on a

$$0 = S_{i,0} \geq S'_{i,0} = 0, \forall i = 0, 1, \ldots, k-1 \quad \text{et} \quad 1 = S_{k,0} \geq S'_{k,0} = 1.$$

Donc, l'Inégalité (7.6) est vraie pour $j = 0$. Supposons qu'elle est vraie à l'ordre j pour tout i fixé, montrons qu'elle reste vraie pour $j+1$. En utilisant la Relation (7.7), on a

$$\begin{aligned} S_{i,j+1} = S_{i,j} + t_{i+1,j}(S_{i+1,j} - S_{i,j}) &\geq S_{i,j} + t'_{i+1,j}(S_{i+1,j} - S_{i,j}), \\ &\quad (\text{car } t_{i+1,j} \geq t'_{i+1,j}) \\ &= (1 - t'_{i+1,j})S_{i,j} + t'_{i+1,j}S_{i+1,j} \\ &\geq (1 - t'_{i+1,j})S'_{i,j} + t'_{i+1,j}S'_{i+1,j}, \\ &\quad (\text{hypothèse de récurrence}) \\ &= S'_{i,j+1} \end{aligned}$$

Alors l'inégalité (7.6) est vraie pour tout $i = 0, \ldots, k$ et $j = 0, \ldots, w$.

En utilisant (7.8) pour $i = k$ on aura

$$\begin{aligned} S'_{k,j+1} = S'_{k,j} + t'_{k+1,j}(S'_{k+1,j} - S'_{k,j}) &= S'_{k,j}, (\text{car } t'_{k+1,j} = 0) \\ &= S'_{k,j-1}, (\text{car } t'_{k+1,j-1} = 0) \\ &= \ldots \\ &= S'_{k,0} = 1 \end{aligned}$$

On peut faire de même pour $S_{k,j}$. □

Grâce au Lemme 7.2.3, on trouve la borne supérieure suivante sur la probabilité d'existence d'annulateurs de E de degrés au plus r

$$\Pr\{Ann_r(E) \neq \{0\}\} = 1 - p_{0,w} = 1 - S_{0,w} \leq 1 - S'_{0,w} \tag{7.9}$$

Pour déduire de la Relation (7.9) une borne supérieure effective de la probabilité d'existence d'annulateur d'un ensemble, nous avons besoin de trouver une borne inférieure sur $S'_{0,w}$. Pour cela, nous écrivons la Relation (7.5) sous forme matricielle. Pour tout $j = 0, \ldots, w-1$, notons C_j le vecteur colonne

$$C_j = \begin{pmatrix} p'_{0,j} \\ \vdots \\ p'_{k,j} \end{pmatrix}$$

La Relation (7.5) est équivalente à

$$C_{j+1} = A_j C_j$$

où A_j est la matrice

$$A_j = \begin{pmatrix} 1-t'_{0,j} & t'_{1,j} & \cdots & 0 \\ 0 & 1-t'_{1,j} & \ddots & \vdots \\ \vdots & \cdots & \ddots & t'_{k,j} \\ 0 & \cdots & \cdots & 1-t'_{k,j} \end{pmatrix}$$

Les valeurs propres de cette matrice et celles de son adjoint A_j^T sont clairement les $1-t'_{i,j}, i = 0, 1, \ldots, k$. Elles sont distinctes car les $t'_{i,j}$ le sont d'après le Théorème 7.1.1. Alors, ces matrices sont diagonalisables et les espaces propres sont de dimension 1. Pour chercher une borne inférieure sur $S'_{0,w}$ (donc une borne supérieure sur $1 - S_{0,w}$), nous aurons considéré les vecteurs propres de A_j^T ; en effet, nous aurons en déduire une borne inférieure sur le produit scalaire entre les vecteurs (colonnes) de x et C_j. Ce produit scalaire égale à $C_j^T x$.

Lemme 7.2.4 ([28])**.** *Pour tout $0 \leq i \leq k, 0 \leq j \leq w$, les vecteurs propres de A_j^T associés à la valeur propre $1 - t'_{i,j}$ sont indépendants de j. Si ϑ_i est le vecteur propre ayant 1 à la première coordonnée non nulle, alors il vérifie*

$$\vartheta_i \cdot C_j \geq 1 - S_{i-1,j}$$

où "$\cdot$" désigne le produit scalaire dans $\mathbb{R}^{k+1}$.

Démonstration. Nous complétons et clarifions la preuve de [28]. La matrice A_j^T a $(k+1)$ valeurs propres distinctes. Si $x = (x_0, x_1, \ldots, x_k)$ est un vecteur propre de A_j^T associé à la valeur propre $(1 - t'_{i,j})$, alors il vérifie le système d'équations linéaires

$$\begin{cases} x_0(1-t'_{0,j}) &= (1-t'_{i,j})x_0 \\ x_0 t'_{1,j} + x_1(1-t'_{1,j}) &= (1-t'_{i,j})x_1 \\ \vdots &= \vdots \\ x_{i-1}t'_{i,j} + x_i(1-t'_{i,j}) &= (1-t'_{i,j})x_i \\ x_i t'_{i+1,j} + x_{i+1}(1-t'_{i+1,j}) &= (1-t'_{i,j})x_{i+1} \\ \vdots &= \vdots \\ x_{k-1}t'_{k,j} + x_k(1-t'_{k,j}) &= (1-t'_{i,j})x_k \end{cases}$$

En résolvant ce système d'équations, on voit que ϑ_i est le vecteur

$$\vartheta_i = \left(\underbrace{0, \ldots, 0}_{i}, 1, \frac{t'_{i+1,j}}{t'_{i+1,j} - t'_{i,j}}, \ldots, \prod_{h=i+1}^{k} \frac{t'_{h,j}}{t'_{h,j} - t'_{i,j}} \right)^T .$$

D'après la Relation (7.3) on a

$$\vartheta_i = \left(\underbrace{0, \ldots, 0}_{i}, 1, \frac{d_{i+1}}{d_{i+1} - d_i}, \ldots, \prod_{h=i+1}^{k} \frac{d_h}{d_h - d_i} \right)^T \tag{7.10}$$

qui est indépendant de j.

Comme chaque coordonnée non nulle de ϑ_i est supérieure ou égale à 1, on peut écrire

$$\vartheta_i \cdot C_j \geq \sum_{h=i}^{k} p'_{h,j} = 1 - \sum_{h=0}^{i-1} p'_{h,j} = 1 - S'_{i-1,j} \geq 1 - S_{i-1,j} \tag{7.11}$$

□

Démonstration du Théorème 7.2.1.

1. On a $d = d_1 = 2^{n-r}$ (la distance minimale de $RM(r,n)$). Si $w > 2^n - 2^{n-r}$ i.e. si $d_1 = 2^{n-r} > 2^n - w$ alors (d'après la croissance des distances de Hamming généralisées et le Lemme 7.2.1) les événements $E_{1,w}, E_{2,w}, ..., E_{k,w}$ sont impossibles, donc nécessairement $\Pr(Ann_r(E) \neq 0) = 0$ et E n'a pas d'annulateur de degré au plus r ; d'où le premier point du Théorème 7.2.1.
2. Le reste de la démonstration est comme dans [28]. Pour tout $i = 0, \ldots, k-1$ et $h = 0, 1, \ldots, w$, le vecteur ϑ_i est un vecteur propre de la matrice A_h^T associé à la valeur propre $(1 - t'_{i,h})$, on a donc, pour tout $j = 0, 1, \ldots, w-1$

$$\begin{aligned} \vartheta_i \cdot C_j = C_j^T \vartheta_i &= C_{j-1}^T A_{j-1}^T \vartheta_i \\ &= (1 - t'_{i,j-1}) C_{j-1}^T \vartheta_i \\ &= \ldots \\ &= \left(\prod_{h=0}^{j-1} (1 - t'_{i,h}) \right) C_0^T \vartheta_i. \end{aligned}$$

ce qui donne (avec l'écriture des $t'_{i,j}$ en termes des distances de Hamming généralisées par la Relation (7.3)

$$\vartheta_i \cdot C_j = \left(\prod_{h=0}^{j-1} \frac{2^n - h - d_i}{2^n - h} \right) \vartheta_i \cdot C_0.$$

Le seul coefficient non nul de C_0 en est le dernier et il vaut 1, en utilisant (7.10) on obtient

$$\vartheta_i \cdot C_0 = \prod_{h=i+1}^{k} \left(\frac{d_h}{d_h - d_i} \right).$$

On en déduit pour $i = 1$ et $j = w$ que

$$\Pr\{Ann_r(E) \neq \{0\}\} = 1 - S_{0,w} \leq \prod_{h=2^n-w+1}^{2^n} \left(\frac{h-d_1}{h}\right) \prod_{h=2}^{k} \left(\frac{d_h}{d_h-d_1}\right).$$

ce qui est le deuxième point du Théorème 7.2.1.

□

Remarquons que l'hypothèse $w \leq 2^n - 2^{n-r}$ est nécessaire, sinon l'Inégalité (7.2) n'aurait pas de sens.

7.2.3 Borne explicite déduite par l'utilisation la cohérence des codes de Reed-Muller

Nous utiliserons la cohérence des codes de Reed-Muller et le Théorème 7.2.1 pour montrer le Théorème 7.2.2. Nous utiliserons aussi la borne de Griesmer dans la démonstration de ce théorème.

Lemme 7.2.5 (Inégalité de Griesmer)**.** *Pour tout code binaire C de dimension k et de distance minimale d, on a*

$$d_k(C) \geq \sum_{h=0}^{k-1} \left\lceil \frac{d}{2^h} \right\rceil$$

Théorème 7.2.2 ([28])**.** *Supposons que E est choisi selon une loi de probabilité uniforme parmi les sous-ensembles de $\mathbb{F}_2^n$ de cardinal égal à w, tel que $w \leq 2^n - 2^{n-r}$, $r \geq 1$, alors il existe une constante a, avec $5 \leq a \leq 11$, telle que*

$$\ln(\Pr\{Ann_r(E) \neq \{0\}\}) \leq d\left[\frac{k}{2^n}\left(\ln\frac{2^n}{d} + a\right) - \ln\left(\frac{2^n+1}{2^n-w+1}\right)\right] \tag{7.12}$$

où $d = 2^{n-r} = d_1$ est la distance minimale de $RM(r,n)$, $k = \dim(RM(r,n)) = \sum_{i=0}^{r}\binom{n}{i}$ et d_1, d_2 sont les deux premières distances de Hamming généralisées de $RM(r,n)$.

Démonstration du Théorème 7.2.2. Nous complétons et clarifions la démonstration de [28].

Considérons le logarithme des deux membres de (7.2), nous obtenons :

$$\ln(\Pr\{Ann_r(E) \neq \{0\}\}) \leq \sum_{i=2^n-w+1}^{2^n} \ln\left(1 - \frac{d_1}{i}\right) - \sum_{i=2}^{k} \ln\left(1 - \frac{d_1}{d_i}\right). \tag{7.13}$$

Cette inégalité a bien un sens puisque, pour tout $i \geq 2^n - w + 1$, on a $d_1 < i$ (ceci est une conséquence directe de l'hypothèse, $w \leq 2^n - 2^{n-r}$, puisque $2^n - w \geq 2^{n-r} = d_1 = d$, où d est la distance minimale de $RM(r,n)$).

En utilisant le fait que si $x < 1$ alors $\ln(1-x) \leq -x$, on en déduit par la comparaison d'une série et de l'intégrale de Riemann que

$$\sum_{i=2^n-w+1}^{2^n} \ln\left(1 - \frac{d_1}{i}\right) \leq -\sum_{i=2^n-w+1}^{2^n} \frac{d_1}{i} \leq -d_1 \int_{2^n-w+1}^{2^n+1} \frac{1}{t} dt$$
$$= d_1 \ln\left(\frac{2^n - w + 1}{2^n + 1}\right).$$

Pour l'autre somme dans (7.13), puisque les distances de Hamming généralisées sont strictement croissantes, on a $d_1/d_i \leq d_1/d_2$ pour tout $i = 2, \ldots, k$. La borne de Griesmer sur les sous-codes C de distance minimale égale à d implique une borne inférieure sur les poids de Hamming généralisés de ces codes

$$d_i(C) \geq \sum_{h=0}^{i-1} \left\lceil \frac{d}{2^h} \right\rceil.$$

On a alors $d_1/d_2 \leq d_1/(d + \lceil \frac{d}{2} \rceil) \leq 2/3$. D'autre part, le théorème des accroissements finis appliqué à la fonction $x \mapsto -\ln(1-x)$ sur les intervalles $[0, \frac{d_1}{d_i}]$ implique que pour tout i il existe une constante positive $c_i \in]0, \frac{d_1}{d_i}[$ telle que

$$-\ln\left(1 - \frac{d_1}{d_i}\right) = \frac{1}{1-c_i}\frac{d_1}{d_i} = \left(1 + \frac{c_i}{1-c_i}\right)\frac{d_1}{d_i} \leq \left(\frac{d_1}{d_i} + \alpha_i \frac{d_1^2}{d_i^2}\right),$$

où $\alpha_i = \frac{1}{1-c_i} > 1$ puisque $0 < c_i \leq \frac{d_1}{d_i} < 1$.

Considérons la somme membre à membre, et notons $\alpha = \max\limits_{2 \leq i \leq k} \alpha_i$, nous avons $1 \leq \alpha = \max\limits_{2 \leq i \leq k} \alpha_i = \max\limits_{2 \leq i \leq k} \frac{1}{1-c_i} \leq \frac{1}{1-d_1/d_2} \leq \frac{1}{1-2/3} = 3$, nous avons donc

$$-\sum_{i=2}^{k} \ln\left(1 - \frac{d_1}{d_i}\right) \leq \sum_{i=2}^{k} \left(\frac{d_1}{d_i} + \alpha \frac{d_1^2}{d_i^2}\right). \tag{7.14}$$

On pose $i_0 = \lfloor \frac{d_1 k}{2^n} \rfloor$. On a $i_0 \neq 0$. En effet, on a $i_0 = \left\lfloor \frac{2^{n-r} \sum_{i=0}^{r} \binom{n}{i}}{2^n} \right\rfloor = \left\lfloor \frac{\sum_{i=0}^{r} \binom{n}{i}}{2^r} \right\rfloor$ et on peut montrer par récurrence sur r que $\frac{\sum_{i=0}^{r} \binom{n}{i}}{2^r} \geq 1$ pour tout $r \leq \lceil \frac{n}{2} \rceil$

- Pour $r = 1$, $\frac{\sum_{i=0}^{r} \binom{n}{i}}{2^r} = \frac{1+n}{2} \geq 1$, pour tout $n \geq 1$,

- Soit $1 \leq r \leq \lceil \frac{n}{2} \rceil - 1$; on suppose que $\frac{\sum_{i=0}^{r} \binom{n}{i}}{2^r} \geq 1$; on a alors

$$\begin{aligned}
\frac{\sum_{i=0}^{r+1} \binom{n}{i}}{2^{r+1}} &= \frac{1}{2} \frac{\sum_{i=0}^{r} \binom{n}{i}}{2^r} + \frac{\binom{n}{r+1}}{2^{r+1}} \\
&\geq \frac{1}{2} + \frac{\binom{n}{r+1}}{2^{r+1}}, (\text{ hypothèse de récurrence }) \\
&\geq \frac{1}{2} + \frac{1}{2} \frac{\binom{n}{r}}{2^r}, (\text{ puisque } r+1 \leq \lceil \frac{n}{2} \rceil) \\
&\geq \frac{1}{2} + \frac{1}{2} \frac{\left(\frac{n}{r}\right)^r}{2^r}, (\text{ Inégalité sur les coefficients binomial }) \\
&\geq \frac{1}{2} + \frac{1}{2} = 1.
\end{aligned}$$

En utilisant la croissance des d_i et le fait que $\frac{i}{d_i} \leq \frac{k}{2^n}$ (la cohérence de $RM(r,n)$), on aura

$$\sum_{i=1}^{k} \frac{1}{d_i} = \sum_{i=1}^{i_0} \frac{1}{d_i} + \sum_{i=i_0+1}^{k} \frac{1}{d_i} \leq \frac{i_0}{d_1} + \frac{k}{2^n} \sum_{i=i_0+1}^{k} \frac{1}{i}.$$

En utilisant la comparaison d'une série et de l'intégrale de Riemann, on a

$$\begin{aligned}
\frac{i_0}{d_1} + \frac{k}{2^n} \sum_{i=i_0+1}^{k} \frac{1}{i} &\leq \frac{k}{2^n} + \frac{k}{2^n} \int_{\lfloor \frac{d_1 k}{2^n} \rfloor}^{k} \frac{1}{x} dx \\
&= \frac{k}{2^n} + \frac{k}{2^n} \left[\int_{\lfloor \frac{d_1 k}{2^n} \rfloor}^{\frac{d_1 k}{2^n}} \frac{1}{x} dx + \int_{\frac{d_1 k}{2^n}}^{k} \frac{1}{x} dx \right] \\
&= \frac{k}{2^n} + \frac{k}{2^n} \left[\ln \frac{\frac{d_1 k}{2^n}}{\lfloor \frac{d_1 k}{2^n} \rfloor} + \ln \frac{k}{\frac{d_1 k}{2^n}} \right] \\
&< \frac{k}{2^n} \left(2 + \ln \frac{2^n}{d_1} \right).
\end{aligned}$$

De la même manière, on a pour la somme des carrés

$$\begin{aligned}
\sum_{i=1}^{k} \frac{1}{d_i^2} \leq \frac{i_0}{d_1^2} + \left(\frac{k}{2^n}\right)^2 \sum_{i=i_0+1}^{k} \frac{1}{i^2} &\leq \frac{k}{2^n} \left(\frac{1}{d_1} + \frac{k}{2^n} \int_{\lfloor \frac{d_1 k}{2^n} \rfloor}^{k} \frac{1}{x^2} dx \right) \\
&= \frac{k}{2^n} \left(\frac{1}{d_1} + \frac{k}{2^n} \int_{\lfloor \frac{dk}{2^n} \rfloor}^{\frac{dk}{2^n}} \frac{1}{x^2} dx + \frac{k}{2^n} \int_{\frac{dk}{2^n}}^{k} \frac{1}{x^2} dx \right) \\
&= \frac{k}{2^n} \left(\frac{1}{d_1} + \frac{k}{2^n} \left(\frac{1}{\lfloor \frac{dk}{2^n} \rfloor} - \frac{1}{\frac{dk}{2^n}} \right) + \frac{1}{d_1} - \frac{1}{2^n} \right) \\
&\leq \frac{k}{2^n} \left[\frac{2}{d_1} - \frac{1}{2^n} + \frac{k}{2^n} \left(\frac{1}{\lfloor \frac{d_1 k}{2^n} \rfloor} - \frac{1}{\frac{d_1 k}{2^n}} \right) \right].
\end{aligned}$$

On a par la Relation (7.14)

$$-\sum_{i=2}^{k} \ln\left(1-\frac{d_1}{d_i}\right) \leq \sum_{i=2}^{k}\left(\frac{d_1}{d_i}+\alpha\frac{d_1^2}{d_i^2}\right)$$
$$\leq \frac{d_1 k}{2^n}\left(2+\ln\frac{2^n}{d_1}\right)+\alpha\frac{d_1 k}{2^n}\left[2-\frac{d_1}{2^n}+\frac{d_1 k}{2^n}\left(\frac{1}{\lfloor\frac{d_1 k}{2^n}\rfloor}-\frac{1}{\frac{d_1 k}{2^n}}\right)\right],$$

et donc

$$-\sum_{i=2}^{k} \ln\left(1-\frac{d_1}{d_i}\right) \leq \frac{d_1 k}{2^n}\left(2+\ln\frac{2^n}{d_1}+3\alpha\right)$$

puisque $\frac{d_1 k}{2^n}\left(\frac{1}{\lfloor\frac{d_1 k}{2^n}\rfloor}-\frac{1}{\frac{d_1 k}{2^n}}\right) \leq 1$, car $\lfloor\frac{d_1 k}{2^n}\rfloor \geq 1$ et $\frac{d_1 k}{2^n} < \lfloor\frac{d_1 k}{2^n}\rfloor + 1$.

On obtient finalement la borne supérieure suivante sur la probabilité d'existence d'annulateurs de E

$$\ln(\Pr\{Ann_r(E) \neq \{0\}\}) \leq d\left[\frac{k}{2^n}\left(\ln\frac{2^n}{d}+a\right)-\ln\left(\frac{2^n+1}{2^n-w+1}\right)\right]$$

où $a = 2+3\alpha$, ce qui termine la démonstration du Théorème 7.2.2. □

Remarquons que
- Si $w = 1$ alors l'Inégalité (7.12) est triviale ;
- Si $w > 1$, le membre de gauche de (7.12) est négatif, et cette inégalité est de plus en plus significative quand w croît.

7.3 Application au comportement asymptotique de l'immunité algébrique

7.3.1 Borne inférieure asymptotique sur l'immunité algébrique d'un ensemble

Nous exploitons maintenant les résultats des sections précédentes pour en déduire une borne inférieure asymptotique sur l'Immunité algébrique d'un sous-ensemble E de $\mathbb{F}_2^n$ de cardinal fixé inférieur à $2^n - 2^{n-r}$. Ce qui étend les résultats de [28], qui sont valables pour les fonctions Booléennes aléatoires et équilibrées (c'est le cas où $|E| = |E^c| = 2^{n-1}$). Cette extension nous amène à introduire un paramètre l dans le théorème suivant.

Théorème 7.3.1. *1. Soit $l \geq 1$ un nombre réel et soit w_n une suite d'entiers telle que $2^{n-l} \leq w_n \leq 2^n - 1$. Supposons que E est choisi selon une loi de probabilité uniforme parmi les sous-ensembles de $\mathbb{F}_2^n$ de cardinal w_n. Alors l'immunité algébrique de E satisfait presque sûrement pour tout $0 < \varepsilon < 1$:*

$$AI(E) \geq \frac{n}{2} - \sqrt{\frac{n}{2} \ln\left[n\delta_l\left(1+\varepsilon\right)\right]} \tag{7.15}$$

quand n tend à l'infini, où $\delta_l = \frac{\ln 2}{\ln\left(\frac{1}{1-2^{-l}}\right)}$.

2. *Supposons que E est choisi selon une loi de probabilité uniforme parmi les sous-ensembles de $\mathbb{F}_2^n$ de cardinal 2^{n-m}, alors*
   - *quand n tend vers l'infini et $m = m_0$ est fixé, l'immunité algébrique de E satisfait presque sûrement (7.15) avec $l = m_0$;*
   - *quand n, m tendent vers l'infini et $n - m$ tend vers une constante non-négative, l'immunité algébrique standard de E est presque sûrement égale à 1 ;*
   - *quand n, m et $n - m$ tendent vers l'infini, l'immunité algébrique standard de E satisfait presque sûrement, pour tout $0 < \varepsilon < 2/3$:*

$$AI(E) \geq \frac{n}{2} - \sqrt{m \ln 2 + \ln\left(\frac{n \ln 2}{2/3 - \varepsilon}\right)}\sqrt{\frac{n}{2}}. \tag{7.16}$$

On utilisera dans la démonstration de ce théorème la borne de Chernoff, voir pour cela [53] par exemple.

Théorème 7.3.2 (Borne de Chernoff)**.** *Soient $0 < p < 1$ et $X_1, X_2, \ldots, X_n$ des variables aléatoires indépendantes qui prennent la valeur 1 avec la probabilité p et 0 avec la probabilité $1 - p$ et soit $S_n = \sum_{i=1}^n X_i$. Alors pour tout $t \geq 0$,*

$$\Pr(|S_n - np| \geq nt) \leq 2e^{-2nt^2}$$

et

$$\Pr(S_n - np \geq nt) \leq e^{-2nt^2} \tag{7.17}$$

Démonstration du Théorème 7.3.1.

1. Nous précisons et généralisons la démonstration de [28] pour le premier point. On déduit de la Relation (7.12) (avec $r = r_n \leq n$) une condition suffisante pour que la probabilité d'existence d'annulateurs non nuls de E de degré au plus r_n tende vers 0 quand n tend vers l'infini ; cette condition est que la distance minimale $d_{1,n} = 2^{n-r_n}$ de

$RM(r_n, n)$ tend vers l'infini et que le terme entre crochets dans (7.12) soit majoré par un nombre constant négatif, i.e.

$$\frac{k_n}{2^n}\left(\ln\frac{2^n}{2^{n-r_n}}+a_n\right)-\ln\left(\frac{2^n+1}{|E^c|+1}\right)<-\varepsilon \tag{7.18}$$

pour un nombre $\varepsilon>0$, où nous notons les constantes k, a de la Relation (7.12) sous la forme k_n, a_n. Nous avons $5\leq a_n\leq 11$.

La condition $|E|\geq 2^{n-l}$ implique que $\ln\left(\frac{2^n+1}{|E^c|+1}\right)\geq\ln\left(\frac{1+2^{-n}}{1-2^{-l}+2^{-n}}\right)$. On note $a=\sup_n a_n$, on a $5\leq a\leq 11$ et on en déduit (en utilisant que $r_n\leq n$) une condition qui implique (7.18) ; cette condition est que

$$\frac{k_n}{2^n}<\frac{\ln\left(\frac{1+2^{-n}}{1-2^{-l}+2^{-n}}\right)-\varepsilon}{n\ln 2+a}=\frac{1-\varepsilon/\ln\left(\frac{1+2^{-n}}{1-2^{-l}+2^{-n}}\right)}{\delta_{n,l}(n+\frac{a}{\ln 2})} \tag{7.19}$$

où $\delta_{n,l}=\ln 2/\ln\left(\frac{1+2^{-n}}{1-2^{-l}+2^{-n}}\right)$. La Relation (7.19) nécessite que $\varepsilon<\ln\left(\frac{1+2^{-n}}{1-2^{-l}+2^{-n}}\right)$ et il est suffisant pour ceci que $\varepsilon<\ln\left(\frac{1}{1-2^{-l}}\right)$ puisque $\ln\left(\frac{1+2^{-n}}{1-2^{-l}+2^{-n}}\right)>\ln\left(\frac{1}{1-2^{-l}}\right)$; on peut faire cette hypothèse sans perte de généralité puisque, si la Relation (7.15) est satisfaite pour une valeur de ε, alors elle est satisfaite pour toutes les valeurs plus grandes que ε.

D'autre part, introduisons une loi de probabilité binomiale X de paramètre $p=1/2$ sur n essais (X est donc la somme de n variables aléatoires identiques suivant une loi de Bernoulli), on a

$$\Pr(X\leq r_n)=\sum_{i=0}^{r_n}\binom{n}{i}\left(\frac{1}{2}\right)^i\left(\frac{1}{2}\right)^{n-i}=\frac{k_n}{2^n}. \tag{7.20}$$

En utilisant la borne de Chernoff (7.17) (avec $t=\frac{\lambda}{2\sqrt{n}}$), on obtient pour tout $\lambda>0$

$$\Pr\left(X-\frac{n}{2}\leq-\lambda\frac{\sqrt{n}}{2}\right)\leq\exp\left(-\frac{\lambda^2}{2}\right). \tag{7.21}$$

Sachant que si $r\leq r'$ alors $\Pr(X\leq r)\leq\Pr(X\leq r')$, on en déduit que, si $r_n\leq\frac{n}{2}-\lambda\frac{\sqrt{n}}{2}$ alors on a

$$\frac{k_n}{2^n}=\Pr(X\leq r_n)\leq\Pr\left(X\leq\frac{n}{2}-\lambda\frac{\sqrt{n}}{2}\right)\leq\exp\left(-\frac{\lambda^2}{2}\right).$$

On en déduit une condition suffisante sur λ telle que k_n satisfait (7.19) ; cette condition est

$$\lambda^2>-2\ln\left(\frac{1-\varepsilon/\ln\left(\frac{1+2^{-n}}{1-2^{-l}+2^{-n}}\right)}{\delta_{n,l}(n+\frac{a}{\ln 2})}\right).$$

Finalement, une condition suffisante pour que la probabilité d'existence d'annulateurs de E de degrés au plus r_n tende vers 0 est

$$r_n < \frac{n}{2} - \sqrt{\ln\left(\frac{\delta_{n,l}(n+\frac{a}{\ln 2})}{1-\varepsilon/\ln\left(\frac{1+2^{-n}}{1-2^{-l}+2^{-n}}\right)}\right)\sqrt{\frac{n}{2}}}.$$

Pour ces valeurs de r_n on a

$$d_{1,n} = 2^{n-r_n} > 2^{\frac{n}{2}+\sqrt{\ln\left(\frac{\delta_{n,l}(n+\frac{a}{\ln 2})}{1-\varepsilon/\ln\left(\frac{1+2^{-n}}{1-2^{-l}+2^{-n}}\right)}\right)\sqrt{\frac{n}{2}}}} \to +\infty \text{ quand } n \to +\infty.$$

D'où la probabilité qu'il n'existe pas d'annulateur de E de degré au plus r_n tend vers 1 si

$$\begin{aligned} r_n &< \frac{n}{2} - \sqrt{\ln\left(\frac{\delta_{n,l}(n+\frac{a}{\ln 2})}{1-\varepsilon/\ln\left(\frac{1+2^{-n}}{1-2^{-l}+2^{-n}}\right)}\right)\sqrt{\frac{n}{2}}} \\ &= \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[\delta_{n,l}\,(n+b)\left(1+\frac{\varepsilon/\ln\left(\frac{1+2^{-n}}{1-2^{-l}+2^{-n}}\right)}{1-\varepsilon/\ln\left(\frac{1+2^{-n}}{1-2^{-l}+2^{-n}}\right)}\right)\right]} \end{aligned}$$

(utilisant que $1/(1-x) = 1 + x/(1-x)$), avec $b = \frac{a}{\ln 2}$. Quand n tend vers l'infini, la racine carrée précédente est équivalente à

$$\sqrt{\frac{n}{2}\ln\left[n\delta_l\left(1+\frac{\varepsilon/\ln\left(\frac{1}{1-2^{-l}}\right)}{1-\varepsilon/\ln\left(\frac{1}{1-2^{-l}}\right)}\right)\right]}, \text{ où } \delta_l = \ln 2/\ln\left(\frac{1}{1-2^{-l}}\right).$$

Ainsi, à une croissance près de la valeur de ε, qui ne réduit pas la généralité puisque ε peut être n'importe quel nombre strictement positif, on obtient le premier point du Théorème 7.3.1.

2. Le point 1 permet de déduire le point 2 dans le premier cas en prenant $l = m_0$ et $|E| = 2^{n-m_0}$. Nous aurons besoin d'améliorer la démonstration précédente pour déduire les autres cas.
- Si $n-m$ tend vers une constante $c \geq 0$ quand n, m tendent vers l'infini, alors (sachant que $AI(E) \leq d_{n,m}$, où $d_{n,m} = \min\{d, \sum_{i=0}^{d}\binom{n}{i} > 2^{n-m}\}$), on a $d_{n,m} = 1$ puisque $\sum_{i=0}^{1}\binom{n}{i} = 1+n \to +\infty$, alors $AI(E) = 1$ asymptotiquement.
- Supposons dans ce qui suit que $n-m$ tend vers l'infini quand n, m tendent vers l'infini. Multiplions et divisons le membre droite de l'Inégalité (7.12) par 2^{-m} (en prenant

$w = |E| = 2^{n-m}$ et $r = r_n$), nous obtenons

$$\begin{aligned}&\ln(\Pr\{Ann_{r_n}(E) \neq \{0\}\})\\ &\leq 2^{n-m-r_n}\left[\frac{k_n}{2^{n-m}}\left(r_n \ln 2 + a_n\right) - 2^m \ln\left(\frac{2^n+1}{2^n - 2^{n-m}+1}\right)\right],\end{aligned} \tag{7.22}$$

où nous notons les constantes k, a de la Relation (7.12) sous la forme k_n, a_n. Nous avons $5 \leq a_n \leq 11$.
On en déduit de la Relation (7.22) une condition suffisante sous laquelle la probabilité d'existence d'annulateurs non nuls de degrés au plus r_n de E tend vers 0 quand n, m tendent vers l'infini : cette condition est que 2^{n-m-r_n} tend vers l'infini et le terme entre crochets dans (7.22) est majoré par une constante négative, c'est à dire,

$$\frac{k_n}{2^{n-m}}\left(r_n \ln 2 + a_n\right) - 2^m \ln\left(\frac{2^n+1}{2^n - 2^{n-m}+1}\right) < -\varepsilon \tag{7.23}$$

avec $\varepsilon > 0$ fixé.

La fonction $(n, m) \mapsto -2^m \ln\left(\frac{2^n+1}{2^n-2^{n-m}+1}\right)$ est une fonction décroissante en n pour tout m ; on peut observer ceci en calculant la dérivée partielle relativement à n

$$\begin{aligned}&\frac{\partial}{\partial n}\left(-2^m \ln\left(\frac{2^n+1}{2^n - 2^{n-m}+1}\right)\right)\\ &= -2^m\left(\frac{2^n \ln 2}{2^n - 2^{n-m}+1} - \frac{(2^n+1)(2^n \ln 2 - 2^{n-m}\ln 2)}{(2^n - 2^{n-m}+1)^2}\right)\frac{2^n - 2^{n-m}+1}{2^n+1}\\ &= -\frac{2^n \ln 2}{(2^n+1)(2^n - 2^{n-m}+1)} < 0.\end{aligned}$$

Alors

$$\begin{aligned}-2^m \ln\left(\frac{2^n+1}{2^n - 2^{n-m}+1}\right) &\leq \sup_m\left[-2^m \ln\left(\frac{3/2}{3/2 - 2^{-m}}\right)\right]\\ &\leq \sup_m\left[-2^m \ln\left(\frac{3/2}{3/2 - 2^{-m}}\right)\right]\\ &= -2/3\end{aligned}$$

(en effet, $-2^m \ln\left(\frac{3/2}{3/2-2^{-m}}\right) = 2^m \ln\left(1 - \frac{2^{-m}}{3/2}\right)$ est une fonction croissante de m "produit de deux fonctions croissantes" et leur limite à $+\infty$ est $-2/3$).

Ainsi (sachant que $r_n \leq n$), une condition suffisante pour avoir (7.23) est

$$\frac{k_n}{2^{n-m}} < \frac{2/3 - \varepsilon}{(n \ln 2 + a)}, \text{ où } a = \sup_n a_n \text{ et } \varepsilon < 2/3. \tag{7.24}$$

Introduisons une loi de probabilité X comme dans la démonstration du premier point et utilisons la borne de Chernoff (7.17) (avec $t = \frac{\lambda}{2\sqrt{n}}$), nous obtenons pour tout $\lambda > 0$

$$\Pr\left(X - \frac{n}{2} \leq -\lambda\frac{\sqrt{n}}{2}\right) \leq \exp\left(-\frac{\lambda^2}{2}\right). \tag{7.25}$$

Si $r_n \leq \frac{n}{2} - \lambda\frac{\sqrt{n}}{2}$ alors nous avons

$$\frac{k_n}{2^{n-m}} = 2^m\frac{k_n}{2^n} = 2^m\Pr(X \leq r_n) \leq 2^m\Pr\left(X \leq \frac{n}{2} - \lambda\frac{\sqrt{n}}{2}\right) \leq 2^m\exp\left(-\frac{\lambda^2}{2}\right).$$

On en déduit alors une condition suffisante sur λ telle que pour tout n fixé, k_n satisfait (7.24), cette condition est

$$\lambda^2 > -2\ln\left(2^{-m}\frac{2/3-\varepsilon}{n\ln 2 + a}\right).$$

Finalement, la probabilité d'existence d' annulateurs non nuls de E de degrés au plus r_n tend vers 0 si

$$r_n < \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2 + a}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}. \tag{7.26}$$

Tout entier m satisfait $\left\lceil\frac{m}{n}\right\rceil n - 1 \leq m = \frac{m}{n}n \leq \left\lceil\frac{m}{n}\right\rceil n$, alors on peut écrire

$$\left(1 - \left\lceil\frac{m}{n}\right\rceil\right)n \leq n - m \leq \left(1 - \left\lceil\frac{m}{n}\right\rceil\right)n + 1.$$

Quand $n - m$ tend vers l'infini lorsque n, m tendent vers l'infini, $\left\lceil\frac{m}{n}\right\rceil$ tend vers une constante positive $c \leq 1$.

Pour les valeurs de r_n satisfaisant (7.26), on a

$$\begin{aligned} 2^{n-m-r_n} &> 2^{\left(1-\left\lceil\frac{m}{n}\right\rceil\right)n + \sqrt{\left\lceil\frac{m}{n}\right\rceil n\ln 2 + \ln\left(\frac{n\ln 2+a}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}} \\ &= 2^{\left[\left(\frac{1}{2}-\left\lceil\frac{m}{n}\right\rceil\right) + \sqrt{\frac{\left\lceil\frac{m}{n}\right\rceil\ln 2}{2} + \ln\left(\frac{n\ln 2+a}{2/3-\varepsilon}\right)\frac{1}{2n}}\right]n}. \end{aligned}$$

Il est facile à montrer que $(\frac{1}{2} - c) + \sqrt{\frac{c\ln 2}{2}} > 0$, où $0 \leq c \leq 1$, alors $2^{n-m-r_n} \to +\infty$ quand $m, n \to +\infty$ et $n - m \to +\infty$.

Ainsi la probabilité qu'il n'existe pas d'annulateur non nul de E de degré au plus r_n tend vers 1 si

$$r_n < \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2 + a}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}.$$

Par conséquent

$$\Pr\left\{AI(E) \geq \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}\right\} \to 1,$$

$$\text{quand } n, m \to +\infty \text{ et } n - m \to +\infty,$$

ce qui termine la démonstration du deuxième point du Théorème 7.3.1. □

7.3.2 Borne inférieure asymptotique sur l'immunité algébrique d'une fonction Booléenne

Avant de voir l'application aux fonctions vectorielles d'une façon générale, nous verrons comment, en utilisant les sections précédentes, on peut montrer que toutes les fonctions booléennes aléatoires équilibrées à n variables ont -asymptotiquement- une immunité algébrique (presque optimale) en au moins $\frac{n}{2}(1-\circ(1))$ quand n tend vers l'infini. La condition f équilibrée est incontournable puisque nous aurons besoin en même temps $2^{n-l} \leq |supp(f)| \leq 2^n - 1$ et $2^{n-l} \leq |supp(f)^c| \leq 2^n - 1$, c'est à dire, $1 \leq |supp(f)| \leq 2^n - 2^{n-l}$, où $supp(f)$ est le support de f.

Une conséquence directe du premier point du Théorème 7.3.1 est le résultat suivant

Corollaire 7.3.1. *Soit f une fonction Booléenne à n variables, aléatoire et équilibrée. Alors l'immunité algébrique de f satisfait presque sûrement, pour tout $0 < \varepsilon < 1$*

$$AI(f) \geq \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[n\left(1+\varepsilon\right)\right]}$$

quand n tend vers l'infini.

Démonstration. Appliquons le Théorème 7.3.1 avec $w_n = 2^{n-1}$. Soit f une fonction booléenne aléatoire équilibrée. Nous avons

$$\Pr\left\{AI(f^{-1}(0)) < \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[n\left(1+\varepsilon\right)\right]}\right\} \to 0,$$

et

$$\Pr\left\{AI(f^{-1}(1)) < \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[n\left(1+\varepsilon\right)\right]}\right\} \to 0.$$

Comme

$$\begin{aligned}
&\Pr\left\{AI(f) < \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[n\left(1+\varepsilon\right)\right]}\right\} \\
\leq &\Pr\left\{AI(f^{-1}(0)) < \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[n\left(1+\varepsilon\right)\right]}\right\} \\
&+ \Pr\left\{AI(f^{-1}(1)) < \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[n\left(1+\varepsilon\right)\right]}\right\},
\end{aligned}$$

alors $\Pr\left\{AI(f) < \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[n\left(1+\varepsilon\right)\right]}\right\} \to 0$ quand $n \to +\infty$, c'est à dire

$$\Pr\left\{AI(f) \geq \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[n\left(1+\varepsilon\right)\right]}\right\} \to 1 \text{ quand } n \to +\infty. \qquad \square$$

7.3.3 Borne inférieure asymptotique de l'immunité algébrique standard d'une fonction vectorielle

Nous étendons maintenant la borne inférieure du Corollaire 7.3.1 en une borne inférieure sur l'immunité algébrique standard d'une fonction booléenne à plusieurs sorties aléatoire et équilibrée. Nous avons le résultat

Corollaire 7.3.2. *Soit F une (n,m)-fonction aléatoire et équilibrée ($1 \leq m \leq n$), alors*

1. *quand n tend vers l'infini et $m = m_0$ est fixé, alors l'immunité algébrique standard de F satisfait presque sûrement, pour tout $0 < \varepsilon < 1$*

$$AI(F) \geq \frac{n}{2} - \sqrt{\frac{n}{2} \ln\left[n\delta_{m_0}(1+\varepsilon)\right]}$$

   où $\delta_{m_0} = \ln 2 / \ln\left(\frac{1}{1-2^{-m_0}}\right)$,

2. *quand n, m tendent vers l'infini et $n - m$ tend vers une constante non-négative, alors l'immunité algébrique standard de F est presque sûrement égale à 1,*

3. *quand n, m et $n-m$ tendent vers l'infini, alors l'immunité algébrique standard de F satisfait presque sûrement, pour tout $0 < \varepsilon < 2/3$*

$$AI(F) \geq \frac{n}{2} - \sqrt{m \ln 2 + \ln\left(\frac{n \ln 2}{2/3 - \varepsilon}\right)} \sqrt{\frac{n}{2}}. \tag{7.27}$$

Remarquons que la dernière borne n'est significative que si le terme droite de (7.27) est positif, c'est à dire si $m \leq \left\lfloor \frac{n/2 - \ln\left(\frac{n \ln 2}{2/3-\varepsilon}\right)}{\ln 2} \right\rfloor$. Quand n, m tendent vers l'infini, cette borne n'est significative que si $\lim_{n,m\to\infty} \frac{m}{n} \leq \frac{1}{2 \ln 2}$.

La fonction F est aléatoire équilibrée, donc pour tout z, le sous-ensemble $F^{-1}(z)$ décrit uniformément l'ensemble des parties de $\mathbb{F}_2^n$ de cardinal $w_{n,m}$ égale à 2^{n-m}, nous sommes donc sous les conditions du deuxième point du Théorème 7.3.1.

Démonstration du Corollaire 7.3.2.

1. On a pour tout z dans $\mathbb{F}_2^m$

$$\begin{aligned}
&\Pr\left\{AI(F^{-1}(z)) < \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[n\delta_m\left(1+\varepsilon\right)\right]}\right\} \\
\leq\ & \sum_{j=1}^{2^m} \Pr\left\{AI(F^{-1}(z_j)) < \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[n\delta_m\left(1+\varepsilon\right)\right]}\right\}, z_j \in \mathbb{F}_2^m \\
\leq\ & 2^m \Pr\left\{AI(F^{-1}(z_{i_0})) < \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[n\delta_m\left(1+\varepsilon\right)\right]}\right\}, z_{i_0} \in \mathbb{F}_2^m.
\end{aligned}$$

Cette expression tend vers 0 quand n tend vers l'infini et m tend vers une constante m_0, compte tenu du Théorème 7.3.1. Alors $\Pr\left\{AI(F^{-1}(z)) \geq \frac{n}{2} - \sqrt{\frac{n}{2}\ln\left[n\delta_{m_0}\left(1+\varepsilon\right)\right]}\right\} \to 1$, quand n tend vers l'infini et $m = m_0$ est fixé, où $\delta_{m_0} = \ln 2/\ln\left(\frac{1}{1-2^{-m_0}}\right)$.

2. Quand n, m tendent vers l'infini et $n - m$ tend vers une constante non-négative, alors pour tout z in $\mathbb{F}_2^m$, l'immunité algébrique standard de $F^{-1}(z)$ est presque sûrement égale à 1, compte tenu du deuxième point du Théorème 7.3.1, alors l'immunité algébrique standard de F est presque sûrement égale à 1 (notons qu'il existe seulement 2^m fonctions ayant une immunité algébrique nulle).

3. Pour le dernier cas on a par le Théorème 7.3.1

$$\Pr\left\{AI(F^{-1}(z)) < \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}\right\} \to 0$$

$$\text{quand } n, m, n-m \to +\infty.$$

D'autre part

$$\begin{aligned}
&\Pr\left\{AI(F) < \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}\right\} \\
\leq\ & 2^m \Pr\left\{AI(F^{-1}(z_{i_0})) < \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}\right\}, z_{i_0} \in \mathbb{F}_2^m,
\end{aligned}$$

et

$$\begin{aligned}
&\Pr\left\{AI(F^{-1}(z_{i_0})) < \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}\right\} \\
\leq\ & \Pr\left\{Ann_{r^0_{n,m}}(F^{-1}(z_{i_0})) \neq \{0\}\right\} \leq \exp(\beta_{n,m}2^{n-m-r^0_{n,m}}) \qquad (7.28)
\end{aligned}$$

avec $r^0_{n,m} = \left\lfloor \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}} \right\rfloor$ et $\beta_{n,m}$ est le terme entre crochets dans (7.22) quand $r_n = r^0_{n,m}$. L'Inégalité (7.28) est satisfaite puisque l'évenement "$AI(F^{-1}(z_{i_0})) < \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}$" est contenu dans l'évenement

$$\text{“}Ann_{r^0_{n,m}}(F^{-1}(z_{i_0})) \neq \{0\}\text{”}.$$

$\beta_{n,m} < -\varepsilon < 0$ dès que on a $r^0_{n,m} < \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}$, (voir pour cela la démonstration du deuxième point du Théorème 7.3.1).

Alors

$$\begin{aligned} & \Pr\left\{AI(F) < \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}\right\} \\ \leq\ & 2^m \exp(-\varepsilon 2^{n-m-r^0_{n,m}}) \\ =\ & \exp(m\ln 2 - \varepsilon 2^{n-m-r^0_{n,m}}) \to 0 \text{ quand } m, n, n-m \to +\infty, \end{aligned}$$

puisque $2^{n-m-r^0_{n,m}} \geq 2^{n-m-r_n} \to +\infty$ pour tout $r_n < \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}$.

Ainsi $\Pr\left\{AI(F) \geq \frac{n}{2} - \sqrt{m\ln 2 + \ln\left(\frac{n\ln 2}{2/3-\varepsilon}\right)}\sqrt{\frac{n}{2}}\right\} \to 1$ quand n, m et $n-m$ tendent vers l'infini, ce qui montre le Corollaire 7.3.2. □

Nous savons que l'immunité algébrique standard d'une (n,m)-fonction F, $(1 \leq m \leq n)$ est au plus $d_{n,m}$, où $d_{n,m} = \min\{d : \sum_{i=0}^{d}\binom{n}{i} > 2^{n-m}\}$, et cette borne est atteinte d'après Feng et al. [30]. Dans le but de comparer notre borne asymptotique sur $AI(F)$ avec cette borne, nous montrons la borne supérieure suivante sur $d_{n,m}$,

Lemme 7.3.1. *Soit $1 \leq m \leq n$ deux entiers positifs. Si $d_{n,m} = \min\{d : \sum_{i=0}^{d}\binom{n}{i} > 2^{n-m}\}$, alors nous avons*

$$d_{n,m} \leq \left\lceil \frac{n-m+1}{2} \right\rceil. \tag{7.29}$$

Démonstration. Posons $k = n - m \geq 0$, nous avons

$$\sum_{i=0}^{\lceil\frac{k+1}{2}\rceil}\binom{n}{i} \geq \sum_{i=0}^{\lceil\frac{k+1}{2}\rceil}\binom{k+1}{i} > \frac{2^{k+1}}{2} = 2^k.$$

Ainsi $d_{n,m} \leq \lceil\frac{k+1}{2}\rceil$ (d'après la définition de $d_{n,m}$), d'où le résultat. □

Suivant la borne (7.29) sur $d_{n,m}$ et le Corollaire 7.3.2 , nous avons pour toute (n,m)-fonction F aléatoire et équilibrée, pour $m \leq n$:

1. Si $m = m_0$ est fixé et n tend vers l'infini, alors l'immunité algébrique standard de F satisfait presque sûrement pour tout $0 < \varepsilon < 1$

$$AI(F) \geq \frac{n}{2} - \sqrt{\frac{n}{2} \ln \left[n\delta_{m_0} (1+\varepsilon) \right]}.$$

   La borne supérieure suivante sur $AI(F)$ est toujours satisfaite :

$$AI(F) \leq d_{n,m_0} \leq \left\lceil \frac{n - m_0 + 1}{2} \right\rceil.$$

   D'autre part, nous avons

$$\lim_{n\to\infty} \frac{\left\lceil \frac{n-m_0+1}{2} \right\rceil}{n/2} = \lim_{n\to\infty} \frac{\frac{n}{2} - \sqrt{\frac{n}{2} \ln \left[n\delta_{m_0} (1+\varepsilon) \right]}}{n/2} = 1.$$

   Alors, quand n tend vers l'infini, $AI(F)$ et d_{n,m_0} sont de l'ordre de $\frac{n}{2}(1 - \circ(1))$;

2. Si n, m tendent vers l'infini et $n - m$ tend vers une constante non-négative, alors l'immunité algébrique standard de F est presque sûrement égale à 1 ;

3. Si $n, m, n - m$ tendent vers l'infini, alors l'immunité algébrique standard de F satisfait presque sûrement pour tout $0 < \varepsilon < 2/3$

$$AI(F) \geq \frac{n}{2} - \sqrt{m \ln 2 + \ln \left(\frac{n \ln 2}{2/3 - \varepsilon} \right)} \sqrt{\frac{n}{2}}.$$

   La borne supérieure suivante sur $AI(F)$ est toujours satisfaite

$$AI(F) \leq d_{n,m} \leq \left\lceil \frac{n - m + 1}{2} \right\rceil.$$

   nous considérons trois cas
   – Si $\lim\limits_{n,m\to\infty} \frac{m}{n} = 0$ alors

$$\lim_{n,m\to\infty} \frac{\left\lceil \frac{n-m+1}{2} \right\rceil}{n/2} = \lim_{n,m\to\infty} \frac{\frac{n}{2} - \sqrt{m \ln 2 + \ln \left(\frac{n \ln 2}{2/3-\varepsilon} \right)} \sqrt{\frac{n}{2}}}{n/2} = 1.$$

   Ainsi $AI(F)$ et $d_{n,m}$ sont de l'ordre de $\frac{n}{2}(1 - \circ(1))$,
   – Si $\lim\limits_{n,m\to\infty} \frac{m}{n} = c \leq \frac{1}{2 \ln 2}$ alors

$$\lim_{n,m\to\infty} \frac{\frac{n}{2} - \sqrt{m \ln 2 + \ln \left(\frac{n \ln 2}{2/3-\varepsilon} \right)} \sqrt{\frac{n}{2}}}{n/2} = 1 - \sqrt{2c \ln 2} \leq \lim_{n,m\to\infty} \frac{\left\lceil \frac{n-m+1}{2} \right\rceil}{n/2} = 1 - c.$$

   Ainsi $AI(F)$ et $d_{n,m}$ sont de l'ordre de $\beta \frac{n}{2}$, où $1 - \sqrt{2c \ln 2} \leq \beta \leq 1 - c$,

– Si $m \sim n$, ($\lim_{n,m\to\infty} \frac{m}{n} = 1$), notre borne (7.27) sur $AI(F)$ n'est pas significative, mais (sachant que $AI(F) \geq 1$ puisque F est équilibrée donc surjective), nous avons $1 \leq AI(F) \leq d_{n,m} \leq \left\lceil \frac{n-m+1}{2} \right\rceil$. D'autre part

$$\lim_{n,m\to\infty} 1/(n/2) = \lim_{n,m\to\infty} \left\lceil \frac{n-m+1}{2} \right\rceil /(n/2) = 0.$$

Ainsi, $AI(F)$ et $d_{n,m}$ sont de l'ordre de $\circ(n)$.

Nous remarquons que dans tous les cas sur n, m notre borne est compatible avec la borne $d_{n,m}$. Alors toutes les (n, m)-fonctions aléatoires et équilibrées sont d'une immunité algébrique presque optimale.

Chapitre 8

Conclusion

Les fonctions Booléennes à une ou à plusieurs sorties jouent un rôle central dans la conception des algorithmes de chiffrement par flot et par blocs. Elles doivent être soigneusement choisies afin d'éviter quelques attaques connues (statistiques ou algébriques). Les cryptographes ont recommandé d'utiliser des fonctions booléennes d'un nombre de variables proche de 20 suite à la découverte des attaques algébriques en 2003, qui sont réalisables contre les schémas par flot et parfois (plus difficilement) contre les schémas par blocs. Un critère nécessaire contre ces attaques est que la fonction utilisée ait une haute immunité algébrique.

Nous avons étudié dans cette thèse la notion d'immunité algébrique d'une fonction Booléenne à une et à plusieurs sorties. Nous avons rappelé que l'immunité algébrique d'une fonction à une sortie est liée à la correction d'erreur des codes de Reed-Muller sur un canal à effacements. La relation entre immunité algébrique, poids et non-linéarité implique qu'une fonction Booléenne de haute immunité algébrique ne peut pas être de mauvaise non-linéarité, mais une haute immunité algébrique n'implique pas une haute non-linéarité.

Il existe trois notions d'immunité algébrique pour une fonction vectorielle, chacune correspond à un mode d'utilisation et un nombre de bits de sortie spécifiques. Pour une (n, m)-fonction :

- l'immunité algébrique standard AI est significative quand m est très petit par rapport à n, $(m \ll n)$, c'est le cas par exemple d'une fonction Booléenne utilisée dans un chiffrement par flot,
- l'immunité algébrique du graphe AI_{gr} est significative quand m est proche de n, $(m \approx n)$, c'est le cas des boite-S dans un chiffrement par blocs.

Nous avons vu des bornes concernant ces notions et nous avons caractérisé des cas d'égalité.

Il a été démontré que la borne supérieure $d_{n,m}$ sur l'immunité algébrique standard AI est atteinte, mais on ne sait pas encore si c'est le cas pour la borne supérieure $D_{n,m}$ sur l'immunité algébrique du graphe AI_{gr}.
Le calcul de l'immunité algébrique n'est pas toujours facile, la stabilité des différentes notions d'immunité algébrique peut nous aider dans le calcul.

En utilisant les distances de Hamming généralisées, on peut montrer une borne sur la probabilité d'existence d'annulateurs d'un sous ensemble ; on en déduit un comportement asymptotique de l'immunité algébrique. Nous avons précisé la preuve de F. Didier [28] concernant une borne inférieure asymptotique sur l'immunité algébrique standard d'une fonction aléatoire et équilibrée, nous avons adapté cette preuve au cas des fonctions vectorielles, en outre, notre borne est compatible avec la borne $d_{n,m}$. On en déduit que les (n, m)-fonctions aléatoires et équilibrées sont d'une immunité algébrique AI presque optimale.

Perspectives de recherche. Nous avons fait porter nos recherches dans cette thèse sur les trois notions existantes d'immunité algébrique des fonctions vectorielles, il reste encore des travaux à faire :

1. Borne supérieure sur AI plus explicite : Nous savons que l'immunité algébrique standard d'une (n, m)-fonction F est au plus $d_{n,m}$ et que cette borne est atteinte. Nous avons vu une borne plus explicite (en fonction de n, m), à savoir que $AI(F)$ est majoré au sens large par $\left\lceil \frac{n-m-1}{2} \right\rceil$, ces deux bornes coïncident lorsque m est égal à n ou égal à 1, mais elles s'éloignent l'une de l'autre lorsque m croît. Trouver dans ce dernier cas une autre borne plus explicite et qui est proche de $d_{n,m}$ est un travail que nous projetons de faire.
2. Il reste à voir si la borne supérieure $D_{n,m}$ sur l'immunité algébrique du graphe est atteinte.
3. Différence entre les valeurs de AI_{comp} de deux fonctions EA équivalente : Nous avons vu que l'immunité algébrique standard (AI) et l'immunité algébrique par composantes (AI_{comp}) ne sont pas stables sous transformations affines étendues. Plus précisément :
   Si F et G sont deux (n, m)-fonctions affinement équivalentes de façon étendue, alors
   $$AI_{comp}(F) - 1 \leq AI_{comp}(G) \leq AI_{comp}(F) + 1 \tag{8.1}$$
   quelle relation existe-t-il pour les valeurs de AI ?

4. Existe -t-il des cas d'égalité entre AI_{gr} et $AI_{comp} + 1$? Nous avons la relation suivante entre l'immunité algébrique du graphe et l'immunité algébrique par composantes d'une (n, m)-fonction F

$$AI_{gr}(F) \leq AI_{comp}(F) + 1$$

   Existe-t-il des cas d'égalité ?

5. Une autre condition pour l'égalité entre AI et AI_{gr} : Nous avons vu une condition suffisante pour l'égalité entre l'immunité algébrique standard et l'immunité algébrique du graphe d'une fonction vectorielle. Existe-t-il une condition moins restrictive ?

Bibliographie

[1] F. Armknecht. Improving Fast Algebraic Attacks. In FSE, volume 3017 of *LNCS*, pages 65–82, 2004.

[2] F. Armknecht, M. Krause. Constructing Single -and multi- Output Boolean Functions with Maximal Algebraic Immunity. Proceedings of *ICALP* 2006, *Lecture Notes of Computer Science* 4052, Springer, pp. 180-191, 2006.

[3] S. Babbage. A Space/Time Tradeoff in Exhaustive Search Attacks on Stream Ciphers. In European *Convention on Security and Detection, IEEE Conference Publication No* 408, 1995.

[4] E.R. Berlekamp. *Algebraic Coding Theory*. Mcgraw-hill, 1968.

[5] E. Biham and A. Shamir. Differential Cryptanalysis of DES-like Cryptosystems. *Journal of Cryptology*, Vol 4, No.1, pp. 3-72, 1991.

[6] A. Biryukov, A. Shamir : Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers, *Asiacrypt 2000, LNCS 2248, Springer*, pp. 1-13.

[7] L. Budaghyan, C. Carlet and A. Pott. New Classes of Almost Bent and Almost Perfect Nonlinear Polynomials. *Proceedings of the Workshop on Coding and Cryptography 2005*, Bergen, pp. 306-315, 2005.

[8] L. Budaghyan, C. Carlet, A. Pott. New Classes of Almost Bent and Almost Perfect Nonlinear Functions. *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 1141-1152, March 2006.

[9] A. Canteaut. On the correlations between a combining function and functions of fewer variables. *Proceedings of the Information Theory Workshop'02*, Bangalore, 2002.

[10] A. Canteaut and M. Trabbia. Improved Fast Correlation Attacks Using Parity-Check Equations of Weight 4 and 5, *Advanced in Cryptology- EUROCRYPT 2000. Lecture notes in computer science* 1807, pp. 573- 588, 2000.

[11] C. Carlet. A method of construction of balanced functions with optimum algebraic immunity. Proceedings of the conference IWCC, Wuyishan, Chine, published by *World Scientific*, series of Coding and Cryptology, pp. 25-43, 2008.

[12] C. Carlet. On the Higher Order Nonlinearities of Algebraic Immune Functions. Proceedings of CRYPTO 2006, *Lecture Notes in Computer Science* 4117, pp. 584-601, 2006.

[13] C. Carlet. Boolean Functions for Cryptography and Error Correcting Codes, Chapter of the monography "Boolean Models and Methods in Mathematics, Computer Science, and Engineering", *Cambridge University Press* (Peter Hammer and Yves Crama editors), pp. 257-397, 2010. Preliminary version available at : http ://www.math.univ-paris13.fr/ carlet/chap-fcts-Bool-corr.pdf

[14] C. Carlet. Vectorial (multi-output) Boolean Functions for Cryptography. Chapter of the monography Boolean Methods and Models in Mathematics, Computer Science, and Engineering", *Cambridge University Press* (Peter Hammer and Yves Crama editors), pp. 398-469, 2010. Preliminary version available at : http ://www.math.univ-paris13.fr/ carlet/chap-vectorial-fcts-corr.pdf

[15] C. Carlet, P. Charpin, and V. Zinoviev. Codes, Bent Functions and Permutations Suitable for DES-like Cryptosystems. *Designs, Codes and Cryptography*, 15(2), pp. 125-156, 1998.

[16] C. Carlet. On the Higher Order Nonlinearities of Boolean Functions and S-boxes, and Their Generalizations. Proceedings of *SETA 2008, Lecture Notes in Computer Science* 5203, pp. 345-367, 2008.

[17] 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.

[18] 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 : Pieprzyk J. (eds.) *ASIACRYPT 2008. LNCS*, vol. 5350, pp. 425–440. *Springer, Heidelberg* (2008).

[19] C. Carlet and K. Feng, An Infinite Class of Balanced Vectorial Boolean Functions with Optimal Algebraic Immunity and Good Nonlinearity, *CODING AND CRYPTOLOGY, Lecture Notes in computer sciences*, 2009, Volume 5557/2009.

[20] 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*, 23, pp. 104 - 116, 2009.

[21] F. Chabaud and S. Vaudenay. Links Between Differential and Linear Cryptanalysis. *Proceedings of EUROCRYPT'94, Lecture Notes in Computer Science* 950, pp. 356-365, 1995.

[22] V. Chepyzhov and B. Smeets. On a Fast Correlation Attack on Certain Stream Ciphers. *Proceedings of EUROCRYPT'91, Lecture Notes in Computer Science* 547, pp. 176-185, 1992.

[23] N. Courtois. Fast Algebraic Attacks on Stream Ciphers with Linear Feedback. In Advances in Cryptology - CRYPTO 2003, volume 2729 of *Lecture Notes in Computer Science*, pages 176-194. Springer-Verlag, 2003.

[24] N. Courtois : Algebraic Attacks on Combiners with Memory and Several Outputs. *Proceedings of ICISC 2004, Lecture notes in computer science* 3506, pp. 3-20, 2005.

[25] N. Courtois And J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations, *advances in cryptology - ASIACRYPT 2002, LNCS 2501. springer-verlag*, 2002.

[26] N.Courtois, W. Meier : Algebraic attacks on stream ciphers with linear feedback, proceedings of *EUROCRYPT 2003, LNCS* 2656, pp. 345-359, *Springer*, 2003. an extended version is available at http ://www.cryptosystem.net/stream/

[27] D.K. Dalai, K.C. Gupta and S. Maitra. Cryptographically Significant Boolean Functions : Construction and Analysis in terms of Algebraic Immunity. *Fast Software Encryption 2005, Lecture Notes in Computer Science* 3557, pp. 98-111, 2005.

[28] F. Didier, A new bound on the block error probability after decoding over the erasure channel. *IEEE Transactions on Information Theory*, 52(10) :4496–4503, October 2006.

[29] J.C. Faugère and G. Ars. An Algebraic Cryptanalysis of Nonlinear Filter Generators using *Gröbner* bases. *Rapport de Recherche INRIA* 4739, 2003.

[30] K. Feng, Q. Liao and J. Yang. Maximal values of generalized algebraic immunity. *Designs, Codes and Cryptography* 50, (2), pp. 243-252, 2009.

[31] S. Fischer and W. Meier. Algebraic Immunity of S-boxes and Augmented Functions. *Proceedings of Fast Software Encryption 2007. Lecture Notes in Computer Science* 4593, pp. 366-381, 2007.

[32] X. Guo-Zhen and J.L. Massey. A Spectral Characterization of Correlation-Immune Combining Functions. *IEEE Trans. Inf. Theory*, vol. 34, no. 3, pp. 569-571, 1988.

[33] P. Hawkes, and G.G. Rose. Rewriting Variables : The Complexity of Fast Algebraic Attacks on Stream Ciphers. In *Advances in Cryptology - CRYPTO 2004*, LNCS 3152, pages 390-406. Springer Verlag, 2004.

[34] T. Iwata and K. Kurosawa. Probabilistic higher order differential attack and higher order bent functions. Proceedings of *ASIACRYPT'99, LNCS* 1716, pp. 62-74, 1999.

[35] T. Jakobsen and L.R. Knudsen. The interpolation attack on block ciphers. *Proceedings of FSE' 97, Lecture Notes in Computer Science* 1267, pp. 28-40, 1997.

[36] E. Key. An analysis of the structure and complexity of nonlinear binary sequence generators. *IEEE Transaction on information theory*, 22(6) :732–736, 1976.

[37] L. Knudsen. Truncated and Higher Order Differentials. In B. Preneel, editor, *Fast Software Encryption – FSE' 94*, volume 1008 of LNCS, pages 196–211. Springer, 1994.

[38] L.R. Knudsen and M.P.J. Robshaw. Non-linear approximations in linear cryptanalysis. *Proceedings of EUROCRYPT'96, Lecture Notes in Computer Science* 1070, pp. 224-236, 1996.

[39] 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. 1994.

[40] R. Lidl and H. Niederreiter. *Finite Fields*, Encyclopedia of Mathematics and its Applications, vol. 20, Addison-Wesley, Reading, Massachus-setts, 1983.

[41] M. Lobanov. Tight bound between nonlinearity and algebraic immunity. *IACR ePrint Archive http ://eprint.iacr.org/ 2005/441.*

[42] F.J. Macwilliams And N.J. Sloane. *The theory of error-correcting codes*, Amsterdam, North Holland. 1977.

[43] J.L. Massey. Shift-register synthesis and BCH decoding . *IEEE transactions on information theory*, 15 :122-127, 1969.

[44] M. Matsui. Linear cryptanalysis method for DES cipher. *Proceedings of EUROCRYPT'93, Lecture Notes in Computer Science* 765, pp. 386- 397, 1994.

[45] W. Meier and O. Staffelbach. Fast correlation attacks on stream ciphers. *Advances in Cryptology, EUROCRYPT'88, Lecture Notes in Computer Science* 330, pp. 301-314, 1988.

[46] W. Meier, E. Pasalic, and C. Carlet, Algebraic Attacks and Decomposition of Boolean Functions, *Advances in Cryptology EUROCRYPT 2004, LNCS* 3027, pp.474-491, Springer-Verlag, 2004.

[47] A. Menezes et al. *Handbook of applied cryptography.* CRC press series on discrete mathematics and its applications. 1997.

[48] 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, no. 8, pp. 3656 - 3662, 2008. Preliminary version available at IACR ePrint Archive http ://eprint.iacr.org/, 2007/117.

[49] K. Nyberg. Perfect non-linear S-boxes. *Proceedings of EUROCRYPT' 91, Lecture Notes in Computer Science* 547, pp. 378-386, 1992.

[50] K. Nyberg. On the construction of highly nonlinear permutations. *Proceedings of EUROCRYPT' 92, Lecture Notes in Computer Science* 658, pp. 92-98, 1993.

[51] 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* 473, pp. 161-173, 1991.

[52] R.L. Rivest, A. Shamir and L.M. Adelman. A method for obtaining digital signatures and public-key cryptosystems. Technical Report mit/lcs/tm-82, 1977.

[53] Robert G. Gallager. *Information Theory and Reliable Communication.* John Wiley and Sons, Inc., New York, NY, USA, 1968.

[54] R.A. Rueppel. *Analysis and design of stream ciphers.* Springer-verlag, 1986.

[55] R.A. Rueppel and O.J. Staffelbach. Products of linear recurring sequences with maximum complexity. *IEEE Transactions on Information theory*, vol. 33, no. 1, 1987.

[56] C.E. Shannon : Communication theory of secrecy systems, *Bell System Technical Journal*, 28 (1949), pp. 656-715.

[57] B. Schneier. *Applied cryptography*. second edition. John Wiley , sons. 1996.

[58] V. M. Sidelnikov. On the mutual correlation of sequences, *Soviet Math. Dokl.* 12, pp. 197-201, 1971.

[59] T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information theory*, vol. 30, no 5, pp. 776-780, 1984.

[60] T. Siegenthaler. Decrypting a class of stream ciphers using ciphertext only. *IEEE transactions on information theory*, 34(1) : 81-84, 1985.

[61] Z.G. SUN, B. Sun and B.C. Li. Algebraic Immunity of Boolean Functions. Progress in Communication Theory and Technology. *Proceedings of the Tenth National Youth Communications Conference*. Beijing : Beijing University of Posts and Telecommunications Press, 2005.

[62] A.F. Webster and S.E. Tavares. On the design of S-boxes. In *Proceedings of CRYPTO'85, Lecture Notes in Computer Science* 219, pp. 523-534, 1985.

[63] V.K. Wei, Generalized Hamming weights for linear codes, *IEEE Transaction on Information Theory*, vol. 37, pp. 1412–1418, Sept. 1991.

[64] Xiao Guo-Zhen, C. Ding and W. Shan. The stability theory of stream ciphers, *Lecture Notes in Computer Science* 561, 1991.

[65] C. Carlet and B. Merabet. Asymptotic lower bound on the algebraic immunity of random balanced multi-output boolean functions. *Advances in Mathematics of Communications*, Volume 7, No. 2, 2013, 197–217.