Critères de sécurité des algorithmes de chiffrement à clé secrète

Marion Videau

THÈSE de DOCTORAT de l'UNIVERSITÉ PARIS 6

Spécialité

INFORMATIQUE

présentée par

Marion VIDEAU

pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PARIS 6

Sujet de la thèse

CRITÈRES de SÉCURITÉ des ALGORITHMES de CHIFFREMENT à CLÉ SECRÈTE

soutenue le 10 novembre 2005

devant le jury composé de

Anne CANTEAUT Pascale CHARPIN	Chargée de recherche à l'INRIA Directrice de recherche à l'INRIA	Directrices de thèse
Tor HELLESETH Matt ROBSHAW	Professeur à l'université de Bergen (Norvège) Senior Security Expert à France Telecom R&D	Rapporteurs
Daniel LAZARD	Professeur émérite à l'université de Paris 6	Président du jury
Antoine JOUX	Délégation Générale pour l'Armement, HDR Professeur associé à l'université de Versailles St-Quentin-en-Yvelines	Examinateur
Didier ALQUIÉ	Délégation Générale pour l'Armement	Invité

CRITÈRES de SÉCURITÉ des ALGORITHMES de CHIFFREMENT à CLÉ SECRÈTE

Marion VIDEAU

CRITÈRES de SÉCURITÉ des ALGORITHMES de CHIFFREMENT à CLÉ SECRÈTE

version du 10/02/2006

À la mémoire de Jean Blanchard

À mes parents

Avant-propos

Cette thèse a été réalisée au sein du projet CODES de l'INRIA-Rocquencourt du 1er octobre 2001 au 31 août 2005. Une partie des travaux a été commencée pendant le stage de DEA qui s'est déroulé dans les mêmes lieux, du 1er mars 2001 au 31 août 2001. J'ai par ailleurs séjourné du 4 octobre 2004 au 3 avril 2005 au centre Selmer, sous la direction de Tor Helleseth, dans le groupe de théorie des codes et de cryptologie de l'université de Bergen en Norvège, site de formation Marie Curie (FASTSEC) dans le cadre du programme Improving Human Research Potential and the Socio-economic Knowledge Base de la Communauté Européenne. Les travaux de cette thèse ont été menés sous la direction d'Anne Canteaut et de Pascale Charpin. Ils portent sur les critères de sécurité des algorithmes de chiffrement à clé secrète. Une première partie a pour objet l'étude de cryptanalyses de chiffrements itératifs par blocs. Elle a donné lieu à des publications dans les actes des conférences *EUROCRYPT 2002* [CV02a], *ISIT 2002* [CV02b] et du *Workshop honoring Bob McEliece on his 60th birthday* [CCV02] ; une version longue est disponible sous forme de rapport de recherche INRIA [CV02c]. La deuxième partie s'intéresse à l'étude des propriétés cryptographiques des fonctions booléennes symétriques. Elle a fait l'objet de publications dans les actes de la conférence *ISIT 2004* [Vid04] d'un *regular paper* dans le journal *IEEE-IT* [CV05] et d'une intervention durant le *Western European Workshop on Research in Cryptology* [Vid05].

Remerciements

La toute première expression de ma gratitude en cet incontournable exercice de remerciements ne saurait être destiné à nul autre qu'à Anne Canteaut, ma directrice de thèse. Son admirable générosité allié à son enthousiasme scientifique intarissable a accompagné et soutenu sans faillir les longues et angoissantes années de thèse qui s'achèvent ici (et qui furent précédées d'un stage de DEA qu'elle a aussi encadré). Elle n'aura nullement fait mentir la fable du petit lapin qui fait une thèse[1] *et je lui dois sans nul doute une part incommensurable de l'élan avec lequel je me prépare à l'*après*... Qu'elle trouve ici l'expression de ma profonde reconnaissance et de mon admiration.*

Je n'oublie pas que la chance m'a pourvue de deux directrices de thèse et je souhaite remercier Pascale Charpin pour m'avoir fait bénéficier de sa grande expérience et de son chaleureux soutien, tout spécialement pendant cette période délicate de la thèse qu'a été la deuxième année. La seconde partie des travaux de cette thèse doit son existence à la forme de sixième sens extra-lucide qui la caractérise. Je souhaite également lui réitérer mes remerciements pour avoir accepté de m'accueillir au projet CODES *pour mon stage de DEA qui a été le point de départ des travaux que je présente ici.*

Je tiens à remercier Tor Helleseth et Matt Robshaw pour m'avoir fait l'honneur d'être rapporteurs de ma thèse. La lecture attentive dont a bénéficié mon manuscrit alors même qu'il est rédigé dans une langue qui n'est pas la leur, représente la meilleure récompense qui se puisse concevoir et je tiens ici à leur exprimer ma profonde gratitude.

Je tiens à réitérer mes remerciements aux membres de mon jury de thèse.

Je remercie Daniel Lazard pour m'avoir fait l'honneur d'accepter de présider le jury de ma thèse.

Je remercie Antoine Joux pour avoir accepté de faire partie du jury de ma thèse.

Je remercie Didier Alquié pour avoir accepté de faire partie du jury de ma thèse et pour ses commentaires sur le manuscrit. Je remercie également la DGA à travers lui pour les trois années d'allocation de recherche dont j'ai bénéficié pour mener à bien mes travaux.

Des remerciements tout particuliers à Isabelle de Lamberterie, sans qui, il m'aurait sans doute été très difficile de soutenir ma thèse dans d'aussi bonnes conditions.

Ces remerciements sont pour moi l'occasion d'un retour nostalgique en arrière. Le chemin parcouru, si incertain à son début, n'aurait pu l'être sans l'intervention de nombreuses personnes que je souhaite ici remercier plus personnellement.

En premier lieu, je souhaite rendre hommage à Jean Blanchard. Il fut mon professeur de mathématiques en classe préparatoire et a su m'encourager à poursuivre dans une voie dont nul sinon lui ne prévoyait qu'elle m'était destinée. Sa générosité et sa gentillesse m'ont été trop tôt ravis ; je lui aurais présenté mes résultats avec beaucoup de fierté et d'émotion. Je dédie tout naturellement cette thèse à sa mémoire.

1. pour ressentir l'indéniable fond de vérité de cette fable il semblerait qu'une thèse soit un pré-requis...

Aux hasards bienveillants de l'existence je dois la rencontre avec Marc Reversat, professeur à l'université Paul Sabatier à Toulouse qui a su orienter mes questionnements vers les bonnes personnes, à commencer par Jean-Marc Couveignes. Je dois à ce dernier de judicieux conseils et la liste des contacts parmi lesquels figurait le projet CODES. *Je remercie également le département maths-info de l'ENSICA pour son soutien à l'orientation inattendue de mes centres d'intérêt.*

Je remercie à nouveau Jean-Marc Couveignes pour avoir accepté, l'année suivant ses conseils, de présider le jury de ma soutenance de DEA. Je garde intactes en mémoire les conditions rocambolesques de non-soutenance, un 21 septembre 2001 à Toulouse, puis rapidement hors de Toulouse... Souvenirs inoubliables rappelés désormais chaque année par la radio.

Je tiens à remercier le projet CODES, *ses permanents et ses* éphémères, *pour le lieu de vie et de recherche exceptionnel qu'il a constitué pendant mes années de thèse. L'enthousiasme, le bonheur évident à la recherche, l'ouverture d'esprit et la bonne humeur qui y règnent ont indéniablement développé mon goût pour la recherche. Je me dois de souligner en particulier le souci remarquable de ses membres aux nombreux problèmes matériels que peut rencontrer un doctorant. Des remerciements tout particuliers à Christelle Guiziou dont l'efficacité, la sollicitude, et l'attention parviennent presque à faire oublier qu'il existe des tracasseries administratives. Merci à la chaleureuse équipe de permanents, Nicolas Sendrier, Daniel Augot, Jean-Pierre Tillich, Pascale Charpin (bis), Anne Canteaut (bis), ainsi que de celles qui ne sont pas vraiment parties, Françoise Lévy-dit-Vehel et Caroline Fontaine. Une pensée particulière à ceux qui furent doctorants quand j'arrivai: Éric, Pierre, Cédric, à ceux qui furent doctorants avec moi Fabien, Carmen, Matthieu, Harold, Ludovic, Magali, Emmanuel, Aline et enfin aux* scarabées *suivants, Raghav, Mathieu, Cédric, Thomas, Yann, Frédéric, Bassem, Andréa et Stéphane. À qui veut savoir ce que signifie* sciences conviviales, *j'indiquerai votre adresse.*

Des remerciements particuliers à Magali pour son aide et son soutien dans la prise en compte des contingences liées à la soutenance de thèse. En effet, ce n'est pas tout d'écrire une thèse, il faut aussi la soutenir, exercice dont la part de complexité administrative ne peut être surestimée.

Nul doute que j'englobe tous les visiteurs du projet, assidus ou inconstants (ou stagiaires), dans ces remerciements. Ils contribuent à l'activité scientifique bouillonnante qui y règne et fait de ce lieu un laboratoire à transformer les rencontres — *autour d'un café bien sûr* — en théorèmes. *Deux chercheurs en particulier ont fait preuve d'une grande bienveillance à mon égard dont je tiens à les remercier spécialement. Je pense en premier lieu à Grischa Kabatiansky, ses cours de codes et son humour tout aussi délicieux que les friandises qu'il n'oublie jamais d'apporter. Je lui dois d'avoir rencontré le professeur Hyun Kwang Kim, que je remercie pour m'avoir accueillie pendant* 5 *semaines au département de mathématiques de l'université des sciences et technologies de Pohang en Corée. D'autre part, je ne saurais oublier de remercier Henk van Tilborg pour l'intérêt constant qu'il a porté à l'avancement de mes travaux.*

Je remercie par ailleurs Michaël Quisquater pour les discussions que nous avons eues ensemble. J'espère que nous parviendrons à faire aboutir le travail que nous avons entamé.

Durant ma thèse, j'ai également séjourné pendant six mois au centre Selmer, dans le groupe de théorie des codes et de cryptologie de l'université de Bergen en Norvège dans le cadre d'un programme européen Marie Curie. Je remercie Tor Helleseth pour m'y avoir chaleureusement accueillie. Ces remerciements s'adressent aussi à tous les membres de l'équipe qui ont contri-

bué, par leur disponibilité et leur gentillesse, à rendre ce séjour plus qu'agréable et à reléguer l'hiver et ses trombes d'eau que le soleil (sic) n'éclairait qu'entre dix heures du matin et trois heures de l'après-midi au plan de curiosités touristiques amusantes. Une pensée toute particulière s'adresse à Matthew Parker dont j'ai pu apprécier l'humour tout britannique et les nombreux commentaires au sujet de mes travaux. J'espère que notre collaboration pourra se poursuivre. Je souhaite également remercier Monika Voit pour le souci constant dont elle a fait preuve pour simplifier toutes les démarches administratives pendant mon séjour. Une telle attention et une telle gentillesse sont de biens précieux alliés lorsqu'on est étranger.

Les nombreux chercheurs et doctorants rencontrés pendant les conférences, les séminaires ou les écoles de jeunes chercheurs ne seront pas oubliés. Je salue en particulier mes acolytes caennais pour un certain carnaval.

Il est bien difficile de ne pas faire que la thèse envahisse peu à peu sa vie privée, à commencer par les collègues qui deviennent des amis — heureusement que les amis ne deviennent pas des collègues... Ces quelques lignes de remerciements plus personnelles s'adressent en premier lieu à Raphaël qui a suivi cette thèse au quotidien et dont l'indéfectible soutien n'est pas étranger à sa réussite. Je tiens aussi à le remercier pour la relecture critique et attentive à laquelle il a soumis ce manuscrit dont le contenu lui était — et lui demeure — parfaitement indifférent, ce qui donne bien plus de valeur à l'exercice.

Mes amis se reconnaîtront certainement et je les remercie de leur présence et de leur soutien.

L'angoisse m'étreint, que ces dernières lignes ne sanctionnent un impardonnable oubli de ceux que je n'aurais pas dû oublier. Qu'ils reçoivent ici l'expression de mon estime et de ma gratitude.

Notice

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

$\mathbf{F}_q$	corps fini à q éléments
$\mathcal{B}_n$	ensemble des fonctions booléennes à n variables (p. 28)
$\mathrm{wt}(x)$	poids de Hamming de x (p. 29)
$d(x,y)$	distance de Hamming entre x et y (p. 29)
$\deg(f)$	degré de la forme algébrique normale d'une fonction booléenne f (p. 30)
$\mathrm{supp}(f)$	support d'une fonction booléenne f (p. 30)
$\mathcal{F}(f)$	déséquilibre d'une fonction booléenne f (correspond au coefficient de Walsh en 0 de f) (p. 32)
φ_a	fonction linéaire définie par $x \mapsto a \cdot x$ (p. 30)
$\mathcal{F}(f+\varphi_a)$	coefficient de Walsh de f en a (correspond au déséquilibre de $f+\varphi_a$) (p. 32)
$\{\mathcal{F}(f+\varphi_a), a \in \mathbf{F}_2^n\}$	spectre de Walsh de f (p. 32)
$\mathcal{L}(f)$	linéarité de f, *i.e.* amplitude maximale des coefficients de Walsh de f (p. 32)
$\mathcal{N}(f)$	non-linéarité de f, *i.e.* distance de f à l'ensemble des fonctions linéaires (p. 32)
$D_a f$	dérivée d'une fonction booléenne f relativement au vecteur a : $D_a f : x \mapsto f(x) \oplus f(x+a)$ (p. 33)
$\{\mathcal{F}(D_a f), a \in \mathbf{F}_2^n\}$	spectre d'auto-corrélation de f (p. 33)
$D_V f$	dérivée d'ordre $\dim(V)$ d'une fonction booléenne f relativement au sous-espace vectoriel V : $D_V f : x \mapsto \bigoplus_{v \in V} f(x+v)$ (p. 33)
$\mathcal{S}ym_n$	ensemble des fonctions symétriques à n variables (p. 103)
S_n	ensemble des permutations de $\{1,\ldots,n\}$ (p. 103)
$v(f)$	vecteur simplifié des valeurs de $f \in \mathcal{S}ym_n$: $v(f) = (v_f(0), v_f(1), \ldots, v_f(n))$, où $\forall x \in \mathbf{F}_2^n$, $f(x) = v_f(\mathrm{wt}(x))$ (p. 103)
$\lambda(f)$	vecteur simplifié de l'ANF de $f \in \mathcal{S}ym_n$: $\lambda(f) = (\lambda_f(0), \lambda_f(1), \ldots, \lambda_f(n))$, où $\forall x \in \mathbf{F}_2^n$, $f(x) = \bigoplus_{i=0}^{n} \lambda_f(i) \bigoplus_{\substack{u \in \mathbf{F}_2^n \\ \mathrm{wt}(u)=i}} \left(\prod_{j=1}^{n} x_j^{u_j} \right)$ (p. 104)
f_{a+V}	restriction d'une fonction booléenne f au sous-espace affine $a+V$: $\forall x \in V$, $f_{a+V}(x) = f(a+x)$ (p. 130)

$\mathsf{F}(f)$	spectre de Walsh simplifié de $f \in \mathcal{S}ym_n$: $\mathsf{F}(f) = (\mathsf{F}_f(0),\mathsf{F}_f(1),\ldots,\mathsf{F}_f(n))$, où $\forall a \in \mathbf{F}_2^n$, $\mathcal{F}(f+\varphi_a) = \mathsf{F}_f(\mathrm{wt}(a))$ (p. 106)
$\mathsf{Ac}(f)$	spectre d'auto-corrélation simplifié de $f \in \mathcal{S}ym_n$: $\mathsf{Ac}(f) = (\mathsf{Ac}_f(0),\ldots,\mathsf{Ac}_f(n))$, où $\forall a \in \mathbf{F}_2^n$, $\mathcal{F}(D_a f) = \mathsf{Ac}(\mathrm{wt}(a))$ (p. 107)
$K_{j,n}$	polynôme générateur des polynômes de Krawtchouk binaires : $K_{j,n}(x) = (1-x)^j(1+x)^{(n-j)}$ (p. 106)
B_n	polynôme générateur des coefficients binomiaux : $B_n(x) = (1+x)^n$ (p. 151)
A_n	polynôme générateur des coefficients binomiaux alternés : $A_n(x) = (1-x)^n$ (p. 151)

Introduction

Les travaux de cette thèse s'inscrivent dans une longue tradition d'évaluation de la sécurité des algorithmes de chiffrement au regard des cryptanalyses dont ils font l'objet. En effet, on date les débuts de la cryptographie aux temps reculés où un être humain voulut communiquer une information écrite à un autre sans qu'un tiers puisse le comprendre, lequel tiers confronta sans plus tarder cette énigme à sa brillante intelligence... autant dire que ce petit jeu débuta avec l'Histoire ! Jeu d'enfants ou jeu diplomatique, élément de la panoplie du cabinet du roi autant que de l'intrigant, cette science mit régulièrement au défi la sagacité de ces tiers exclus, indésirables et incorrigiblement curieux qui n'eurent de cesse de soumettre les chiffres utilisés à toute technique permettant de les casser et de lire les messages dont ils protégeaient le secret. Et l'imagination romanesque n'eut plus qu'à s'envoler à la suite de récits haletants de messages au chiffre de la reine destinés à l'amant de l'autre côté de la Manche... L'ère moderne de cette science, qui prit naissance en 1949 suite à la publication de l'article fondateur de Claude E. Shannon [Sha49], perdit certes à cet instant beaucoup en superbe romanesque, mais vit la mise en place de la recherche académique en cryptologie rendant publiques la conception et l'analyse des algorithmes cryptographiques. Par ailleurs, la généralisation dans ce que d'aucuns nomment la *société de l'information* des applications mettant en jeu de la cryptographie rend d'autant plus indispensable la recherche publique en cryptologie soumettant tout nouveau système à l'expertise de la communauté scientifique. En effet, la migration accélérée de toute forme d'information vers des données numériques demande des systèmes dûment éprouvés et par là-même réputés sûrs, adaptés aux nombreuses contraintes des divers domaines auxquels elle s'applique. Il faut en effet répondre à des utilisations dans des contextes aussi contrastés que ceux, par exemple, de la carte bancaire, du téléphone cellulaire, des transactions commerciales en ligne ou des transmissions satellitaires.

La *cryptologie* ou science du secret présente deux visages complémentaires : la *cryptographie* et la *cryptanalyse*. La cryptographie, étymologiquement écriture secrète, est devenue par extension l'étude de cet art ; il s'agit de déterminer les manières les moins faillibles possibles de *chiffrer* des messages susceptibles d'être interceptés lors de leur transmission. La cryptanalyse en est son contrepoint : alors que pour le destinataire légitime du *cryptogramme*, il s'agit de le *déchiffrer* pour prendre connaissance du contenu du message, l'attaquant qu'est le cryptanalyste cherche à *décrypter* un message chiffré, c'est-à-dire à connaître le contenu du message sans posséder les codes ou les clés nécessaires à son déchiffrement. Ces deux aspects sont indissociables. En effet, la conception de systèmes cryptographiques sûrs ne saurait se faire sans connaissance sur les cryptanalyses antérieures d'autres systèmes.

La cryptographie est de prime abord connue pour ses applications militaires et diplomatiques. Elle recouvre en fait des domaines plus étendus que la simple confidentialité des données assurée par le chiffrement qu'elle évoque en premier lieu. Dans un contexte plus général de protection de l'information contre des tentatives d'accès non-autorisé à des données, on peut

classer les principales fonctionnalités offertes par la cryptographie comme suit :

- *le chiffrement*, qui est utilisé pour protéger le contenu d'un message contre sa divulgation à toute personne ne possédant pas le secret lui permettant de le lire en clair ;
- *l'identification*, qui permet de s'assurer de l'identité de la personne dont est issu le message reçu ;
- *l'authentification*, qui assure que les données reçues n'ont subi aucune modification depuis leur émission première ;
- *la signature*, qui comme une signature manuscrite protège la provenance, l'intégrité et garantit la non-répudiation d'un message.

Des applications cryptographiques permettent de combiner plusieurs des fonctionnalités ci-dessus. Nous nous intéresserons dans cette thèse au chiffrement. Fonctionnalité la plus ancienne et de ce fait la plus étudiée, elle se décline en *chiffrement symétrique* et *chiffrement asymétrique* aux contextes d'étude radicalement différents dont je vais décrire brièvement les principes. Mes travaux portent sur le chiffrement symétrique.

Principe général du chiffrement

Le cas d'école met en présence deux protagonistes, *Alice* et *Bob*. Alice souhaite envoyer un message m à Bob à travers un canal de communication susceptible d'être espionné par un tiers à tout instant. Dans cette situation les personnages ont besoin :

- d'un algorithme de chiffrement E qui prend en paramètre une quantité secrète appelée clé de chiffrement ;
- d'un algorithme de déchiffrement D qui prend en paramètre une quantité secrète appelée clé de déchiffrement ;
- d'une clé de chiffrement K_e ;
- d'une clé de déchiffrement K_d.

Une formalisation simple des principes de chiffrement et de déchiffrement peut être donnée par les relations suivantes : $c = E_{K_e}(m)$ et $m = D_{K_d}(c)$ où c représente le message chiffré. On peut modéliser le principe par la figure 1.

Le système tel qu'il est modélisé 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. On distingue en général trois types de chiffrement :

- *Le chiffrement à algorithme restreint*: c'est le type de chiffrement le plus ancien. La confidentialité du message chiffré repose sur le secret de l'algorithme de chiffrement. Ce type de chiffrement n'est pas utilisé à grande échelle et a émané pendant très longtemps d'instances diplomatiques ou militaires ;
- *Le chiffrement à clé secrète*, encore appelé *chiffrement symétrique* ou *chiffrement conventionnel*: c'est le type de chiffrement le plus répandu. Dans le schéma précédent, il correspond à des clés de chiffrement et de déchiffrement identiques ou facilement déductibles l'une de l'autre, connues uniquement par l'émetteur et le destinataire. L'algorithme est quant à lui public et la confidentialité du message échangé repose uniquement sur le secret de la clé partagée. Se posent alors inévitablement deux questions, d'une part celle de l'échange sécurisé de la clé et d'autre part celle du nombre de clés à générer dans un réseau à n points où chaque personne doit posséder une clé différente pour chacune des communications avec les $(n-1)$ autres personnes (d'où un total de $n(n-1)/2$ clés pour

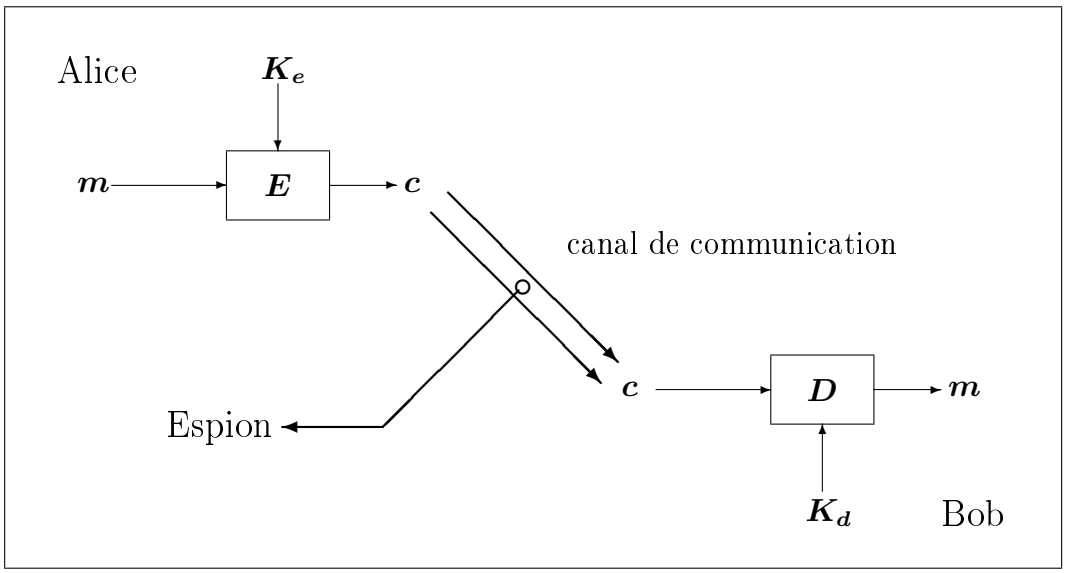


FIG. 1 – *Principe du chiffrement et du déchiffrement*

le réseau). Malgré ces inconvénients, ce type de système a l'avantage, pour des raisons algorithmiques, d'être extrêmement rapide (comparativement à sa contrepartie à clé publique) à un moindre coût, permettant d'atteindre des débits de l'ordre de centaines de mégabits par seconde ;

– *Le chiffrement à clé publique*, ou *chiffrement asymétrique* : ce type de chiffrement est de découverte bien plus récente puisqu'on date son invention de 1976 [DH76]. Idée révolutionnaire dont le principe fut curieusement publié avant toute instance la réalisant en pratique, ce type de chiffrement diffère radicalement du précédent. Les clés de chiffrement et de déchiffrement sont différentes. La clé de chiffrement ainsi que l'algorithme sont connus de tous. La sécurité du système repose sur le secret de la clé de déchiffrement et sur l'impossibilité (au moins en pratique) de déduire la clé de déchiffrement (dite clé privée) de la connaissance de la clé de chiffrement (dite clé publique). Chaque protagoniste possède une paire de clés, la clé publique étant publiée et la clé privée n'étant connue que de son propriétaire. Le nombre de clés à générer pour un réseau à n points est ainsi réduit à $2n$. Réponse indéniable à la question de l'échange sécurisé de la clé, le système en pose néanmoins une autre concernant l'authenticité du possesseur de la clé publique que l'on veut utiliser. Ces algorithmes sont en outre affligés d'une grande lenteur, avec des débits maximums de l'ordre de centaines de kilobits par seconde, qui les rend inaptes à une utilisation en ligne pour échanger des messages longs.

Les travaux de cette thèse portent uniquement sur les chiffrements symétriques. Ainsi dans les chapitres suivants nous ne considérerons que le cas où $K_e = K_d = K$.

Afin de tirer parti des avantages des algorithmes à clé secrète et des algorithmes à clé publique, les systèmes de chiffrement modernes les plus couramment utilisés sont des systèmes *hybrides* constitués à la fois d'algorithmes à clé secrète et à clé publique. L'échange de la clé secrète s'effectue grâce à l'algorithme à clé publique apportant une réponse à la question de l'échange sécurisé de la clé. La communication qui s'ensuit est chiffrée grâce à l'algorithme à clé secrète, ce qui permet de bénéficier de systèmes rapides pouvant traiter d'importants

volumes de données.

Contributions et organisation du document

Les travaux de cette thèse portent sur les critères de sécurité des algorithmes de chiffrement à clé secrète. Dans une première partie introductive, chapitre 1, je présente les principes du chiffrement symétrique et les différents contextes et formalisations des attaques contre ces systèmes. Je poursuis cette présentation en détaillant les deux familles de chiffrements résultant du traitement apporté au texte en clair :

- soit bit-à-bit par addition d'une suite chiffrante produite à partir de la clé secrète au texte clair, on parle alors de *chiffrement à flot synchrone* (ou encore chiffrement par flux, à la volée, au fil de l'eau...) ;
- soit par paquets de n bits en appliquant une transformation paramétrée par la clé secrète au groupe de n bits du texte clair, on parle alors de *chiffrement par blocs* et je m'intéresse en particulier aux *chiffrements itératifs par blocs* résultant de r itérations successives d'une même *fonction de tour* paramétrée par une sous-clé dérivée de la clé secrète.

Ces deux types de chiffrements conduisent à des modélisations mathématiques différentes mais présentent de nombreux points communs, par les fonctions sous-jacentes qui les composent, au travers de la théorie des fonctions booléennes à laquelle je me reporterai fréquemment au cours de mon étude. Une première partie est consacrée aux généralisations de la cryptanalyse différentielle sur les chiffrements itératifs par blocs, en particulier la cryptanalyse différentielle d'ordre supérieur ainsi qu'aux faiblesses introduites par l'utilisation de fonctions de non-linéarité maximale. La deuxième partie porte, quant à elle, sur l'étude des propriétés cryptographiques des fonctions booléennes symétriques, famille de fonctions qui peuvent se révéler avantageuses à utiliser en tant que fonction de filtrage pour des chiffrements à flot.

Généralisations de la cryptanalyse différentielle et attaque différentielle d'ordre supérieur sur les chiffrements de Feistel à 5 tours

Les chiffrements itératifs par blocs constituent un ensemble de briques de base largement utilisés dans divers contextes grâce aux différents modes opératoires disponibles. Ils peuvent ainsi être à l'origine de chiffrements à flot, d'applications d'authentification de message ou d'entité, de fonctions à sens unique ou de fonctions de hachage. D'un point de vue historique, on compte les standards ouverts de chiffrement que sont le DES et l'AES dans cette famille. Publié dans les années 1970 par le National Bureau of Standards américain, le DES [FIP99] (Data Encryption Standard) a été le standard mondial de facto en matière de chiffrement jusqu'à la fin des années 1990. Cependant les 56 bits de sa clé mis en regard de l'augmentation continue de la puissance de calcul des machines l'ont finalement rendu vulnérable aux attaques par recherche exhaustive, ce qui a conduit à son remplacement. Ainsi l'algorithme belge *Rijndael* rebaptisé AES [FIP01] (Advanced Encryption Standard) lui a succédé en tant que standard américain. Ces standardisations et les procédures y conduisant ont encouragé une recherche active concernant ces algorithmes sur lesquels repose la grande majorité des applications sécurisées disponibles actuellement.

Longtemps algorithme de référence, le DES a vu le développement de deux attaques génériques particulièrement efficaces, *la cryptanalyse différentielle* publiée en 1991 par Biham

et Shamir [BS91] et la *cryptanalyse linéaire* introduite en 1991 par Tardy-Corfdir et Gilbert [TCG91] et appliquée au DES en 1993 par Matsui [Mat93, Mat94]. Le concept de *sécurité démontrable* a vu le jour à leur suite, même si le terme démontrable devrait immédiatement être suivi de *contre les attaques différentielle et linéaire*. Démontrable donc dans la mesure où des bornes supérieures sur les probabilités de transition différentielle et linéaire ont pu être établies [NK95] ou dans le cadre de la théorie de la décorrélation [Vau98]. La prise en compte de ces critères fait ainsi partie de l'évaluation minimale de la sécurité de tout nouvel algorithme, algorithmes dont la conception est pour certains directement inspirée de ces théories. C'est le cas de MISTY1 ou de sa variante KASUMI [3GP] qui a été choisi comme algorithme de chiffrement et d'authentification de messages pour les mobiles de troisième génération très précisément pour sa sécurité démontrable et pour les facilités d'implémentation matérielle prévues. Les critères de résistance liés à ces deux attaques concernent essentiellement la partie non-linéaire des fonctions de tour des chiffrements itératifs par blocs. Les fonctions de tour pour lesquelles l'entrée et la sortie ont la même taille présentent une résistance maximale aux attaques différentielles si et seulement si elles sont *presque parfaitement non-linéaires* [NK93] et aux attaques linéaires si et seulement si elles sont *presque courbes* [CV95]. Ces dernières fonctions sont également presque parfaitement non-linéaires [Nyb91]. Il faut cependant garder à l'esprit que la sécurité démontrable des systèmes utilisant de telles fonctions ne garantit en aucun cas leur résistance contre d'autres types d'attaques. Les différentielles d'ordre supérieur introduite par Lai [Lai94] ont mis à la disposition des cryptanalystes d'autres outils d'analyse des systèmes permettant d'élaborer des attaques [Knu95].

Ces trois cryptanalyses peuvent être modélisées sous la forme d'une attaque sur le dernier tour avec distingueur, ce que je présente dans le chapitre 2. Le chiffrement MISTY1 sans fonction FL réduit à 5 tours est vulnérable à une attaque différentielle d'ordre supérieur [THK99] de cette forme et cette attaque reste valable lorsqu'on substitue à la fonction presque courbe utilisée toute autre fonction puissance pour laquelle l'exposant est de même poids de Hamming [BF00]. Les résultats que j'apporte dans les chapitres 3 et 4 sont directement liés à l'utilisation de telles fonctions. En effet, dans le chapitre 3, je montre comment le principe de l'attaque repose sur l'évaluation du degré de la composée de deux fonctions puissance presque courbes. D'autre part, on connaît une caractérisation de ses fonctions sous forme de codes cycliques [CCD99]. Ainsi, le théorème de McEliece reformulé dans le cas des codes cycliques à deux non-zéros [CCD00b] permet de majorer le degré de la composée de deux fonctions puissance. Cette propriété étend le champ de l'attaque différentielle d'ordre supérieur à toutes les fonctions puissance pour lesquelles la divisibilité du code cyclique associé est élevée, ce qui est le cas de toutes les fonctions puissance presque courbes connues. En fait, cette propriété est plus largement valable pour des fonctions booléennes vectorielles non nécessairement équivalentes à des fonctions puissance dans le corps fini associé. Je montre ainsi dans le chapitre 4 que dès que le spectre de Walsh de la fonction de tour est divisible par une grande puissance de 2 (ce qui est par définition le cas des fonctions presque courbes), la majoration sur le degré de la composée de deux fonctions s'applique, rendant possible une attaque différentielle d'ordre supérieur inspirée de celle de MISTY1, attaque valable pour tout chiffrement de Feistel à 5 tours. Cette cryptanalyse conduit naturellement à un critère de sécurité : une fonction de tour est résistante à l'attaque différentielle d'ordre supérieur proposée si son spectre de Walsh est faiblement divisible, ce qui signifie en particulier qu'elle ne doit pas être presque courbe. Cette dernière remarque souligne le caractère paradoxal des fonctions extrémales pour un critère donné. En effet, ces fonctions possèdent alors de si fortes structures algébriques qu'elles en deviennent les points faibles du système. Cette partie de mes travaux a fait l'objet de publi-

cations dans les actes des conférences *EUROCRYPT 2002* [CV02a], *ISIT 2002* [CV02b] et du *Workshop honoring Bob McEliece on his 60th birthday* [CCV02]. Une version longue est disponible sous forme de rapport de recherche INRIA [CV02c].

Propriétés cryptographiques des fonctions booléennes symétriques

En l'absence de standards dans leurs rangs, les chiffrements à flot ne bénéficient pas de la même publicité que leurs homologues par bloc d'autant que nombre d'entre eux sont propriétaires et/ou confidentiels. Ils n'en demeurent pas moins des primitives dont l'intérêt indéniable se matérialise dans les procédures récentes d'appels à candidature et d'évaluation de chiffrements à flot telles que celles lancées dans le cadre des projets NESSIE [NES01] et ECRYPT [ECR05] en Europe ou CRYPTREC [Cry01] au Japon. Ils se révèlent en effet indispensables dès qu'on cherche à atteindre des débits importants pour des coûts logiciels ou matériels limités. En outre, leur nature même leur fournit l'avantage de ne pas propager les erreurs. On les retrouve ainsi fréquemment dans des applications de télécommunications.

La majeure partie des chiffrements à flot sont construits suivant le principe d'un état interne dont l'évolution à chaque instant est contrôlée par une fonction de transition et qui fournit une entrée à une fonction de filtrage produisant une suite chiffrante. Dans le cas où la fonction de transition est linéaire, la sécurité de ces systèmes repose sur les propriétés cryptographiques de la fonction booléenne utilisée pour *filtrer* l'état interne à l'instant t. Outre la nécessité d'étudier des attaques potentielles contre de tels systèmes se pose la question des fonctions booléennes disponibles vérifiant les multiples critères à la fois cryptographiques et imposés par les implémentations de l'algorithme. Ainsi, dans la deuxième partie de ce document — à partir du chapitre 5, j'expose les résultats portant sur l'étude des *fonctions booléennes symétriques*.

Les fonctions booléennes symétriques possèdent la bonne propriété d'être représentables de manière peu coûteuse que ce soit sous forme logicielle ou matérielle. Candidates naturelles pour qui veut implémenter de tels systèmes, elles doivent avant d'entrer dans la conception d'un chiffrement à flot vérifier les nombreux critères issus des cryptanalyses connues. J'apporte dans les chapitres qui y sont consacrés une étude systématique des principaux critères requis : degré algébrique, équilibre, résilience, critère de propagation et non-linéarité (l'immunité algébrique fait l'objet d'une étude en cours et ne fait pas partie de cette thèse). Cette exploration se fonde sur la démonstration d'une propriété structurelle d'un vecteur de représentation des fonctions symétriques qui est périodique et dont la période est reliée au degré algébrique des fonctions.

Les travaux de Mitchell [Mit90] et de Yang et Guo [YG95] sur l'énumération de fonctions booléennes vérifiant simultanément plusieurs critères cryptographiques (parmi lesquelles on compte la symétrie) ont apportés les premiers résultats concernant l'équilibre et la résilience de ces fonctions. La question de l'existence de telles fonctions a été résolue par la détermination de plusieurs familles infinies de fonctions symétriques équilibrées [vzGR97, SM03], résilientes [GHS93, vzGR97] et sans corrélation [SM03]. La question qui demeure en suspens à ce propos concerne l'ordre de résilience maximal de ces fonctions qui est conjecturé ne pas dépasser 2. Dans ce contexte, en combinant la propriété de périodicité du vecteur représentant une fonction symétrique et celles de ses restrictions, j'ai pu déterminer une nouvelle borne sur l'ordre maximal de résilience atteignable par les fonctions symétriques qui améliore significativement les bornes existantes pour des fonctions de faible degré. De même, l'étude des propriétés des dérivées des fonctions symétriques m'a permis de caractériser les fonctions symétriques vérifiant le critère de propagation de degré 2 ($PC(2)$) — résultat également démontré par Aline Gouget [Gou04b] — de caractériser partiellement les fonctions vérifiant $PC(1)$ et

d'améliorer les résultats existant sur les structures linéaires [DW97]. Les définitions et propriétés des fonctions symétriques ainsi que ces résultats font l'objet du chapitre 5. J'expose dans le chapitre 6 la caractérisation complète des fonctions symétriques de degré 2 et 3, c'est-à-dire le calcul exhaustif de leur spectre de Walsh et leur spectre d'auto-corrélation obtenu grâce à la propriété de périodicité. Cette propriété me permet par ailleurs de déterminer toutes les fonctions booléennes symétriques équilibrées de degré inférieur ou égal à 7. Le chapitre 7 concerne la non-linéarité. En effet, on connaît la caractérisation complète des fonctions symétriques à n variables de non-linéarité maximale : lorsque n est pair, les fonctions courbes (de non-linéarité $2^{n-1} - 2^{\frac{n}{2}-1}$) sont les fonctions de degré 2 [Sav94]; lorsque n est impair, la non-linéarité maximum vaut $2^{n-1} - 2^{\frac{n-1}{2}}$ et est atteinte par des fonctions dont le spectre de Walsh est tri-valué, qui sont aussi dans ce cas les fonctions quadratiques [MS02]. Ces fonctions de faible degré algébrique ne sont pas utilisables dans un système de chiffrement, ce qui signifie qu'il est nécessaire d'étudier le cas des fonctions de non-linéarité légèrement plus faible mais de degré algébrique plus élevé. Je caractérise ainsi les fonctions symétriques à n variables de non-linéarité supérieure à $2^{n-1} - 2^{\lfloor \frac{n+1}{2} \rfloor} - 2^{t+1}$ pour $0 \leq t \leq \lfloor \frac{n+1}{2} \rfloor$. Je donne en particulier la description de toutes les fonctions de non-linéarité sous-optimale et de degré algébrique n ou $n-1$. Les fonctions symétriques sont ainsi un bon moyen d'atteindre un nombre de variables élevés tout en conservant une complexité et un encombrement mémoire raisonnable. Les valeurs de critères tels que la non-linéarité dépendent toujours directement de n et ne trouvent pas de simplification grâce à la symétrie. Au regard de la remarque sur les fonctions presque courbes de la première partie, il faut néanmoins toujours garder à l'esprit qu'une fonction fortement structurée est susceptible d'introduire des faiblesses dans le système qui l'utilise. La conception d'attaques exploitant le caractère symétrique des fonctions utilisées dans différents contextes reste donc un problème ouvert très important. Ces travaux ont fait l'objet de publications dans les actes de la conférence *ISIT 2004* [Vid04], dans le journal *IEEE-IT* [CV05] et d'une intervention durant le *Western European Workshop on Research in Cryptology* [Vid05].

Chapitre 1

Introduction à la cryptographie symétrique

Ce chapitre a pour ambition d'établir un état de l'art sur le chiffrement symétrique; sans prétendre à l'exhaustivité, mon objectif est qu'il soit suffisamment complet pour qu'une personne non-initiée aux arcanes de la cryptanalyse perçoive les motivations de mes travaux. Une première partie est consacrée à la définition moderne du niveau de sécurité requis d'un chiffrement symétrique et des modélisations auxquelles conduisent les différents contextes d'attaques que nous prendrons en compte. Nous nous intéresserons ensuite à la présentation des chiffrements à flot et des chiffrements par blocs. Dans les deux cas, nous nous attacherons à présenter brièvement les principes du système, les outils d'analyse disponibles, la modélisation mathématique sur laquelle il se fonde, les cryptanalyses dont il fait l'objet et les critères de sécurité qui en sont déduits. Pour clore ce préambule et avant de plonger plus avant dans l'analyse de la sécurité des algorithmes, il est indispensable de ne pas perdre de vue que la finalité d'un cryptosystème est d'être utilisé ce qui signifie que la sécurité est un paramètre primordial qui n'en demeure pas moins tributaire de sa faisabilité.

1.1 Utilisation de systèmes de chiffrement à grande échelle, premiers critères de sécurité

Il est courant de comparer un système de chiffrement symétrique à un coffre-fort. Un système de chiffrement à algorithme secret serait comme un coffre-fort à combinaison unique : si un voleur découvre la combinaison, il faut changer le coffre-fort. Force est d'admettre que le système devient ainsi vite contraignant. Aussi dans le cadre d'une utilisation à grande échelle, il n'est pas envisageable d'utiliser ce genre d'algorithmes dont on peut estimer que le secret sera forcément éventé à plus ou moins long terme. La version contemporaine de cette mésaventure est illustrée par l'histoire de l'algorithme *RC*4 au code indisponible, implémenté dans le protocole *ssl* et dont le secret fut révélé en septembre 1994, 7 ans après son invention par un message posté sur un newsgroup [Ano94]. La morale de cette histoire plaide pour la publication des algorithmes puisque des faiblesses ont rapidement été détectées conduisant l'entreprise propriétaire (RSA Data Security Inc., en l'occurrence) à apporter une réponse [Riv01]— qui constitue par ailleurs une reconnaissance de fait de l'algorithme publié, surnommé *alleged RC*4.

Les cryptosystèmes modernes obéissent à un certain nombre d'exigences nommées *desiderata de Kerckhoffs*. Publiées en 1883 dans *La cryptographie militaire* d'Auguste Kerckhoffs von Nieuvenhof, ces règles restent pour la plupart d'actualité. Ayant pour origine les changements apportés aux communications militaires par le télégraphe qui en étendait la portée et

la rapidité, elles prennent en compte pour la première fois dans une courte série de préceptes clairement définis les exigences d'un déploiement à grande échelle de la cryptographie et l'importance grandissante de la cryptanalyse mettant à l'épreuve les procédés de chiffrement. Il fut ainsi conduit à énoncer les conditions suivantes comme règles minimales de conception d'un chiffrement symétrique :

1. le système doit être matériellement, sinon mathématiquement, indécryptable ;
2. il faut qu'il n'exige pas le secret et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;
3. la clé doit pouvoir en être communiquée et retenue sans le secours de notes écrites et être changée et modifiée au gré des correspondants ;
4. il faut qu'il soit applicable à la correspondance télégraphique ;
5. il faut qu'il soit portatif et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes ;
6. le système doit être d'un usage facile ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.

La règle 2 souligne bien la nécessité des systèmes de chiffrement à algorithmes publics dont la sécurité ne repose que sur le secret de la clé. Par ailleurs, le fait de rendre publiques les méthodes de chiffrement et de déchiffrement offre une certaine garantie sur la sécurité d'un système, dans la mesure où tout nouvel algorithme peut immédiatement bénéficier de l'expertise de la communauté scientifique.

1.2 Modélisation d'un système de chiffrement

Pour préciser le modèle décrit à la figure 1.1 et afin de fixer le vocabulaire et les notations pour la suite du document, nous reprenons la description du principe général du chiffrement décrit au chapitre d'introduction, adapté au cas du chiffrement symétrique. Nous nous inspirons pour cela du chapitre 1 du *Handbook of cryptography* [MvOV97].

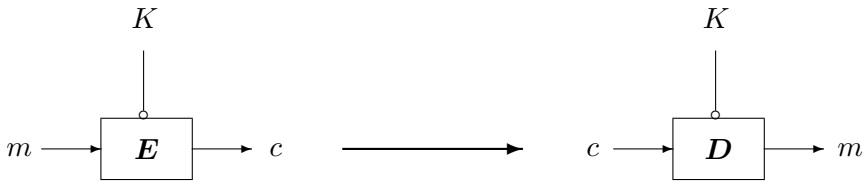


FIG. 1.1 – *Système de chiffrement symétrique*

La définition d'un *système de chiffrement* (parfois appelé *chiffre* anciennement *code secret*) comprend la donnée de :

- un *espace des textes en clair* (dits aussi clairs), noté $\mathcal{M}$;
- un *espace des textes chiffrés* (dits aussi chiffrés), noté $\mathcal{C}$;
- un *espace des clés* noté $\mathcal{K}$;
- un ensemble de transformations de chiffrement $\{E_K, K \in \mathcal{K}\}$;
- un ensemble de transformations de déchiffrement $\{D_K, K \in \mathcal{K}\}$.

Les espaces des clairs, chiffrés et clés sont définis dans notre cas sur l'alphabet binaire {0,1}. Étant donnée cette définition, il devient ainsi évident que la taille de chacun de ces espaces joue également un rôle dans la sécurité du système (il suffit pour cela de s'imaginer un espace des clairs de taille 2 qui ne contiendrait par exemple que les mots *oui* et *non*...). Cette remarque permet d'expliquer l'attention portée à des paramètres tels que la taille de la clé (afin d'éviter une recherche exhaustive) ou la taille des blocs pour un chiffrement par blocs (pour éviter une attaque par dictionnaire).

La modélisation mathématique d'un système de chiffrement symétrique repose ainsi sur quelques éléments principaux. Les messages à chiffrer sont donc représentés sous forme binaire. L'ensemble {0,1} des valeurs prises par un bit est représenté par le corps fini à deux éléments $\mathbf{F}_2$ dont $\oplus$ désigne l'opération additive (OU-EXCLUSIF) et $\cdot$ (ou rien) représente l'opération multiplicative (ET). Les messages sont découpés en blocs logiques dont la taille dépend du type de chiffrement considéré (à flot ou par blocs). Les messages de taille n sont représentés par des vecteurs de $\mathbf{F}_2^n$ ou parfois par des éléments du corps fini à 2^n éléments, $\mathbf{F}_{2^n}$. Les opérations dans ces ensembles seront usuellement notées $+$ pour l'opération additive et $\cdot$ pour l'opération multiplicative. Nous aurons par ailleurs à considérer la famille des applications opérant sur ces ensembles, les *fonctions booléennes*, ensemble des fonctions $f : \mathbf{F}_2^n \to \mathbf{F}_2^m$. Nous donnons une introduction aux fonctions booléennes au paragraphe 1.5 page 28.

1.3 Attaques sur les systèmes de chiffrement

Le chiffrement correspond certainement à l'aspect le plus visible de ce qui constitue la sécurité dans un environnement numérique. Il ne représente néanmoins qu'un maillon de la chaîne permettant d'assurer la sécurité d'informations confidentielles. Si nous reprenons la métaphore du coffre-fort, le chiffrement se matérialiserait dans la serrure ; ainsi, la solidité des parois du coffre, par exemple, ne relève pas de notre propos. En d'autres termes, cela signifie que de tout temps, une des meilleures façons d'obtenir des informations secrètes a été de soudoyer des personnes et d'exploiter l'erreur humaine. Nous ne traitons bien sûr pas de ces aspects dans ce document.

Un autre paramètre essentiel à prendre en compte pour la protection de données chiffrées est le temps. Ce paramètre est relatif aux questions concernant la durée d'utilisation envisagée pour un système, le temps de validité d'un message chiffré, la fréquence d'envoi des messages ou, de la même manière, la quantité d'information chiffrée avec la même clé en un laps de temps donné. Les réponses à ces questions permettent d'évaluer le matériel chiffré à la disposition de l'attaquant, l'avantage que lui confère un décryptement selon l'instant auquel il intervient et le danger que représente la détermination de la clé secrète utilisée à une date donnée.

Selon les données à la disposition du cryptanalyste, ce dernier peut être en mesure de retrouver différentes informations, classées ici par ordre décroissant d'importance :

- la clé secrète. Dans ce cas le système est dit complètement cassé ;
- un algorithme globalement équivalent au chiffrement ou au déchiffrement, c'est-à-dire une application $f = E_K$ ou $g = D_K$ pour l'instance de K attaquée ;
- le clair correspondant à un chiffré donné. Dans ce cas le système est dit partiellement cassé ;
- un distingueur. Cet algorithme permet, dans le cas des chiffrements à flot, de déterminer si la chaîne binaire est produite par une source réelle d'aléa ou par un chiffrement donné. Dans le cas des chiffrements par blocs, il s'agit en revanche de discriminer le

sous-ensemble des permutations déterminées par la clé secrète d'un sous-ensemble de permutations tirées aléatoirement. Dans tous les cas, il s'agit de trouver un moyen de reconnaître un ensemble déterminé par les clés de chiffrement possibles, d'un sous-ensemble de même cardinal tiré aléatoirement.

Les systèmes à clé secrète, systèmes dont l'existence est la plus ancienne, ne font que depuis peu l'objet de *preuves de sécurité* (au sens de la théorie de l'information). En outre, ces résultats ne permettent que rarement de se prononcer sur les chiffrements existants. La seule preuve formelle de sécurité a été énoncée pour le *masque jetable*. Longtemps la seule constatation que le chiffrement n'avait pas été cassé a constitué la seule démonstration empirique de sa sécurité. À l'heure actuelle, la conception de systèmes à clé secrète sûrs repose essentiellement sur une démarche heuristique. Les attaques apparues durant les 20 dernières années ont permis de formaliser un certain nombre de critères qu'un chiffrement doit vérifier afin d'être réputé sûr. Malheureusement, si on sait démontrer qu'un algorithme résiste aux attaques *classiques*, on ne peut garantir sa résistance contre de nouveaux types d'attaques. Ainsi et assez paradoxalement, le niveau de sécurité supposé des systèmes de chiffrement repose pour une part non-négligeable sur la foi qu'on leur accorde et la réputation d'invulnérabilité que les chercheurs et les informaticiens leur reconnaissent.

1.3.1 Un système inconditionnellement sûr : le masque jetable

La théorie de l'information permet de démontrer qu'il existe un système de chiffrement inconditionnellement sûr : le masque jetable ou encore *one time pad*. On peut classer ce chiffrement dans la catégorie des chiffrements à flot, le message en clair est additionné par un OU-EXCLUSIF bit à bit à une suite aléatoire de même taille que lui, qui constitue la clé secrète.

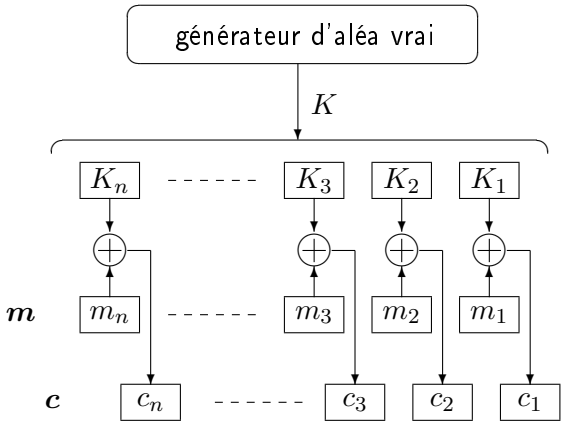


FIG. 1.2 – *Principe du masque jetable*

Le caractère purement aléatoire de la suite chiffrante garantit que toutes les caractéristiques du texte clair sont noyées par la clé. L'entropie du chiffré étant supérieure à celle du texte clair, on ne peut retrouver aucune information par la connaissance du chiffré seul. Ce système séduisant par sa sécurité inconditionnelle présente néanmoins des désavantages majeurs :

– la clé doit être au moins aussi longue que le message en clair. Ainsi le chiffrement destiné à sécuriser la communication d'un message nécessite l'échange sécurisé d'un autre message

au moins aussi long que lui... Il est donc difficile dans ce cas-là de respecter la condition de Kerckhoffs : *la clé doit pouvoir en être communiquée et retenue sans le secours de notes écrites et être changée et modifiée au gré des correspondants* ;
- la sécurité n'est garantie que si la clé n'est utilisée qu'une seule fois, d'où le nom de *masque jetable*. En effet, si on chiffre deux messages m_1 et m_2 avec la même clé K et que les chiffrés correspondants c_1 et c_2 sont interceptés, l'attaquant est alors en mesure de 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.

Ces deux inconvénients sont inhérents à la propriété de sécurité inconditionnelle comme l'a démontré Shannon [Sha49]. Ce système n'est ainsi utilisé que pour des applications requérant un niveau de sécurité justifiant de tels inconvénients, c'est-à-dire rarement. Nous nous intéressons ainsi à des systèmes utilisant des clés de taille raisonnable (actuellement autour de 128 bits).

1.3.2 Classification des attaques

Étant donné un système de chiffrement symétrique paramétré par une clé K, seul paramètre secret, la classification des types d'attaques dépend du matériel additionnel à la disposition du cryptanalyste :

- *attaque à chiffré seul* : le cryptanalyste ne dispose que de matériel chiffré intercepté passivement ;
- *attaque à clair connu* : le cryptanalyste dispose à la fois du matériel chiffré et des messages en clair correspondants. Cette configuration n'est pas si exceptionnelle qu'on pourrait le croire car il est rare qu'un cryptogramme ne laisse pas filtrer grâce à des indices extérieurs, taille du message, heure de transmission, événements précédents et suivants au moins quelques indices sur le contenu du message en clair ;
- *attaque à clair choisi* : le cryptanalyste dispose soit de la boîte noire que constitue E_K sans moyen d'extraire K, soit d'un moyen de faire chiffrer à l'insu de l'émetteur des messages dont il peut observer le transit sur le canal de communication qu'il espionne. Il se trouve ainsi en possession de couples clairs-chiffrés dont il a pu choisir les propriétés ;
- *attaque à clair choisi adaptative* : elle correspond au cas précédent auquel on ajoute l'avantage pour le cryptanalyste de pouvoir faire évoluer les propriétés des clairs soumis au chiffrement selon les résultats obtenus sur les chiffrés observés ;
- *attaque à chiffré choisi* (adaptative ou non) : le cryptanalyste est en possession d'un certain nombre de couples clairs-chiffrés correspondant à des chiffrés de son choix (situation duale de l'attaque à clair choisi) ;
- *attaque à clés liées* : le cryptanalyste a accès à des chiffrés obtenus grâce à un ensemble de clés inconnues mais possédant des relations connues entre elles.

Lorsque le cryptanalyste parvient à retrouver soit les messages en clair soit la clé de déchiffrement de manière systématique et effective (sans soudoyer quelqu'un qui possède ses informations) on dit que le système est cassé.

Enfin, les attaques se distinguent selon les aspects du système sur lequel elles portent. On peut attaquer le système en le considérant comme une boîte noire. Cette approche correspond au degré le plus faible de connaissances sur le système. Elle lui reste entièrement extérieure en ne considérant que ses sorties et éventuellement ses entrées. On classe la recherche exhaustive de clé dans cette catégorie. À l'inverse, l'étude détaillée des propriétés internes d'un chiffrement

permet de produire des attaques qui exploitent tous les aspects structurels de l'algorithme. Les cryptanalyses les plus connues et dont sont déduits des critères de conception appartiennent à cette catégorie. Nous nous intéressons à ce type d'attaques dans ce document. La classe des *attaques par canaux secondaires* s'intéresse aux implémentations des algorithmes. En effet, que ce soit sous forme logicielle ou matérielle, une implémentation d'un algorithme a des répercussions physiques mesurables telles que le temps d'exécution, la température d'un circuit ou son rayonnement magnétique. Si de telles grandeurs sont dépendantes de la clé, il est alors possible de retirer de l'information de leur mesure. De la même manière on peut aussi interagir avec le système par *injection de fautes* et déduire des informations de son comportement en réaction.

1.3.3 Recherche exhaustive de la clé

Un paramètre essentiel pour la sécurité d'un système à clé secrète est la taille de l'espace des clés, c'est-à-dire le nombre de clés possibles. En effet, il est toujours possible de mener sur un algorithme de chiffrement une attaque dite *exhaustive* pour retrouver la clé. L'attaque consiste simplement à énumérer l'ensemble des clés possibles et à les essayer successivement pour déchiffrer un message. Il suffit ensuite de détecter le message en clair correspondant parmi les résultats du déchiffrement ce qui nécessite quelques couples clairs-chiffrés ou des informations statistiques sur les clairs d'origine (format de codage des caractères, langue d'origine du document...). Si on considère la clé d'un système de chiffrement symétrique comme un vecteur de k bits, le nombre moyen de clés à fournir à la fonction de déchiffrement pour mener à bien cette attaque est alors 2^{k-1}. La faisabilité d'une telle attaque dépend évidemment de l'évolution de la technologie. On rappelle en effet que la *loi de Moore* est une constatation empirique qui affirme que la puissance de calcul des processeurs double tous les 18 mois. On considère actuellement que l'espace des clés est suffisamment grand si la clé comporte au minimum 80 bits, c'est pourquoi les algorithmes actuels proposent en général au minimum des clés de 128 bits. Ainsi, l'algorithme de chiffrement à clé secrète le plus utilisé jusqu'à récemment, le DES (Data Encryption Standard), est désormais vulnérable à une attaque exhaustive puisqu'il utilise une clé secrète de 56 bits. Une telle attaque, demandant en moyenne 2^{55} chiffrements DES, a été 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[1]. Le coût d'une telle machine était alors estimé à $ 210 000. C'est pourquoi le DES a été remplacé par un nouveau standard de chiffrement à clé secrète, l'AES (Advanced Encryption Standard)[FIP01]. L'AES 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 chercheurs belges, V. Rijmen et J. Daemen [DR99]. Il opère sur des blocs de message de 128 bits et est disponible pour trois tailles de clé différentes, 128, 192 et 256 bits, ce qui le met à l'abri des attaques exhaustives.

1.3.4 Complexité des attaques

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

1. http://www.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/

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 données mesure la quantité de données (nombre de couples clairs-chiffrés par exemple ou nombre de bits de suite chiffrante à observer) nécessaire à la réalisation d'une attaque ;
- la complexité en mémoire mesure la quantité de mémoire nécessaire au cours de l'attaque ;
- la complexité en temps mesure le nombre d'unités de temps nécessaires pour mener à bien l'attaque. Dans la plupart des cas, l'unité de temps correspond à une opération de chiffrement ou à un cycle d'horloge.

La *complexité d'une attaque* correspond à la partie dominante des complexités ci-dessus. On considère qu'un chiffrement à clé secrète présente une bonne sécurité s'il n'existe pas d'attaque dont la complexité soit significativement inférieure à la recherche exhaustive sur la clé ou la recherche exhaustive sur les textes. Une attaque générique cherchant à trouver un compromis entre ces diverses complexités est l'*attaque par compromis temps-mémoire-données*, originellement *compromis temps-mémoire* de Hellman (time-memory tradeoff) [Hel80]. Cette technique peut se décliner en plusieurs variantes suivant le type de système attaqué (cf. [HS05] pour un état de l'art récent sur le sujet).

Tout nouveau chiffrement qui veut être considéré comme une proposition sérieuse s'accompagne ainsi de garanties sur le fait que les attaques connues qui lui sont appliquées présentent une complexité proche de celle de la recherche exhaustive. Aussi, il est de plus en plus difficile de présenter une attaque effective sur des versions complètes des algorithmes de chiffrement actuellement en usage. Ainsi un autre moyen de mesurer la marge de sécurité qu'offre une système de chiffrement consiste à évaluer la complexité des attaques connues sur des versions simplifiées (réduction du nombre de tours, cadencement régulier...).

Nous allons maintenant présenter les deux grandes familles de chiffrements symétriques. Nous nous intéressons d'abord aux chiffrements à flot et aux familles d'attaques qui s'appliquent à ces systèmes et dont sont issus les critères cryptographiques que nous étudions pour les fonctions booléennes objets des derniers chapitres de ce document. Nous présentons ensuite le chiffrement par blocs ainsi que rapidement les attaques dont il fait l'objet. Nous réservons au chapitre 2 la description détaillée de la modélisation des attaques sur le dernier tour ainsi que les critères déduits des attaques différentielle et linéaire.

1.4 Le chiffrement à flot

Le masque jetable se révélant être un chiffrement bien trop contraignant, on cherche à conserver le même type de schéma avantageux dans certains contextes en remplaçant le générateur d'aléa vrai par un générateur pseudo-aléatoire, c'est-à-dire un système générant des suites présentant des caractéristiques statistiques d'aléa satisfaisant les tests communément en cours à partir d'une *graine* de petite taille. Ce générateur est alors paramétré par une clé secrète de taille plus faible que le message à chiffrer. La sécurité n'est plus inconditionnelle puisque l'entropie de la clé est alors plus faible que celle du message à chiffrer, mais en imposant au générateur pseudo-aléatoire de vérifier des propriétés cryptographiques, on espère néanmoins produire une suite chiffrante issue d'un système calculatoirement sûr.

Le principe d'un chiffrement à flot repose sur la combinaison — généralement l'addition bit-à-bit, dans ce cas le chiffrement est dit *additif* — du message clair avec une suite chiffrante produite par un générateur paramétré

- soit uniquement par la clé secrète K, on parle alors de *chiffrement à flot synchrone* (je ne m'intéresserai ici qu'à ce type de systèmes) ;
- soit par la clé secrète K et un nombre donné de bits du clair (ou de manière équivalente du chiffré), on parle alors de *chiffrement à flot auto-synchronisant*.

Dans un système de chiffrement à flot, la fonction de chiffrement varie au fur et à mesure de son application au message en clair ; c'est la raison pour laquelle ce type de système est dit avoir de la mémoire et est parfois appelé *chiffrement à état*. Au contraire un chiffrement par blocs est dit *sans mémoire* car le résultat du chiffrement d'un bloc est déterminé uniquement par la clé et le bloc de clair, indépendamment du temps. Par ailleurs, le chiffrement à flot se distingue en général du chiffrement par blocs par le fait que la taille du bloc logique de traitement du message en clair peut être très petite (1 bit dans les chiffrements additifs). Le chiffrement à flot est particulièrement adapté aux deux types d'applications suivantes :

- les applications logicielles qui requièrent un débit de chiffrement et de déchiffrement très élevé. En effet, les ordres de grandeur en nombre de cycles par octet sont comparativement de 5 pour un chiffrement à flot orienté logiciel (ex. Py [BS05] : 2,85 sur un Pentium3, SNOW2.0 [EJ02] : 4,2 sur un Pentium4, SOSEMANUK [BBC+05b] : 3,15 sur un G4) contre 20 pour un chiffrement par blocs (ex. AES : 14 sur un Atlon) ;
- les applications matérielles fortement contraintes, notamment en termes de surface de circuit et de consommation (cas des systèmes embarqués). La complexité d'implémentation des chiffrements à flot dédiés au matériel est en effet plus faible que celle des chiffrements par blocs.

Dans ces deux types d'environnements on privilégie donc des chiffrements à flot dédiés plutôt que des chiffrements par blocs (même utilisés dans un mode opératoire qui les transforme en chiffrement à flot comme les modes OFB ou CTR).

1.4.1 Principe général d'un chiffrement à flot synchrone

Dans un chiffrement à flot synchrone, la suite chiffrante est engendré à partir de la clé secrète K par un générateur pseudo-aléatoire. Le destinataire du message partageant cette clé, il peut ainsi produire la même suite chiffrante et retrouver le message en clair en la combinant au message chiffré. Le traitement du message par unités de faible taille permet de limiter la propagation des erreurs.

Un générateur pseudo-aléatoire est un automate à états finis qui génère une suite en produisant à chaque instant un ou plusieurs bits calculés à partir de son état interne. On modélise en général un tel générateur cryptographique à partir de trois fonctions principales décrites à la figure 1.3 :

- une procédure d'initialisation (notée init sur le schéma) qui détermine l'état initial du générateur à partir de la clé K (et éventuellement d'un vecteur d'initialisation connu noté IV) ;
- une fonction de transition (notée Φ sur le schéma) qui fait évoluer l'état interne entre les instants t et $t+1$. Cette fonction peut dépendre de la clé, de l'IV et du temps, mais elle est en général fixe dans tous les chiffrements dédiés à des environnements matériels pour d'évidentes raisons de simplicité et d'encombrement ;

– une fonction de filtrage (notée f sur le schéma) qui, à partir de l'état interne à l'instant t, produit un ou plusieurs bits de la suite chiffrante z. Tout comme la fonction de transition la fonction de filtrage peut dépendre de la clé, de l'IV et du temps, mais, pour les mêmes raisons, elle est en général fixe dans tous les chiffrements dédiés à des environnements matériels.

La fonction h qui sert à combiner la sortie de la fonction filtrante et le message doit être inversible pour permettre le déchiffrement. Pour simplifier la présentation nous nous limiterons dans ce qui suit au cas où la fonction filtrante ne produit qu'un bit et où la fonction h est l'addition binaire.

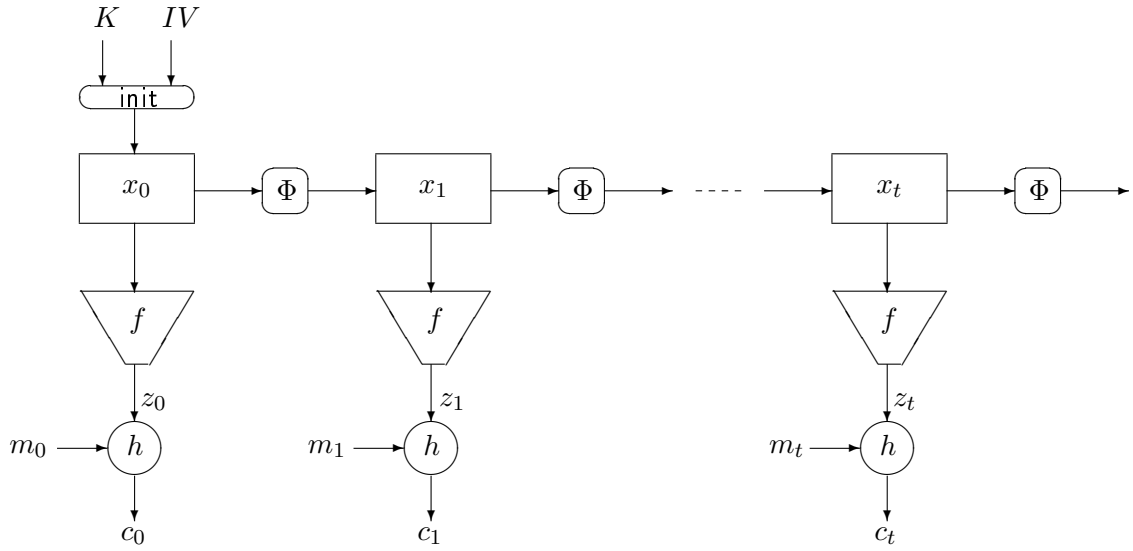


FIG. 1.3 – *Principe général d'un chiffrement à flot synchrone*

La sécurité d'un système de chiffrement à flot repose sur les caractéristiques du générateur de la suite chiffrante. À partir de l'état initial paramétré par la clé secrète et éventuellement un vecteur d'initialisation, il doit produire une suite binaire ayant de bonnes propriétés statistiques (caractère pseudo-aléatoire) et que l'on ne sait pas relier de manière simple à la clé secrète (propriétés cryptographiques). On demande aux suites ainsi produites de ne pouvoir être distinguées d'une suite réellement aléatoire. On distingue deux types d'attaques sur ces systèmes :

– les attaques par recouvrement de la clé qui visent à retrouver la clé secrète K ou de manière équivalente l'état initial ou un état interne complet du générateur ;
– les attaques par distingueur où il s'agit alors de déterminer si une suite arbitraire de bits de suite chiffrante provient d'un générateur pseudo-aléatoire donné ou s'il s'agit d'aléa véritable. En effet, si k est la taille de la clé et que le générateur produit des suites de taille n, il sélectionne 2^k suites binaires parmi les 2^n possibles. Idéalement, il doit être impossible de distinguer l'ensemble des 2^k suites ainsi définies de 2^k suites binaires de taille n tirées aléatoirement parmi les 2^n possibles.

Les attaques par distingueur sont bien entendu plus faibles que celles visant à recouvrer la clé secrète. En effet, elles ne permettent que d'obtenir des informations partielles sur le message

(par exemple vérifier si un chiffré intercepté correspond à un clair donné) alors que retrouver la clé secrète permet de déchiffrer tous les messages interceptés. Les cryptanalyses consistant, à partir de la connaissance de N bits de suite chiffrante, à prédire la valeur du bit suivant (*next bit prediction attack*) pourraient constituer une catégorie intermédiaire d'attaques, mais on peut démontrer que leur existence est en fait équivalente à celle d'une attaque par distingueur [BM84].

1.4.2 Les familles de chiffrement à flot

Pour qu'un générateur pseudo-aléatoire puisse être utilisé dans un système cryptographique, il doit être impossible en pratique de retrouver la clé secrète à partir de la suite chiffrante produite, ce qui nous permet de déduire immédiatement quelques propriétés génériques que doivent vérifier les chiffrements construits selon le principe précédent :

- la taille de l'état interne doit être suffisamment grande afin de contrer d'une part bien sûr la recherche exhaustive, mais également les attaques par compromis temps-mémoire. Ces attaques permettent en effet de retrouver la valeur de l'état interne pour une complexité en racine carrée du nombre d'états, ce qui impose que la taille de l'état interne soit au moins deux fois celle de la clé ;
- la fonction de transition doit garantir une période élevée pour la suite chiffrante, ce qui revient à dire que pour tout état initial x_0 la suite $x_t = \Phi^t(x_0)$ ne doit pas avoir de cycles courts ;
- la fonction de filtrage ne doit pas perturber les bonnes propriétés statistiques des états en entrée. Ainsi il est nécessaire pour que la suite binaire produite ne soit pas biaisée que la distribution en sortie de la fonction de filtrage f soit uniforme. Lorsque f est une fonction booléenne scalaire qui vérifie cette propriété, on dit qu'elle est *équilibrée*. Dans le cas contraire, il existe une attaque par distingueur nécessitant la connaissance de $\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$ bits de la suite chiffrante avec $\varepsilon = \left|\mathbf{P}[f(X) = 1] - \frac{1}{2}\right|$;
- la fonction de filtrage doit s'interposer entre l'état interne et la suite produite de manière à ne fournir aucune information sur l'état interne.

La classification des différents types de chiffrements à flot est une tâche délicate dans la mesure où les contraintes liées à l'environnement de destination du chiffrement jouent un rôle déterminant dans sa conception et que certains détails destinés à améliorer le caractère aléatoire de la suite produite permettent souvent de classer les chiffrements dans plusieurs catégories (ou dans aucune...). Nous allons tenter néanmoins de dresser un inventaire volontairement imprécis de cette population. Le critère de la classification peut porter sur divers éléments, on peut par exemple souligner l'orientation logicielle ou matérielle du chiffrement, porter son attention sur la nature ou la taille de l'état interne, sur le type de fonction de transition (à laquelle va être lié le type de fonction de filtrage). Nous distinguons, selon le type de fonction de transition utilisée, deux grandes familles de chiffrements à flot :

- celle dont la fonction de transition est linéaire. On compte notamment dans cette catégorie les incontournables générateurs à base de *registres à décalage à rétroaction linéaire* ;
- celle dont la fonction de transition est non-linéaire, famille plus hétéroclite que nous allons détailler.

Chiffrements à base de registre à décalage à rétroaction linéaire Le registre à décalage à rétroaction linéaire (LFSR pour *Linear Feedback Shift Register*) est un composant

élémentaire bien adapté pour des 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 LFSR est certainement 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 que l'on sait relier de manière linéaire à son initialisation. La suite produite aisément à partir de ce composant ne peut pas être utilisée telle quelle dans un but cryptographique. En améliorer les propriétés consiste donc essentiellement à introduire de la non-linéarité dans le but de rendre impossible une recherche aisée de la valeur d'initialisation à partir de l'observation de la suite chiffrante et ce sans altérer les bonnes propriétés statistiques de la suite produite par un LFSR. Cette tâche est dévolue à une fonction booléenne utilisée soit pour combiner plusieurs registres soit pour filtrer l'état interne d'un unique registre.

Chiffrements à base d'états à évolution non-linéaire Afin de parer à la linéarité des registres à rétroaction linéaire, on peut chercher à créer des états dont la transition est non-linéaire. Les solutions retenues, tout comme leur complexité (dont dépend évidemment le choix) présentent des différences nettes selon si le chiffrement est destiné à une implémentation logicielle ou matérielle.

Un procédé d'orientation logicielle uniquement consiste à considérer des chiffrements possédant un état interne de grande taille qui évolue de manière permanente et non-linéaire. L'exemple type est celui de RC4 qui repose sur un grand tableau dont les valeurs sont modifiées à chaque itération de l'algorithme. On peut citer également le chiffrement Py (à prononcer *roo*) proposé par E. Biham et J. Seberry. Ces chiffrements ne permettent en général pas de calculer de manière formelle la période de la suite chiffrante, mais cet inconvénient est compensé par la grande taille de l'état interne (10400 bits pour Py, par exemple) qui rend les cycles courts improbables.

Lorsqu'on prend en compte les contraintes liées à un environnement matériel, qui plus est souvent fortement contraint en surface ou en consommation, il est impératif de revenir à des tailles d'états internes raisonnables, ce qui implique qu'en contrepartie il faut pouvoir évaluer une borne inférieure sur la longueur des cycles de la suite produite. Cette catégorie inclut les systèmes à base de registres à rétroaction non-linéaire (NLFSR) (voir par exemple la proposition Achterbahn [GGK05] et ceux à base de registres à décalage à rétroaction à retenue (FCSR) [AB05]. Une autre catégorie de fonctions de transition non-linéaires introduites récemment est la classe des fonctions-T [KS02]. Il s'agit de fonctions d'implémentation aisée dans un environnement logiciel pour lesquelles on sait calculer la période de la suite produite. Les caractéristiques de ces systèmes ont fait l'objet d'études moins nombreuses que celles des systèmes qui reposent sur des LFSR, mais ils pourraient être amenés à se développer au regard des nouvelles attaques publiées contre ces derniers, même si certaines faiblesses ont été mises en évidence récemment par Molland et Helleseth [MH05].

Les conceptions hybrides Comme écrit précédemment, une classification générale peut difficilement rendre compte des détails de conception d'un chiffrement donné qui cherche à élaborer le meilleur compromis possible entre toutes les contraintes d'origines diverses auxquelles il doit apporter une réponse. Il existe ainsi des systèmes dont l'état interne est divisé en deux parties, l'une comportant une fonction de transition linéaire et l'autre une fonction de transition non-linéaire (par exemple Grain [HJM05]). Toutefois, lorsque la fonction de tran-

sition non-linéaire opère sur un état de petite taille, on assimile le générateur à un système à transition linéaire, opérant sur la première partie de l'état interne, muni d'une mémoire, qui correspond à la deuxième partie de l'état interne. Des systèmes tels que SNOW2.0 ou SOSEMANUK sont conçus sur ce modèle.

Les systèmes présentés précédemment utilisent en général un cadencement régulier des registres, régularité dont le cryptanalyste peut tirer avantage. Aussi peut-on chercher à introduire un facteur augmentant l'imprévisibilité du comportement du système en utilisant un cadencement irrégulier. Cela signifie que la décision de l'instant de transition de l'état interne ou du nombre de transitions à opérer est prise irrégulièrement par un autre composant du système. On parle alors de systèmes à *horloge contrôlée*. Des exemples de tels systèmes sont LILI-128 [DCG+00], A5/1 [BGW99], le *Shrinking generator* [CKM94] ou DECIM [BBC+05a]. Cependant la mise en place de cette solution dans le cadre d'applications matérielles présente l'inconvénient majeur de nécessiter une mémoire tampon garantissant un débit de chiffrement constant, ce qui accroît la complexité de l'implémentation.

1.4.3 Chiffrements à flot conçus à partir de LFSR

Nous présentons en exemple cette famille de chiffrements à flot. En effet, de nombreux résultats et outils mathématiques sont disponibles pour la conception et l'analyse de tels systèmes et ils permettent d'illustrer les propriétés cryptographiques requises des fonctions booléennes que nous étudions dans la deuxième partie de cette thèse.

De manière générale, les attaques contre les systèmes utilisant le registre à décalage à rétroaction linéaire comme composant élémentaire cherchent à tirer partie de la structure algébrique sous-jacente du système.

Le 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. La figure 1.4 illustre le principe d'un tel composant. à chaque cycle d'horloge, les L bits du registre

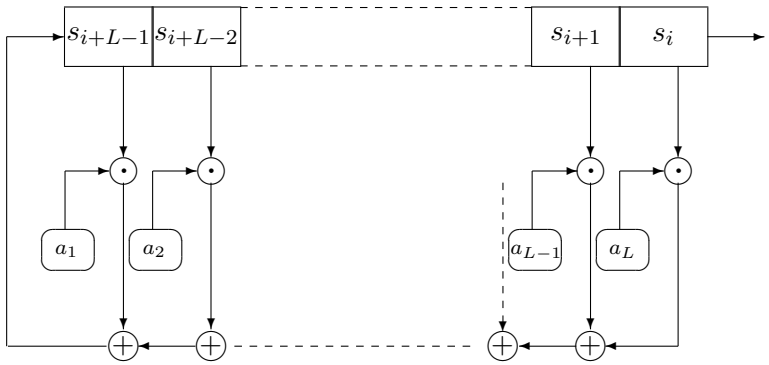


FIG. 1.4 – *Principe d'un registre à décalage à rétroaction linéaire (LFSR)*

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 :

$$s_{i+L} = a_1 s_{i+L-1} \oplus a_2 s_{i+L-2} \oplus \cdots \oplus a_L s_i.$$

La suite s produite par une telle récurrence linéaire est *ultimement périodique*, c'est-à-dire qu'il existe une pré-période n_0 telle que la suite $(s_n)_{n\geq n_0}$ est périodique — on a $n_0 = 0$ si $a_L = 1$. À toute suite récurrente linéaire on associe un polynôme f_s que l'on appelle polynôme de rétroaction :

$$f_s \in \mathbf{F}_2[X], \quad f_s(X) = 1 + a_1X + a_2X^2 + \cdots + a_LX^L.$$

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 $\Lambda(s)$. Lorsque le polynôme de rétroaction minimal est primitif et que l'état initial $(s_0,s_1,\ldots,s_{L-1})$ est non-nul, la période de la suite s est maximale et égale à $2^{\Lambda(s)} - 1$. Les suites périodiques engendrées par des polynômes de rétroaction primitifs sont appelées m-séquences ou suites ML (de longueur maximale) .

Il est évident qu'un LFSR seul produit une suite aux piètres propriétés cryptographiques. En effet, si nous mettons de côté le fait que les L premiers bits de la suite sont l'initialisation (qu'il est toujours possible de ne pas utiliser par exemple), dans le cadre d'une attaque à clair connu, la connaissance du polynôme de rétroaction (qui est en général fixé pour une implémentation matérielle) permet de produire, à partir de l'observation de L bits consécutifs de la suite s, tous les bits ultérieurs. Le cas où le polynôme de rétroaction est inconnu (déterminé à partir de la clé par exemple) ne garantit pas une meilleure sécurité puisque la linéarité de la récurrence permet de retrouver les L coefficients du polynôme de rétroaction grâce à la résolution d'un système linéaire obtenu par l'observation de $2L$ éléments de la suite. Il existe même une manière plus efficace de 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 [Ber68, Mas69] détermine le LFSR équivalent, c'est-à-dire le polynôme de rétroaction minimal d'une suite de complexité linéaire $\Lambda(s)$ grâce à l'observation de $2\Lambda(s)$ éléments consécutifs de la suite et ce sans connaissance préalable de $\Lambda(s)$. 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, pour se mettre à l'abri d'une attaque par l'algorithme de Berlekamp-Massey) et de grande période en utilisant des LFSR 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 n LFSR combinés par une fonction booléenne à n variables ou un seul LFSR dont on choisit n bascules avec des espacements soigneusement déterminés, que l'on filtre par une fonction booléenne à n variables. 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 [Rue86]. Les deux systèmes sont représentés par les schémas 1.5 et 1.6.

Si on note z_t le bit de suite chiffrante à l'instant t, on a alors :

$$z_t = f(x_t^1,x_t^2,\ldots,x_t^n)$$

où $x_t^1,x_t^2,\ldots,x_t^n$ correspondent soit aux sorties de n LFSR 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 (SFINKS [BLM+05] est une proposition récente de système de ce type ; son état interne est constitué d'un LFSR de 256 bits dont 17 bits sont les entrées de la fonction de filtrage).

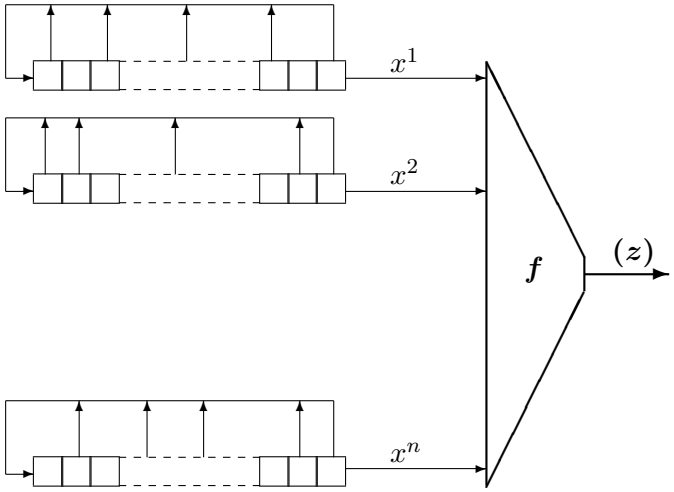


FIG. 1.5 – *LFSR combinés*

1.4.4 Attaques sur les chiffrements à flot

Nous nous intéressons ici aux attaques permettant de définir des critères de sécurité pour la fonction de filtrage. Nous ne ferons donc pas mention d'autres attaques telles que celles du type *guess-and-determine* ou celles portant sur le chargement de la clé ou sur le changement de l'IV. Pour simplifier, nous nous limiterons en règle générale au cas où la fonction de filtrage ne produit qu'un bit, *i.e.* $f : \mathbf{F}_2^n \to \mathbf{F}_2$ est une fonction booléenne scalaire.

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, soit de données statistiques, c'est le cas des attaques par distingueur et par corrélation. Dans le cas de systèmes à base de LFSR, cette dernière classe d'attaques modélise la sortie du générateur en un mot de code bruité par un canal binaire symétrique et s'attache donc à le décoder. Les attaques décrites ne peuvent bien sûr plus que rarement s'appliquer telles qu'elles sur les systèmes existant qui ont intégrés les critères qui en ont été déduits dans leur conception. Elles restent néanmoins de bons moyens d'évaluer la marge de sécurité des algorithmes lorsqu'elles sont appliquées sur des versions simplifiées ou une base pour élaborer des attaques dédiées.

Attaques par distingueur

Le premier biais statistique à prendre en compte est bien sûr la distribution des sorties de la fonction de filtrage. Nous avons déjà mentionné que la sortie de cette fonction doit être uniformément distribuée (équilibrée) afin d'éviter une attaque par distingueur. Il existe par ailleurs d'autres types de biais statistiques exploitables.

Lorsque les bits d'entrées de la fonction de filtrage correspondent à des bits fixés de l'état interne, on peut chercher à déterminer quand une configuration de bits donnée se reproduit sur d'autres entrées. Plus formellement, si on note $x_t = (x_t^1, x_t^2, \ldots, x_t^n)$ l'entrée de la fonction de filtrage à l'instant t, on cherche à déterminer un décalage τ tel que, pour tout t, les ensembles

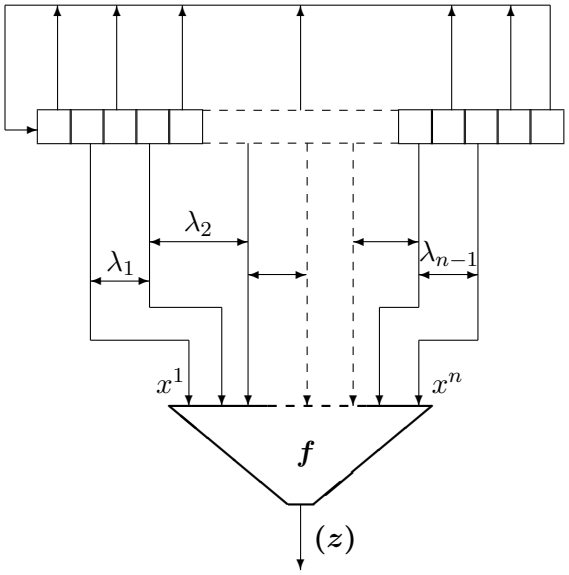


FIG. 1.6 – *LFSR filtré*

des entrées aux instants t et $t+\tau$ aient une intersection non-nulle :

$$\exists\tau\ \forall t, \quad \{x_t^1,x_t^2,\ldots,x_t^n\} \cap \{x_{t+\tau}^1,x_{t+\tau}^2,\ldots,x_{t+\tau}^n\} \neq \emptyset$$

Pour illustrer la situation, on peut se représenter un système composé d'un registre à décalage (à rétroaction linéaire ou non) dont un certain nombre de bascules fixées fournissent les entrées de la fonction de filtrage (situation du registre filtré dédié à un environnement matériel). Dans ce cas, la valeur d'une bascule en entrée de f intervient systématiquement dans une bascule située en aval également en entrée de f. Par exemple, en reprenant les notations de la figure 1.6, la $i$ème entrée de la fonction f à l'instant t correspond à la $(i+1)$ème entrée à l'instant $t+\lambda_i$.

On peut alors modéliser l'attaque de la manière suivante lorsque les entrées de f aux instants t et $t+\tau$ possèdent ℓ valeurs communes. Considérons des vecteurs de variables aléatoires mutuellement indépendantes à valeurs dans $\{0,1\}$ uniformément distribuées, $X = (X_1,X_2,\ldots,X_\ell)$ d'une part, $Y = (Y_1,Y_2,\ldots,Y_{n-\ell})$ et $Z = (Z_1,Z_2,\ldots,Z_{n-\ell})$ d'autre part. Pour toute permutation σ de l'ensemble S_n des permutations de $\{1,\ldots,n\}$ et pour tout vecteur $v = (v_1,v_2,\ldots,v_n)$, on note $\sigma(v)$ le vecteur $(v_{\sigma(1)},v_{\sigma(2)},\ldots,v_{\sigma(n)})$. Nous nous intéressons aux vecteurs à n composantes $(X\|Y)$ et $\sigma(X\|Z)$, où $\|$ représente la concaténation de deux vecteurs.

Pour tout $\sigma \in \mathsf{S}_n$, on a alors :

$$\begin{aligned}\mathbf{P}[z_t = z_{t+\tau}] &= \mathbf{P}_{X,Y,Z}[f(X\|Y) = f(\sigma(X\|Z))] \\ &= \frac{1}{2^\ell} \sum_{x\in\mathbf{F}_2^\ell} \sum_{\varepsilon\in\mathbf{F}_2} \mathbf{P}_Y[f(x\|Y)=\varepsilon] \cdot \mathbf{P}_Z[f(\sigma(x\|Z)))=\varepsilon]\end{aligned}$$

Dans l'hypothèse défavorable pour l'attaque, cette probabilité doit valoir $\frac{1}{2}$. Pour que cette condition soit remplie, il faut que la sortie de la fonction de filtrage reste équilibrée quand on fixe un sous-ensemble quelconque de taille ℓ de ses entrées. La fonction est alors dite

ℓ-résiliente. Dans ce cas on a :

$$\mathbf{P}[z_t = z_{t+\tau}] = \frac{1}{2^\ell} \sum_{x \in \mathbf{F}_2^\ell} \sum_{\varepsilon \in \mathbf{F}_2} \left(\frac{1}{2}\right)^2 = \frac{1}{2}$$

Dans le cas des registres à décalage, en général, les espacements entre deux positions consécutives des bascules en entrée de la fonction de filtrage sont deux à deux premiers entre eux [Gol96]. Cette règle garantit que l'intersection entre deux ensembles d'entrées de f est au plus de taille 1. Ainsi, pour se prémunir d'une éventuelle attaque par distingueur sur un tel système, la fonction de filtrage doit être au moins 1-résiliente.

Attaques par corrélation

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. Dans le cas du chiffrement à flot, elles ont été originellement introduites contre les systèmes par combinaison par Siegenthaler [Sie84, Sie85]. Cette cryptanalyse est en fait valable dès que l'état interne du générateur est décomposable en plusieurs parties — par exemple dès que la fonction de transition agit sur des sous-ensembles de l'état interne. On peut alors rechercher la valeur d'une partie indépendamment des autres. L'attaque repose sur l'existence d'éventuelles corrélations entre la sortie de la fonction de filtrage et la partie incriminée de l'état interne ce qui revient à considérer les corrélations entre la sortie de f et un sous-ensemble de ses entrées.

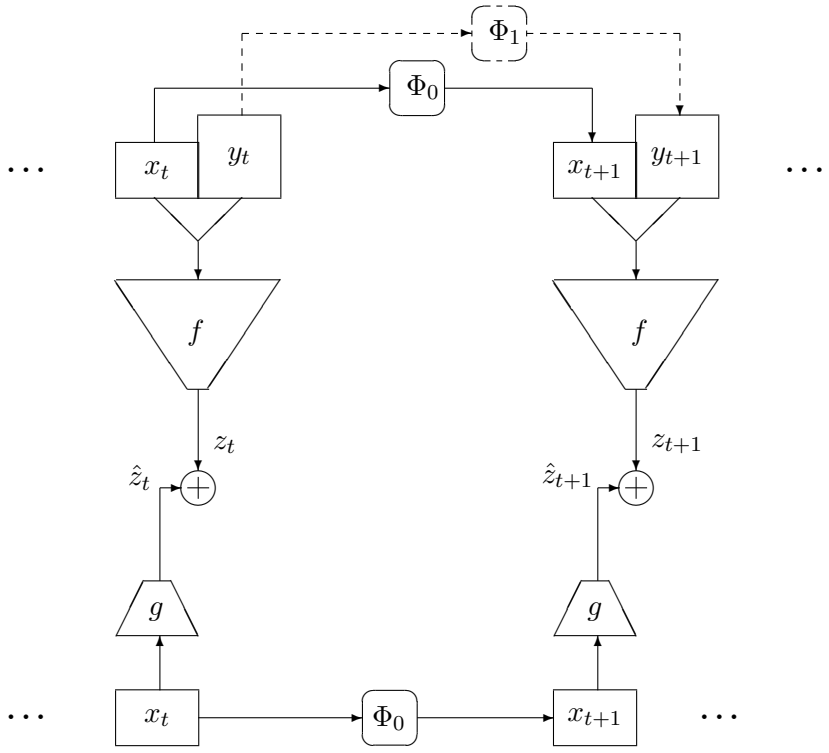


FIG. 1.7 – *Exemple d'attaque par corrélation*

Pour décrire plus précisément le principe de l'attaque, supposons, comme sur la figure 1.7, que l'on peut séparer l'état interne à l'instant t en deux parties x_t et y_t mises à jour indépendamment par Φ_0 et Φ_1 respectivement. On peut ainsi exprimer la fonction de filtrage f sous la forme

$$\begin{array}{rccc} f: & \mathbf{F}_2^\ell \times \mathbf{F}_2^{n-\ell} & \to & \mathbf{F}_2 \\ & (x,y) & \mapsto & f(x,y)\ . \end{array}$$

S'il existe une fonction $g : \mathbf{F}_2^\ell \to \mathbf{F}_2$ telle que pour X, Y deux variables aléatoires indépendantes uniformément distribuées on ait :

$$p_g = \mathbf{P}_{X,Y}[f(X,Y) = g(X)] > \frac{1}{2},$$

on peut alors mener une attaque par corrélation qui effectue une recherche exhaustive sur les ℓ premiers bits de l'état initial.

Algorithme 1.1 *Pour tout $x_0 \in \mathbf{F}_2^\ell$:*

1. *Calculer les N premiers bits de la suite $\hat{z}$, $\hat{z}_t = g \circ \Phi^t(x_0)$;*
2. *Calculer la corrélation entre les suites z et $\hat{z}$ sur N bits :*

$$\mathsf{c}(z,\hat{z}) = \sum_{t=0}^{N-1} (-1)^{z_t \oplus \hat{z}_t};$$

Si $\mathsf{c}(z,\hat{z})$ dépasse un certain seuil, alors x_0 est un candidat pour l'état initial.

L'attaque repose sur le fait que lorsque la valeur de x_0 essayée n'est pas celle de l'initialisation correcte, alors les suites z et $\hat{z}$ ne sont pas corrélées et $\mathsf{c}(z,\hat{z}) \simeq 0$. En revanche, lorsque x_0 correspond à l'initialisation correcte, on a

$$\mathsf{c}(z,\hat{z}) = 2N\left(p_g - \frac{1}{2}\right).$$

Dans ce cas, distinguer entre une distribution uniforme et la distribution de la variable aléatoire $f(X,Y) + g(X)$ nécessite la connaissance d'un nombre de bits de suite chiffrante de l'ordre de

$$N = \mathcal{O}\left(\frac{1}{\left(p_g - \frac{1}{2}\right)^2}\right).$$

La complexité en temps pour retrouver les ℓ bits de l'état initial x_0 par recherche exhaustive est alors de l'ordre de

$$N2^\ell = \mathcal{O}\left(\frac{2^\ell}{\left(p_g - \frac{1}{2}\right)^2}\right).$$

L'attaque par corrélation soulève naturellement deux questions complémentaires pour l'attaquant et le concepteur du système :

- Comment choisir la fonction g qui maximise le biais $p_g - \frac{1}{2}$?
- Comment choisir la fonction de filtrage f pour résister à cette attaque, c'est-à-dire pour que $p_g = \frac{1}{2}$ pour toute fonction $g : \mathbf{F}_2^\ell \to \mathbf{F}_2$?

Pour répondre à ces questions, il nous faut calculer la probabilité p_g. Pour X, Y deux variables aléatoires indépendantes uniformément distribuées on a :

$$\begin{aligned} p_g &= \mathbf{P}_{X,Y}[f(X,Y) = g(X)] \\ &= \frac{1}{2^\ell} \sum_{x \in \mathbf{F}_2^\ell} \mathbf{P}_Y[f(x,Y) = g(x)] \end{aligned}$$

Si on note par p_x la probabilité que $f(x,\cdot) = 1$, on obtient alors :

$$\begin{aligned} p_g &= \frac{1}{2^\ell} \left(\sum_{x \in g^{-1}(0)} (1 - p_x) + \sum_{x \in g^{-1}(1)} p_x \right) \\ &= \frac{1}{2^\ell} \left((2^\ell - \mathrm{wt}(g)) - \sum_{x \in \mathbf{F}_2^\ell} (-1)^{g(x)} p_x \right) \\ &= \frac{1}{2^\ell} \left(2^{\ell-1} + \sum_{x \in \mathbf{F}_2^\ell} (-1)^{g(x)} \left(\frac{1}{2} - p_x \right) \right) . \end{aligned}$$

On en déduit aisément que p_g est maximale lorsque tous les termes de la somme sont positifs, ce qui nous permet de déterminer la fonction qui maximise le biais utilisable pour l'attaque :

$$\begin{cases} g(x) = 1 & \text{si } p_x > \frac{1}{2} \\ g(x) = 0 & \text{si } p_x < \frac{1}{2} \\ g(x) \text{ quelconque} & \text{si } p_x = \frac{1}{2} \end{cases} .$$

Dans ce cas on obtient :

$$\max_{g \in \mathcal{B}_\ell} p_g = \frac{1}{2} + \frac{1}{2^\ell} \sum_{x \in \mathbf{F}_2^\ell} \left| \frac{1}{2} - p_x \right| .$$

Cette expression nous permet également de répondre à la question du concepteur qui cherche à se prémunir contre cette attaque : pour que ce biais soit nul, il faut et il suffit que pour tout $x \in \mathbf{F}_2^\ell$, $p_x = \frac{1}{2}$. Cela signifie donc que f est équilibrée quand on fixe ses ℓ premières entrées. On dit alors que f est *sans corrélation relativement à ses ℓ premières variables* (ou ℓ-résiliente).

Dans le cas particulier des générateurs par combinaison de LFSR, chaque entrée de la fonction de filtrage (appelée alors fonction de combinaison) correspond à la sortie d'un LFSR. Il est impossible de mener une attaque par corrélation qui effectue une recherche exhaustive sur l'initialisation de ℓ LFSR parmi les n qui composent le système à partir du moment où la fonction est ℓ-résiliente. Pour espérer attaquer un tel système, il faudra donc considérer $\ell + 1$ LFSR simultanément. On peut montrer dans ce cas [CT00] que la probabilité p_g est maximale quand la fonction g correspond à une fonction booléenne affine (plus précisément à la somme des $\ell + 1$ variables considérées plus une constante binaire). L'attaque repose ainsi sur la probabilité $p_g = \mathbf{P}_{X,Y}[f(X,Y) = \mathbf{1} \cdot X \oplus \varepsilon]$. Minimiser cette probabilité revient à maximiser la distance de la fonction de filtrage aux fonctions affines à $\ell + 1$ variables . Ce critère apparaît incontournable au vu des améliorations de l'attaque par corrélation en attaque par corrélation rapide.

Attaques par corrélation rapides

L'attaque par corrélation d'origine utilise la valeur de cette corrélation comme un moyen de détecter la bonne initialisation des registres au cours d'une recherche exhaustive. Cette technique a été améliorée en *attaque par corrélation rapide* par Meier et Staffelbach en 1988 [MS88, MS89] . La recherche exhaustive est alors remplacée par des techniques de correction d'erreurs.

On peut formaliser le principe de l'attaque de la manière suivante. Quand la fonction de transition Φ est linéaire et que la fonction g de l'attaque par corrélation est affine, on peut représenter le vecteur constitué des N premiers bits de la suite $\hat{z}$ comme un mot d'un code linéaire $\mathcal{C}$ de longueur N et de dimension ℓ, où ℓ est la taille de la partie de l'état initial que l'on souhaite retrouver. Si on note G la matrice génératrice du code, on a alors :

$$(x_0, \ldots, x_{\ell-1})G = (\hat{z}_0, \ldots, \hat{z}_{N-1}).$$

Lorsque la suite $\hat{z}$ est corrélée à la suite z, c'est-à-dire lorsque $\mathbf{P}[z_t = \hat{z}_t] = p_g > \frac{1}{2}$, on peut assimiler z au résultat de la transmission de $\hat{z}$ à travers un canal binaire symétrique de probabilité de transition $1 - p_g$. Ainsi, pour retrouver les ℓ bits de la partie d'état initial visée, $(x_0, \ldots, x_\ell)$, il suffit de décoder le mot de code reçu $(z_0, \ldots, z_N)$ relativement au code $\mathcal{C}$.

Dans ce contexte, l'attaque par corrélation de Siegenthaler opère un décodage à maximum de vraisemblance puisqu'elle calcule les 2^ℓ mots de code et choisit le plus proche du mot observé $(z_0, \ldots, z_N)$. Les attaques par corrélation rapides utilisent des algorithmes de décodage plus rapides (itératif, par liste, turbo...) qui permettent de réduire la complexité en temps de l'attaque — on obtient ainsi une complexité notablement inférieure par rapport à la recherche exhaustive sur les ℓ bits de la partie d'initialisation visée — au prix d'une complexité en données plus élevée. Il est possible de transposer l'attaque dans le cas d'un système à registre filtré. On peut classer de nombreuses attaques dans cette catégorie (par exemple [JJ99b, JJ99a, JJ00, CJS00, CT00, CJM02]), les variantes portant sur les différentes techniques de décodage utilisées, prenant en compte plus ou moins complètement les spécifications du système [Lev04].

Attaques algébriques

Attaque algébrique directe Il est toujours possible d'écrire directement l'expression des bits de la suite chiffrante sous la forme d'un système de N équations faisant intervenir l'état initial recherché x_0, la fonction de transition et la fonction de filtrage :

$$\begin{cases} z_0 = f(x_0) \\ z_1 = f \circ \Phi(x_0) \\ \vdots \\ z_N = f \circ \Phi^N(x_0) \end{cases}$$

Ce système est ensuite résolu pour retrouver les bits de x_0. On se ramène ainsi à un problème de résolution de système d'équations, ce qui est aisé dans le cas linéaire — on utilise par exemple un pivot de Gauss. Dans le cas où les équations sont de degré supérieur à 1, il est possible de résoudre efficacement un tel système s'il est surdéterminé par le biais de techniques de linéarisation ou avec des algorithmes calculant des bases de Gröbner. La complexité de ces techniques augmente cependant exponentiellement avec le degré des équations. Afin d'exclure la possibilité d'utiliser de telles techniques sur le système ci-dessus, il est primordial que soit la fonction de filtrage soit la fonction de transition ait un degré algébrique élevé. Ce critère

est d'autant plus important dans le cas où la fonction de transition est linéaire comme dans un LFSR. On rappelle en outre que pour les systèmes à base de LFSR, le degré algébrique de la fonction de filtrage doit être élevé afin d'augmenter la complexité linéaire de la suite chiffrante.

Attaque algébrique améliorée L'attaque directe décrite ci-dessus récemment améliorée par Courtois et Meier sous le nom d'*attaque algébrique* [CM03] a donné de remarquables résultats. Elle a permis de cryptanalyser certains systèmes dont la fonction de transition est linéaire même quand la fonction de filtrage a un degré algébrique élevé [FA03]. L'idée consiste à résoudre non pas le système d'équations décrivant directement le chiffrement mais un système formé par des relations de degré plus faible existant entre l'entrée et la sortie de la fonction de filtrage. Ainsi, s'il existe une fonction g de petit degré telle

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

on en déduit alors que

$$\forall t \in \mathbf{N} \quad g(\Phi^t(x_0)) \cdot z_t = 0.$$

Cela signifie donc que dès qu'on observe un bit de suite chiffrante z_t qui vaut 1, l'état initial x_0 vérifie l'équation

$$g(\Phi^t(x_0)) = 0$$

qui est de degré $\deg(g)$, c'est-à-dire faible quand la fonction de transition Φ est linéaire. Il est donc important pour le cryptanalyste d'arriver à isoler des équations de bas degré. Ces attaques ont donné lieu à un critère appelé *immunité algébrique* qui correspond au plus petit degré du polynôme annulateur des fonctions booléennes f ou $(1+f)$ [MPC04]. Une amélioration de cette attaque appelé *attaque algébrique rapide* a été proposée dans [Cou03] puis dans [Arm04].

1.5 Introduction aux fonctions booléennes

On peut faire appel aux outils d'étude des fonctions booléennes dès qu'on cherche à modéliser des applications binaires. Dans le cas du chiffrement à flot, l'intervention d'une fonction booléenne se matérialise immédiatement dans la fonction de filtrage. Dans le cas du chiffrement par blocs, nous ferons également appel à ces objets au travers des composantes booléennes d'une fonction vectorielle. Nous présentons ici les définitions et les propriétés qui nous sont utiles par la suite en rappelant brièvement le critère cryptographique auxquelles elles correspondent.

1.5.1 Fonctions booléennes, définitions - notations

Une *fonctions booléenne vectorielle* à n variables et m composantes est une application de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$. Ainsi, nous appellerons *fonction booléenne scalaire* (ou simplement *fonction booléenne* lorsqu'il n'y aura aucun risque de confusion) une application de $\mathbf{F}_2^n$ dans $\mathbf{F}_2$. L'ensemble des fonctions booléennes à n variables et m composantes est un espace vectoriel sur $\mathbf{F}_2$ que nous noterons $\mathcal{B}_n^m$ et $\mathcal{B}_n$ lorsque $m = 1$.

Dans la suite du document, nous considérerons une fonction booléenne vectorielle à m composantes comme étant un ensemble de m fonctions booléennes scalaires, c'est pourquoi nous nous attacherons essentiellement à présenter les propriétés des fonctions booléennes scalaires.

Nous rappelons quelques notations, en soulignant le fait que nous avons voulu séparer le symbole de l'addition dans $\mathbf{F}_2$ et les autres formes d'addition dans la mesure où nous considérons parfois les éléments de $\mathbf{F}_2$ comme des éléments de $\mathbf{Z}$. Ainsi, dans l'ensemble {0,1} des valeurs prises par un bit et représenté par le corps fini à deux éléments $\mathbf{F}_2$,

- $\oplus$ désigne l'opération additive (OU-EXCLUSIF) ;
- $\cdot$ (ou rien) représente l'opération multiplicative (ET).

Les vecteurs de $\mathbf{F}_2^n$ sont parfois représentés comme des éléments du corps fini à 2^n éléments, $\mathbf{F}_{2^n}$ grâce à un isomorphisme canonique entre les deux ensembles. Les opérations dans l'espace vectoriel $\mathbf{F}_2^n$ sont notées

- $+$ pour l'addition de deux vecteurs (addition bit à bit) ;
- $\cdot$ pour le produit scalaire de deux vecteurs :

$$\forall\ x,\ y \in \mathbf{F}_2^n, \quad x \cdot y = \bigoplus_{i=1}^{n} x_i y_i\ .$$

Dans le corps fini les opérations sont également notées $+$ pour l'addition entre deux éléments et $\cdot$ (ou rien) pour le produit de deux éléments.

Le vecteur nul est naturellement noté 0, et le vecteur tout-à-un est noté $\mathbf{1}$.

Définition 1.2 *Le* poids *de Hamming d'un vecteur $x = (x_1, \ldots, x_n)$ de $\mathbf{F}_2^n$ est le nombre de ses composantes non-nulles, il est noté* $\mathrm{wt}(x)$ *et vaut :*

$$\mathrm{wt}(x) = \sum_{i=1}^{n} x_i\ .$$

Le support de x est l'ensemble des indices de ses composantes non-nulles :

$$\mathrm{supp}(x) = \{i \in \{1, \ldots, n\}, x_i \neq 0\}\ .$$

La distance de Hamming entre deux vecteurs est le nombre de composantes en lesquelles leurs valeurs diffèrent. La distance entre x et y notée $d(x,y)$ vaut

$$d(x,y) = \mathrm{wt}(x + y).$$

Enfin, en ce qui concerne les ensembles de fonctions — typiquement $\mathcal{B}_n$ — nous notons également l'addition par $+$ et la multiplication par $\cdot$ (ou rien).

Représentations d'une fonction booléenne

Une manière évidente de représenter une application réside dans la description explicite des correspondances entre une entrée et la valeur de cette fonction en cette entrée. Dans le cas d'une fonction booléenne, cette représentation est appelée *table de vérité*.

Définition 1.3 *Soit f une fonction booléenne à n variables. On appelle* table de vérité *de f l'ensemble :*

$$\{(x, f(x)), x \in \mathbf{F}_2^n\}\ .$$

Dans de nombreux cas, on ne travaillera qu'avec le vecteur :

$$(f(P_0),\ldots,f(P_{2^n-1}))$$

où $P_0,\ldots,P_{2^n-1}$ sont les éléments de $\mathbf{F}_2^n$ rangés suivant un ordre quelconque mais fixé (par exemple l'ordre lexicographique). Nous appellerons ce vecteur le *vecteur des valeurs* de f.

Définition 1.4 *Le support de la fonction booléenne f est l'ensemble des vecteurs de $\mathbf{F}_2^n$ pour lesquels la valeur de la fonction est non-nulle :*

$$\mathrm{supp}(f) = \{x \in \mathbf{F}_2^n, f(x) \neq 0\} \ .$$

Le poids *de f est le poids de son vecteur des valeurs, c'est-à-dire le cardinal de son support. On le note* $\mathrm{wt}(f)$.

Le théorème d'interpolation de Lagrange nous permet de dire que toute fonction booléenne admet une unique représentation sous la forme d'un polynôme multivarié. Ce polynôme est appelé *forme algébrique normale* de la fonction.

Définition 1.5 *Soit f une fonction booléenne à n variables. La forme algébrique normale (ANF pour* Algebraic Normal Form*) de f est l'unique polynôme de $\mathbf{F}_2[x_1,\ldots,x_n]/(x_1^2 - x_1,\ldots,x_n^2 - x_n)$ représentant f :*

$$\forall\ x \in \mathbf{F}_2^n, \quad f(x) = \bigoplus_{u \in \mathbf{F}_2^n} c_f(u)\ x^u, \quad c_f(u) \in \mathbf{F}_2, \text{ avec } x^u = x_1^{u_1} x_2^{u_2} \cdots x_n^{u_n} \ .$$

Le degré de f, noté $\deg(f)$, *est alors le degré de sa forme algébrique normale.*

On remarque que le degré d'une fonction correspond au poids maximum du vecteur u tel que le coefficient $c_f(u)$ de la forme algébrique normale de f est non-nul :

$$\deg(f) = \max_{\substack{u \in \mathbf{F}_2^n \\ c_f(u) \neq 0}} \mathrm{wt}(u) \ .$$

Une fonction de degré 1 est appelée fonction *affine* et lorsque sa valeur en 0 est nulle, la fonction est *linéaire*. Nous noterons par φ_a, la fonction linéaire définie par $\varphi_a : x \mapsto a \cdot x$.

Exemple : Soit f une fonction booléenne à 3 variables, $f \in \mathcal{B}_3$, telle que sa forme algébrique normale soit :

$$f(x) = f(x_1,x_2,x_3) = x_1 \oplus x_2 \oplus x_2x_3 \oplus x_1x_2x_3 \ .$$

La table de vérité de f ainsi que les coefficients de son ANF sont présentés dans le tableau 1.1.

Le support de f est $\{(1,0,0),(0,1,0),(1,0,1)\}$, son poids est $\mathrm{wt}(f) = 3$ et son degré est $\deg(f) = 3$. ◇

Il existe d'autres représentations de fonctions booléennes. Mentionnons la *forme numérique normale* [CG99] que nous définirons et utiliserons pour les fonctions booléennes symétriques. Dans le cadre de l'étude des circuits, il existe également une *forme disjonctive normale* et une *forme conjonctive normale*.

$x = (x_1,x_2,x_3)$	$f(x)$	$c_f(x)$
(0, 0, 0)	0	0
(1, 0, 0)	1	1
(0, 1, 0)	1	1
(1, 1, 0)	0	0
(0, 0, 1)	0	0
(1, 0, 1)	1	0
(0, 1, 1)	0	1
(1, 1, 1)	0	1

TAB. 1.1 – *Table de vérité et coefficients de la forme algébrique normale de la fonction* $f(x_1,x_2,x_3) = x_1 \oplus x_2 \oplus x_2x_3 \oplus x_1x_2x_3$

Distance entre deux fonctions

La distance entre deux fonctions booléennes est le nombre de vecteurs en lesquels leurs valeurs diffèrent. Plus formellement, soit f et g deux fonctions de $\mathcal{B}_n$ alors la distance entre f et g notée $d(f,g)$ vaut

$$d(f,g) = \mathrm{wt}(f+g).$$

Ainsi la distance d'une fonction f à un ensemble de fonctions $\mathcal{E}$ s'exprime sous la forme :

$$d(f,\mathcal{E}) = \min_{g\in\mathcal{E}} \mathrm{wt}(f+g).$$

Spectre de Walsh d'une fonction booléenne

Les fonctions booléennes admettent également une représentation spectrale. On peut en effet appliquer dans ce cas la théorie de Fourier dans les groupes abéliens finis (pour une étude mathématique détaillée de cette théorie appliquée aux fonctions booléennes, on peut par exemple se reporter à la thèse de Michael Quisquater [Qui04]). Dans le cas des fonctions booléennes, on s'intéresse pratiquement à la transformée de la fonction signe de $f \in \mathcal{B}_n$ qui est définie par $(-1)^f$. Outre les justifications mathématiques, cette fonction est une manière commode de *centrer* les valeurs prises par la fonction f. Les dénominations n'étant pas toujours clairement fixées, nous parlerons dans notre cas exclusivement de *transformée de Walsh* (mais il est tout à fait possible de trouver dans la littérature les termes *transformée de Fourier, de Walsh-Hadamard, de Walsh, de Hadamard* , avec des définitions qui portent soit sur la fonction elle-même soit sur sa fonction signe, avec une valeur normalisée ou non). Pour les besoins de cette thèse, on peut introduire plus simplement cette notion en considérant la notion intuitive de *corrélation* empruntée aux statistiques et au traitement du signal. Pour cela, nous utilisons ce que nous notons par $\mathcal{F}(f)$ qui correspond à la corrélation de $(-1)^f$ avec la fonction signe de la constante nulle (ou plus précisément à la valeur de l'inter-corrélation de $(-1)^f$ avec $(-1)^0$ en 0). Nous appelons parfois cette quantité, improprement, *poids d'une fonction* ou poids centré de f car on peut aisément la relier au véritable poids d'une fonction booléenne telle que définie par la définition 1.5.1.

$$\mathcal{F}(f) = \sum_{x\in\mathbf{F}_2^n} (-1)^{f(x)} = 2^n - 2\,\mathrm{wt}(f).$$

Elle mesure en fait le *biais* ou le *déséquilibre* de f car la fonction f produit autant de fois la valeur 0 et la valeur 1 si et seulement si $\mathcal{F}(f) = 0$. De plus, la quantité $|\mathcal{F}(f)|$ mesure également la distance de f à l'ensemble des fonctions équilibrées. En effet,

$$|\mathcal{F}(f)| = 2 \min_{\substack{g \in \mathcal{B}_n \\ g \text{ équilibrée}}} d(f,g).$$

Nous définissons alors la *transformée de Walsh* d'une fonction booléenne f comme la corrélation de $(-1)^f$ avec la fonction signe d'une fonction linéaire. Nous parlerons plus commodément de corrélation *centrée* de f avec une fonction linéaire.

Définition 1.6 *Soit f une fonction booléenne à n variables. La* transformée de Walsh *de f est une fonction notée $\mathcal{F}(f + \varphi_{(\cdot)})$ définie par*

$$\begin{array}{rccc} \mathcal{F}(f+\varphi_{(\cdot)}): & \mathbf{F}_2^n & \to & \mathbf{Z} \\ & a & \mapsto & \displaystyle\sum_{x \in \mathbf{F}_2^n} (-1)^{f(x) \oplus a \cdot x} = \mathcal{F}(f + \varphi_a) \end{array}$$

La valeur $\mathcal{F}(f + \varphi_a)$ est appelée coefficient de Walsh de f en a *et l'ensemble*

$$\{\mathcal{F}(f + \varphi_a), a \in \mathbf{F}_2^n\}$$

est appelé spectre de Walsh de f.

La transformée de Walsh vue comme la transformée de Fourier de la fonction signe de f est inversible ce qui garantit que la transformée de Walsh d'une fonction booléenne f en est une représentation complète.

Nous notons également par $\mathcal{L}(f)$ l'amplitude maximale des coefficients de Walsh d'une fonction booléenne f :

$$\mathcal{L}(f) = \max_{a \in \mathbf{F}_2^n} |\, \mathcal{F}(f + \varphi_a)\,|\,.$$

Cette valeur correspond donc à la corrélation maximale de f avec une fonction affine. Cette valeur est naturellement reliée à la *non-linéarité de f* qui mesure la distance de f à l'ensemble des fonctions de degré inférieur ou égal à 1.

Définition 1.7 *Soit f une fonction booléenne à n variables. La* non-linéarité de f, *notée $\mathcal{N}(f)$ est la distance de f à l'ensemble des fonctions booléennes affines :*

$$\mathcal{N}(f) = \min_{a \in \mathbf{F}_2^n} (\mathrm{wt}(f + \varphi_a), \mathrm{wt}(f + \varphi_a + 1)) = 2^{n-1} - \frac{\mathcal{L}(f)}{2}.$$

Proposition 1.8 *Soit $f \in \mathcal{B}_n$. Alors $\mathcal{L}(f) \geq 2^{\frac{n}{2}}$. Les fonctions atteignant cette borne n'existent que pour n pair et sont dites* courbes.

Les fonctions courbes sont particulièrement étudiées car elles correspondent à des objets extrémaux dans diverses théories. Elles correspondent par exemple aux mots de longueur 2^n dont la distance au code de Reed-Muller[2] d'ordre 1 est égale au rayon de recouvrement de ce code [Rot76]. Une caractérisation importante de ces fonctions également très utilisée dans des applications de télécommunications concerne leur spectre d'auto-corrélation. Les coefficients d'auto-corrélation d'une fonction courbe sont en effet nuls excepté en 0 [MS90].

2. Une manière de caractériser le code de Reed-Muller d'ordre r et de longueur 2^m, noté $R(r,m)$, est de dire qu'il correspond à l'ensemble des vecteurs des valeurs des fonctions booléennes à m variables et de degré au plus r.

Dérivées d'une fonction booléenne

La *dérivée d'une fonction booléenne* est une fonction qui apparaît naturellement lorsqu'on considère sa *fonction d'auto-corrélation*. En fait elle intervient aussi pour le cas vectoriel dans la cryptanalyse différentielle.

Considérons la fonction de translation τ_a définie pour $a \in \mathbf{F}_2^n$ et tout $x \in \mathbf{F}_2^n$ par $\tau_a : x \mapsto a+x$. L'auto-corrélation d'une fonction f mesure la corrélation de f avec ses décalés $f \circ \tau_a$. On peut définir la somme $f + f \circ \tau_a$ comme étant la dérivée de f relativement au vecteur a pour souligner l'aspect de différence discrète entre deux versions décalées de la même fonction.

Définition 1.9 *Soit f une fonction booléenne à n variables. Pour tout a de $\mathbf{F}_2^n$, on définit $D_a f$, la* dérivée de f relativement au vecteur a *par*

$$\begin{array}{rccl} D_a f : & \mathbf{F}_2^n & \to & \mathbf{F}_2 \\ & x & \mapsto & f(x) \oplus f(x+a). \end{array}$$

On peut ainsi définir le spectre d'auto-corrélation d'une fonction booléenne.

Définition 1.10 *Soit f une fonction booléenne à n variables. On appelle* spectre d'auto-corrélation de f *l'ensemble*

$$\{\mathcal{F}(D_a f), a \in \mathbf{F}_2^n\}.$$

Ce qui nous permet de formaliser la caractérisation des fonctions courbes par leur spectre d'auto-corrélation.

Proposition 1.11 *Soit $f \in \mathcal{B}_n$. Alors f est* courbe *si et seulement si $\mathcal{F}(D_a f) = 0$ pour tout $a \neq 0$.*

Autrement dit, les fonctions courbes sont les fonctions dont toutes les dérivées (à part la dérivée relativement au vecteur nul) sont équilibrées.

On peut naturellement étendre récursivement la définition de la dérivée au cas des *dérivées d'ordre supérieur* [Dil74] [Lai94].

Définition 1.12 *Soit f une fonction booléenne à n variables et $a_1, \ldots, a_i$ des vecteurs linéairement indépendants de $\mathbf{F}_2^n$. Notons par $\langle a_1, \ldots, a_i \rangle$ le sous-espace vectoriel engendré par ces i vecteurs. On appelle* dérivée d'ordre i de f relativement au sous-espace $\langle a_1, \ldots, a_i \rangle$ *la fonction définie par :*

$$\begin{array}{rccl} D_{\langle a_1,\ldots,a_i \rangle} f : & \mathbf{F}_2^n & \to & \mathbf{F}_2 \\ & x & \mapsto & D_{a_i}\left(D_{\langle a_1,\ldots,a_{i-1} \rangle} f\right)(x) = \displaystyle\bigoplus_{v \in \langle a_1,\ldots,a_i \rangle} f(x+v) \end{array}$$

La dernière égalité montre clairement que la valeur d'une dérivée d'ordre i ne dépend que du sous-espace vectoriel engendré par i vecteurs indépendants et aucunement de la base ainsi formée.

1.5.2 Propriétés cryptographiques des fonctions booléennes

Ainsi définies, les fonctions booléennes possèdent certaines propriétés qui sont recherchées pour la conception d'un algorithme de chiffrement. Outre le *degré algébrique* et la *non-linéarité* que nous avons définis au paragraphe précédent, les critères cryptographiques courants pour les fonctions booléennes sont les suivants.

Fonctions équilibrées Pour d'évidentes raisons statistiques, on privilégie les fonctions dont la sortie prend autant de fois la valeur 0 que la valeur 1. De telles fonctions sont dites équilibrées.

Définition 1.13 *Soit $f \in \mathcal{B}_n$. La fonction f est dite* équilibrée *si sa sortie est uniformément distribuée,* i.e.

$$\#\{x \in \mathbf{F}_2^n, f(x) = 0\} = \#\{x \in \mathbf{F}_2^n, f(x) = 1\} = 2^{n-1}$$

Lorsque f est équilibrée sa transformée de Walsh en 0 est nulle : $\mathcal{F}(f) = 0$.

Fonctions sans corrélation d'ordre ℓ Afin de résister aux attaques par corrélation, une fonction doit être sans corrélation avec un sous-ensemble de ses entrées.

Définition 1.14 *Soit $f \in \mathcal{B}_n$. On note $X = (X_1, \ldots, X_n)$ un ensemble de variables aléatoires mutuellement indépendantes. La fonction f est dite*

- sans corrélation par rapport à l'ensemble d'indices $L \in \{1, \ldots, n\}$ *si la distribution de probabilités de sa sortie est inchangée quand les entrées $(X_i)_{i \in L}$ sont fixées et que les $(X_i)_{i \notin L}$ forment un ensemble de variables aléatoires mutuellement indépendantes uniformément distribuées. Autrement dit*

$$\forall a \in \mathbf{F}_2^{|L|}, \mathbf{P}\left[f(X) = 0 \mid \forall i \in L, X_i = a_i\right] = \mathbf{P}\left[f(X) = 0\right] \ ;$$

- sans corrélation d'ordre ℓ *si, pour tout ensemble d'indices L de cardinal inférieur ou égal à ℓ, f est sans corrélation par rapport à L ;*
- *ℓ-résiliente si f est sans corrélation d'ordre ℓ et équilibrée.*

L'ordre de résilience d'une fonction booléenne peut également être défini grâce à ses coefficients de Walsh.

Proposition 1.15 *[XM88] La fonction $f \in \mathcal{B}_n$ est ℓ-résiliente si et seulement si*

$$\forall a \in \mathbf{F}_2^n,\ 0 \leq \mathrm{wt}(a) \leq \ell,\ \mathcal{F}(f + \varphi_a) = 0.$$

Il est important de noter qu'il existe un compromis entre le degré algébrique maximal d'une fonction et l'ordre d'immunité aux corrélations maximal.

Proposition 1.16 Borne de Siegenthaler [Sie84]
Soit $f \in \mathcal{B}_n$. Si f est sans corrélation d'ordre ℓ, alors son degré algébrique noté d, vérifie la relation suivante :

$$d + \ell \leq n.$$

Si de plus elle est équilibrée et que $\ell < n - 1$, alors

$$d + \ell \leq n - 1.$$

Critère de propagation, structure linéaire Le critère de propagation correspond à un critère cryptographique qui ne paraît essentiel que pour les chiffrements par blocs et les fonctions de hachage. Il semble néanmoins que certaines attaques contre des chiffrements à flot utilisent des propriétés de dérivées de la fonction de filtrage [GCD00]. Du point de vue d'un chiffrement par blocs — rappelons qu'ils sont aussi dits *sans mémoire* — le résultat du chiffrement d'un bloc étant déterminé uniquement par ce bloc et la clé, il apparaît légitime de chercher à quantifier la modification apportée au résultat du chiffrement d'un bloc x par la complémentation de certains bits de x. À l'origine, l'étude d'une telle modification portait sur un bit complémenté. On cherche bien sûr à ce qu'une telle modification ne mette pas en lumière un biais. Lorsque pour une fonction booléenne f et pour tout vecteur e_i de poids 1, la probabilité que $f(x+e_i)$ diffère de $f(x)$ est exactement $1/2$, on dit que f satisfait le *critère d'avalanche strict* (SAC) [WT86] . Le *critère de propagation* correspond à la généralisation de ce critère pour les vecteurs de poids supérieur.

Définition 1.17 *[PLL+91] Soit $f \in \mathcal{B}_n$. On dit que f vérifie le* critère de propagation de degré k, *(PC(k)), si pour tout $a \in \mathbf{F}_2^n$ tel que $1 \leq \mathrm{wt}(a) \leq k$ la fonction $D_a f : x \mapsto f(x) \oplus f(x+a)$ est équilibrée.*

Cela signifie en d'autres termes que pour tout $a \in \mathbf{F}_2^n$, $1 \leq \mathrm{wt}(a) \leq k$, les coefficients d'autocorrélation de f sont nuls : $\mathcal{F}(D_a f) = 0$. Ainsi les fonctions *courbes* de $\mathcal{B}_n$ sont les fonctions qui vérifient $PC(n)$.

D'une certaine manière, le contrepoint du critère de propagation correspond aux *structures linéaires*, ce sont les vecteurs tels que la dérivée d'une fonction relativement à eux est une constante.

Définition 1.18 *Soit $f \in \mathcal{B}_n$. On dit que $a \in \mathbf{F}_2^n$ est une structure linéaire de f si $D_a f$ est une fonction constante.*

L'ensemble des structures linéaires d'une fonction forme naturellement un sous-espace vectoriel de $\mathbf{F}_2^n$.

On pourra naturellement parler de *structure linéaire d'ordre supérieur* pour désigner les sous-espaces vectoriels tels que la dérivée d'une fonction relativement à eux est une constante.

Définition 1.19 *Soit $f \in \mathcal{B}_n$. On dit que V sous-espace vectoriel de $\mathbf{F}_2^n$ est une structure linéaire d'ordre* $\dim(V)$ *de f si $D_V f$ est une fonction constante.*

Immunité algébrique d'une fonction booléenne Un des paramètres permettant de quantifier la résistance aux attaques algébriques d'une fonction de filtrage pour un chiffrement à flot est l'*immunité algébrique*. En effet, le principe de l'attaque réside dans la possibilité de résoudre le système d'équations reliant les bits de sortie et l'état initial dès que le degré de ces relations est faible. Soit $AN(f)$ l'idéal annulateur de f, c'est-à-dire

$$AN(f) = \{g \in \mathcal{B}_n,\, g \neq 0 \mid g \cdot f = 0\}\,.$$

Alors l'immunité algébrique de f est le plus petit degré atteint par une fonction de $AN(f) \cup AN(1+f)$ [MPC04].

Définition 1.20 *Soit $f \in \mathcal{B}_n$. On définit l'*immunité algébrique de f *que l'on note $AI(f)$ par :*

$$AI(f) = \min_{g \in AN(f) \cup AN(1+f)} \deg(g).$$

Le nombre de fonctions de degré au plus d dans $AN(f)$ est égal à 2^κ où κ est la dimension du noyau de la matrice obtenue après restriction du code de Reed-Muller d'ordre d et de longueur 2^n au support de f. Les lignes d'une telle matrice correspondent donc aux évaluations des monômes de degré au plus d en les valeurs de x telles que $f(x) = 1$. Cette matrice ayant $\sum_{i=0}^{d} \binom{n}{i}$ lignes et $\mathrm{wt}(f)$ colonnes, son noyau est non-nul dès que

$$\sum_{i=0}^{d} \binom{n}{i} > \mathrm{wt}(f) \ .$$

De même $AN(1+f)$ contient des fonctions de degré au plus d si

$$\sum_{i=0}^{d} \binom{n}{i} > 2^n - \mathrm{wt}(f) \ .$$

Ainsi, comme il est souligné dans [DGM04], l'immunité algébrique d'une fonction est liée à son poids. On peut ainsi remarquer que pour un nombre impair de variables, seules les fonctions équilibrées sont susceptibles de présenter une immunité algébrique maximale. On peut également en déduire en corollaire immédiat que pour toute fonction booléenne f à n variables, $AI(f) \leq \lceil n/2 \rceil$. Par un raisonnement similaire, on peut également établir un lien entre l'immunité algébrique d'une fonction et sa non-linéarité [DGM04]. On peut démontrer que pour toute fonction linéaire φ, l'immunité algébrique de $f + \varphi$ est au plus $AI(f) + 1$. Ainsi, toute fonction $f \in \mathcal{B}_n$ d'immunité algébrique au moins d vérifie :

$$\mathcal{N}(f) \geq \sum_{i=0}^{d-2} \binom{n}{i}.$$

Cela signifie en particulier que toute fonction dont l'immunité algébrique est optimale présente également une haute non-linéarité, et plus précisément :

$$\mathcal{N}(f) \geq \begin{cases} 2^{n-1} - \binom{n}{\frac{n-1}{2}} & \text{si } n \text{ est impair} \\ 2^{n-1} - \frac{1}{2}\binom{n}{\frac{n}{2}} - \binom{n}{\frac{n}{2}-1} & \text{si } n \text{ est pair.} \end{cases}$$

1.6 Le chiffrement symétrique itératif par blocs

Les chiffrements par blocs sont des primitives cryptographiques largement répandues. Ils peuvent en effet être vus comme des composantes cryptographiques élémentaires utilisables dans de nombreux contextes grâce à des combinaisons adaptées (chiffrement à flot, fonction de hachage, fonction à sens unique, authentification de message ou d'entité). D'autre part, ils comptent parmi eux les standards de chiffrement que sont le DES désormais remplacé par l'AES. La très large utilisation de ces chiffrements a motivé leur étude et conduit en une quarantaine d'années à la formalisation, certes incomplète, de critères de conception stables s'appuyant sur un nombre réduit de schémas de base et l'ensemble des attaques publiées à ce jour.

1.6.1 Le traitement par blocs des données

Un chiffrement est dit par blocs s'il divise le texte en clair en blocs de taille fixe (généralement 64 ou 128 bits) et applique la même fonction aux blocs successifs.

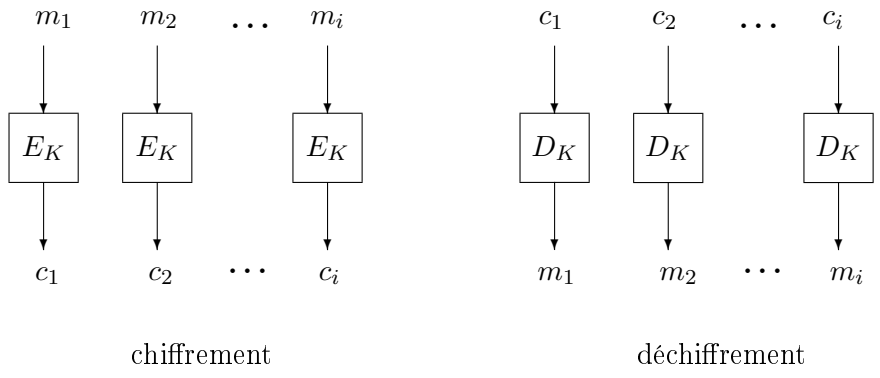


FIG. 1.8 – *Le mode opératoire ECB*

La méthode la plus simple pour chiffrer un message dont la longueur dépasse la taille d'un bloc consiste à diviser ce message en plusieurs blocs qu'on chiffre séparément. Ce mode opératoire appelé ECB (Electronic Codebook) présente de nombreux inconvénients, dont le plus évident réside dans le fait que deux blocs de clair identiques produisent deux blocs chiffrés identiques. Cette propriété permet à un attaquant de détecter d'éventuelles répétitions dans le texte clair.

Dans la pratique, les chiffrements par blocs sont utilisés avec d'autres modes opératoires plus sûrs dans lesquels chaque bloc de chiffré dépend de tous les blocs de clair précédents. On considère que ces modes opératoires sont un moyen de transformer un chiffrement par blocs en chiffrement à flot synchrone ou auto-synchronisant selon le mode [MvOV97]. On peut citer par exemple les modes opératoires CBC (Cipher Block Chaining), CFB (Cipher Feedback) et OFB (Output Feedback) dont une variante importante est le mode compteur (CTR). On décrit à la figure 1.9, le mode CBC. Sur ce schéma, IV est un vecteur d'initialisation constant.

Les modes opératoires ne doivent pas introduire de faiblesses dans l'utilisation d'un chiffrement par blocs (le mode ECB en est un exemple). Certains paramètres président donc au choix d'un mode opératoire, tels la sécurité, la propagation des erreurs, la nécessité de synchronisation et le débit obtenu.

Considérons un bloc de message m de taille n bits. Nous ne nous intéressons dans ce qui suit qu'aux propriétés de la primitive sous-jacente que constitue le chiffrement par blocs :

$$\begin{array}{rccc} E: & \mathbf{F}_2^k \times \mathbf{F}_2^n & \rightarrow & \mathbf{F}_2^n \\ & (K,m) & \mapsto & E_K(m) = c. \end{array}$$

Pour que le chiffrement E_K soit sûr, il faut qu'il soit impossible de retrouver m à partir de c sans aucune connaissance sur K. L'application qui relie m au chiffré $c = E_K(m)$ doit être une bijection afin d'assurer l'unicité du déchiffrement. Ainsi E_K est une permutation sur l'ensemble des vecteurs de $\mathbf{F}_2^n$. Il existe $(2^n)!$ telles permutations indexées par les 2^k valeurs possibles de la clé K. Concevoir un système de chiffrement par blocs E revient ainsi à faire

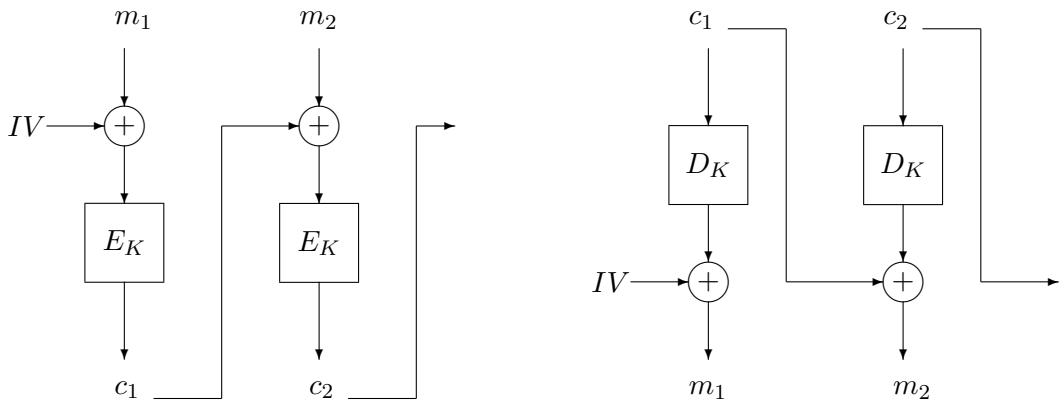


FIG. 1.9 – *Le mode opératoire CBC*

le choix du sous-ensemble de permutations définies par la clé K. Idéalement, un tel ensemble de 2^k permutations ne devrait pas être distinguable d'un ensemble de 2^k permutations tirées aléatoirement parmi les $(2^n)!$ possibles.

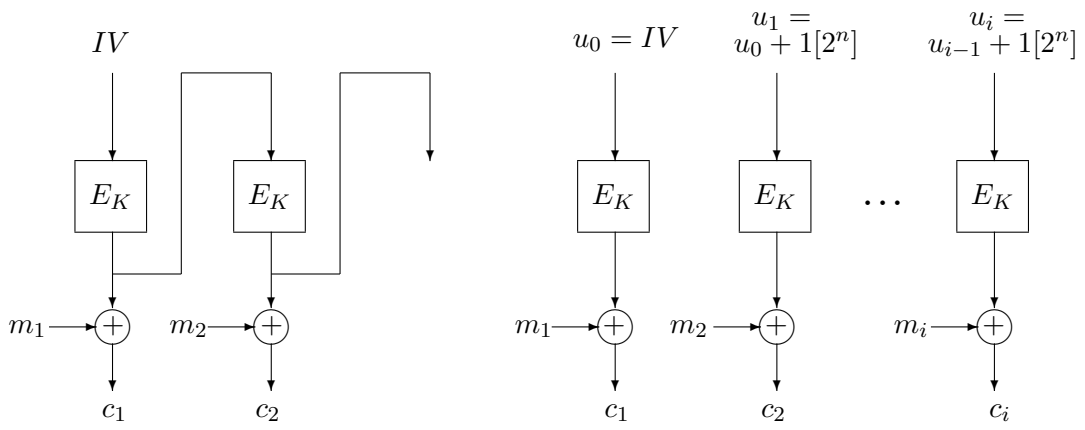
1.6.2 Principe général d'un chiffrement itératif

Les chiffrements par blocs actuels sont des *chiffrements par composition*. Le principe d'un chiffrement par composition repose sur l'intuition que la conjonction de plusieurs unités élémentaires de chiffrement soigneusement choisies et aux propriétés bien définies permet d'obtenir un chiffrement résultant cryptographiquement plus résistant que chacune des unités prises séparément. Cette technique fut formalisée par Horst Feistel au début des années 70, lorsqu'il travaillait sur l'algorithme LUCIFER qui servit de base à la conception du DES. Les notions fondamentales proviennent de l'article fondateur de Claude Shannon, *The communication theory of secrecy systems* [Sha49], qui établit les bases mathématiques d'un système de communication chiffrée à partir de la théorie de l'information. Deux notions capitales s'imposent alors, la *diffusion* et la *confusion*. 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 classe de chiffrements par composition la plus fréquemment rencontrée est constituée par les *chiffrements itératifs par blocs*.

De manière formelle, un chiffrement itératif par blocs consiste à itérer r fois une fonction interne F (cf. Figure 1.11). À chacun des r tours, la fonction F est paramétrée par une quantité secrète $k^{(i)}$, la clé de tour. Pour que le chiffrement soit inversible, la fonction itérée F doit être une permutation pour chaque valeur possible du paramètre $k^{(i)}$. Les r clés $(k^{(1)}, \cdots, k^{(r)})$ sont en général dérivées d'une unique clé-maître par un algorithme de cadencement de clé.

Ainsi le chiffrement E_K résulte de la composition des r fonctions de tours $F_{k^{(i)}}$:

$$E_K = F_{k^{(r)}} \circ F_{k^{(r-1)}} \circ \cdots \circ F_{k^{(2)}} \circ F_{k^{(1)}},$$



OFB à rétroaction complète mode compteur

FIG. 1.10 – *Variantes du mode opératoire OFB*

où chacune des fonctions $F_{k^{(i)}}$ est cryptographiquement faible mais permet d'assurer un chiffrement fort après r itérations. Comment fixe-t-on le nombre d'itérations ? La borne inférieure est fixée par la contrainte de sécurité que constitue la faiblesse de la fonction de tour ; nous citons le proverbe cryptographique :*la plupart des chiffrements sont sûrs après application de suffisamment de tours.* D'autre part, la borne supérieure est quant à elle limitée par les performances du système ; la phrase précédente appelant toujours la fameuse réplique : *la plupart des chiffrements sont trop lents après application de suffisamment de tours*[3].

La plupart des algorithmes à clé secrète par blocs sont ainsi construits suivant le modèle itératif. Dans le DES, par exemple, les blocs comportent 64 bits. La fonction itérée F est une permutation de $\mathbf{F}_2^{64}$ paramétrée par une clé de 48 bits. Cette fonction est itérée 16 fois et les 16 sous-clés $(k^{(1)}, \cdots, k^{(16)})$ sont dérivées d'une clé-maître de 56 bits. Quant à l'AES, pour une clé de 128 bits, il itère 10 fois une permutation de $\mathbf{F}_2^{128}$ paramétrée par une sous-clé de 128 bits.

Les algorithmes itératifs par blocs se répartissent essentiellement en deux grandes familles, suivant la structure de la fonction interne F. Les deux structures principalement utilisées sont le *schéma de Feistel* (utilisée dans le DES), que nous détaillons par la suite, et la structure de *réseau de substitution-permutation* (utilisée dans l'AES). Il est bien évident que ces deux catégories ne permettent pas de classer de manière stricte les chiffrements qui s'en inspirent pour leur conception. On compte en général de nombreuses variantes construites à partir d'elles et dont les divergences s'expliquent par la prise en compte de contraintes d'implémentation, de performances, d'attaques dédiées, etc.

1.6.3 Chiffrement de type Feistel

Un chiffrement de Feistel se caractérise par le traitement spécifique sous forme de moitiés du bloc d'entrée qui permet d'obtenir l'inversibilité du chiffrement dans tous les cas.

3. La première phrase est due à Luke O'Connor, la réponse est due à James Massey.

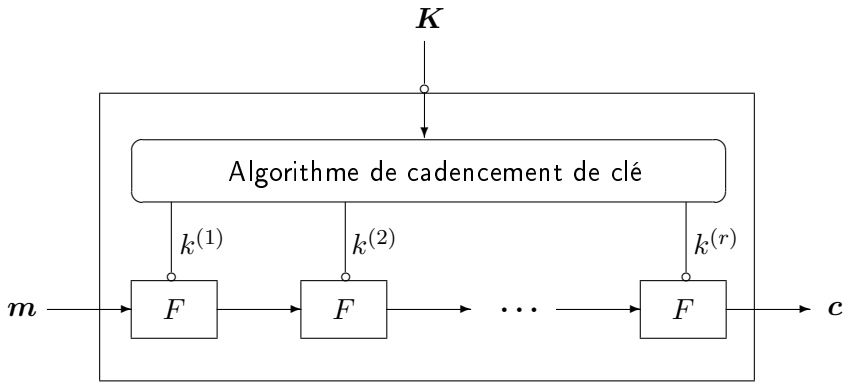


FIG. 1.11 – *Principe d'un chiffrement itératif par blocs*

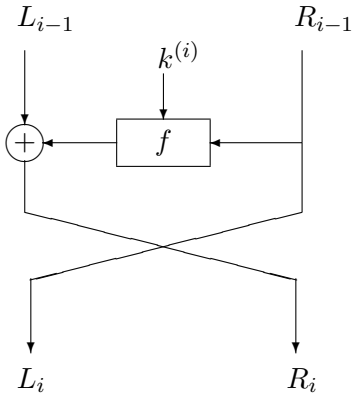


FIG. 1.12 – *Fonction itérée d'un chiffrement de Feistel*

Définition 1.21 Chiffrement de Feistel
Un chiffrement de Feistel est un chiffrement itératif par blocs opérant sur des blocs de $2n$ *bits. La fonction itérée* $F_{k^{(i)}}$ *est définie par :*

$$\begin{array}{rccc} F_{k^{(i)}} : & \mathbf{F}_2^n \times \mathbf{F}_2^n & \rightarrow & \mathbf{F}_2^n \times \mathbf{F}_2^n \\ & (L_{i-1},R_{i-1}) & \mapsto & (L_i,R_i) \end{array}$$

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* $k^{(1)},\cdots,k^{(r)}$.

Description du DES Historiquement incontournable, le DES est naturellement un des premiers exemples de structure de Feistel. La structure globale du chiffrement est décrite à la figure 1.13. Nous ne résumons ici que brièvement les caractéristiques du chiffrement, pour de plus amples détails nous renvoyons le lecteur à la description du standard [FIP99].

Dans le DES, le bloc compte 64 bits considérés sous la forme de deux mots de 32 bits. La clé-maître comporte 56 bits codés en 7 bits dans 8 octets, le dernier élément de chaque octet

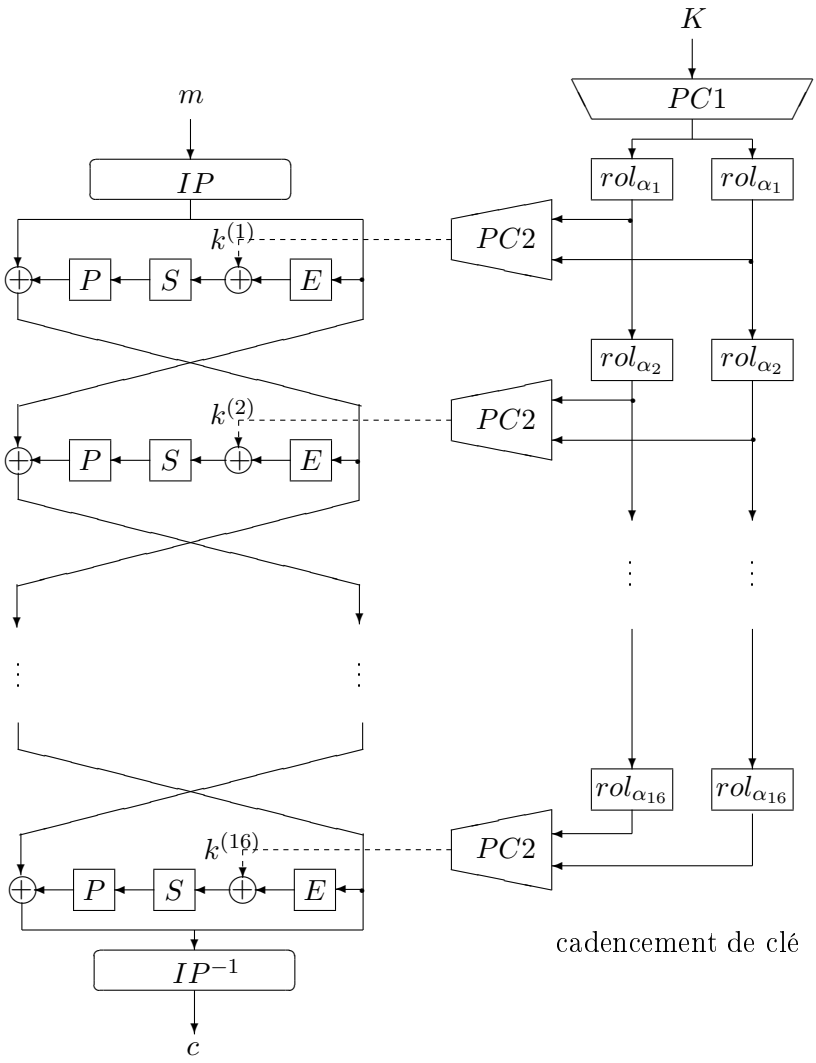


FIG. 1.13 – *Le DES*

jouant le rôle de bit de parité. L'algorithme de cadencement de clé produit à chaque tour une sous-clé de 48 bits. La structure de la fonction itérée du DES se décompose en quatre parties, une expansion affine E, l'insertion de la sous-clé par addition bit-à-bit, une fonction non-linéaire S dans laquelle on retrouve les 8 fameuses *boîtes S du DES* et enfin une permutation P. Finalement pour compléter la description de l'algorithme, nous signalons que IP est une permutation sur l'ordre des bits du clair, $PC1$ est une permutation sur les 56 bits de la clé, rol_{α_i} est un décalage circulaire de α_i bits et $PC2$ est une fonction de compression simple de 56 bits vers 48.

1.6.4 Chiffrement de type substitution-permutation

Nous nous intéressons aux chiffrements qui ont une structure itérative. Contrairement aux chiffrements de Feistel dont la structure même garantit qu'ils sont inversibles et permet d'utiliser des fonctions internes quelconques, les chiffrements de type substitution-permutation

imposent des contraintes sur les fonctions internes utilisables.

Définition 1.22 Réseau de substitution-permutation
Un réseau de substitution-permutation itératif opère sur des blocs de n bits. La fonction itérée $F_{k^{(i)}}$ 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 $F_{k^{(i)}}$ doit être une bijection.

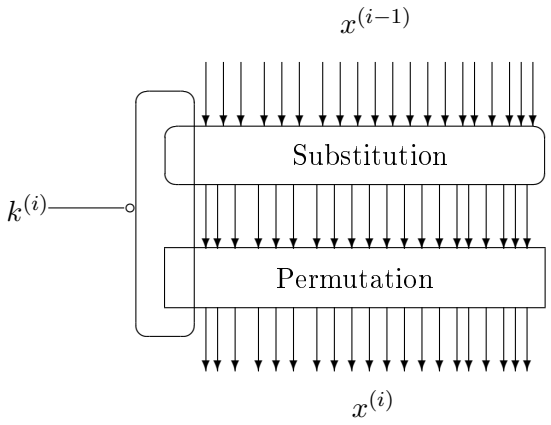


FIG. 1.14 – *Une itération dans un réseau de substitution-permutation*

La définition très large ne permet pas d'avoir une représentation très précise du principe (en particulier l'insertion de la clé). Nous allons illustrer cette famille de chiffrements grâce à l'AES.

Description de l'AES L'*Advanced Encryption Standard* est issu d'une compétition et d'une expertise qui s'échelonna de l'appel à candidature par le NIST en 1997 jusqu'à la sélection finale du candidat *Rijndael* en octobre 2000. Tout comme son prédécesseur, ce standard a été l'occasion de formaliser des critères de conception réunissant l'état de l'art des dernières connaissances en cryptographie. Outre le standard [FIP01] auquel on peut se reporter, un livre a également été publié afin d'expliquer et de justifier les propriétés choisies pour le système [DR02]. La structure globale de l'AES à 128 bits de clé est décrite à la figure 1.15.

Dans l'AES, le bloc comporte 128 bits. La clé peut compter 128, 196 ou 256 bits, ce qui influence le nombre de tours du chiffrement. La version à 128 bits comporte 10 itérations. L'algorithme de cadencement de clé produit des sous-clés de la même taille que la clé-maître. La première itération est précédée par l'addition de la sous-clé numéro 0 qui correspond à la clé-maître et la dernière itération ne comporte pas de `MixColumns`. La fonction itérée se compose du quadruplet de fonctions (`SubBytes`, `ShiftRows`, `MixColumns`, `AddRoundKey`), cette dernière fonction étant simplement l'addition bit-à-bit de la sous-clé au bloc courant. On y retrouve trois étapes, conformément aux principes fondamentaux de confusion et de diffusion énoncés par Shannon. La première étape, dite de confusion, la fonction `SubBytes`, consiste à appliquer à chacun des 16 octets de l'entrée une même permutation S. Cette fonction correspond (à une application affine près) à la fonction inverse dans le corps fini à 2^8 éléments (dans la pratique, elle est mise en table) ; elle assure la résistance de l'algorithme aux attaques différentielle

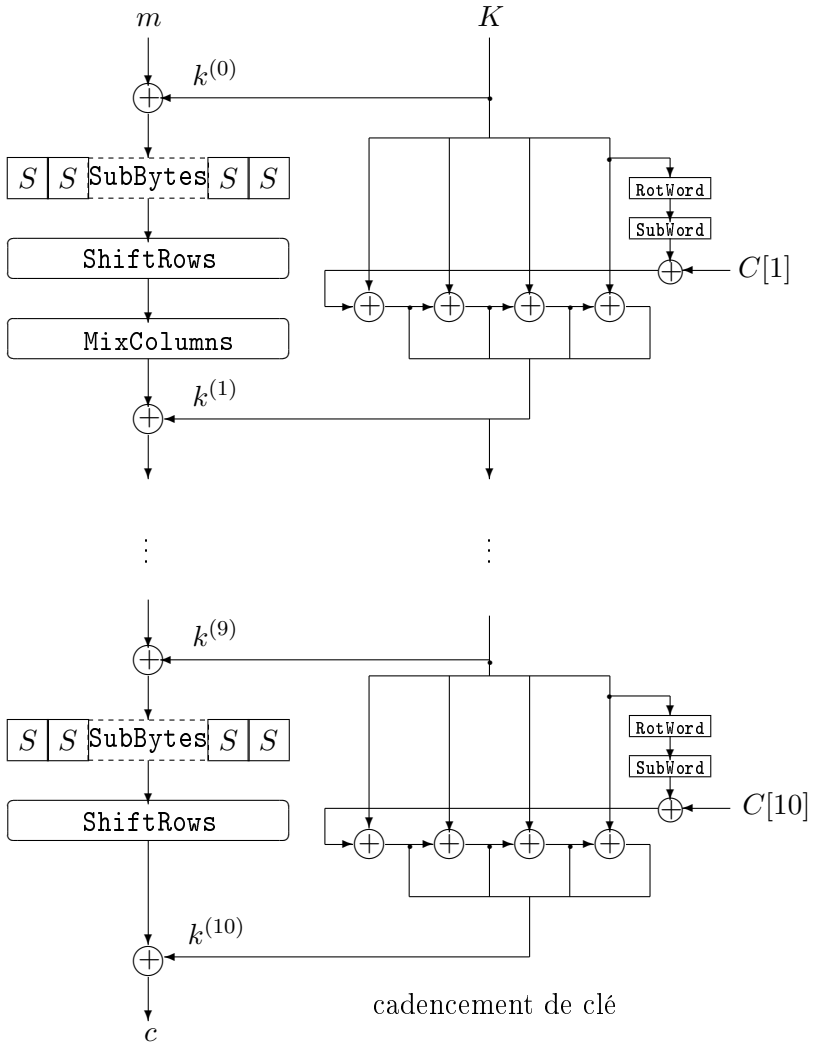


FIG. 1.15 – *L'AES*

et linéaire. La phase de diffusion est composée des fonctions `ShiftRows` et `MixColumns` qui représentent des opérations simples sur le corps à 2^8 éléments. Enfin, on effectue un OU EXCLUSIF bit-à-bit entre le résultat et la sous-clé de l'itération.

Les sous-clés de 128 bits, numérotées de 0 à 10, sont dérivées de la clé secrète de la manière suivante : la sous-clé numéro 0 correspond à la clé secrète ; ensuite, la sous-clé numéro i (utilisée à la $i$ème itération) est obtenue à partir de la sous-clé numéro $(i-1)$ grâce à l'algorithme décrit à la figure 1.15. On permute de manière circulaire, par la fonction `RotWord`, les quatre derniers octets de la clé numéro $(i-1)$, puis on leur applique la fonction `SubWord` composée de 4 permutations S. Après avoir ajouté une constante (dépendant de i) au premier octet (les trois autres octets de la constante $C[i]$ du schéma sont nuls), on effectue une addition bit-à-bit entre les quatre octets ainsi obtenus et les quatre premiers octets de la sous-clé précédente. Les trois autres blocs de quatre octets de la clé numéro i sont ensuite simplement le résultat d'un OU EXCLUSIF entre le bloc correspondant de la sous-clé $(i-1)$ et le bloc précédent de la sous-clé i.

1.6.5 Attaques structurelles sur les chiffrements itératifs par blocs

Lorsque la taille du bloc du chiffrement est suffisante ainsi que la taille de la clé, les attaques ayant pour objet les chiffrements itératifs par blocs reposent sur l'analyse approfondie des spécifications du système et des informations qu'il peut laisser filtrer. Si nous mettons de côté les attaques *physiques* par canaux secondaires, les principales attaques et les critères généraux de conception pour ces systèmes reposent sur l'étude de la structure du chiffrement dont on cherche à déduire des propriétés pouvant conduire à des faiblesses. Dans cette optique, on peut séparer deux types d'approches.

Une approche algébrique déterministe peut essayer de reconstruire le chiffrement comme un polynôme dont les coefficients dépendent de la clé. Si le nombre de coefficients est insuffisamment élevé, le système peut alors être cassé sans retrouver la clé elle-même ; l'attaque par interpolation de Jakobsen et Knudsen repose sur ce principe [JK97]. On peut également essayer de représenter l'algorithme sous la forme d'un système d'équations faisant intervenir les bits du clair, du chiffré et de la clé. Si le système peut être résolu de manière significativement plus rapide que la recherche exhaustive, on retrouve la clé et le système est cassé. Aucun système actuel n'est sensible à la version triviale de l'attaque précédente. Néanmoins des tentatives d'exploiter des structures algébriques fortes des chiffrements pour obtenir des systèmes d'équations surdéterminés alliés à des méthodes de résolution efficaces de systèmes ont été imaginés [CP02]. Si ces attaques ont fait leurs preuves contre les chiffrements à flot, elles demandent encore à être améliorées dans le cas des chiffrements par blocs afin de savoir si elles sont ou non effectives.

Une approche statistique peut également être privilégiée. On se rappelle que concevoir un chiffrement par blocs revient à sélectionner un sous-ensemble de permutations indexées par les clés K possibles. L'idée générale des attaques statistiques est de déterminer un biais dans la distribution statistique d'une grandeur liée au système permettant de distinguer le sous-ensemble des permutations issues du chiffrement ou d'une partie du chiffrement d'un sous-ensemble de permutations tirées aléatoirement. La plus grande partie des attaques publiées appartient à cette catégorie. Les attaques algébriques précédemment décrites peuvent également être modifiées afin de produire des attaque statistiques. Lorsqu'une propriété statistique est déterminée il est alors possible de l'utiliser pour retrouver tout ou partie de la clé, principalement dans le cadre d'attaques du type *diviser pour régner* que nous modélisons dans le chapitre suivant sous la forme d'attaque *sur le dernier tour*.

Au sein des attaques statistiques on distingue clairement les cryptanalyses différentielle et linéaire qui ont été les premières attaques génériques publiées à conduire à des critères de conception généraux largement applicables. La modélisation de ces attaques et l'étude des critères de sécurité qui en ont été déduits font l'objet du chapitre 2 où nous nous intéressons aux généralisations de l'attaque différentielle.

Chapitre 2

Attaques sur le dernier tour et généralisations de la cryptanalyse différentielle

La première partie des travaux de ma thèse porte sur la sécurité des chiffrements itératifs par blocs. Comme nous l'avons vu dans le chapitre précédent, un chiffrement itératif par blocs est constitué de r itérations ou tours d'une fonction paramétrée par une clé de tour. Cette conception repose sur le principe du chiffrement par composition où la conjonction de plusieurs composants cryptographiquement faibles doit produire un chiffrement résultant résistant. Les cryptanalyses publiées durant les 20 dernières années ont permis de formaliser des critères de conception pour les chiffrements itératifs par blocs. Ces critères correspondent à des propriétés mathématiques que doivent vérifier les fonctions de tour. Ainsi la présence de fonctions de non-linéarité maximale dans les fonctions de tour assure au système une résistance optimale aux attaques linéaires [Mat93, Mat94]. De telles fonctions sont appelées *fonctions presque courbes* [CV95] . Elles n'existent que pour un nombre impair de variables mais elles garantissent en revanche également une résistance optimale aux attaques différentielles. De telles fonctions sont par exemple utilisées dans le chiffrement MISTY [Mat97] dont je détaille et justifie, dans le chapitre suivant, la cryptanalyse différentielle d'ordre supérieur dont il a fait l'objet dans [BF00]. L'étude de ce cas particulier permet d'en déduire une généralisation que je présente dans le chapitre 4. En fait et cette idée revient de manière récurrente au fil des cryptanalyses et des critères auxquelles elles donnent lieu, les fonctions optimales garantissant une résistance maximale à une attaque donnée sont des fonctions fortement structurées. Ces structures, tout en renforçant un aspect du système, introduisent une autre faiblesse dans le chiffrement. En ce qui concerne les fonctions presque courbes, la faiblesse provient du fait que le spectre de Walsh d'une fonction presque courbe est divisible par une grande puissance de 2. J'utilise cette propriété pour présenter dans le chapitre 4 une généralisation d'attaque différentielle d'ordre supérieur pour les chiffrements de Feistel à 5 tours.

2.1 Cryptanalyses sur le dernier tour

Considérons une famille $(F_k)_{k\in\mathcal{K}}$ de permutations de $\mathbf{F}_2^n$, espace des vecteurs de n bits. Le paramètre k, vecteur binaire de taille ℓ appartient à l'ensemble $\mathcal{K} \subset \mathbf{F}_2^\ell$ appelé *espace des clés de tour*. Le chiffrement est le résultat de la composition de r permutations $F_{k^{(i)}}$ où $k^{(i)} \in \mathcal{K}$, pour $1 \leq i \leq r$:

$$\begin{aligned} x^{(r)} &= F_{k^{(r)}} \circ F_{k^{(r-1)}} \circ \cdots F_{k^{(2)}} \circ F_{k^{(1)}}(x^{(0)}) \\ &= E_K(x^{(0)}), \end{aligned}$$

où on note également par $x^{(i)}$ le résultat obtenu après le i-ème tour, c'est-à-dire $x^{(i)} = F_{k^{(i)}}(x^{(i-1)})$ pour $1 \leq i \leq r$. Le vecteur $(k^{(1)},k^{(2)},\ldots,k^{(r)})$ est la clé du chiffrement et ses composantes sont les clés de tour. Ces clés de tours peuvent être obtenues par un algorithme de cadencement de clé à partir d'une unique clé maître, K, de taille inférieure à celle de la concaténation des clés de tour.

2.1.1 Principe des attaques statistiques sur le dernier tour

Les principales techniques de cryptanalyse concernant les chiffrements itératifs par blocs sont des attaques du type *diviser pour régner*. Elles visent à retrouver la sous-clé $k^{(r)}$ utilisée dans la dernière itération de l'algorithme par la connaissance d'un certain nombre de couples clair-chiffré. On obtient ainsi une information sur la clé de chiffrement utilisée. Le restant $\mathbf{k} = (k^{(1)},k^{(2)},\ldots,k^{(r-1)}) \in \mathcal{K}^{r-1}$ de la clé ou bien la clé maître peut être retrouvé soit par application itérative de cette technique aux chiffrements obtenus successivement en retirant le dernier tour, soit par une recherche exhaustive sur les bits inconnus restant (ou bien sûr par une combinaison des deux techniques).

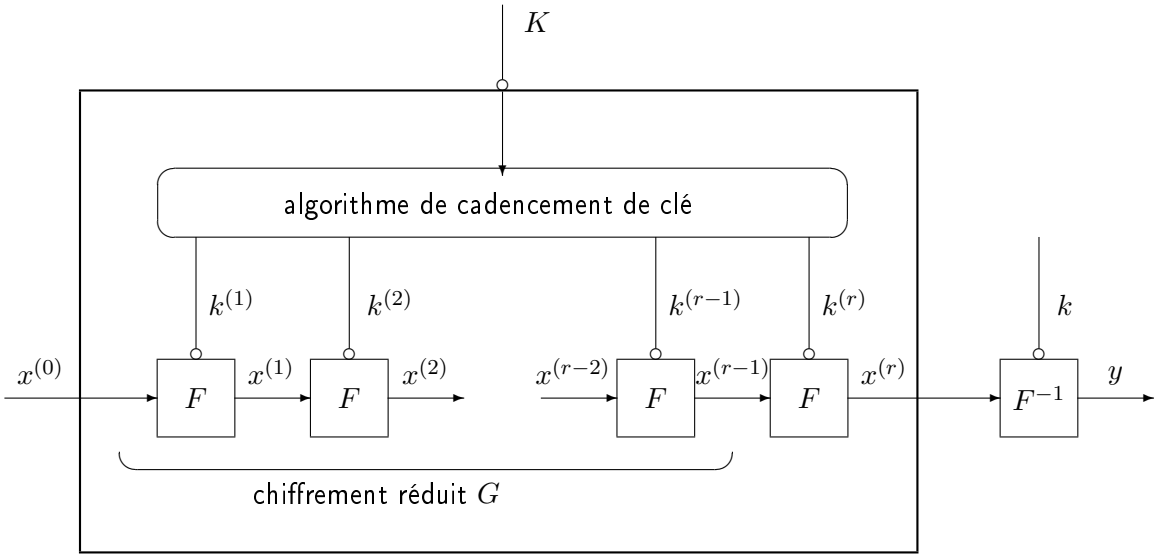


FIG. 2.1 – *Principe d'une attaque sur le dernier tour*

Les attaques statistiques sur le dernier tour exploitent le fait que la fonction itérée $F : (k,x) \mapsto y = F_k(x)$ d'un chiffrement itératif par blocs est généralement *cryptographiquement faible*. Cela signifie qu'une relation sur un ensemble d'entrées $\mathcal{M}_0 \subset \mathbf{F}_2^n$ peut conduire à l'observation de statistiques non-uniformes sur l'ensemble de sorties correspondant $\{F_k(x), x \in \mathcal{M}_0\}$. L'idée est de suivre l'évolution de telles caractéristiques au cours des itérations afin de pouvoir définir la distribution des sorties, après un certain nombre d'itérations, correspondant à une distribution donnée des entrées. On applique ce principe pour une attaque sur le dernier tour en recherchant de telles relations sur le *chiffrement réduit*, c'est-à-dire la fonction $G_{\mathbf{k}}$ composée des $(r-1)$ premières itérations de la fonction de chiffrement originale :

$$G_{\mathbf{k}} = F_{k^{(r-1)}} \circ \cdots \circ F_{k^{(1)}}.$$

On rappelle que $\mathbf{k} = (k^{(1)}, \ldots, k^{(r-1)}) \in \mathcal{K}^{r-1}$ détermine un ensemble de permutations sur $\mathbf{F}_2^n$. Si on parvient à distinguer une telle famille de permutations d'un ensemble de permutations tirées aléatoirement, c'est-à-dire à déterminer une grandeur liée au chiffrement réduit $G_{\mathbf{k}}$ mais qui soit indépendante de la valeur de la clé $\mathbf{k}$ et qui ne soit pas uniformément distribuée, on a défini une procédure qui fournit de l'information sur la clé $k^{(r)}$.

2.1.2 Les distingueurs

De manière formelle, un distingueur est une machine de Turing probabiliste avec oracle qui fournit une réponse binaire après un certain nombre de requêtes. Nous allons essayer d'utiliser cette notion de manière informelle et intuitive pour modéliser les attaques sur le dernier tour. Dans notre cas, le distingueur $\mathcal{D}$ doit répondre avec une certaine probabilité à la question : une permutation sur $\mathbf{F}_2^n$, dont on peut obtenir un nombre limité de valeurs par un oracle, appartient-elle à une famille de permutations $\mathcal{P}$ dont on connaît les caractéristiques ? Le caractère aléatoire provient du fait qu'on ne sait pas quelle permutation l'oracle implémente : soit $f \in \mathcal{P}$, soit $\pi \in_R \mathrm{S}_{2^n}$, où $\in_R$ signifie que π est tirée aléatoirement dans S_{2^n}.

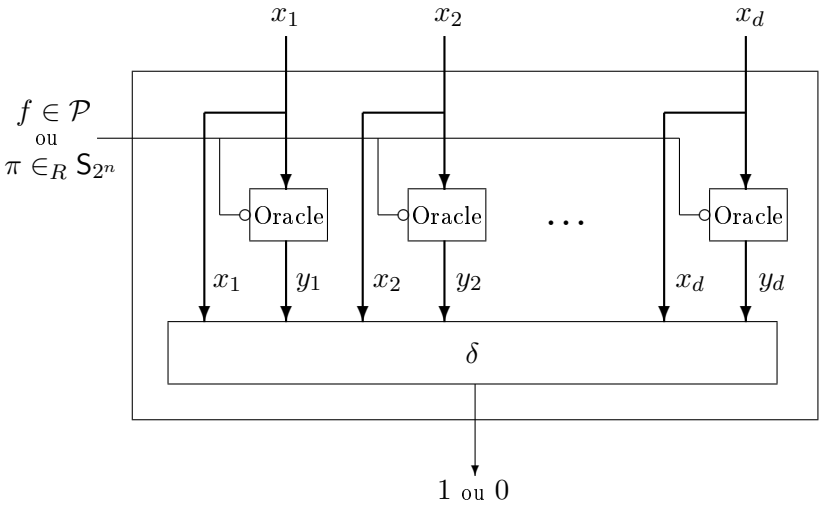


FIG. 2.2 – *Schéma d'un distingueur*

Définition 2.1 Distingueur non-adaptatif à d requêtes
Soient S_{2^n} *l'ensemble des permutations de* $\mathbf{F}_2^n$, $\mathcal{P}$ *un sous-ensemble de* S_{2^n}. *Un* distingueur *de* $\mathcal{P}$ *relativement à un sous-ensemble* $\{x_1, \ldots, x_d\}$ *de* $\mathbf{F}_2^n$, *disposant d'un oracle implémentant une permutation* O, *est une fonction*

$$\begin{array}{rccl} \mathcal{D}: & (\mathbf{F}_2^n)^d & \to & \mathbf{F}_2 \\ & (x_1, \ldots, x_d) & \mapsto & \delta(x_1, \ldots, x_d, \mathsf{O}(x_1), \ldots, \mathsf{O}(x_d)) \end{array}$$

où δ *est une fonction de test, telle qu'appliquée à une permutation* $f \in \mathcal{P}$, *il existe* $\varepsilon > 0$ *pour lequel l'*avantage, $\mathsf{Adv}_{\mathcal{D}}$, *du distingueur vérifie :*

$$\mathsf{Adv}_{\mathcal{D}}(f,\pi) = \Big| \mathbf{P}\left[\mathcal{D}(x_1, \ldots, x_d) = 1 \,\middle|\, \mathsf{O} = f \in \mathcal{P}\right] - \mathbf{P}\left[\mathcal{D}(x_1, \ldots, x_d) = 1 \mid \mathsf{O} = \pi \in_R \mathrm{S}_{2^n}\right] \Big| > \varepsilon.$$

Les grandeurs qui déterminent l'avantage d'un distingueur portent les dénominations suivantes :

- probabilité d'*erreur de type* I notée $\alpha = \mathbf{P}\left[\mathcal{D}(x_1,\ldots,x_d) = 0 \middle|\ \mathsf{O} = f \in \mathcal{P}\right]$;
- probabilité d'*erreur de type* II notée $\beta = \mathbf{P}\left[\mathcal{D}(x_1,\ldots,x_d) = 1 \middle|\ \mathsf{O} = \pi \in_R \mathrm{S}_{2^n}\right]$.

L'avantage s'écrit alors : $\mathsf{Adv}_{\mathcal{D}}(f,\pi) = |\, 1 - (\alpha + \beta)\,|$. On pourra supposer que $\mathsf{Adv}_{\mathcal{D}}(f,\pi) = 1 - (\alpha + \beta)$ car, dans le cas contraire, il suffit de considérer le distingueur complémentaire (c'est-à-dire celui qui répond 0 quand le précédent répondait 1) pour retrouver cette situation. La *vraisemblance* des hypothèses $\{\mathsf{O} = f \in \mathcal{P}\}$ ou $\{\mathsf{O} = \pi \in_R \mathrm{S}_{2^n}\}$ relativement à la valeur de $\mathcal{D}(x_1,\ldots,x_d)$ est la probabilité de l'événement $\{\mathcal{D}(x_1,\ldots,x_d) = 1\}$ sous une des hypothèses $\{\mathsf{O} = f \in \mathcal{P}\}$ ou $\{\mathsf{O} = \pi \in_R \mathrm{S}_{2^n}\}$; elle vaut respectivement $(1-\alpha)$ et β. On définit ainsi le *rapport de vraisemblance* noté $\mathrm{lr}_{\mathcal{D}}$ comme le rapport de ces deux probabilités :

$$\begin{aligned}\mathrm{lr}_{\mathcal{D}} &= \frac{1-\alpha}{\beta} \\ &= 1 + \frac{\mathsf{Adv}_{\mathcal{D}}(f,\pi)}{\beta}.\end{aligned}$$

On définit également deux autres types d'erreurs :

- la non-détection, c'est-à-dire le fait de répondre que la permutation $f \in \mathcal{P}$ n'appartient pas à $\mathcal{P}$: $\{\mathcal{D}(x_1,\ldots,x_d) = 0\}$ et $\{\mathsf{O} = f \in \mathcal{P}\}$
- la fausse alarme, c'est-à-dire le fait de répondre que la permutation est $f \in \mathcal{P}$ alors qu'elle est tirée aléatoirement dans S_{2^n} : $\{\mathcal{D}(x_1,\ldots,x_d) = 1\}$ et $\{\mathsf{O} = \pi \in_R \mathrm{S}_{2^n}\}$.

La probabilité d'erreur totale du distingueur est alors égale à :

$$\mathbf{P}_{\mathrm{err}} = \mathbf{P}[\{\mathcal{D}(x_1,\ldots,x_d) = 0\} \text{ et } \{\mathsf{O} = f \in \mathcal{P}\}] + \mathbf{P}[\{\mathcal{D}(x_1,\ldots,x_d) = 1\} \text{ et } \{\mathsf{O} = \pi \in_R \mathrm{S}_{2^n}\}].$$

Ce que nous pouvons développer grâce aux formules de Bayes :

$$\mathbf{P}_{\mathrm{err}} = \mathbf{P}[\mathcal{D}(x_1,\ldots,x_d) = 0 \mid \mathsf{O} = f \in \mathcal{P}]\ \mathbf{P}[\mathsf{O} = f \in \mathcal{P}] + \mathbf{P}[\mathcal{D}(x_1,\ldots,x_d) = 1 \mid \mathsf{O} = \pi \in_R \mathrm{S}_{2^n}]\ \mathbf{P}[\mathsf{O} = \pi \in_R \mathrm{S}_{2^n}].$$

On sait que l'oracle implémente soit l'une soit l'autre des éventualités de cette alternative. Ainsi $\mathbf{P}[\mathsf{O} = f \in \mathcal{P}] = p_f$ et $\mathbf{P}[\mathsf{O} = \pi \in_R \mathrm{S}_{2^n}] = p_\pi = 1 - p_f$.

En guise d'exemple, si on se place dans le cas où l'oracle ne fournit aucune information a priori, on peut considérer le cas symétrique :

$$\mathbf{P}[\mathsf{O} = f \in \mathcal{P}] = \mathbf{P}[\mathsf{O} = \pi \in_R \mathrm{S}_{2^n}] = \frac{1}{2}.$$

On a donc

$$\mathbf{P}_{\mathrm{err}} = \frac{\alpha}{2} + \frac{\beta}{2}.$$

Ainsi l'avantage du distingueur est égal à

$$\mathsf{Adv}_{\mathcal{D}}(f,\pi) = 1 - 2\ \mathbf{P}_{\mathrm{err}}.$$

L'étude des propriétés de l'ensemble $\mathcal{P}$ permet d'obtenir des informations pour le différencier d'un ensemble tiré aléatoirement dans S_{2^n}. On peut ainsi obtenir des valeurs pour α et

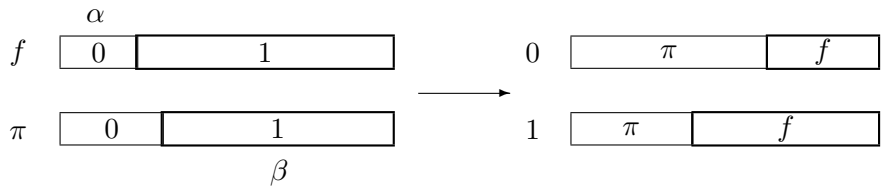


FIG. 2.3 – *Probabilités avant et après l'expérience*

β. Pour l'observateur qui se situe après la réalisation du test avec le distingueur, les valeurs intéressantes sont $\mathbf{P}[\mathsf{O} = \pi \in_R \mathrm{S}_{2^n} \mid \mathcal{D}(x_1,\ldots,x_d) = 0]$ ou $\mathbf{P}[\mathsf{O} = f \in \mathcal{P} \mid \mathcal{D}(x_1,\ldots,x_d) = 1]$. En effet ce sont elles qui permettent de prendre une décision suite au résultat du distingueur (typiquement considérer qu'une clé est valide ou non). L'évaluation de ces probabilités nécessite la connaissance de α et β d'une part et de p_f d'autre part. Considérons le calcul de $\mathbf{P}[\mathsf{O} = f \in \mathcal{P} \mid \mathcal{D}(x_1,\ldots,x_d) = 1]$:

$$\mathbf{P}[\mathsf{O} = f \in \mathcal{P} \mid \mathcal{D}(x_1,\ldots,x_d) = 1] = \frac{(1-\alpha)\times p_f}{\mathbf{P}[\mathcal{D}(x_1,\ldots,x_d) = 1]}$$
$$\mathbf{P}[\mathsf{O} = \pi \in_R \mathrm{S}_{2^n} \mid \mathcal{D}(x_1,\ldots,x_d) = 1] = \frac{\beta\times(1-p_f)}{\mathbf{P}[\mathcal{D}(x_1,\ldots,x_d) = 1]}$$

On a également

$$\mathbf{P}[\mathsf{O} = f \in \mathcal{P} \mid \mathcal{D}(x_1,\ldots,x_d) = 1] + \mathbf{P}[\mathsf{O} = \pi \in_R \mathrm{S}_{2^n} \mid \mathcal{D}(x_1,\ldots,x_d) = 1] = 1$$

car les deux événements sont exclusifs l'un de l'autre et sont les seuls possibles. On obtient ainsi

$$\mathbf{P}[\mathcal{D}(x_1,\ldots,x_d) = 1] = (1-\alpha)p_f + \beta(1-p_f),$$

ce qui nous permet d'écrire :

$$\mathbf{P}[\mathsf{O} = f \in \mathcal{P} \mid \mathcal{D}(x_1,\ldots,x_d) = 1] = \frac{(1-\alpha)p_f}{(1-\alpha)p_f + \beta(1-p_f)}.$$

Ainsi le rapport des probabilités après l'observation du résultat du distingueur est égal à :

$$\frac{\mathbf{P}[\mathsf{O} = f \in \mathcal{P} \mid \mathcal{D}(x_1,\ldots,x_d) = 1]}{\mathbf{P}[\mathsf{O} = \pi \in_R \mathrm{S}_{2^n} \mid \mathcal{D}(x_1,\ldots,x_d) = 1]} = \frac{1-\alpha}{\beta} \times \frac{p_f}{1-p_f}$$

Il correspond donc au *rapport de vraisemblance* lorsque $p_f = p_\pi = \frac{1}{2}$. On retrouve ainsi le fait que l'approche baysienne et l'approche fréquentiste classique conduisent aux mêmes valeurs lorsque la distribution de probabilités a priori est uniforme.

On peut alors exprimer $\mathbf{P}[\mathsf{O} = f \in \mathcal{P} \mid \mathcal{D}(x_1,\ldots,x_d) = 1]$ en fonction de ce rapport :

$$\mathbf{P}[\mathsf{O} = f \in \mathcal{P} \mid \mathcal{D}(x_1,\ldots,x_d) = 1] = \frac{1}{1 + \frac{1-p_f}{p_f}\frac{1}{\mathrm{lr}_\mathcal{D}}}.$$

Ainsi, si $\mathrm{lr}_\mathcal{D} = \frac{1}{p_f} - 1$, la probabilité $\mathbf{P}[\mathsf{O} = f \in \mathcal{P} \mid \mathcal{D}(x_1,\ldots,x_d) = 1]$ ne nous est d'aucune aide pour le choix d'une éventualité pour O. Dans le cas où $p_f = \frac{1}{2}$, cela correspond naturellement au cas où l'avantage, $\mathsf{Adv}_\mathcal{D}(f,\pi) = 1 - (\alpha+\beta)$ est nul. On en déduit également que pour un

avantage donné, on peut augmenter la valeur du rapport de vraisemblance si on parvient à diminuer la valeur de β.

La décision d'opter pour l'une ou l'autre des éventualités de l'alternative sera bien sûr plus fiable si on dispose d'un nombre d'observations suffisamment grand. On souhaite donc répéter l'expérience plusieurs fois. On cherche ainsi à amplifier l'avantage en soumettant au distingueur élémentaire un nombre d de requêtes puis en itérant N fois ce processus de manière indépendante. On définit ainsi la notion de *distingueur itératif d'ordre d et de complexité N* [Vau99].

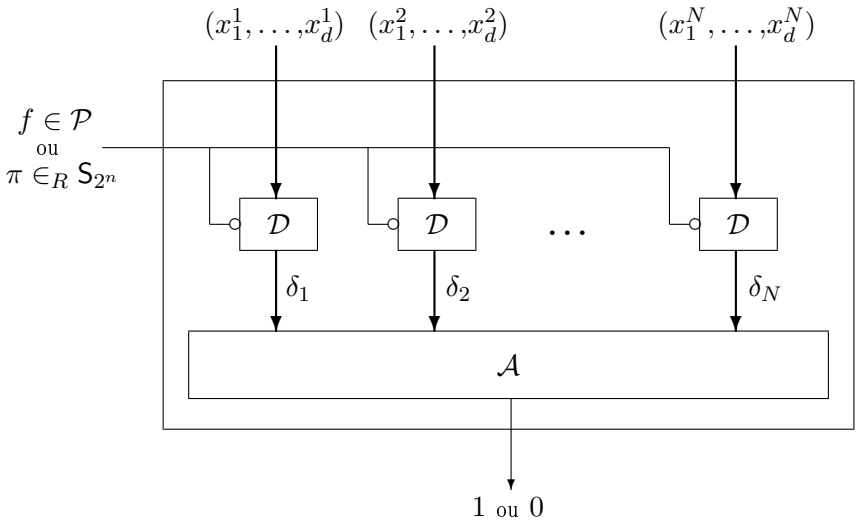


FIG. 2.4 – *Schéma d'un distingueur non-adaptatif itératif d'ordre d et de complexité N*

Définition 2.2 Distingueur non-adaptatif itératif d'ordre d et de complexité N
Soient d et N deux entiers. Un distingueur non-adaptatif itératif d'ordre d et de complexité N *de $\mathcal{P}$ est défini à partir d'un distingueur élémentaire $\mathcal{D}$ par la donnée*

- *d'une loi de probabilités μ sur $(\mathbf{F}_2^n)^d$;*
- *d'une fonction d'acceptation $\mathcal{A}$ de $\mathbf{F}_2^N$ vers $\mathbf{F}_2$.*

Le procédure qui définit le distingueur $\mathcal{D}_N^d$ est alors la suivante :

1. *Pour i de 1 à N faire*
   (a) *sélectionner $\mathbf{x} \leftarrow (x_1,\ldots,x_d)$ suivant la loi μ ;*
   (b) *évaluer $\delta_i \leftarrow \mathcal{D}(\mathbf{x})$;*
2. *Retourner $\mathcal{A}(\delta_1,\ldots,\delta_N)$.*

Plusieurs paramètres entrent ainsi en jeu pour le distingueur itératif ; outre les probabilités d'erreur qui définissent l'avantage et la probabilité d'erreur totale pour le distingueur élémentaire, on doit déterminer la complexité N qui permet d'obtenir un avantage non-négligeable pour le distingueur itératif. Les cryptanalyses sont souvent définies selon ce modèle : par exemple pour la cryptanalyse linéaire, l'ordre du distingueur vaut $d = 1$, pour la cryptanalyse différentielle, $d = 2$ et pour la cryptanalyse différentielle d'ordre supérieur, $d = 2^{\dim(V)}$ où V est le sous-espace vectoriel sur lequel on calcule la différentielle.

2.1.3 Attaques sur le dernier tour

Plaçons-nous maintenant dans le cadre d'une cryptanalyse sur le dernier tour. Supposons qu'il existe un distingueur de la famille de chiffrements réduits

$$\mathcal{G} = \{G_{\mathbf{k}}, \mathbf{k} = (k^{(1)}, \ldots, k^{(r-1)}) \in \mathcal{K}^{r-1}\}.$$

La fonction de tour F étant connue, si nous appliquons ce distingueur à toutes les fonctions

$$\begin{aligned} F_k^{-1} \circ E_K &= F_k^{-1} \circ (F_{k^{(r)}} \circ F_{k^{(r-1)}} \circ \cdots F_{k^{(2)}} \circ F_{k^{(1)}}) \\ &= F_k^{-1} \circ (F_{k^{(r)}} \circ G_{\mathbf{k}}), \quad \forall k \in \mathcal{K}, \end{aligned}$$

le chiffrement réduit $G_{\mathbf{k}}$ obtenu lorsque $k = k^{(r)}$ peut être reconnu parmi les chiffrements résultant des cas où $k \neq k^{(r)}$. Un tel distingueur est modélisé par la figure 2.5. L'oracle implémente une permutation constituée par la composition de la fonction de chiffrement attaquée E_K avec un tour de déchiffrement paramétré par une sous-clé $\hat{k}$ tirée au hasard dans $\mathcal{K}$.

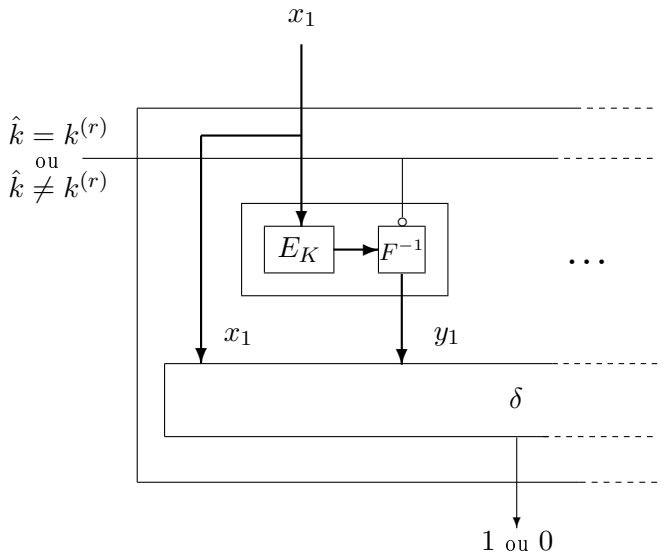


FIG. 2.5 – *Distingueur pour une attaque sur le dernier tour*

En fait, cette formalisation suppose en pratique que la famille de fonctions $F_k^{-1} \circ (F_{k^{(r)}} \circ G_{\mathbf{k}})$ avec $k \neq k^{(r)}$ se comporte comme une famille de permutations tirées aléatoirement ; on se retrouve alors dans le cas de l'alternative simple où il est possible d'évaluer l'avantage. Cette hypothèse s'appuie sur le fait que de telles fonctions étant le résultat du chiffrement réduit suivi de deux itérations de chiffrement supplémentaires, elles se comportent *davantage* comme des permutations aléatoires. C'est l'*hypothèse de répartition aléatoire par fausse clé* (hypothesis of wrong-key randomization) [Har96, Kuk99]. D'autre part, pour appliquer le modèle du distingueur, l'avantage doit être une valeur commune à tous les chiffrements réduits quelle que soit la clé $\mathbf{k}$: c'est l'*hypothèse d'équivalence de clé fixée* (hypothesis of fixed-key equivalence) [Har96, HKM95, Kuk99] pour la cryptanalyse linéaire ou l'*hypothèse d'équivalence stochastique* (hypothesis of stochastic equivalence) pour la cryptanalyse différentielle [LMM91]. En pratique, cela signifie qu'on considère que la probabilité $\mathbf{P}[\mathcal{D}(x_1, \ldots, x_d) = 1 \mid \hat{k} = k^{(r)}]$ à partir de laquelle est construite le distingueur est égale à l'espérance sur l'ensemble des

clés $\mathbf{k}$ du chiffrement réduit des probabilités que $\mathcal{D}$ retourne 1 lorsque la permutation est un chiffrement réduit $G_{\mathbf{k}}$; Pour que l'hypothèse soit valable, il faut également que la variance soit faible. Ainsi le distingueur doit retourner la même réponse pour presque tous les chiffrements réduits de la famille $\mathcal{G}$. Cette hypothèse fournit une bonne approximation lorsque le chiffrement ne comporte pas de composantes fortement dépendantes de la clé comme pour les chiffrements où la clé de tour est insérée par addition, *i.e.* $F_k(x) = F(x + k)$, ce qui est le cas de la majorité des chiffrements connus tels que le DES ou l'AES. On calcule l'avantage en comparant cette valeur à la probabilité $\mathbf{P}[\mathcal{D}(x_1,\ldots,x_d) = 1 \mid \hat{k} \neq k^{(r)}]$ qu'on considère être l'espérance sur l'ensemble des permutations de $\mathbf{F}_2^n$ que $\mathcal{D}$ retourne 1 lorsque la permutation est tirée aléatoirement dans S_{2^n}.

Il est clair que ce qui différencie les types d'attaques réside dans la définition de la fonction de test δ. Le choix de cette fonction repose sur une étude détaillée des caractéristiques du chiffrement qui permet de relever les éléments qui le différencient suffisamment d'une permutation aléatoire de S_{2^n}. On entend par suffisamment le fait que l'avantage du distingueur qu'on peut élaborer à partir de cette étude conduise à une probabilité de succès significative pour une complexité raisonnable (nettement inférieure à la recherche exhaustive pour que l'attaque soit considérée comme valable). Dans le cas d'un distingueur itératif, la fonction d'acceptation $\mathcal{A}$ influence également la probabilité de succès. Dans la plupart des cas, elle correspond au test qui compare la valeur d'une variable compteur U avec un seuil τ. Si on modélise la sortie de la fonction de test δ par une variable aléatoire Δ, celle-ci suit une loi de Bernoulli de paramètre $(1-\alpha)$ si $\hat{k} = k^{(r)}$ et β si $\hat{k} \neq k^{(r)}$ et la variable aléatoire U suit une loi binomiale de paramètre $(N,\, 1-\alpha)$ et $(N,\, \beta)$ respectivement. Ainsi un algorithme générique d'attaque sur le dernier tour utilisant un distingueur non-adaptatif itératif d'ordre d et de complexité N peut être décrit de la manière suivante :

Algorithme 2.3 Attaque sur la sous-clé du dernier tour avec un distingueur $\mathcal{D}_N^d$
ENTRÉE: *Un nombre N de d-uplets de clairs $\mathbf{x_1} = (m_1^1,\ldots,m_d^1),\ldots,\mathbf{x_N} = (m_1^N,\ldots,m_d^N)$ et les N d-uplets de chiffrés correspondant $(c_1^1,\ldots,c_d^1),\ldots,(c_1^N,\ldots,c_d^N)$.*
SORTIE: *Un ensemble de candidats pour $k^{(r)}$, clé du dernier tour.*

Pour tout $\hat{k} \in \mathcal{K}$ faire

1. *Initialiser un compteur $\mathsf{u}[\hat{k}]$;*
2. *Pour i de 1 à N faire*
   (a) $\mathbf{y}_i \leftarrow \left(F_{\hat{k}}^{-1}(c_1^i),\ldots,F_{\hat{k}}^{-1}(c_d^i)\right)$;
   (b) $\mathsf{u}[\hat{k}] \leftarrow \mathsf{u}[\hat{k}] + \delta(\mathbf{x}_i,\, \mathbf{y}_i)$;
3. *Si $\mathsf{u}[\hat{k}] \geq \tau$ alors retourner $\hat{k}$.*

Cet algorithme retourne un ensemble de clés pour lesquelles la valeur du compteur est supérieure à un certain seuil. Il est intéressant de remarquer que l'algorithme prend une décision binaire à la suite de la fonction d'acceptation (retenir ou non une clé) mais que son déroulement nous permettrait d'obtenir des résultats plus élaborés. En effet la décision porte sur le rapport de vraisemblance (ou sur le signe du logarithme de ce rapport):

$$\mathrm{lr}_{\mathcal{D}_N^d} = \frac{\mathbf{P}[\mathsf{U} \geq \tau \mid \hat{k} = k^{(r)}]}{\mathbf{P}[\mathsf{U} \geq \tau \mid \hat{k} \neq k^{(r)}]} = \frac{\mathbf{P}[\hat{k} = k^{(r)} \mid \mathsf{U} \geq \tau]}{\mathbf{P}[\hat{k} \neq k^{(r)} \mid \mathsf{U} \geq \tau]},$$

si $\mathbf{P}[\hat{k} = k^{(r)}] = \mathbf{P}[\hat{k} \neq k^{(r)}] = \frac{1}{2}$. Il faut donc au préalable se fixer un seuil d'acceptation τ. On peut ainsi envisager de retourner une unique clé qui est celle pour laquelle la valeur $\mathsf{u}[\hat{k}] - \tau$ est maximale. On peut également retourner une liste de clés ordonnée suivant la valeur de $\mathsf{u}[\hat{k}] - \tau$. Ces deux procédures d'acceptation sont illustrées par le second et troisième algorithme de Matsui [Mat93, Mat94] pour la cryptanalyse linéaire du DES. On peut également définir un autre rapport de vraisemblance noté $\mathrm{lr}_{\mathcal{D}_N^d}(u)$ qui permet de prendre une décision pour une valeur de compteur donnée :

$$\mathrm{lr}_{\mathcal{D}_N^d}(u) = \frac{\mathbf{P}[\mathsf{U} = u \mid \hat{k} = k^{(r)}]}{\mathbf{P}[\mathsf{U} = u \mid \hat{k} \neq k^{(r)}]} = \frac{(1-\alpha)^u \alpha^{N-u}}{\beta^u (1-\beta)^{N-u}}.$$

On peut ainsi envisager de retourner une unique clé qui est celle dont la valeur du compteur maximise le rapport de vraisemblance $\mathrm{lr}_{\mathcal{D}_N^d}(u)$. Ce rapport de vraisemblance est lié à la quantité définie sous le nom de *rapport signal à bruit* (SNR pour Signal to Noise Ratio) par Biham et Shamir [BS93]. On peut également retourner une liste ordonnée d'un ensemble de clés pour lesquelles le rapport est supérieur à un seuil. Il est bien sûr théoriquement possible de retourner la liste ordonnée de toutes les clés rangées dans l'ordre décroissant du rapport de vraisemblance, ce qui permettrait de conserver le maximum d'information sur le déroulement de l'attaque ; néanmoins, la quantité de données à mémoriser à cette fin est trop importante pour envisager l'appliquer en pratique. Il serait intéressant en revanche de déterminer une grandeur tirant partie du maximum d'information disponible au cours et à l'issue du déroulement de l'algorithme et qui permettrait d'optimiser la procédure de choix. Notre présentation se restreint au cas où l'attaque effectue une recherche exhaustive sur la clé du dernier tour. Toutefois, certains distingueurs utilisent une fonction δ qui ne fait intervenir qu'une portion de la clé $k^{(r)}$ (usuellement quelques bits). Dans ce cas, l'attaque consiste à diviser l'espace des sous-clés en classes d'équivalence et à déterminer la classe à laquelle appartient $k^{(r)}$. Dans ce type de situations, le nombre de bits de la sous-clé retrouvés par l'attaque est donc plus faible, mais la complexité en temps est réduite.

Une fois la famille de sous-clés candidates obtenue après une instance de l'attaque, on peut exploiter cette information dans d'autres attaques (autres distingueurs, recherche exhaustive, etc.) pour rechercher le restant $\mathbf{k}$ de la clé de chiffrement. La vérification finale s'effectue sur quelques couples clairs-chiffrés. L'attaque nécessite N d-uplets de clairs et de chiffrés. Son coût moyen est $\#\mathcal{K}(NdT_{F^{-1}} + T_{\mathcal{A}})$, où $\#\mathcal{K}$ est le cardinal de l'espace des clés de tour, $T_{\mathcal{A}}$ est le coût moyen de la fonction d'acceptation $\mathcal{A}$ et $T_{F^{-1}}$ est le nombre moyen d'opérations nécessaires à l'évaluation de F_k^{-1}. Le nombre de requêtes N à soumettre au distingueur itératif dépend de la probabilité d'erreur admissible, $\mathbf{P}_{\mathrm{err}}$ et donc du rapport de vraisemblance :

$$\mathrm{lr}_{\mathcal{D}} = \frac{1-\alpha}{\beta} = 1 + \frac{\mathsf{Adv}_{\mathcal{D}}(G_{\mathbf{k}}, \pi)}{\beta}.$$

On cherche typiquement des valeurs de probabilité de non-détection de l'ordre de 0,01 et une probabilité de fausse alarme faible. On a ainsi [Gil97, Min02] :

$$\begin{cases} \text{si } \frac{\mathsf{Adv}_{\mathcal{D}}(G_{\mathbf{k}}, \pi)}{\beta} \gg 1 & \text{alors } N = \mathcal{O}\left(\frac{1}{1-\alpha}\right) \\ \text{si } \frac{\mathsf{Adv}_{\mathcal{D}}(G_{\mathbf{k}}, \pi)}{\beta} \ll 1 & \text{alors } N = \mathcal{O}\left(\frac{\beta(1-\beta)}{(1-\alpha-\beta)^2}\right) \end{cases}$$

Le premier cas correspond à la cryptanalyse différentielle où $(1-\alpha) = \frac{1}{2^n - 1} + \varepsilon$ avec n la taille de bloc considérée, et $\beta = \frac{1}{2^n - 1}$. Le deuxième cas correspond à la cryptanalyse linéaire où $\beta = \frac{1}{2}$ et $\alpha = \frac{1}{2} - \varepsilon$. Le nombre de requêtes au distingueur est alors de l'ordre de $\frac{1}{\varepsilon^2}$.

Les cas extrêmes sont ceux d'événements soit certains soit impossibles pour le chiffrement réduit. On peut se ramener au cas où $\alpha = 1$. On voit alors que $\mathrm{lr}_{\mathcal{D}_N^d}(u) \geq 1$ si et seulement si la valeur du compteur correspond au nombre de requêtes effectuées N. On a donc $\tau = N$ dans ce cas. Le rapport signal sur bruit s'écrit

$$\begin{aligned}\mathrm{lr}_{\mathcal{D}_N^d}(N) &= \frac{\mathbf{P}[\hat{k} = k^{(r)} | U = N]}{\mathbf{P}[\hat{k} \neq k^{(r)} | U = N]} \\ &= \frac{\mathbf{P}[U = N | \hat{k} = k^{(r)}]}{\mathbf{P}[U = N | \hat{k} \neq k^{(r)}]} \\ &= \frac{1}{\beta^N}.\end{aligned}$$

Le nombre de requêtes N à effectuer doit donc être de l'ordre de $\frac{1}{-\log(\beta)}$. Dans le cas où $\beta = 1 - \varepsilon$ (ce qui est souvent le cas en pratique, car le biais entre le chiffrement réduit et une permutation aléatoire est très faible), on obtient $N = \mathcal{O}(\varepsilon^{-1})$. Ce cas extrême correspond à la situation de la cryptanalyse différentielle impossible et de nombreuses cryptanalyses différentielles tronquées ou d'ordre supérieur. Par ailleurs, dans cette situation, on peut aisément diminuer la complexité de l'attaque en diminuant le nombre de requêtes à l'oracle grâce à un distingueur dit séquentiel [Jun05]. Cette procédure est particulièrement adaptée au cas précédent car on dispose d'une règle immédiate de rejet ou d'acceptation du candidat. Ainsi pour la même probabilité de succès, on peut diminuer la complexité de l'attaque. On peut également étendre son utilisation quand les événements sont hautement improbables ou hautement probable bien que ni certains ni impossibles. On peut également trouver un exemple d'une telle modélisation dans [Jun05].

Définition 2.4 Distingueur non-adaptatif séquentiel limité à N requêtes
Soit N un entier. Un distingueur non-adaptatif séquentiel limité à N requêtes *de $\mathcal{P}$, appliqué à une permutation f de S_{2^n} est défini par la donnée*

- *de fonctions $\mathcal{A}_j$ de $\mathbf{F}_2^j$ vers $\mathbf{F}_2$, $1 \leq j \leq N$;*
- *de fonctions $\mathcal{R}_j$ de $\mathbf{F}_2^j$ vers $\mathbf{F}_2$, $1 \leq j \leq N$.*

Le procédure qui définit le distingueur est alors la suivante :

1. *$j \leftarrow 1$;*
2. *Tant que $j \leq N - 1$*
   (a) *sélectionner de manière non-adaptative x_j ;*
   (b) *calculer $y_j = \mathsf{f}(x_j)$;*
      i. *Si $\mathcal{A}_j(y_1, \ldots, y_j) = 1$ alors retourner 1 et s'arrêter ;*
      ii. *Sinon, si $\mathcal{R}_j(y_1, \ldots, y_j) = 0$ alors retourner 0 et s'arrêter ;*
   (c) *$j \leftarrow j + 1$;*
3. *Retourner $\mathcal{A}_N(y_1, \ldots, y_N)$.*

On retrouve ainsi l'idée qu'outre l'avantage, la valeur de α ou de β influe sur la complexité en nombre de requêtes au distingueur.

On détermine généralement un distingueur pour $G_{\mathbf{k}}$ à partir d'un écart à la loi uniforme dans le comportement de la permutation $G_{\mathbf{k}}$ qui est indépendant de $\mathbf{k}$. Ainsi, la cryptanalyse différentielle, introduite par Biham et Shamir en 1991 [BS91], s'applique dès que pour tout

$\mathbf{k} \in \mathcal{K}^{r-1}$, la fonction $G_{\mathbf{k}}$ possède une dérivée, $D_a G_{\mathbf{k}} : m \mapsto G_{\mathbf{k}}(m+a)+G_{\mathbf{k}}(m)$, dont la sortie ne suit pas la loi uniforme. La cryptanalyse linéaire, introduite en 1991 par Tardy-Corfdir et Gilbert [TCG91] et appliquée au DES en 1993 par Matsui [Mat93, Mat94] exploite l'existence d'une relation linéaire entre les bits du texte clair m et ceux de $G_{\mathbf{k}}(m)$, qui soit satisfaite avec la probabilité la plus élevée possible. Nous décrivons ces attaques de manière plus détaillée dans la suite. Nous verrons en particulier que la formalisation de ces deux attaques [LMM91, HKM95] montre que leur succès dépend directement de certaines propriétés de la fonction itérée F_k. On a ainsi pu énoncer des critères précis de conception des algorithmes itératifs par blocs résistants à ces deux attaques. En effet, l'utilisation d'une fonction itérée F_k presque parfaitement non-linéaire [NK93] et presque courbe [CV95] garantit une résistance optimale aux cryptanalyses différentielle et linéaire. Ces résultats ont notamment été utilisés dans l'AES [FIP01], dans lequel la fonction de substitution satisfait des propriétés similaires.

2.1.4 Cas des chiffrements de Feistel

La structure de schéma de Feistel fournit une simplification préalable à toute détermination de biais. En effet, la première caractéristique que l'on peut souligner dans un chiffrement de Feistel à r tours, est que la structure qui coupe les blocs en deux moitiés pour n'en modifier qu'une permet, en considérant une attaque à clair connu, de réduire le nombre de passages dans les fonctions de tour. Pour attaquer un chiffrement de Feistel à r tours, il suffit d'être capable de distinguer la fonction qui, à tout bloc de clair associe la moitié droite de la sortie du tour $(r-2)$, notée R_{r-2}, d'une fonction aléatoire. En effet, comme on peut le voir sur la figure 1.12 et sachant qu'en général on ne permute pas les deux moitiés du chiffré au dernier tour, on a $R_r = R_{r-1}$ et

$$L_r = L_{r-1} + F(R_{r-1},k^{(r)}), \text{ c'est-à-dire } L_{r-1} = L_r + F(R_r,k^{(r)}).$$

Comme $L_{r-1} = R_{r-2}$ on a $R_{r-2} = L_r + F(R_r,k^{(r)})$. Si on parvient à distinguer la fonction $(L_0,R_0) \mapsto R_{r-2}$ d'une fonction aléatoire, on obtient alors une manière de tester les clés potentielles du dernier tour. Ceci conduit à chercher des caractéristiques de R_{r-2}. Si, de plus, on considère une attaque à clair choisi qui ne fait intervenir que des blocs de clair dont la partie droite, R_0, est constante, il suffit alors de ne prendre en compte que $(r-3)$ passages dans la fonction de tour.

2.2 Cryptanalyse linéaire

Bien que la présentation de ce premier exemple de cryptanalyse statistique ne respecte pas l'ordre chronologique habituel, elle permet de mettre en évidence une certaine évolution logique entre les attaques.

2.2.1 Principe de l'attaque

Le principe de la cryptanalyse linéaire repose sur l'approximation de la fonction itérée F_k par une fonction affine. Cela signifie qu'on cherche une expression de la forme :

$$a \cdot F_k(x) = b \cdot x \oplus c \cdot k, \quad a, b \in \mathbf{F}_2^n, c \in \mathbf{F}_2^\ell,$$

qui soit vérifiée pour une proportion significative de vecteurs $x \in \mathbf{F}_2^n$. En chaînant de telles équations, on aboutit à une approximation linéaire de $(r-1)$ tours de l'algorithme de la forme :

$$a \cdot x^{(0)} \oplus b \cdot x^{(r-1)} \oplus c \cdot (k_1, \ldots, k_{r-1}) = 0,$$

(les vecteurs a, b, c sont bien sûrs différents de ceux de l'équation précédente) c'est-à-dire

$$a \cdot x \oplus b \cdot G_{\mathbf{k}}(x) = c \cdot \mathbf{k}, \quad a, b \in \mathbf{F}_2^n, c \in (\mathbf{F}_2^\ell)^{r-1}.$$

Si cette approximation est valable pour un nombre suffisant de valeurs de $x \in \mathbf{F}_2^n$, on peut appliquer un distingueur entre deux sources aléatoires binaires, l'une suivant une loi uniforme (probabilité $\frac{1}{2}$) liée à une permutation aléatoire et l'autre suivant une loi de Bernoulli de probabilité $\frac{1}{2} + \varepsilon$, liée au chiffrement réduit. De nombreuses variantes et améliorations de l'attaque existent, on peut en trouver une présentation détaillée dans [Jun05]. On peut également mentionner des variantes exploitant notamment l'utilisation simultanée de plusieurs approximations linéaires [KR94, BCQ04]. Nous ne présentons ici que le principe, c'est-à-dire la forme de la fonction de test δ utilisée dans le distingueur pour une attaque de la forme décrite au paragraphe 2.1.3. Ainsi pour une cryptanalyse linéaire la fonction de test δ pour une relation linéaire de paramètres (a, b) est définie par :

$$\begin{array}{rccc} \delta_{(a,b)} \ : & \mathbf{F}_2^n \times \mathbf{F}_2^n & \rightarrow & \mathbf{F}_2 \\ & (x, y) & \mapsto & a \cdot x \oplus b \cdot y, \end{array}$$

et le distingueur linéaire élémentaire peut être modélisé par $\mathcal{D}(x) = a \cdot x \oplus b \cdot \mathsf{O}(x)$.

Déterminer la proportion de valeurs de $x \in \mathbf{F}_2^n$ pour lesquelles une combinaison linéaire des composantes de la fonction de chiffrement réduite $\varphi_b \circ G_{\mathbf{k}}$ et une approximation affine $(\varphi_a + c)$ coïncident est équivalent à déterminer la distance de $\varphi_b \circ G_{\mathbf{k}}$ à $(\varphi_a + c)$ ou de la même manière à calculer la corrélation entre $\varphi_b \circ G_{\mathbf{k}}$ et l'approximation affine. On est donc amené à considérer la transformée de Walsh du chiffrement réduit. Ainsi, déterminer la meilleure approximation revient à déterminer le coefficient d'amplitude la plus élevée dans le spectre de Walsh de $G_{\mathbf{k}}$.

Définition 2.5 *Le* spectre de Walsh *d'une fonction vectorielle F de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$ est composée des spectres de Walsh des fonctions booléennes $\varphi_\beta \circ F : x \mapsto \beta \cdot F(x)$, $\beta \in \mathbf{F}_2^m \setminus \{0\}$. Il correspond ainsi à l'ensemble*

$$\{\mathcal{F}(\varphi_\beta \circ F + \varphi_\alpha),\ \beta \in \mathbf{F}_2^m \setminus \{0\},\ \alpha \in \mathbf{F}_2^n\}\ .$$

Le coefficient de Walsh d'amplitude maximale est noté $\mathcal{L}(F)$ et la non-linéarité de F est définie par

$$\mathcal{N}(F) = 2^{n-1} - \frac{\mathcal{L}(F)}{2}.$$

Naturellement, la condition qu'on impose au chiffrement réduit pour résister à une telle attaque est que sa distance à l'ensemble des fonctions affines doit être la plus grande possible. Cela signifie que la non-linéarité de la fonction doit être maximale. Plus précisément le biais ε exploité dans l'attaque, *i.e.* la distance entre les deux probabilités testées par le distingueur est donnée par

$$\varepsilon = \frac{\mathcal{L}(G_{\mathbf{k}})}{2^{n+1}}.$$

Le nombre de couples clairs-chiffrés nécessaires à la cryptanalyse est donc de l'ordre de $\frac{2^{2n+2}}{\mathcal{L}(G_{\mathbf{k}})^2}$. Des bornes précises sur le nombre de requêtes nécessaires à un distingueur itératif de complexité N utilisé dans une cryptanalyse linéaire afin de rendre significative l'avantage du distingueur sont données dans [Jun05]. Elles correspondent ainsi à la valeur ε^{-2}.

2.2.2 Résistance des chiffrements itératifs à la cryptanalyse linéaire

Une attaque linéaire est possible sur un chiffrement itératif si pour tout $k \in \mathcal{K}$, la fonction itérée F_k de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ admet une approximation par une fonction linéaire avec une probabilité éloignée de $\frac{1}{2}$. Pour que le système résiste à la cryptanalyse linéaire, il faut donc que la fonction F_k soit telle que pour tout $\alpha \in \mathbf{F}_2^n$ et pour tout $\beta \in \mathbf{F}_2^n \setminus \{0\}$

$$\lambda_{F_k} = \max_{\substack{\alpha \in \mathbf{F}_2^n \\ \beta \in \mathbf{F}_2^n \setminus \{0\}}} \left| \#\{x \in \mathbf{F}_2^n,\ \beta \cdot F_k(x) + \alpha \cdot x = 1\} - 2^{n-1} \right| = \frac{\mathcal{L}(F_k)}{2}$$

soit le plus petit possible. En d'autres termes cela signifie que dans les cas où la clé est insérée par addition on cherche à ce que la non-linéarité de F :

$$\mathcal{N}(F) = 2^{n-1} - \frac{\mathcal{L}(F)}{2}$$

soit la plus élevée possible.

Comme la non-linéarité d'une fonction est invariante par composition avec une transformation affine, ce critère porte en fait sur la fonction de substitution (ou de confusion) qui est la seule partie non-linéaire de la fonction de tour. Dans les constructions itératives classiques de chiffrement par blocs (structures de Feistel ou de substitution-permutation), une haute non-linéarité de la fonction de tour (*i.e.*, de la fonction de substitution) suffit à garantir une bonne résistance à l'attaque linéaire. En effet, la non-linéarité de la fonction de tour fournit une borne supérieure sur celle du chiffrement réduit pour ces deux types de schémas, comme il a été démontré dans [Nyb94] pour les structures de Feistel et par exemple dans [Kel05] pour les réseaux de substitution-permutation.

Afin d'assurer une résistance maximale à la cryptanalyse linéaire, on choisit donc une fonction de substitution ayant la plus haute non-linéarité possible.

Proposition 2.6 [Sid71, CV95] *Pour toute fonction* $F : \mathbf{F}_2^n \to \mathbf{F}_2^n$,

$$\mathcal{L}(F) \geq 2^{\frac{n+1}{2}} \ .$$

Lorsque la borne est atteinte, F *est dite* presque courbe *(AB pour almost bent).*

L'utilisation a priori surprenante du terme *presque courbe* pour qualifier les fonctions optimales provient du fait que la borne précédente n'est pas valide lorsque le nombre d'entrées et de sorties de la fonction diffèrent. Pour une fonction F de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$, on a

$$\mathcal{L}(F) \geq 2^{\frac{n}{2}}$$

et les fonctions pour lesquelles la borne est atteinte sont dites *courbes* dans [Nyb91]. Dans le cas où $m = 1$, on retrouve les fonctions booléennes courbes définies à la proposition 1.5.1 page 32.

La valeur minimale de $\mathcal{L}(F)$ lorsque F est une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ ne peut être atteinte que lorsque n est impair. Lorsque n est pair, on connaît des fonctions telles que $\mathcal{L}(F) = 2^{\frac{n}{2}+1}$. On conjecture en fait que cette valeur est la valeur minimale [Dob98c, SP80].

Une propriété particulière des fonctions presque courbes réside dans le profil remarquable de leur spectre de Walsh :

Proposition 2.7 *[CV95] Le spectre de Walsh d'une fonction presque courbe F définie de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ ne contient que les valeurs 0 et $\pm 2^{\frac{n+1}{2}}$.*

Notons enfin que l'utilisation d'approximations linéaires de la fonction $\varphi_b \circ G_{\mathbf{k}}$ dans l'attaque n'est justifiée que par le fait que de telles approximations peuvent être chaînées facilement à travers les tours du chiffrement. Une approximation de degré supérieur pourrait évidemment être utilisée de la même manière, mais de telles approximations restent généralement très difficile à trouver (cf. [KR96] pour le cas des approximations quadratiques et [Cou04] pour le cas d'approximations bilinéaires) et ne vérifient en général pas l'hypothèse d'équivalence de clé fixée.

2.3 Cryptanalyse différentielle

2.3.1 Principe de la cryptanalyse différentielle

La cryptanalyse différentielle, introduite par Biham et Shamir en 1991 [BS91], est une attaque à clair choisi applicable à tous les chiffrements à clé secrète itératifs par blocs. Elle s'applique dès que le chiffrement réduit possède une dérivée dont les valeurs ne sont pas uniformément distribuées.

Définition 2.8 *Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$. Pour tout $a \in \mathbf{F}_2^n$, la* dérivée de F relativement au vecteur *a est la fonction définie pour tout $x \in \mathbf{F}_2^n$ par*

$$D_a F(x) = F(x+a) + F(x) \ .$$

La cryptanalyse différentielle repose sur l'observation de la probabilité que la dérivée du chiffrement réduit relativement à un vecteur donné a, $a \neq 0$ prenne la valeur b, $b \neq 0$.

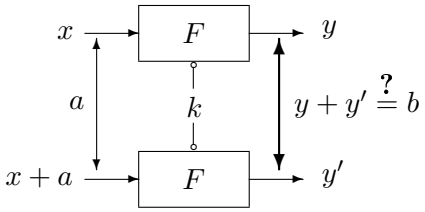


FIG. 2.6 – *Détermination d'une différentielle à un tour*

Définition 2.9 Différentielle
Une différentielle au tour i d'un chiffrement itératif de fonction de tour F paramétrée par une

clé de tour, est un couple $(a,b) \neq (0,0)$ *tel qu'il existe* $x \in \mathbf{F}_2^n$ *pour lequel*

$$D_a\left(F_{k_i} \circ \cdots \circ F_{k_1}\right)(x) = b.$$

La probabilité de la différentielle (a,b) *au tour* i *est définie par*

$$\mathrm{Dp}^{(i)}(a,b) = \mathbf{P}_X[D_a\left(F_{k_i} \circ \cdots \circ F_{k_1}\right)(X) = b].$$

Afin d'élaborer une attaque sur le dernier tour, il faut considérer une différentielle au tour $(r-1)$ (cf. Figure 2.7). Si la probabilité de la différentielle est éloignée de la distribution uniforme, il devient possible de concevoir un distingueur afin de retrouver la clé du dernier tour $k^{(r)}$. La probabilité d'une différentielle dans le cas uniforme correspond à l'espérance des probabilités d'une différentielle sur l'ensemble des permutations de $\mathbf{F}_2^n$. Elle vaut $\frac{1}{2^n-1}$.

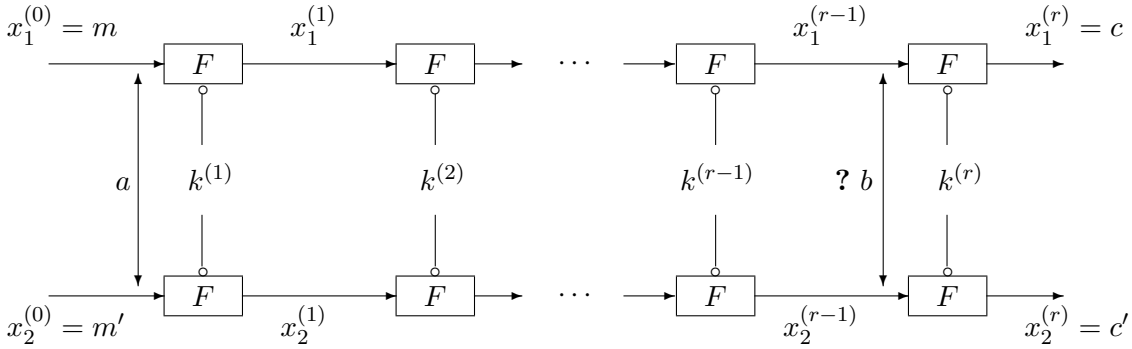


FIG. 2.7 – *Principe de la cryptanalyse différentielle*

La fonction de test δ d'un distingueur pour une différentielle $(a,\ b)$ est donc définie par :

$$\begin{array}{rccc} \delta_{(a,b)} & : & (\mathbf{F}_2^n)^2 \times (\mathbf{F}_2^n)^2 & \to & \mathbf{F}_2 \\ & & (x_1,\, x_2,\, y_1,\, y_2) & \mapsto & \phi_b(y_1 + y_2), \end{array}$$

où ϕ_b est l'indicatrice de b c'est-à-dire la fonction définie par :

$$\begin{array}{rccc} \phi_b & : & \mathbf{F}_2^n & \to & \mathbf{F}_2 \\ & & x & \mapsto & \begin{cases} 1 & \text{si } x = b \\ 0 & \text{sinon.} \end{cases} \end{array}$$

Le distingueur différentiel élémentaire peut alors être modélisé par $\mathcal{D}(x,x+a) = \phi_b\left(D_a\mathsf{O}(x)\right)$.

Un chiffrement est résistant aux attaques différentielles si l'avantage du distingueur différentiel est le plus faible possible. Déterminer la proportion de valeurs de $x \in \mathbf{F}_2^n$ pour lesquelles $D_a G_{\mathbf{k}}(x) = b$ revient à calculer le poids de la fonction booléenne $\phi_b \circ D_a G_{\mathbf{k}}$. Ainsi déterminer la meilleure différentielle au tour $(r-1)$ revient à déterminer $(a,b) \neq (0,0)$ tels que pour tout $\mathbf{k} \in \mathcal{K}^{r-1}$,

$$\#\{x \in \mathbf{F}_2^n,\, G_{\mathbf{k}}(x+a) + G_{\mathbf{k}}(x) = b\} = \mathrm{wt}\left(\phi_b \circ D_a G_{\mathbf{k}}\right)$$

est maximal. Ainsi, le biais ε exploité dans l'attaque, *i.e.* la distance entre les deux probabilités testées par le distingueur différentiel pour un couple $(a,b) \neq (0,0)$ donné est lié aux poids des

fonctions $\phi_b \circ D_a G_{\mathbf{k}}$ et ϕ_b :

$$\begin{aligned}\varepsilon &= \frac{\mathrm{wt}\left(\phi_b \circ D_a G_{\mathbf{k}}\right) - \mathrm{wt}(\phi_b)}{2^n - 1} \\ &= \frac{\mathcal{F}(\phi_b) - \mathcal{F}\left(\phi_b \circ D_a G_{\mathbf{k}}\right)}{2(2^n - 1)}.\end{aligned}$$

Comme nous l'avons vu au paragraphe 2.1.1, la probabilité dans le cas uniforme étant très faible, on se trouve dans le cas où la complexité de l'attaque est de l'ordre de ε^{-1}. Une manière de diminuer en pratique la complexité de l'attaque lorsqu'on considère plusieurs différentielles, consiste à prendre les requêtes x et la différence a en entrée dans un sous-espace vectoriel. On peut ainsi calculer une fois pour toute l'ensemble des chiffrés correspondant aux éléments du sous-espace vectoriel puis considérer ensuite toutes les différentielles de différence en entrée appartenant à ce sous-espace sans avoir à recalculer les chiffrés correspondants.

2.3.2 Résistance des chiffrements itératifs à la cryptanalyse différentielle

La condition sur le distribution des valeurs de la dérivée $D_a G_{\mathbf{k}}$ pour tout $\mathbf{k} \in \mathcal{K}^{r-1}$ permet de définir un critère pour que le chiffrement résiste aux attaques différentielles. Il faut que le poids de $\phi_b \circ D_a G_{\mathbf{k}}$ soit le plus petit possible. Une manière de s'en assurer réside ainsi dans la détermination d'une condition suffisante qui porte sur la fonction de tour : si la fonction de tour vérifie le critère de résistance, alors on veut montrer que le chiffrement résultant présente une résistance optimale à la cryptanalyse différentielle. Nyberg et Knudsen ont ainsi démontré dans le cas des chiffrements de Feistel que cette condition est suffisante [NK93]. Ainsi pour toute sous-clé $k \in \mathcal{K}$, la fonction de tour F_k d'un chiffrement itératif par blocs doit être telle que la valeur

$$\delta_{F_k} = \max_{a,\ b \neq 0} \mathrm{wt}\left(\phi_b \circ D_a F_k\right)$$

soit faible afin que le chiffrement itératif résultant résiste aux attaques différentielles. Le nombre de solutions de l'équation $D_a F_k(x) = b$ étant pair — en effet, si x_0 est une solution, alors $x_0 + a$ est également solution — une borne inférieure de δ_{F_k} est 2.

Proposition 2.10 *[NK93] Pour toute fonction F de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$, on a :*

$$\delta_F \geq 2 \ .$$

En cas d'égalité, la fonction F est dite presque parfaitement non-linéaire *(APN pour almost perfect nonlinear).*

Il est à noter que le terme presque parfaitement non-linéaire provient de la borne générale

$$\delta_F \geq 2^{n-m}$$

vérifiée pour les fonctions F de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$, où les fonctions atteignant la borne sont dites *parfaitement non-linéaires* (perfect nonlinear) [MS90]. De telles fonctions n'existent que si n est pair et $n \geq 2m$ [Nyb91]. Elles correspondent exactement aux fonctions courbes définies page 57 [MS90]. Notons que, dans le cas où $m = 1$, *i.e.* pour les fonctions booléennes scalaires, cette équivalence est exactement celle qui fait l'objet de la proposition 1.5.1 page 32. Les fonctions APN peuvent également être définies en termes de dérivées d'ordre 2 :

Proposition 2.11 *Une fonction F de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ est APN si et seulement si pour tout vecteurs non-nuls a et b de $\mathbf{F}_2^n$, où $a \neq b$, on a*

$$\forall x \in \mathbf{F}_2^n,\ D_a D_b F(x) \neq 0.$$

Toutes les permutations APN connues actuellement ont un nombre impair de variables. De fait, on conjecture que pour toute fonction F de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ avec n pair, on a

$$\delta_F \geq 4\ .$$

Des cas particuliers sont prouvés en particulier pour les fonctions puissance [Can97, CTZ97].

Le critère de résistance à la cryptanalyse linéaire portant sur la fonction de tour F montre qu'une fonction presque courbe est également presque parfaitement non-linéaire [CV95]. Ainsi choisir une fonction de tour dans cette classe extrémale de fonctions garantit une résistance optimale contre les attaques différentielles et linéaires. Ces propriétés liées à la possibilité d'établir des bornes pour les probabilités des différentielles et des approximations linéaires pour un chiffrement itératif en fonction de celles que vérifient une fonction de tour a donné naissance au concept de *sécurité démontrable*. Elles ont ainsi motivé l'emploi de ces fonctions dans de nombreux systèmes conçus par la suite, ce qui est le cas, par exemple, de MISTY.

2.4 Cryptanalyse par différentielle impossible

Cette variante de la cryptanalyse différentielle a été proposée par Biham, Biryukov et Shamir en 1999 [BBS99] . D'après la description du distingueur différentiel elle peut être considérée comme étant sa complémentation. En effet, l'attaque repose sur des différentielles ayant une faible probabilité de se produire, voire une probabilité nulle, pour le chiffrement. La fonction de test δ d'un distingueur par différentielle impossible est donc définie par :

$$\begin{array}{rccc} \delta_{(a,b)} & : & (\mathbf{F}_2^n)^2 \times (\mathbf{F}_2^n)^2 & \rightarrow & \mathbf{F}_2 \\ & & (x_1,\, x_2,\, y_1,\, y_2) & \mapsto & \phi_b(y_1 + y_2) \oplus 1, \end{array}$$

où ϕ_b est l'indicatrice de b. Le distingueur par différentielle impossible élémentaire peut alors être modélisé par $\mathcal{D}(x,x+a) = \phi_b\left(D_a \mathsf{O}(x)\right) \oplus 1$. Le distingueur répond ainsi 1 avec une grande probabilité, voire une probabilité 1, quand la différence en sortie n'est pas b. Le biais se calcule en fonction de la probabilité dans le cas uniforme qui est donc $1 - \frac{1}{2^n - 1}$.

2.5 Cryptanalyse par différentielle tronquée

La notion de différentielle tronquée a été introduite en 1995 par Knudsen [Knu95]. Elle permet de considérer des différentielles constituées par des portions de mots binaires et cette simplification produit des attaques de systèmes préalablement résistants à l'attaque différentielle classique. De manière formelle, Knudsen définit une différentielle tronquée par analogie avec une différentielle classique : si $(a,b) \neq (0,0)$ est une différentielle au tour i, on appelle différentielle tronquée au tour i le couple (a',b') où a' et b' sont des sous-séquences de a et b respectivement. Notons par $T_{(i_1,\ldots,i_t)}$, où $i_1 < \cdots < i_t$, la fonction de projection de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^t$ telle que

$$\forall (x_1,\ldots,x_n) \in \mathbf{F}_2^n,\ T_{(i_1,\ldots,i_t)}(x_1,\ldots,x_n) = (x_{i_1},\ldots,x_{i_t}).$$

Nous simplifions la notation en T_t lorsque les indices de projection n'ont pas besoin d'être spécifiés. En pratique, considérer une différentielle tronquée (a',b') revient à considérer l'ensemble des différentielles (a,b) telles que $T_{t_1}(a) = a'$ et $T_{t_2}(b) = b'$. On a ainsi pour les différentielles tronquées :

$$(a',b') \in \mathbf{F}_2^{t_1} \times \mathbf{F}_2^{t_2} \sim \left\{(a,b') \in \mathbf{F}_2^n \times \mathbf{F}_2^{t_2},\, T_{t_1}(a) = a'\right\}$$
$$(a,b') \in \mathbf{F}_2^n \times \mathbf{F}_2^{t_2} \sim \left\{(a,b) \in \mathbf{F}_2^n \times \mathbf{F}_2^n,\, T_{t_2}(b) = b'\right\}$$

Seul le second cas nous intéresse dans la mesure où tronquer la différence d'entrée ne conduit pas à modifier la fonction de test δ mais revient à fournir des requêtes au distingueur pour lesquelles la différence appartient à un sous-espace vectoriel. Nous adaptons donc la définition d'une différentielle tronquée dans notre cas :

Définition 2.12 Différentielle tronquée
Une différentielle tronquée à t bits au tour i d'un chiffrement itératif de fonction de tour F paramétrée par une clé de tour, est un couple (a,b'), $a \neq 0$ de $\mathbf{F}_2^n \times \mathbf{F}_2^t$ tel qu'il existe $x \in \mathbf{F}_2^n$ et une fonction de projection T_t pour lesquels

$$T_t\left(D_a\left(F_{k_i} \circ \cdots \circ F_{k_1}\right)(x)\right) = b'.$$

La probabilité de la différentielle tronquée à t bits (a,b') au tour i est définie par

$$\mathrm{TDp}^{(i)}(a,b') = \mathbf{P}_X[T_t\left(D_a\left(F_{k_i} \circ \cdots \circ F_{k_1}\right)(X)\right) = b'].$$

On retrouve le cas de la différentielle classique pour $t = n$. Une attaque par différentielle tronquée repose sur la détermination d'un couple $(a,b') \in \mathbf{F}_2^n \times \mathbf{F}_2^t$, $a \neq 0$ tels que la probabilité de la différentielle tronquée s'éloigne de celle de la distribution uniforme. Le cas extrême pouvant correspondre à une probabilité 1 si on s'intéresse à un nombre t très faible de bits et après peu de tours. Afin d'élaborer une attaque sur le dernier tour, il faut considérer une différentielle tronquée au tour $(r-1)$. Si la probabilité de la différentielle tronquée est éloignée de la distribution uniforme, il devient possible de concevoir un distingueur afin de retrouver la clé du dernier tour $k^{(r)}$. La probabilité d'une différentielle tronquée à t bits dans le cas uniforme correspond à la somme sur les différences de sorties dont la troncature à t bits est égale au vecteur b' (les événements étant indépendants) de l'espérance des probabilités des différentielles ainsi définies sur l'ensemble des permutations de $\mathbf{F}_2^n$. Elle vaut $\frac{2^{n-t}-1}{2^n-1}$ si $b' = 0$ et $\frac{2^{n-t}}{2^n-1}$ si $b' \neq 0$. Le cas $t = 1$ revient à distinguer deux sources binaires et le cas $t = n$ correspond à la cryptanalyse différentielle classique. La fonction de test δ d'un distingueur pour une différentielle tronquée à t bits (a,b'), est donc définie par :

$$\begin{array}{rccc} \delta_{(a,b')} : & (\mathbf{F}_2^n)^2 \times (\mathbf{F}_2^n)^2 & \to & \mathbf{F}_2 \\ & (x_1, x_2, y_1, y_2) & \mapsto & \phi_{b'}(T_t(y_1 + y_2)), \end{array}$$

où $\phi_{b'}$ est l'indicatrice de b' c'est-à-dire la fonction définie par :

$$\begin{array}{rccc} \phi_{b'} : & \mathbf{F}_2^t & \to & \mathbf{F}_2 \\ & x & \mapsto & \begin{cases} 1 & \text{si } x = b' \\ 0 & \text{sinon.} \end{cases} \end{array}$$

Le distingueur par différentielle tronquée à t bits élémentaire peut alors être modélisé par $\mathcal{D}(x,x+a) = \phi_{b'}\left(T_t(D_a\mathsf{O}(x))\right)$.

Un chiffrement est résistant aux attaques par différentielles tronquées si l'avantage du distingueur par différentielle tronquée à t bits pour tout $1 \leq t \leq n$, est le plus faible possible. Déterminer la proportion de valeurs de $x \in \mathbf{F}_2^n$ pour lesquelles $T_t(D_aG_{\mathbf{k}}(x)) = b'$ revient à calculer le poids de la fonction booléenne $\phi_{b'} \circ T_t \circ D_aG_{\mathbf{k}}$. Ainsi déterminer la meilleure différentielle tronquée au tour $(r-1)$ revient à déterminer (a,b') tels que pour tout $\mathbf{k} \in \mathcal{K}^{r-1}$,

$$\#\{x \in \mathbf{F}_2^n,\, T_t(G_{\mathbf{k}}(x+a) + G_{\mathbf{k}}(x)) = b'\} = \mathrm{wt}\left(\phi_{b'} \circ T_t \circ D_aG_{\mathbf{k}}\right)$$

est maximal. Plus précisément le biais ε exploité dans l'attaque, *i.e.* la distance entre les deux probabilités testées par le distingueur par différentielle tronquée pour une différentielle (a,b') est donnée par (dans le cas où $b' \neq 0$) :

$$\begin{aligned}\varepsilon &= \frac{\mathrm{wt}\left(\phi_{b'} \circ T_t \circ D_aG_{\mathbf{k}}\right) - \mathrm{wt}(\phi_{b'} \circ T_t)}{2^n - 1} \\ &= \frac{\mathcal{F}(\phi_{b'} \circ T_t) - \mathcal{F}\left(\phi_{b'} \circ T_t \circ D_aG_{\mathbf{k}}\right)}{2(2^n - 1)}.\end{aligned}$$

On peut déduire comme condition de résistance à l'attaque par différentielle tronquée que

$$\max_{\substack{1 \leq t \leq n \\ a \in \mathbf{F}_2^n \setminus \{0\},\ b' \in \mathbf{F}_2^t}} \mathrm{wt}\left(\phi_{b'} \circ T_t \circ D_aG_{\mathbf{k}}\right)$$

doit être le plus petit possible.

Pratiquement, en ne considérant que des portions des vecteurs d'entrée et de sortie, on augmente la probabilité d'erreur de type II. On augmente donc l'ordre de grandeur du nombre de couples clairs-chiffrés nécessaires relativement à l'avantage du distingueur et à une probabilité d'erreur totale donnée. Néanmoins la structure du chiffrement peut rendre l'avantage bien plus significatif que dans le cas de la cryptanalyse différentielle classique. Un bon exemple est celui de l'algorithme de chiffrement Skipjack sur lequel la cryptanalyse différentielle tronquée s'est avérée extrêmement efficace [KRW99].

Exemple du DES [Knu95]: Afin d'illustrer cette attaque, intéressons-nous à un tour de DES. On montre qu'en ce qui le concerne, il est possible de trouver des différentielles tronquées de probabilité 1. En effet, le traitement appliqué au mot entrant dans une fonction de tour par les boîtes S se fait indépendamment d'une boîte à l'autre. Ainsi, il est possible d'isoler une boîte et de ne chercher des différentielles que sur elle. On constate en étudiant le fonctionnement de la permutation P, que la sortie d'une boîte S affecte au plus 6 boîtes S au tour suivant.

Reprenons en détail les transformations subies par le vecteur de 32 bits entre les tours i et $i+1$. Après un premier passage dans les boîtes S, le vecteur $(s_1,s_2,\ldots,s_{32})$ subit la permutation P puis on lui ajoute la moitié gauche L_{i-1} ; il entre ensuite dans l'expansion E et la clé de tour $i+1$ est ajoutée. Si on note s_i^j le bit numéro i (toujours en lisant les bits par la gauche) de la j-ième boîte S, on obtient après l'expansion E un vecteur dont chaque composante est constituée d'un bit de boîte S et d'un bit de la moitié gauche L_{i-1}. Nous ne signalons dans le tableau que les bits des boîtes S.

De cette expression on tire qu'en entrée de tour la boîte S_1 n'est pas affectée par les boîtes S_1 et S_3 du tour précédent. Le récapitulatif pour l'ensemble des boîtes est effectué ci-après :

tour i					tour $i+1$			
S_1	s_1, s_2, s_3, s_4		$s_{16}, s_7, s_{20}, s_{21}$			$s_1^7, s_4^4, s_3^2, s_4^5, s_1^6, s_1^8$		S_1
S_2	s_5, s_6, s_7, s_8		$s_{29}, s_{12}, s_{28}, s_{17}$			$s_1^6, s_1^8, s_4^3, s_4^7, s_1^5, s_1^1$		S_2
S_3	$s_9, s_{10}, s_{11}, s_{12}$		$s_1, s_{15}, s_{23}, s_{26}$			$s_1^5, s_1^1, s_3^4, s_3^6, s_2^7, s_1^2$		S_3
S_4	$s_{13}, s_{14}, s_{15}, s_{16}$	P	$s_5, s_{18}, s_{31}, s_{10}$	$+L_{i-1}$	E	$s_2^7, s_1^2, s_2^5, s_3^8, s_2^3, s_2^1$	$+k^{(i+1)}$	S_4
S_5	$s_{17}, s_{18}, s_{19}, s_{20}$		s_2, s_8, s_{24}, s_{14}			$s_2^3, s_2^1, s_4^2, s_4^6, s_2^4, s_4^8$		S_5
S_6	$s_{21}, s_{22}, s_{23}, s_{24}$		s_{32}, s_{27}, s_3, s_9			$s_2^4, s_4^8, s_3^7, s_3^1, s_1^3, s_3^5$		S_6
S_7	$s_{25}, s_{26}, s_{27}, s_{28}$		$s_{19}, s_{13}, s_{30}, s_6$			$s_1^3, s_3^5, s_1^4, s_2^8, s_2^2, s_2^6$		S_7
S_8	$s_{29}, s_{30}, s_{31}, s_{32}$		$s_{22}, s_{11}, s_4, s_{25}$			$s_2^2, s_2^6, s_3^3, s_4^1, s_1^7, s_4^4$		S_8

L'entrée de la boîte	n'est pas affectée par les boîtes
S_1	S_1, S_3
S_2	S_2, S_4
S_3	S_3, S_8
S_4	S_4, S_6
S_5	S_5, S_7
S_6	S_2, S_6
S_7	S_1, S_7
S_8	S_5, S_8

De ce fait, il est possible de trouver pour le DES une différentielle tronquée à 4 tours de probabilité 1. Cette différentielle tronquée porte sur 8 bits de chiffré après 4 tours.

Reprenons le schéma d'un chiffrement de Feistel (cf. Figure 2.8). Il s'agit dans ce cas d'une attaque à clairs choisis, pour laquelle on considère une paire de clairs égaux pour leurs moitiés droites et tels que leurs moitiés gauches diffèrent de α. On compare ainsi les différences entre les deux chiffrements consécutifs de (L_0, R_0) et de $(L_0 + \alpha, R_0)$. La différence α est choisie de manière que les entrées des boîtes S entre les deux chiffrements soient égales, sauf pour une, supposons la boîte S_1, après l'expansion E. Comme la différence porte sur la moitié gauche, le premier tour maintient la même configuration et au second tour, les sorties de toutes les boîtes S entre les deux chiffrements sont égales excepté pour la boîte S_1. A l'entrée du troisième tour, les entrées des boîtes S_1 et S_7 entre les deux chiffrements sont égales puisqu'elles ne sont pas affectées par la sortie de la boîte S_1 (cf. Tableau 2.5). Les sorties des boîtes S_1 et S_7 sont ainsi égales et on connaît donc 8 bits de différence en sortie du troisième tour. Comme on connaît la différence au tour précédent et que sur ce schéma $\boldsymbol{c}_R = R_3 = R_4$, on connaît donc 8 bits de la différence de la sortie du quatrième tour.

Cette différentielle tronquée peut donc être utilisée pour élaborer une attaque sur le dernier tour. Dans [Knu95], l'attaque est une généralisation qui porte sur 6 tours de DES où on cherche cette fois-ci le dernier couple de clés. Knudsen montre qu'il existe une attaque différentielle du DES à 6 tours nécessitant 46 textes clairs choisis et dont le coût est équivalent à 3500 chiffrements, c'est-à-dire qu'elle ne nécessite que quelques secondes sur un PC.

Autres fonctions booléennes pour une différentielle : L'évolution des attaques entre la différentielle classique et la différentielle tronquée montre qu'on s'intéresse à la répartition de la dérivée du chiffrement réduit au travers de ce qu'on peut considérer comme un filtre booléen. Dans le cas de la différentielle classique, on utilise l'indicatrice de la différence de

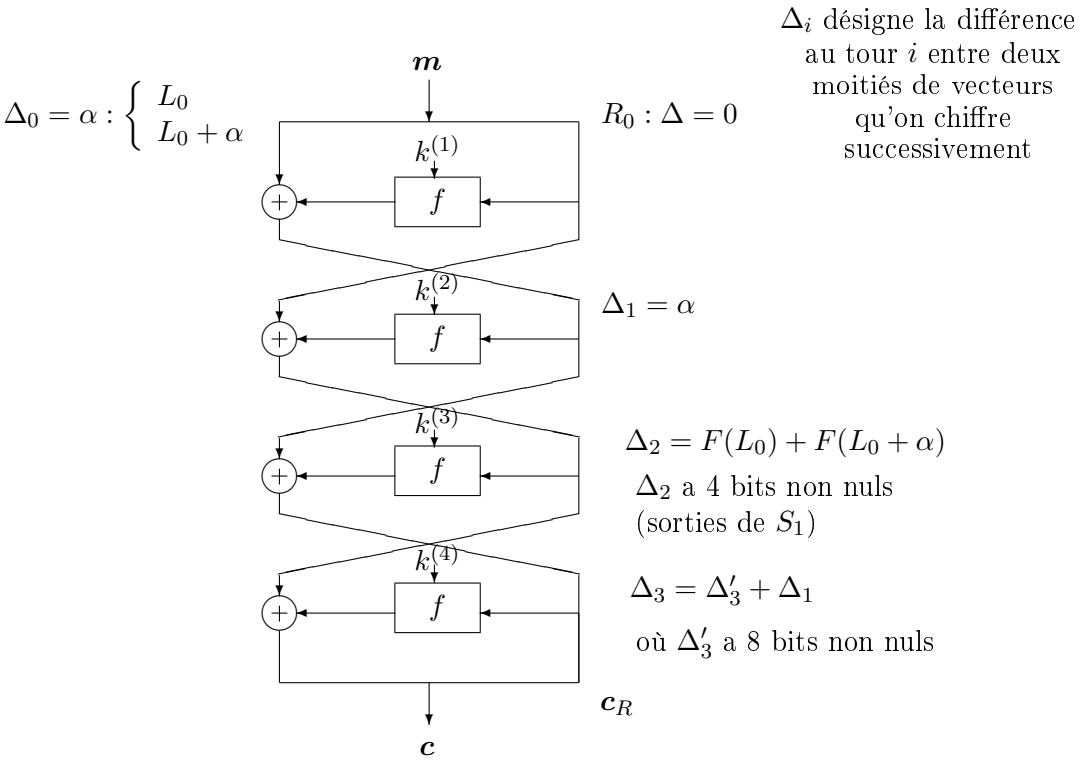


FIG. 2.8 – *Différentielle tronquée à 4 tours*

sortie fixée alors que dans le cas de la différentielle tronquée on considère la composition d'une fonction de troncature avec l'indicatrice de la différence tronquée de sortie attendue. Il est alors légitime de se demander si l'utilisation d'autres fonctions booléennes pour ce filtre permettrait de mettre en lumière d'autres types de faiblesses pour le chiffrement réduit et les propriétés de résistance correspondantes pour les fonctions de tour.

2.6 Cryptanalyse différentielle d'ordre supérieur

La différentielle classique repose sur l'observation de l'influence sur la différence entre deux chiffrés de la différence entre les clairs correspondants. Les différentielles tronquées permettent de n'observer ces variations que sur une portion des bits de sortie. On cherche à améliorer les idées précédentes en détaillant de manière plus précise les variations du chiffré en fonction de celles du clair. On utilise alors la notion de différentielle d'ordre supérieur. Au lieu de considérer uniquement une différence fixée entre des couples de clairs et la distribution des différences des chiffrés correspondants, on soumet au chiffrement une famille de 2^i clairs dont la variation est fixée par une famille de i différences dont on considère les 2^i combinaisons linéaires. On observe ensuite de manière similaire les différences sur la famille des 2^i chiffrés ainsi générés.

On généralise de manière naturelle la notion de dérivée d'ordre supérieur pour les fonctions

booléennes vectorielles.

Définition 2.13 *[Lai94] Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$ et $a_1,\ldots,a_t$ des vecteurs linéairement indépendants de $\mathbf{F}_2^n$. Notons par $\langle a_1,\ldots,a_t\rangle$ le sous-espace vectoriel engendré par ces t vecteurs. On appelle* dérivée d'ordre t de F relativement au sous-espace $\langle a_1,\ldots,a_t\rangle$ *la fonction définie par :*

$$\begin{array}{rccl} D_{\langle a_1,\ldots,a_t\rangle}F: & \mathbf{F}_2^n & \to & \mathbf{F}_2^m \\ & x & \mapsto & D_{a_t}\left(D_{\langle a_1,\ldots,a_{t-1}\rangle}F\right)(x) = \displaystyle\sum_{v\in\langle a_1,\ldots,a_t\rangle} F(x+v) \end{array}$$

.

La dérivée d'ordre 0 de F est bien sûr définie comme étant la fonction elle-même.

Définition 2.14 Différentielle d'ordre t
Une différentielle d'ordre t au tour i d'un chiffrement itératif de fonction de tour F paramétrée par une clé de tour, est un couple (V,b), V sous-espace vectoriel de dimension t, $b\in\mathbf{F}_2^n$, tel qu'il existe $x\in\mathbf{F}_2^n$ pour lequel

$$D_V\left(F_{k_i}\circ\cdots\circ F_{k_1}\right)(x) = b.$$

La probabilité de la différentielle d'ordre supérieur (V,b) au tour i est définie par

$$\mathrm{HDp}^{(i)}(a,b) = \mathbf{P}_X[D_V\left(F_{k_i}\circ\cdots\circ F_{k_1}\right)(X) = b].$$

Les définitions des dérivées et différentielles introduites pour l'attaque différentielle initiale correspondent donc aux dérivées et différentielles d'ordre 1.

La cryptanalyse différentielle d'ordre supérieur a été introduite par Knudsen [Knu95]. Elle repose originellement sur l'existence de dérivées d'ordre t constantes pour le chiffrement réduit $G_{\mathbf{k}}$. On peut en généraliser l'idée en définissant un distingueur pour une différentielle d'ordre supérieur. La fonction de test δ d'un distingueur pour une différentielle d'ordre supérieur (V,b) où $\dim(V)=t$ et $b\in\mathbf{F}_2^n$, est alors définie par :

$$\begin{array}{rccl} \delta_{(V,b)}: & (\mathbf{F}_2^n)^{2^t}\times(\mathbf{F}_2^n)^{2^t} & \to & \mathbf{F}_2 \\ & (x_1,\ldots,x_{2^t},y_1,\ldots,y_{2^t}) & \mapsto & \phi_b\left(\displaystyle\sum_{k=0}^{2^t} y_k\right), \end{array}$$

où ϕ_b est l'indicatrice de b. Le distingueur par différentielle d'ordre t élémentaire peut alors être modélisé par

$$\mathcal{D}(x+V) = \phi_b\left(\sum_{v\in V}\mathsf{O}(x+v)\right).$$

Pour mener à bien une attaque par différentielle d'ordre supérieur, il faut donc déterminer un sous-espace V de dimension t et un vecteur b tels que la probabilité de la différentielle d'ordre t, (V,b) s'éloigne notablement du cas uniforme dont la probabilité est $\frac{1}{2^n-1}$. Ainsi, si on peut déterminer un sous-espace vectoriel V de $\mathbf{F}_2^n$ de dimension t tel que pour tout $\mathbf{k}=(k_1,\ldots,k_{r-1})\in\mathcal{K}^{r-1}$ on ait :

$$D_V G_{\mathbf{k}} = c, \tag{2.1}$$

où c est une constante qui ne dépend pas de $\mathbf{k}$, alors la différentielle a pour probabilité 1. Le biais utilisé pour l'attaque est ainsi le plus élevé de tous ceux définis précédemment. En contrepartie, l'attaque dépend de la dimension du sous-espace V puisqu'elle nécessite la connaissance de 2^t clairs choisis.

Ainsi le paramètre principal de l'attaque est le sous-espace V. Trouver un tel sous-espace vérifiant l'équation 2.1 et de dimension la plus faible possible permet d'améliorer l'efficacité de l'attaque. Un candidat naturel, mais pas forcément de dimension minimale pour V se déduit naturellement du degré du chiffrement réduit lorsque ce degré est connu.

Si on définit le degré d'une fonction booléenne vectorielle comme le maximum des degrés de la forme algébrique normale de ses composantes booléennes, nous pouvons utiliser les propriétés suivantes pour déterminer un sous-espace V.

Proposition 2.15 *Soit* $\deg(F)$ *le degré d'une fonction vectorielle* $F : \mathbf{F}_2^n \to \mathbf{F}_2^m$. *Alors pour tout* $a \in \mathbf{F}_2^n \setminus \{0\}$, *on a*

$$\deg(D_a F) \leq \deg(F) - 1.$$

Ainsi on en déduit aisément la proposition suivante :

Proposition 2.16 *[Lai94] Soit* F *une fonction de* $\mathbf{F}_2^n$ *dans* $\mathbf{F}_2^n$ *de degré* d. *Alors pour tout sous-espace* $V \subset \mathbf{F}_2^n$ *de dimension* $(d+1)$, *on a*

$$D_V F(x) = 0 \quad \textit{pour tout } x \in \mathbf{F}_2^n \ .$$

Il faut bien noter que le plus petit sous-espace V vérifiant l'équation 2.1 peut être de dimension bien plus faible que $\deg(G_{\mathbf{k}}) + 1$. Un problème à résoudre reste donc l'estimation du degré du chiffrement réduit. Il existe une première borne supérieure triviale :

$$\max_{\mathbf{k} \in \mathcal{K}^{r-1}} \deg(G_{\mathbf{k}}) \leq \left(\max_{k \in \mathcal{K}} \deg(F_k) \right)^{r-1} .$$

On peut donc élaborer une attaque différentielle d'ordre supérieur utilisant cette borne supérieure comme dimension du sous-espace V seulement dans le cas où F_k est de degré très faible. Cette propriété est utilisée par Jakobsen et Knudsen [JK97] afin de casser un exemple de chiffrement proposé dans [NK93], dont la fonction de tour est une permutation presque courbe quadratique. Néanmoins un critère déduit d'une telle attaque est trop peu contraignant. Nous nous attachons à déterminer une borne supérieure plus précise pour le degré du chiffrement réduit dans le chapitre 4.

Il existe par ailleurs plusieurs pistes pour élargir le champ d'application de cette attaque, on peut ainsi considérer :

- pour un ordre t donné, déterminer la différentielle d'ordre t de probabilité la plus éloignée possible du cas uniforme (procédure similaire à une différentielle classique) ;
- pour un ordre t donné déterminer la différentielle d'ordre t tronquée de probabilité la plus éloignée possible du cas uniforme (procédure similaire à une différentielle tronquée) ;
- déterminer l'ordre t qui maximise la probabilité de la différentielle d'ordre t de l'une ou l'autre des propositions ci-dessus, pour une valeur de t *raisonnable*.

Nous avons résumé les différentes probabilités impliquées dans les attaque présentées dans ce chapitre sur le schéma 2.9.

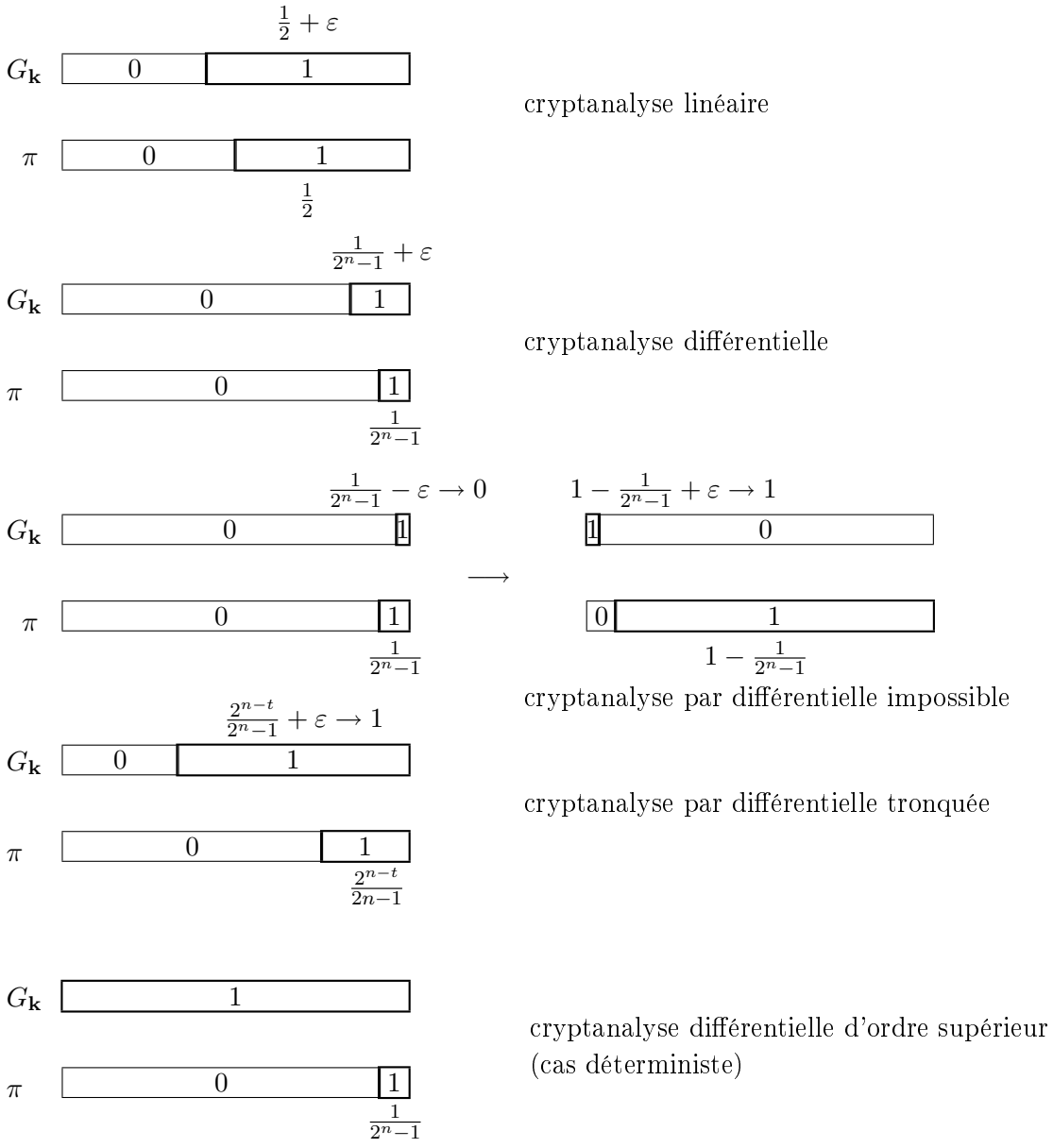


FIG. 2.9 – *Quelques attaques sur le dernier tour*

Chapitre 3

Généralisation de la cryptanalyse différentielle d'ordre supérieur sur l'algorithme MISTY1

Proposés en 1996 par Mitsuru Matsui, MISTY1 et MISTY2 [Mat97] sont des algorithmes de chiffrement symétriques itératifs par blocs. La taille des blocs est de 64 bits et la taille de la clé de 128 bits. Le nombre de tours, variable et multiple de 4, est au minimum de 8 (recommandation de l'auteur).

A la base de la conception de ces systèmes se trouve la notion de sécurité démontrable (*provable security*) contre les attaques linéaire et différentielle, notion développée par Nyberg et Knudsen [NK93] . Pour cela, on utilise des fonctions de confusion (notées ici S_7 et S_9) qui sont presque-courbes. D'autre part, ont été prises en considération les possibilités d'implémentation matérielle efficace. En effet, une des applications visées par MISTY est le réseau ATM où les débits peuvent atteindre des valeurs de plusieurs centaines de mégabits par seconde. Toutefois, la version allégée de MISTY1, appelée $M'1$, qui fait l'objet de la preuve de sécurité pour les attaques différentielle et linéaire est vulnérable à une cryptanalyse différentielle d'ordre supérieur mise en évidence par Tanaka et al. [THK99] . Par ailleurs, comme l'ont montré Babbage et Frisch [BF00] , cette attaque reste valable lorsqu'on remplace la fonction de substitution S_7 par n'importe quelle autre permutation puissance presque courbe de degré 3.

Je vais ici généraliser l'attaque de Tanaka et al. sur $M'1$ pour une taille de bloc quelconque et montrer qu'elle trouve son origine dans l'utilisation de boîtes S presque courbes. Autrement dit, la vulnérabilité de l'algorithme à l'attaque différentielle d'ordre supérieur réside dans sa résistance optimale aux attaques différentielle et linéaire.

3.1 Présentation générale du système de chiffrement

MISTY1 et MISTY2 utilisent des fonctions similaires, nommées FO, FI, FL. L'ordre d'application de ces fonctions diffèrent d'un système à l'autre. La structure de MISTY1 sans fonction FL (cadre de l'étude qui va suivre) est celle d'un chiffrement de Feistel. La différence des schémas influence nettement les vitesses de chiffrement et de déchiffrement. MISTY2 chiffre plus rapidement, mais en contrepartie, la vitesse de la phase de déchiffrement dépend du mode opératoire utilisé pour le chiffrement. Ainsi, MISTY1 est plus indiqué pour les modes ECB (Electronic CodeBook) et CBC (Cipher-Block Chaining), alors que MISTY2 l'est davantage pour les modes OFB (Output Feedback) et CFB (Cipher Feedback). Une variante de MISTY1

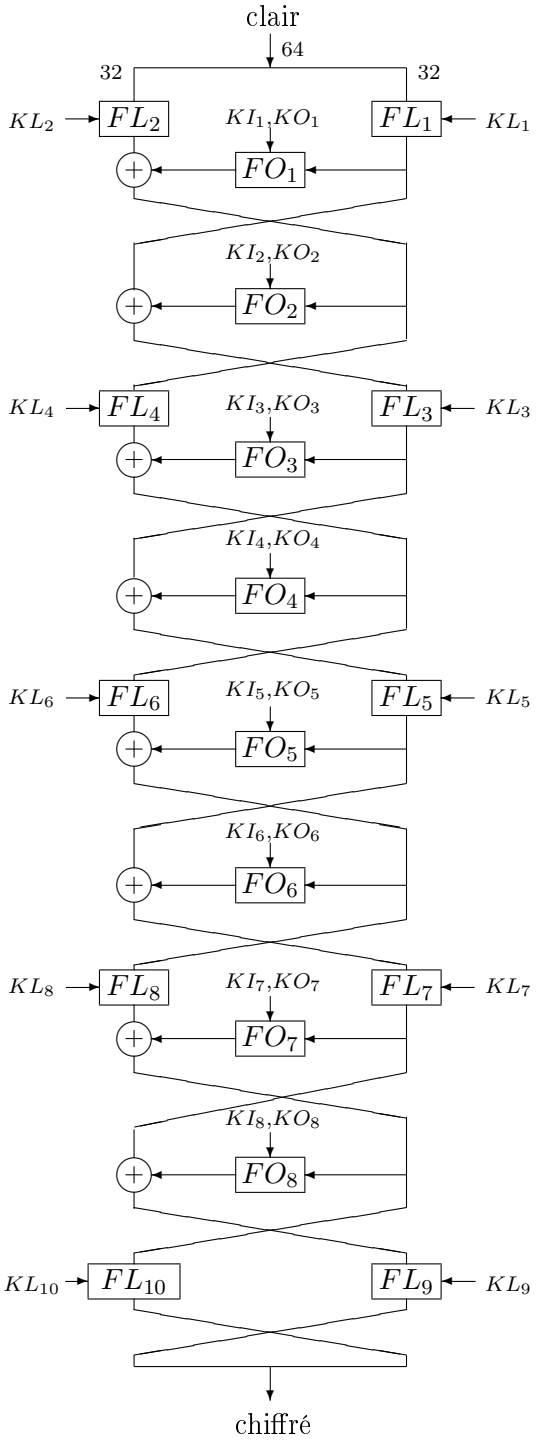


FIG. 3.1 – *Le système de chiffrement MISTY1*

utilisée en mode CTR, appelée KASUMI [3GP] a été choisie pour les applications de confidentialité et d'intégrité des données des mobiles de troisième génération. Les raisons principales de ce choix par le groupe de spécifications techniques du 3GPP (3^{rd} Generation Partnership Project) ont été la sécurité démontrable et les facilités d'implémentation matérielle.

La structure originelle de MISTY1 à 8 tours est présentée à la figure 3.1. Les fonctions FI utilisent deux fonctions linéaires simples de manipulation de données binaires appelées *truncate* et*zero-extend* qui transforment respectivement un bloc de 9 bits en un bloc de 7 bits en supprimant ses 2 bits de poids fort, et un bloc de 7 bits en un bloc de 9 bits en ajoutant des 0 aux 2 positions de poids fort.

La résistance de MISTY1 contre les attaques différentielle et linéaire a été démontrée par Matsui sur le système dépourvu des fonctions FL. C'est donc sur cette version simplifiée de MISTY1 que porte l'étude qui va suivre.

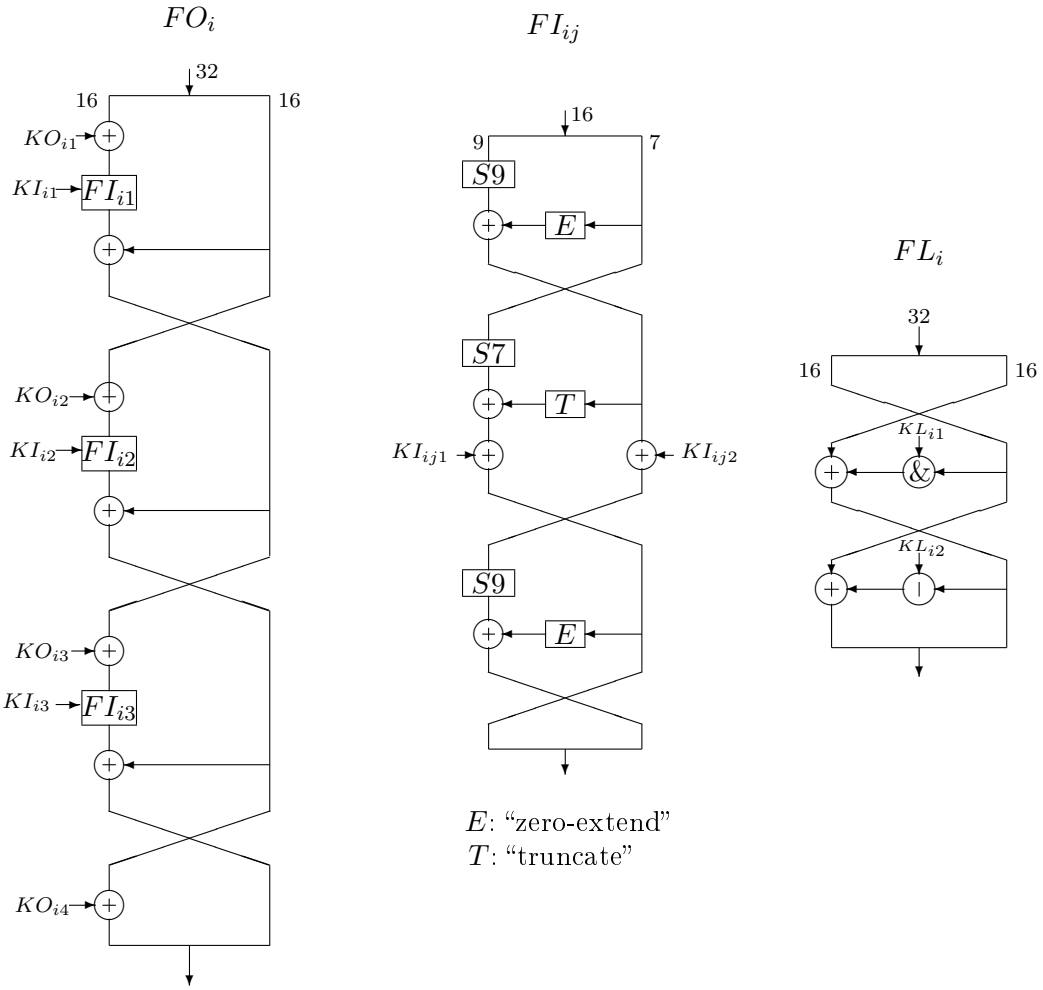


FIG. 3.2 – *Composition des fonctions FO, FI et FL*

Les fonctions linéaires FL n'étant rajoutées que pour contrer d'éventuelles attaques d'autres types et ne permettant pas de démontrer un quelconque apport supplémentaire de sécurité, il est légitime de se limiter à la partie démontrable et donc de ne considérer que le système

sans les fonctions FL afin de le soumettre à d'autres attaques. Dans [THK99], Tanaka et al. ont montré que MISTY1 sans fonction FL réduit à cinq tours était vulnérable à une attaque différentielle d'ordre 7. S. Babbage et L. Frisch [BF00] ont montré que cette attaque était valable si on remplace la boîte S_7 par n'importe quelle autre fonction puissance $x \mapsto x^e$ telle que $\mathrm{wt}(e) = 3$ sur $\mathbf{F}_{2^7}$ qui est presque-courbe.

L'objet de ce chapitre est de mettre en place une généralisation de ce type d'attaque, qui va permettre tout d'abord d'expliquer l'origine de la faiblesse de l'algorithme et par la suite de mettre en évidence une famille d'algorithmes vulnérables à cette famille de cryptanalyse.

3.2 Présentation de $M'1$, version de MISTY1 soumise à l'attaque

3.2.1 Les notations

Nous nous intéressons ici à un cadre plus général pour cet algorithme de chiffrement. On considère $M'1$, chiffrement itératif par blocs, de taille de bloc $16m$, construit sur le modèle de MISTY1. Toutes les fonctions utilisées dans MISTY1 sont alors à reconsidérer en fonction du paramètre m. L'attaque porte sur cinq tours de chiffrement n'utilisant pas les fonctions FL. Le système est présenté à la figure 3.3. Dans ce cas général, les fonctions *truncate* et *zero-extend* sont définies de la manière suivante :

Notation 3.1 *On définit la fonction* zero-extend *par :*

$$\begin{array}{rccc} E: & \mathbf{F}_2^{2m-1} & \to & \mathbf{F}_2^{2m+1} \\ & (u_1, \cdots, u_{2m-1}) & \mapsto & (u_1, \cdots, u_{2m-1}, 0, 0) \end{array}$$

et la fonction truncate *par :*

$$\begin{array}{rccc} T: & \mathbf{F}_2^{2m+1} & \to & \mathbf{F}_2^{2m-1} \\ & (u_1, \cdots, u_{2m+1}) & \mapsto & (u_1, \cdots, u_{2m-1}). \end{array}$$

Notons que ces deux fonctions sont linéaires et que :

$$\forall x \in \mathbf{F}_2^{2m-1}, \quad T \circ E(x) = x.$$

On donne ici les principales définitions et notations nécessaires pour la suite de ce chapitre.

Notation 3.2 *On appelle $M'1$ la version de MISTY1 réduite à 5 tours, sans fonction FL.*

x_1 est la partie droite de $8m$ bits du vecteur de $16m$ bits entrant dans $M'1$, x_0 la partie gauche et (x_{i+1},x_i) la valeur intermédiaire dans $M'1$ après i tours.

Notation 3.3 Manipulation de bits
Soit u un vecteur de $16m$ bits. On désigne par u^L, u^R, u^{L_k}, u^{R_k}, respectivement, la moitié gauche de u, la moitié droite de u, les k bits les plus à gauche de u et les k bits les plus à droite de u.

On note par $\|$ l'opération de concaténation de deux vecteurs binaires.

Notation 3.4 *On note x les $(2m-1)$ bits les plus à droite de x_0 ($x = x_0^{R_{2m-1}}$), (x_0,x_1) étant le vecteur entrant dans $M'1$.*

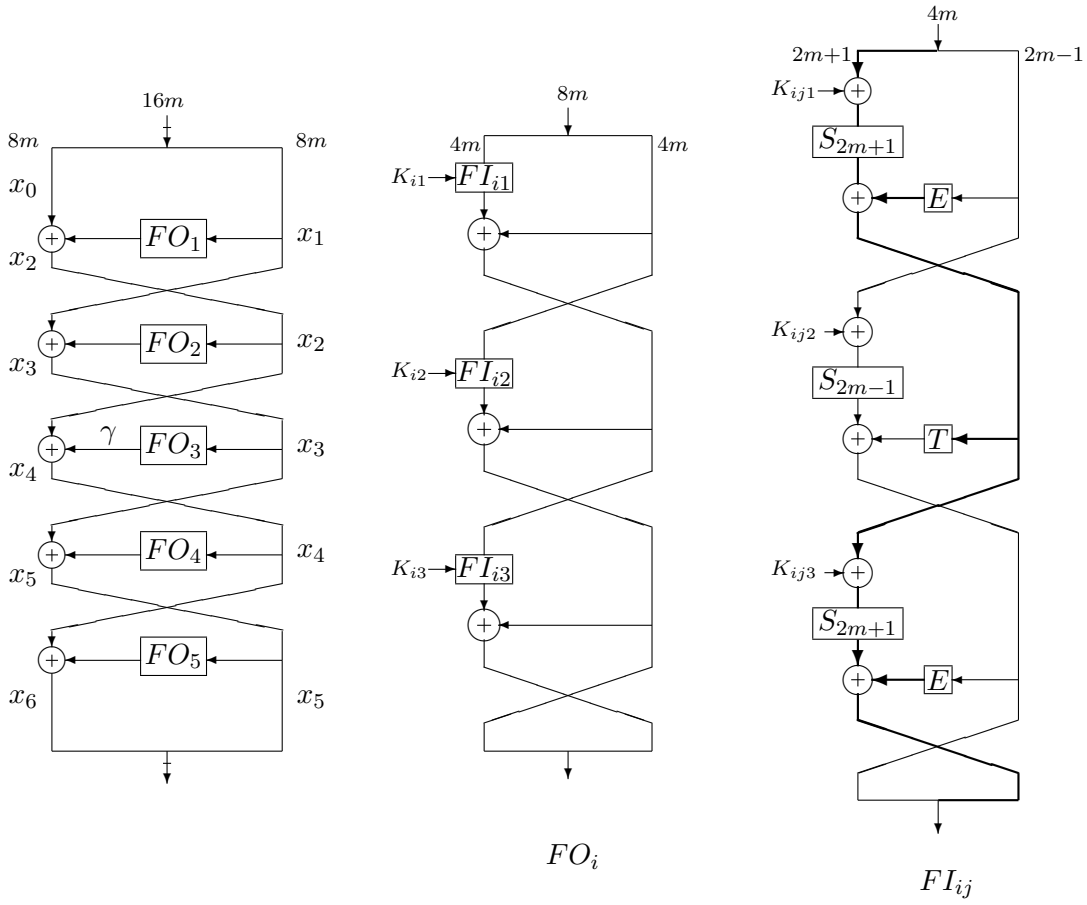


FIG. 3.3 – *Les* 5 *premiers tours de* $M'1$ *sans fonction FL avec cadencement équivalent de clés*

3.2.2 Principe général de l'attaque

Notation 3.5 *On note* V *le sous-espace vectoriel de dimension* $(2m-1)$ *des blocs de clairs formés par les vecteurs de* $16m$ *bits de la forme* $(0_{6m+1} \parallel x \parallel 0_{8m})$ *où* x *parcourt* $\mathbf{F}_2^{2m-1}$ *et* W *le sous-espace vectoriel supplémentaire de* V, i.e. *tel que* $V \times W = \mathbf{F}_2^{16m}$.

On s'intéresse alors au chiffrement des blocs de message de la forme :

$$(\underbrace{0_{6m+1} \parallel x \parallel 0_{8m}}_{P}) + (\underbrace{w_0 \parallel 0_{2m-1} \parallel w_1}_{w})$$

où $P \in V$ et w est une constante fixée dans W qui s'écrit $(w_0 \parallel w_1)$. L'entrée du 1^{er} tour de la fonction de chiffrement correspond au vecteur (x_0,x_1) avec $x_1 = w_1$ et $x_0 = (w_0 \parallel x)$ où w_1 et w_0 sont deux constantes.

On considère la fonction f_K qui à tout bloc de message de $16m$ bits associe les $(2m-1)$ bits les plus à gauche de x_4.

$$\begin{array}{rccc} f_K: & \mathbf{F}_2^{16m} & \to & \mathbf{F}_2^{2m-1} \\ & (x_0,x_1) & \mapsto & x_4^{L_{2m-1}}. \end{array}$$

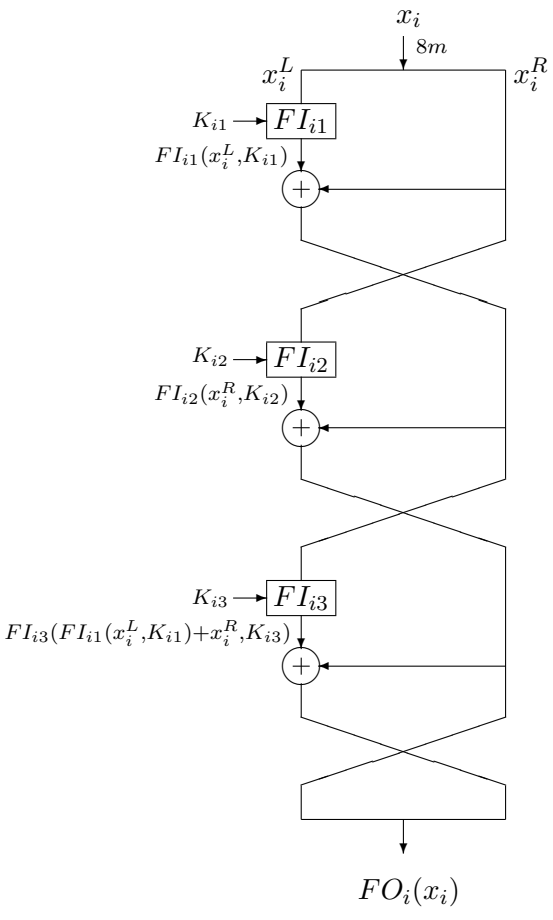
Dans l'attaque différentielle d'ordre supérieur portant sur $M'1$, avec $m = 4$ et les boîtes S_7 et S_9 proposées par Matsui, Tanaka et al. [THK99] ont montré que la dérivée d'ordre 7 de f_K relativement à V est une constante indépendante de la clé secrète K utilisée. Autrement dit, on a :

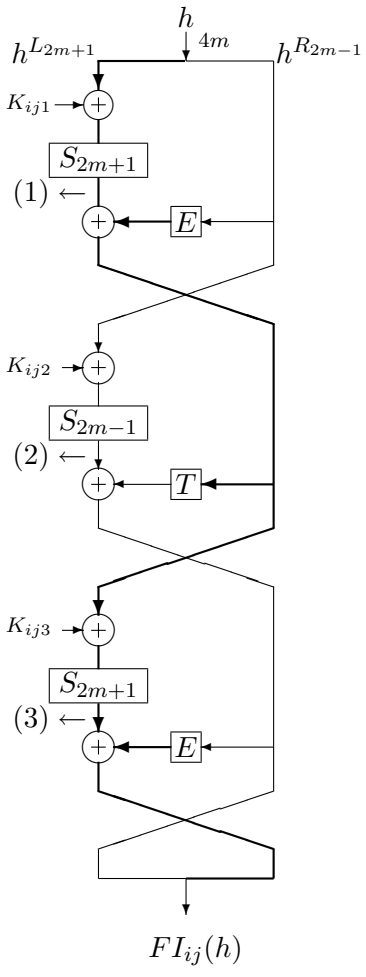
$$\forall w \in W, \quad \sum_{x \in V} f_K(x+w) = c. \tag{3.1}$$

Babbage et Frisch [BF00] ont montré expérimentalement que la propriété (3.1) reste vraie pour $m = 4$ si la boîte S_7 est une fonction puissance presque courbe de degré 3.

3.2.3 Expressions générales des sorties des fonctions FO_i et FI_{ij}

Pour déterminer l'expression générale de la fonction f_K, nous allons détailler les formules qui donnent la valeur des sorties des fonctions FO_i et FI_{ij}. En reprenant leur construction, nous obtenons ainsi les expressions suivantes.





On a pour $FO_i(x_i)$:

$$[FO_i(x_i)]^L = FI_{i2}(x_i^R,K_{i2}) + FI_{i1}(x_i^L,K_{i1}) + x_i^R$$
$$[FO_i(x_i)]^R = FI_{i3}(FI_{i1}(x_i^L,K_{i1}) + x_i^R,K_{i3}) + [FO_i(x_i)]^L .$$

Pour $FI_{ij}(h)$, on cherche d'abord les expressions des vecteurs aux trois points (1), (2) et (3) indiqués sur la figure 3.2.3 :

$$(1) : \ S_{2m+1}(h^{L_{2m+1}} + K_{ij1})$$
$$(2) : \ S_{2m-1}(h^{R_{2m-1}} + K_{ij2})$$
$$(3) : \ S_{2m+1}\left(S_{2m+1}(h^{L_{2m+1}} + K_{ij1}) + E(h^R_{2m-1}) + K_{ij3}\right)$$

ce qui nous donne finalement :

$$[FI_{ij}(h)]^{L_{2m-1}} = S_{2m-1}(h^{R_{2m-1}} + K_{ij2}) + T \circ S_{2m+1}(h^{L_{2m+1}} + K_{ij1}) + h^{R_{2m-1}}$$
$$[FI_{ij}(h)]^{R_{2m+1}} = S_{2m+1}\left(S_{2m+1}(h^{L_{2m+1}} + K_{ij1}) + E(h^{R_{2m-1}}) + K_{ij3}\right) + E\left([FI_{ij}(h)]^{L_{2m-1}}\right).$$

3.3 Étude détaillée de x_4

On va présenter ici les calculs qui conduisent à l'expression détaillée de x_4. On note c_i, $i \in \mathbf{N}$ des constantes. Reprenons le schéma de $M'1$ (figure 3.3).

Au premier tour

$$x_2 = FO_1(x_1,K_1) + x_0 = (\underbrace{c_2}_{4m} \parallel \underbrace{c_1}_{2m+1} \parallel \underbrace{x + c_0}_{2m-1}).$$

Au deuxième tour

$$x_3 = FO_2(x_2,K_2) + x_1 = (\mu \parallel \lambda) + x_1,$$

où $\mu = [FO_2(x_2,K_2)]^L$ et $\lambda = [FO_2(x_2,K_2)]^R$. On a :

$$\begin{aligned}\mu &= FI_{22}(x_2^R,K_{22}) + FI_{21}(x_2^L,K_{21}) + x_2^R \\ &= FI_{22}(c_1 \parallel x + c_0) + \underbrace{cte + (c_1 \parallel x + c_0)}_{c_3 \parallel x + c_4},\end{aligned}$$

ce qu'on obtient par $FI_{21}(x_2^L,K_{21}) + x_2^R = cte + (c_1 \parallel x + c_0) = (c_3 \parallel x + c_4)$. Par ailleurs,

$$\lambda = FI_{23}(c_3 \parallel x + c_4) + FI_{22}(c_1 \parallel x + c_0) + (c_3 \parallel x + c_4).$$

Il nous faut maintenant détailler les expressions de $FI_{23}(c_3 \parallel x + c_4)$ et $FI_{22}(c_1 \parallel x + c_0)$. On a d'une part :

$$\begin{aligned}[FI_{22}(c_1 \parallel x + c_0)]^{L_{2m-1}} &= S_{2m-1}(x + c_0 + K_{222}) + T \circ S_{2m+1}(c_1 + K_{221}) + x + c_0 \\ &= S_{2m-1}(x + c_5) + x + c_6,\end{aligned}$$

d'où on tire

$$\mu^{L_{2m-1}} = S_{2m-1}(x + c_5) + x + c_9 \ ;$$

où $c_9 = c_6 + c_3^{L_{2m-1}}$. D'autre part, on a :

$$\begin{aligned}[FI_{23}(c_3 \parallel x + c_4)]^{L_{2m-1}} &= S_{2m-1}(x + c_4 + K_{232}) + T \circ S_{2m+1}(c_3 + K_{231}) + x + c_4 \\ &= S_{2m-1}(x + c_7) + x + c_8,\end{aligned}$$

d'où on tire

$$\lambda^{L_{2m-1}} = S_{2m-1}(x + c_7) + S_{2m-1}(x + c_5) + c_{10}.$$

On calcule aussi :

$$\begin{aligned}[FI_{22}(c_1 \parallel x + c_0)]^{R_{2m+1}} &= S_{2m+1}\left(S_{2m+1}(c_1 + K_{221}) + E(x + c_0) + K_{223}\right) \\ &\quad + E \circ S_{2m-1}(x + c_5) + E(x) + E(c_6) \\ &= S_{2m+1}(E(x) + c_{11}) + E \circ S_{2m-1}(x + c_5) + E(x) + c_{12} \\ [FI_{23}(c_3 \parallel x + c_4)]^{R_{2m+1}} &= S_{2m+1}\left(S_{2m+1}(c_3 + K_{221}) + E(x + c_4) + K_{223}\right) \\ &\quad + E \circ S_{2m-1}(x + c_7) + E(x) + E(c_8) \\ &= S_{2m+1}(E(x) + c_{13}) + E \circ S_{2m-1}(x + c_7) + E(x) + c_{14},\end{aligned}$$

et on obtient donc

$$\begin{aligned}\mu^{R_{2m+1}} &= S_{2m+1}\left(E(x) + c_{11}\right) + E \circ S_{2m-1}(x + c_5) + c_{15} \\ \lambda^{R_{2m+1}} &= S_{2m+1}\left(E(x) + c_{13}\right) + E \circ S_{2m-1}(x + c_7) + E(x) + c_{14} \\ &\quad + S_{2m+1}\left(E(x) + c_{11}\right) + E \circ S_{2m-1}(x + c_5) + E(x) + c_{12} + E(x) + E(c_4) \\ &= S_{2m+1}\left(E(x) + c_{13}\right) + E \circ S_{2m-1}(x + c_7) \\ &\quad + S_{2m+1}\left(E(x) + c_{11}\right) + E \circ S_{2m-1}(x + c_5) + E(x) + c_{16}.\end{aligned}$$

Finalement, on obtient les expressions de μ et λ :

$$\begin{aligned}\mu &= (S_{2m-1}(x + c_5) + x + c_9 \parallel S_{2m+1}\left(E(x) + c_{11}\right) + E \circ S_{2m-1}(x + c_5) + c_{15}) \\ \lambda &= (S_{2m-1}(x + c_7) + S_{2m-1}(x + c_5) + c_{10} \parallel S_{2m+1}\left(E(x) + c_{13}\right) + E \circ S_{2m-1}(x + c_7) \\ &\quad + S_{2m+1}\left(E(x) + c_{11}\right) + E \circ S_{2m-1}(x + c_5) + E(x) + c_{16}).\end{aligned}$$

Comme $x_3 = (\mu \parallel \lambda) + x_1$, on a donc

$$x_3 = (\mu + c_{17} \parallel \lambda + c_{18}).$$

Au troisième tour On s'intéresse donc à x_4 et plus précisément à ses $(2m-1)$ bits les plus à gauche sur lesquels portent la propriété de différentielle d'ordre supérieur. On commence donc par évaluer sur la moitié gauche de la sortie de FO_3, notée γ :

$$\gamma = [FO_3(x_3)]^L = FI_{32}(\lambda + c_{18}) + FI_{31}(\mu + c_{17}) + \lambda + c_{18}.$$

Le vecteur binaire qui nous intéresse est $\gamma^{L_{2m-1}}$. On a :

$$\begin{aligned}[FI_{31}(\mu + c_{17})]^{L_{2m-1}} &= S_{2m-1}(\mu^{R_{2m-1}} + c_{17}^{R_{2m-1}} + K_{312}) \\ &\quad + T \circ S_{2m+1}(\mu^{L_{2m+1}} + c_{17}^{L_{2m+1}} + K_{311}) + \mu^{R_{2m-1}} + c_{17}^{R_{2m-1}} \\ [FI_{32}(\lambda + c_{18})]^{L_{2m-1}} &= S_{2m-1}(\lambda^{R_{2m-1}} + c_{18}^{R_{2m-1}} + K_{322}) \\ &\quad + T \circ S_{2m+1}(\lambda^{L_{2m+1}} + c_{18}^{L_{2m+1}} + K_{321}) + \lambda^{R_{2m-1}} + c_{18}^{R_{2m-1}},\end{aligned}$$

d'où :

$$\gamma^{L_{2m-1}} = \mu^{R_{2m-1}} + \lambda^{R_{2m-1}} + \lambda^{L_{2m-1}} + c_{19} + T \circ S_{2m+1}(\mu^{L_{2m+1}} + c_{20}) + T \circ S_{2m+1}(\lambda^{L_{2m+1}} + c_{21}) + S_{2m-1}(\mu^{R_{2m-1}} + c_{22}) + S_{2m-1}(\lambda^{R_{2m-1}} + c_{23}).$$

Comme

$$x_4^{L_{2m-1}} = \gamma^{L_{2m-1}} + \underbrace{x_2^{L_{2m-1}}}_{=cte},$$

on obtient finalement :

$$\begin{aligned} x_4^{L_{2m-1}} = \; & \mu^{R_{2m-1}} + \lambda^{R_{2m-1}} + \lambda^{L_{2m-1}} + c_{24} + T \circ S_{2m+1}(\mu^{L_{2m+1}} + c_{20}) \\ & + T \circ S_{2m+1}(\lambda^{L_{2m+1}} + c_{21}) + S_{2m-1}(\mu^{R_{2m-1}} + c_{22}) + S_{2m-1}(\lambda^{R_{2m-1}} + c_{23}). \end{aligned} \tag{3.2}$$

Ce qu'on cherche alors à déterminer sur cette expression est la nature des termes de degré $(2m-1)$ de chacun des bits de $x_4^{L_{2m-1}}$ vus comme des fonctions booléennes à $(2m-1)$ variables, correspondant aux bits de x.

3.4 Analyse du monôme de plus haut degré dans x_4

Les boîtes S ont la structure générale suivante : $x \mapsto L(x^e)$ où L est une permutation linéaire et e est un exposant de fonction presque courbe. Cette dernière propriété est très importante car elle impose une forte structure aux boîtes S (cf. partie 3.5), structure sur laquelle s'appuie l'explication de l'origine de la propriété de différentielle d'ordre supérieur.

On appelle e l'exposant de la boîte S_{2m-1} et d son degré (c'est-à-dire le poids de Hamming de e), de même, e_2 désigne l'exposant de la boîte S_{2m+1} et d_2 son degré. Les conditions de l'attaque sur l'algorithme MISTY1 d'origine sont telles que $d_2 \leq d$ (on a $d_2 = 2$ et $d = 3$). Par ailleurs, pour permettre de suivre le même type de schéma d'attaque, il faut qu'on puisse ne pas avoir à considérer les termes provenant de $T \circ S_{2m+1}(\mu^{L_{2m+1}} + c_{20}) + T \circ S_{2m+1}(\lambda^{L_{2m+1}} + c_{21})$. Pour cela, il faut pouvoir considérer une différentielle d'ordre suffisant, c'est-à-dire que $dd_2 < 2m-1$. On se limite au cas où $d_2 = 2$, comme pour l'attaque d'origine et on s'intéresse au paramètre d. On utilise la proposition suivante :

Proposition 3.6 [CCD99, cor.4] et [CCZ98, th.1]
Soit n un entier impair. Si f est presque courbe sur $\mathbf{F}_2^n$, alors son degré est au plus $\frac{n+1}{2}$. Donc si la fonction puissance $f : x \mapsto x^s$ sur $\mathbf{F}_{2^n}$ est presque courbe, alors

$$\deg(f) = \mathrm{wt}(s) \leq \frac{n+1}{2}$$

où $\mathrm{wt}(s)$ désigne le poids de Hamming du vecteur binaire s.

Comme la fonction S_{2m-1} est presque courbe sur $\mathbf{F}_{2^{2m-1}}$, on a donc $d \leq \frac{2m-1+1}{2} = m$. Cette inégalité doit être comparée à la condition $dd_2 < 2m-1$, ce qui donne $d < m - \frac{1}{2}$, soit $d \leq m-1$. On peut noter que l'attaque originale correspond effectivement à un tel cas de figure : $d = 3$ pour $m = 4$.

Dans la suite du chapitre, on considérera donc que $d_2 = 2$ et que $d \leq m-1$.

Comme μ et λ sont de degré au plus d en x, les termes de degré $(2m-1)$ dans l'expression (3.2) ne proviennent que de $S_{2m-1}(\mu^{R_{2m-1}} + c_{22}) + S_{2m-1}(\lambda^{R_{2m-1}} + c_{23})$. Les expressions à détailler sont donc $S_{2m-1}(\mu^{R_{2m-1}} + c_{22})$ et $S_{2m-1}(\lambda^{R_{2m-1}} + c_{23})$.

Notation 3.7 *Soit f une fonction booléenne à n variables. On note $[f]_d$ la somme des termes de degré au moins d dans la forme algébrique normale de f. Par exemple,*

$$[x_0x_2x_4x_5 + x_0x_1x_2 + x_1x_2x_3 + x_0x_1 + 1]_3 = x_0x_2x_4x_5 + x_0x_1x_2 + x_1x_2x_3.$$

Pour une fonction booléenne vectorielle, l'opérateur s'applique à chaque composante.

Expression de $\left[S_{2m-1}(\mu^{R_{2m-1}} + c_{22})\right]_{2m-1}$

On obtient :

$$\begin{aligned}\left[S_{2m-1}(\mu^{R_{2m-1}} + c_{22})\right]_{2m-1} &= \left[S_{2m-1}\left(S_{2m-1}(x + c_5) + T \circ S_{2m+1}(E(x) + c_{11}) + T(c_{15})\right)\right]_{2m-1} \\ &= \left[S_{2m-1}\left(S_{2m-1}(x) + T \circ S_{2m+1}(E(x) + c_{25}) + c_{26}\right)\right]_{2m-1}\end{aligned}$$

et on appelle $t(x)$ l'expression :

$$t(x) = S_{2m-1}(x) + T \circ S_{2m+1}(E(x) + c_{25}) + c_{26}.$$

On a effectué un changement de variables car on calcule une différentielle de même degré que la dimension du sous-espace considéré.

L'écriture explicite des fonctions puissance presque courbes $S_{2m+1}(x) = x^{e_2}$ et $S_{2m-1}(x) = x^e$ permet d'écrire $t(x)$ sous la forme

$$t(x) = x^e + \underbrace{Q(x)}_{\text{quadratique}} + \underbrace{A(x,c_{25})}_{\text{affine}} + c_{26}$$

où Q ne contient que des termes quadratiques et A des termes affines en x, puisque c_{25} n'apparaît que dans des termes de degré au plus 1 et c_{26} n'apparaît que dans des termes constants.

Donc, tout terme de $[S_{2m-1}(t(x))]_{2m-1}$ est formé par le produit de β_1 termes en x^e, de β_2 termes en $Q(x)$, de β_3 termes en $A(x,c_{25})$ et de β_4 termes constants, avec $\beta_1+\beta_2+\beta_3+\beta_4 = d$. Autrement dit, les termes sont de la forme

$$x^{e\lambda_1} \cdot x^{\lambda_2} \cdot x^{\lambda_3} \cdot c$$

où λ_1, λ_2 et λ_3 sont des entiers inférieurs ou égaux à 2^{2m-1}, vérifiant : $\mathrm{wt}(\lambda_1) = \beta_1$, $\mathrm{wt}(\lambda_3) = \beta_3$ et $\mathrm{wt}(\lambda_2) = 2\beta_2$ puisque λ_2 est la somme de β_2 entiers de poids 2.

Le terme de degré maximum en x s'obtient si on considère la configuration où $\beta_4 = 0$, auquel cas on a $\beta_1 + \beta_2 + \beta_3 = d$. Par ailleurs, un tel terme dépend d'une des constantes uniquement si $\beta_3 \neq 0$. Son degré est donc donné par :

$$\mathrm{wt}\left((e\lambda_1 + \lambda_2 + \lambda_3) \bmod \left(2^{2m-1} - 1\right)\right)$$

et l'attaque serait réalisable à partir du moment où $\mathrm{wt}\left((e\lambda_1 + \lambda_2 + \lambda_3) \bmod \left(2^{2m-1} - 1\right)\right) < 2m - 1$.

Expression de $\left[S_{2m-1}(\lambda^{R_{2m-1}}+c_{23})\right]_{2m-1}$

On traite de même l'expression $S_{2m-1}(\lambda^{R_{2m-1}}+c_{23})$. On a :

$$\begin{aligned}\left[S_{2m-1}(\lambda^{R_{2m-1}}+c_{23})\right]_{2m-1} &= [S_{2m-1}\left(S_{2m-1}(x+c_5)+S_{2m-1}(x+c_7)\right.\\ &\quad +T\circ S_{2m+1}(E(x)+c_{11})+T\circ S_{2m+1}(E(x)+c_{13})\\ &\quad \left.+x+T(c_{16})\right)]_{2m-1}.\end{aligned}$$

On effectue également un changement de variable en posant :

$$g(x)=\underbrace{S_{2m-1}(x)+S_{2m-1}(x+c_{28})}_{=D_{c_{28}}S_{2m-1}(x)\rightarrow\deg=(d-1)}+\underbrace{T\circ S_{2m+1}(E(x)+c_{29})+T\circ S_{2m+1}(E(x)+c_{30})}_{=D_{c_{30}}T\circ S_{2m+1}(E(x+c_{29}))\rightarrow\deg=1}+x+c_{31}$$

et

$$\left[S_{2m-1}(\lambda^{R_{2m-1}}+c_{23})\right]_{2m-1}=[S_{2m-1}(g(x))]_{2m-1}$$

On peut également écrire

$$g(x)=D_{c_{28}}S_{2m-1}(x)+\underbrace{A(x,c_{32})}_{\text{affine}}+c_{31}.$$

Comme $\deg(g)\leq(d-1)$, on en déduit immédiatement que $[S_{2m-1}(g(x))]_{2m-1}=0$ dès que $d(d-1)<2m-1$, ce qui est le cas pour les paramètres originaux du chiffrement ($m=4$ et $d=3$). En résumé, il faut maintenant étudier la structure de $[t(x)^e]_{2m-1}$ et $[g(x)^e]_{2m-1}$. Pour cela, il est nécessaire de détailler les caractéristiques des fonctions utilisées dans les boîtes S, à savoir les fonctions puissance presque courbes.

3.5 Fonctions presque courbes et codes correcteurs

Les fonctions presque courbes et les fonctions presque parfaitement non-linéaire peuvent être caractérisées par des propriétés de codes linéaires. Dans le cas particulier des fonctions puissance, on peut ainsi appliquer des résultats concernant les codes cycliques pour mettre en évidence des propriétés de divisibilité des poids de telles fonctions.

3.5.1 Codes linéaires associés à une fonction

Comme nous l'avons vu dans le chapitre 2, les biais intervenant dans les cryptanalyses différentielle et linéaire sont proportionnels aux quantités suivantes où f est une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ (typiquement f correspond à une fonction de substitution) :

$$\begin{aligned}\delta_f &= \max_{\alpha\neq 0,\beta}\#\{x\in\mathbf{F}_2^n\ ,\ f(x+\alpha)+f(x)=\beta\}\\ &= \max_{\alpha\neq 0,\beta}\mathrm{wt}(\phi_\beta\circ D_\alpha f)\\ \lambda_f &= \left|\max_{\alpha,\beta\neq 0}\#\{x\in\mathbf{F}_2^n\ ,\ \alpha\cdot x+\beta\cdot f(x)=1\}-2^{n-1}\right|\\ &= \frac{\mathcal{L}(f)}{2}.\end{aligned}$$

Les propriétés des fonctions presque courbes et presque parfaitement non-linéaires peuvent être exprimées en termes de codes correcteurs d'erreurs.

Tout sous-espace vectoriel de $\mathbf{F}_2^n$ de dimension k est appelé code linéaire binaire de longueur n et de dimension k et est noté $[n,k]$. à tout $[n,k]$-code linéaire binaire $\mathcal{C}$ on associe le $[n,n-k]$-code dual, noté $\mathcal{C}^\perp$ et défini par :

$$\mathcal{C}^\perp = \{x \in \mathbf{F}_2^n,\, x \cdot c = 0,\, \forall c \in \mathcal{C}\}.$$

Toute matrice H de dimension $r \times n$ définit un $[n,n-r]$-code linéaire binaire $\mathcal{C}$:

$$\mathcal{C} = \{c \in \mathbf{F}_2^n,\, cH^T = 0\}$$

où H^T désigne la matrice transposée de H. H est alors appelée matrice de parité de $\mathcal{C}$.

On identifiera désormais l'espace vectoriel $\mathbf{F}_2^n$ au corps fini $\mathbf{F}_{2^n}$ et on étudiera uniquement les fonctions f de $\mathbf{F}_{2^n}$ telles que $f(0) = 0$. En effet, toute fonction g de $\mathbf{F}_{2^n}$ vérifiant $g(0) = c \neq 0$ peut être obtenue à partir d'une fonction f vérifiant $f(0) = 0$ en la composant avec la permutation linéaire de $\mathbf{F}_{2^n}$ définie par $\ell(x) = x + c$. Les fonctions f et g ont donc les mêmes valeurs de δ et λ.

À une telle fonction f de $\mathbf{F}_{2^n}$ on associe le code linéaire binaire $\mathcal{C}_f$ de longueur $2^n - 1$ ayant pour matrice de parité la matrice binaire à $2n$ lignes et $(2^n - 1)$ colonnes

$$H_f = \begin{bmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{2^n-2} \\ f(1) & f(\alpha) & f(\alpha^2) & \dots & f(\alpha^{2^n-2}) \end{bmatrix}$$

où chaque élément de $\mathbf{F}_{2^n}$ est représenté par un vecteur colonne binaire de n bits et α est un élément primitif de $\mathbf{F}_{2^n}$.

Théorème 3.8 [CCZ98]
Soit une fonction $f : \mathbf{F}_{2^n} \to \mathbf{F}_{2^n}$ telle que $f(0) = 0$ et $\mathcal{C}_f$ son code associé. Alors :

(i) $\lambda_f = 2^{n-1}$ *si et seulement si* $\dim \mathcal{C}_f > 2^n - 1 - 2n$ *ou* $\mathcal{C}_f^\perp$ *contient le vecteur tout-à-un.*

(ii) Si $\dim \mathcal{C}_f = 2^n - 1 - 2n$,

$$\lambda_f = \max_{c \in \mathcal{C}_f^\perp,\, c \neq 0} \left| 2^{n-1} - \mathrm{wt}(c) \right|.$$

En particulier, pour n impair, f est presque courbe si et seulement si pour tout mot de code $c \in \mathcal{C}_f^\perp$,

$$2^{n-1} - 2^{\frac{n-1}{2}} \leq \mathrm{wt}(c) \leq 2^{n-1} + 2^{\frac{n-1}{2}}.$$

(iii) f est presque parfaitement non-linéaire si et seulement si la distance minimale du code $\mathcal{C}_f$ est supérieure ou égale à 5.

Le tableau 3.1 donne la liste des exposants s tels que la fonction puissance $x \mapsto x^s$ est presque parfaitement non-linéaire (APN) ou presque courbe (AB). Il s'agit des seules fonctions puissance presque courbes connues (à équivalence près par composition avec une permutation linéaire de $\mathbf{F}_{2^n}$). Notons qu'une permutation presque courbe qui n'est pas équivalente à une fonction puissance à récemment été construite pour la première fois [BCP05].

nom	exposant t	APN	AB	référence
quadratique	2^i+1 avec pgcd$(n,i)=1$	oui	oui	[Nyb93]
Kasami	$2^{2i}-2^i+1$ avec pgcd$(n,i)=1$	oui	oui	[Kas71]
Welch	$2^{\frac{n-1}{2}}+3$	oui	oui	[Dob98b, CCD00a]
Niho	$2^{\frac{n-1}{2}}+2^{\frac{n-1}{4}}-1$ avec $n\equiv 1 \bmod 4$	oui	oui	[Dob98a, HX01]
	$2^{\frac{n-1}{2}}2+^{\frac{3n-1}{4}}-1$ avec $n\equiv 3 \bmod 4$	oui	oui	[Dob98a, HX01]
inverse	2^n-2	oui	non	[Nyb93]
Dobbertin	$2^{\frac{4n}{5}}+2^{\frac{3n}{5}}+2^{\frac{2n}{5}}+2^{\frac{n}{5}}-1$ avec $n\equiv 0 \bmod 5$	oui	non	[Dob98a]

TAB. 3.1 – *Fonctions puissances presque parfaitement non-linéaires et presque courbes sur* $\mathbf{F}_{2^n}$ *avec* n *impair*

3.5.2 Divisibilité des poids des codes cycliques

Dans le cas où la fonction f considérée est une fonction puissance de $\mathbf{F}_{2^n}$, *i.e.* $f(x)=x^s$, la matrice H_f est une matrice de parité d'un code cyclique. Les propriétés des codes cycliques permettent d'énoncer des théorèmes concernant les fonctions puissance presque courbes.

Définition 3.9 *Un code linéaire binaire* $\mathcal{C}$ *de longueur* n *est cyclique si pour tout mot de code* $c=(c_0,\cdots,c_{n-1})$ *de* $\mathcal{C}$ *le vecteur* $(c_{n-1},c_0,\cdots,c_{n-2})$ *est aussi dans* $\mathcal{C}$.

Si à tout vecteur $(c_0,\cdots,c_{n-1})\in\mathbf{F}_2^n$ on associe le polynôme $c(X)=\sum_{i=0}^{n-1}c_iX^i$ de $\mathcal{R}_n=\mathbf{F}_2[X]/(X^n-1)$, alors tout code cyclique binaire de longueur n est un idéal de $\mathcal{R}_n$. Les idéaux de $\mathbf{F}_2[X]/(X^n-1)$ sont les images par la surjection canonique des idéaux de $\mathbf{F}_2^n[X]$ qui est un anneau intègre principal. Ainsi tout idéal du quotient $\mathbf{F}_2[X]/(X^n-1)$ est monogène, c'est-à-dire qu'il existe un g de degré minimal qui l'engendre. On peut donc associer à tout code cyclique $\mathcal{C}$ de longueur n un unique polynôme unitaire g, de degré minimal. Ce polynôme est appelé le polynôme générateur du code et ses racines sont appelées zéros du code. L'ensemble de définition de $\mathcal{C}$ est alors l'ensemble :

$$I(\mathcal{C})=\{i\in\{0,\cdots,2^n-2\}\mid\ \alpha^i \text{ est un zéro de } \mathcal{C}\}$$

où α est un élément primitif de $\mathbf{F}_{2^n}$. Comme $\mathcal{C}$ est un code binaire, son ensemble de définition est une réunion de classes cyclotomiques relatives à 2 modulo (2^n-1), $Cl(a)$ où $Cl(a)=\{2^ja \bmod (2^n-1)\}$.

Dans toute la suite on identifiera l'ensemble de définition d'un code cyclique binaire de longueur (2^n-1) avec les représentants de la classe cyclotomique relative à 2 modulo (2^n-1). Le code linéaire $\mathcal{C}_f$ associé à la fonction puissance $f:x\mapsto x^s$ sur $\mathbf{F}_{2^n}$ est défini par la matrice de parité suivante :

$$H_f=\begin{bmatrix}1 & \alpha & \alpha^2 & \cdots & \alpha^{2^n-2}\\ 1 & \alpha^s & \alpha^{2s} & \cdots & \alpha^{(2^n-2)s}\end{bmatrix}.$$

Le code consiste donc en tous les vecteurs binaires c de longueur (2^n-1) tels que $cH_f^T=0$, c'est-à-dire tels que :

$$c(\alpha)=\sum_{i=0}^{2^n-2}c_i\alpha^i=0 \ \text{ et } \ c(\alpha^s)=\sum_{i=0}^{2^n-2}c_i\alpha^{is}=0.$$

Le code $\mathcal{C}_f$ est donc le code cyclique binaire de longueur (2^n-1) et d'ensemble de définition $\{1,s\}$.

Les fonctions puissance presque courbes et presque parfaitement non-linéaires correspondent donc à des codes cycliques ayant des propriétés métriques exceptionnelles. Ces objets optimaux apparaissent également dans d'autre contextes applicatifs que la cryptologie et la correction d'erreur. Ainsi en télécommunications, les exposants des fonctions AB correspondent aux valeurs des décimations pour lesquelles une séquence ML et sa décimée ont une corrélation mutuelle parfaite [HK98, Hel76, Cha98].

Définition 3.10 *Un code binaire $\mathcal{C}$ est dit 2^ℓ-divisible si le poids de tous ses mots de code est divisible par 2^ℓ. De plus, $\mathcal{C}$ est dit exactement 2^ℓ-divisible si, en outre, il contient au moins un mot de code dont le poids n'est pas divisible par $2^{\ell+1}$.*

Le théorème suivant dû à McEliece [McE72] réduit la détermination de la divisibilité exacte des poids des codes cycliques binaires à un problème de combinatoire.

Théorème 3.11 *Un code cyclique binaire est exactement 2^ℓ-divisible si et seulement si ℓ est le plus petit nombre tel que $\ell+1$ non-zéros de $\mathcal{C}$ (avec répétitions possibles) ont pour produit 1.*

On s'intéresse en particulier aux codes cycliques primitifs à deux zéros et à la divisibilité exacte de leurs duaux. On note $\mathcal{C}_{1,s}$ le code cyclique binaire de longueur (2^n-1) et d'ensemble de définition $Cl(1)\cup Cl(s)$. Les non-zéros du code cyclique $\mathcal{C}_{1,s}^\perp$ sont les α^{-i} où $i \in Cl(1)\cup Cl(s)$. Donc $(\ell+1)$ non-zéros de $\mathcal{C}_{1,s}^\perp$ ont pour produit 1 si et seulement si il existe $I_1 \subset Cl(s)$ et $I_2 \subset Cl(1)$ tels que $|I_1|+|I_2| = \ell+1$ et

$$\prod_{k\in I_1\cup I_2} \alpha^{-k} = 1 \Longleftrightarrow \sum_{k\in I_1\cup I_2} k \equiv 0 \bmod (2^n-1).$$

Considérons les entiers u et v définis par leur expansion en base 2 :

$$u = \sum_{i=0}^{n-1} u_i 2^i \quad \text{et} \quad v = \sum_{i=0}^{n-1} v_i 2^i$$

où $u_i = 1$ si et seulement si $2^i s \bmod (2^n-1) \in I_1$ et $v_i = 1$ si et seulement si $2^i \bmod (2^n-1) \in I_2$. On obtient alors :

$$\sum_{k\in I_1\cup I_2} k \equiv \sum_{i=0}^{n-1} u_i 2^i s + \sum_{i=0}^{n-1} v_i 2^i \bmod (2^n-1) \equiv 0 \bmod (2^n-1).$$

La taille de I_1 (resp. I_2) correspond à $\mathrm{wt}(u)$, poids de Hamming de u (resp. $\mathrm{wt}(v)$). On peut alors énoncer le théorème de McEliece dans le cas qui nous intéresse de la manière suivante :

Corollaire 3.12 *Le code cyclique $\mathcal{C}_{1,s}^\perp$ de longueur (2^n-1) est exactement 2^ℓ-divisible si et seulement si pour tout (u,v), $0 \le u \le 2^n-1$ et $0 \le v \le 2^n-1$, tel que $us+v \equiv 0 \bmod (2^n-1)$, on a :*

$$\mathrm{wt}(u) + \mathrm{wt}(v) \ge \ell+1.$$

Comme $v \leq 2^n - 1$, la condition $us + v \equiv 0 \bmod (2^n - 1)$ peut s'écrire $v = (2^n - 1) - (us \bmod (2^n - 1))$, ce qui conduit à la formulation équivalente suivante :

Corollaire 3.13 *Le code cyclique $\mathcal{C}_{1,s}^{\perp}$ de longueur $(2^n - 1)$ est exactement 2^ℓ-divisible si et seulement si pour tout u tel que $0 \leq u \leq 2^n - 1$,*

$$\mathrm{wt}(A(u)) \leq \mathrm{wt}(u) + m - 1 - \ell$$

où $A(u) = us \bmod (2^n - 1)$.

3.6 Caractérisation des fonctions presque courbes

Comme on a pu le voir, la non-linéarité d'une fonction de $\mathbf{F}_{2^n}$ dans $\mathbf{F}_{2^n}$ est liée à la distribution des poids de certains codes linéaires binaires de longueur $(2^n - 1)$ et de dimension $2n$. On s'intéresse donc à la distribution des poids des codes admettant ces paramètres. On appelle énumérateur des poids du code linéaire $\mathcal{C}$ de longueur n le vecteur $(A_0, \cdots, A_n)$ où A_i est le nombre de mots de code de poids i dans le code $\mathcal{C}$.

Théorème 3.14 [CCD99]
Soit $\mathcal{C}$ un $[2^n - 1, 2^n - 2n - 1]$-code linéaire binaire de distance minimale $d \geq 3$. Supposons que le dual $\mathcal{C}^{\perp}$ de ce code ne contient pas le vecteur tout-à-un : $\mathbf{1} = (1, \cdots, 1)$. Soit $A = (A_0, \cdots, A_{2^n-1})$ (resp. $B = (B_0, \cdots, B_{2^n-1})$) l'énumérateur des poids de $\mathcal{C}^{\perp}$ (resp. $\mathcal{C}$). On a alors :

(i) Si w_0 est tel que $A_w = A_{2^n-w} = 0$ pour tout $0 < w < w_0$ alors

$$6(B_3 + B_4) \leq (2^n - 1)\left[(2^{n-1} - w_0)^2 - 2^{n-1}\right]$$

avec égalité si et seulement si $A_w = 0$ pour tout $w \notin \{0, w_0, 2^{n-1}, 2^n - w_0\}$.

(ii) Si w_1 est tel que $A_w = A_{2^n-w} = 0$ pour tout $w_1 < w < 2^{m-1}$ alors

$$6(B_3 + B_4) \geq (2^n - 1)\left[(2^{n-1} - w_1)^2 - 2^{n-1}\right]$$

avec égalité si et seulement si $A_w = 0$ pour tout $w \notin \{0, w_1, 2^{n-1}, 2^n - w_1\}$.

La preuve de ce théorème (cf. [CCD99]) repose sur les quatre premières égalités des identités de Pless :

$$\begin{aligned}
\sum_{w=0}^{2^n-1} wA_w &= 2^{2n-1}(2^n - 1), \\
\sum_{w=0}^{2^n-1} w^2 A_w &= 2^{3n-2}(2^n - 1), \\
\sum_{w=0}^{2^n-1} w^3 A_w &= 2^{2n-3}\left((2^n - 1)^2(2^n + 2) - 3!B_3\right), \\
\sum_{w=0}^{2^n-1} w^4 A_w &= 2^{2n-4}\left(2^n(2^n - 1)(2^{2n} + 3 \cdot 2^n - 6) + 4!(B_4 - (2^n - 1)B_3)\right).
\end{aligned}$$

On peut alors énoncer le théorème suivant qui caractérise les fonctions presque courbes par la divisibilité des poids des codes linéaires qui leur sont associés :

Théorème 3.15 *Soit n un entier impair et soit f une fonction de $\mathbf{F}_{2^n}$ dans $\mathbf{F}_{2^n}$ telle que $\lambda_f \neq 2^{n-1}$. f est presque courbe si et seulement si f est presque parfaitement non-linéaire et si le code $\mathcal{C}_f^{\perp}$ qui lui est associé est $2^{\frac{n-1}{2}}$-divisible.*

Dans le cas où f est une fonction puissance, $f : x \mapsto x^s$, le code $\mathcal{C}_f$ correspondant est le code cyclique binaire $\mathcal{C}_{1,s}$ de longueur $(2^n - 1)$ et d'ensemble de définition $\{1,s\}$. La divisibilité des poids du code dual correspondant peut donc être obtenue grâce au théorème de McEliece sous sa forme du corollaire 3.12. On obtient ainsi la caractérisation suivante des fonctions puissance presque courbes :

Corollaire 3.16 *Soit $f : x \mapsto x^s$ une permutation sur $\mathbf{F}_{2^n}$. Alors f est presque courbe sur $\mathbf{F}_{2^n}$ si et seulement si f est presque parfaitement non-linéaire sur $\mathbf{F}_{2^n}$ et*

$$\forall u,\ 1 \leq u \leq 2^n - 1,\ \ \mathrm{wt}\,(us \bmod (2^n - 1)) \leq \frac{n-1}{2} + \mathrm{wt}(u). \tag{3.3}$$

3.7 Faiblesse des boîtes S utilisant des fonctions puissance presque courbes

Comme on a pu le constater dans la partie précédente, le fait qu'une fonction soit presque courbe lui impose une forte structure, c'est-à-dire, en particulier, que le code dual du code linéaire qui lui est associé admet une très grande divisibilité des poids ainsi qu'une répartition très localisée de la distribution de ses poids (on dit que son spectre a des raies puisque tous les mots du code $\mathcal{C}_f^{\perp}$ ont leur poids dans $\{0,2^{n-1},2^{n-1} \pm 2^{\frac{n-1}{2}}\}$). Lorsqu'on considère le théorème 3.15 sous la forme du corollaire 3.16 concernant les fonctions puissance presque courbes, on constate qu'il fournit une majoration pour le degré de fonctions booléennes correspondant au produit de composantes de fonctions puissance presque courbe. Cette propriété fournit l'explication de la faiblesse autorisant l'emploi d'une différentielle d'ordre 7 dans l'attaque originale de $M'1$ et nous permet d'amorcer une étude plus générale de la famille de chiffrements suivant le schéma de $M'1$, paramétrés par la taille de bloc $16m$.

3.7.1 Cas particulier de l'attaque de [BF00] pour $m = 4$

Dans le cas particulier de cette attaque, comme $\deg(g) = 2$ et $d = \mathrm{wt}(e) = 3$, $S_{2m-1}(\lambda^{R_{2m-1}} + c_{23})$ a un degré au plus 6 ce qui signifie que $[g(x)^e]_{(2m-1)=7} = 0$. On ne s'intéresse alors qu'au terme $[t(x)^e]_{(2m-1)=7}$. Obtenir des degrés supérieurs à 7 pour l'expression :

$$t(x)^e = (x^e + Q(x) + A(x,c_{25}) + c_{26})^e$$

implique de multiplier au minimum deux termes de degré 3 de $t(x)$. Or les termes de degrés 3 de $t(x)$ proviennent du terme x^e, donc d'une fonction puissance presque courbe. En utilisant la majoration énoncée dans le corollaire 3.16, pour un produit de deux termes de $t(x)$ on a :

$$\forall u,\quad \mathrm{wt}(u) = 2,\quad \mathrm{wt}(ue(\mathrm{mod} 2^7 - 1)) \leq \frac{7-1}{2} + \mathrm{wt}(u) = 5.$$

De cette majoration on tire donc la démonstration du Fait 2 dans [BF00] : les termes de degré six du produit de deux bits de sortie de $S_7(x)$ sont toujours nuls, le produit de 2 bits de la sortie d'une boîte S_7 est de degré au plus 5. Donc le seul terme de degré 7 de $t(x)^e$ provient du degré 5 du produit de deux bits de x^e et du degré 2 du terme quadratique de $Q(x)$. Ce terme de degré 7 est donc forcément indépendant des constantes et permet donc de monter l'attaque différentielle d'ordre 7 sur $M'1$.

3.7.2 Cas général : majoration des degrés

Si on revient aux expressions impliquées dans la cryptanalyse de la version généralisée de $M'1$ avec des blocs de $16m$ bits, nous devons étudier le degré de

- $[t(x)^e]_{2m-1}$ avec

$$t(x) = S_{2m-1}(x) + T \circ S_{2m+1}(E(x) + c_{25}) + c_{26},$$

ce que l'écriture explicite des fonctions puissance presque courbes $S_{2m+1}(x) = x^{e_2}$ et $S_{2m-1}(x) = x^e$ permet d'écrire $t(x)$ sous la forme

$$t(x) = x^e + \underbrace{Q(x)}_{\text{quadratique}} + \underbrace{A(x,c_{25})}_{\text{affine}} + c_{26} \ ;$$

- $[g(x)^e]_{2m-1}$ avec

$$g(x) = D_{c_{28}} S_{2m-1}(x) + D_{c_{30}} T \circ S_{2m+1}(E(x + c_{29})) + x + c_{31},$$

qu'on utilise sous la forme

$$g(x) = D_{c_{28}} S_{2m-1}(x) + A(x,c_{32}) + c_{31}.$$

Nous allons voir qu'il est possible de fournir certaines majorations systématiques du degré de $[t(x)^e]_{2m-1}$. Pour ce qui est du terme où $g(x) = D_{c_{28}} S_{2m-1}(x) + A(x,c_{32}) + c_{31}$, un problème différent se pose dans la mesure où il n'est pour l'instant pas connu de propriétés caractéristiques des dérivées de fonctions presque courbes. Une première approche a consisté en une simulation des degrés maximum obtenus lorsqu'on effectue le produit de composantes d'une dérivée de fonction presque courbe. S'il se trouve que le degré de telles fonctions est soumis à une majoration, ou qu'elles prennent une forme particulière ne permettant que certains degrés, l'étude systématique effectuée sur le terme $[t(x)^e]_{2m-1}$ pourrait également être appliquée à $g(x) = D_{c_{28}} S_{2m-1}(x) + A(x,c_{32}) + c_{31}$.

On peut toutefois remarquer que tous les termes de $g(x)^e$ correspondent au produit de β_1 composantes de $D_{c_{28}} S_{2m-1}$ et de β_2 composantes de $A(x,c_{32})$ avec $\beta_1 + \beta_2 \leq d$. Le degré d'un tel terme est donc majoré par $\beta_1(d-1) + (d-\beta_1)$ car $\deg(D_{c_{28}} S_{2m-1}) \leq d-1$. Dans le cas où $\beta_1 = d$ (et donc $\beta_2 = 0$), ce terme correspond à un produit de dérivées relativement à c_{28}. Il prend donc la même valeur en x et en $x + c_{28}$ pour tout $x \in \mathbf{F}_2^{2m-1}$ et ne peut par conséquent pas être de degré $2m-1$. Les termes de degré $2m-1$ de $(g(x))^e$ ne peuvent donc provenir que des cas où $\beta_1 \leq d-1$. Leur degré est alors majoré par $(d-1)^2 + 1$. Il s'ensuit que $(g(x))^e$ ne contient aucun terme de degré $2m-1$ si

$$d < 1 + \sqrt{2m-2}.$$

C'est notamment le cas pour les paramètres suivants :

$$\begin{array}{lll} d=3 & \text{et} & m \geq 4 \\ d=4 & \text{et} & m \geq 6 \\ d=3 & \text{et} & m \geq 10 \end{array}$$

3.8 Présentation des résultats concernant le terme $[t(x)^e]_{2m-1}$: classement des valeurs des exposants de la boîte S_{2m-1}

Comme indiqué au paragraphe 3.4, on peut écrire :

$$t(x) = x^e + \underbrace{Q(x)}_{\text{quadratique}} + \underbrace{A(x,c_{25})}_{\text{affine}} + c_{26}$$

où Q ne contient que des termes quadratiques et A des termes affines en x, puisque c_{25} n'apparaît que dans des termes de degré au plus 1 et c_{26} n'apparaît que dans des termes constants.

Donc, tout terme de $[S_{2m-1}(t(x))]_{2m-1}$ est formé par le produit de β_1 termes en x^e, de β_2 termes en $Q(x)$, de β_3 termes en $A(x,c_{25})$ et de β_4 termes constants, avec $\beta_1+\beta_2+\beta_3+\beta_4 = d$. Autrement dit, les termes sont de la forme

$$x^{e\lambda_1} \cdot x^{\lambda_2} \cdot x^{\lambda_3} \cdot c$$

où λ_1, λ_2 et λ_3 sont des entiers inférieurs ou égaux à 2^{2m-1}, vérifiant : $\mathrm{wt}(\lambda_1) = \beta_1$, $\mathrm{wt}(\lambda_3) = \beta_3$ et $\mathrm{wt}(\lambda_2) = 2\beta_2$ puisque λ_2 est la somme de β_2 entiers de poids 2.

Le terme de degré maximum en x s'obtient si on considère la configuration où $\beta_4 = 0$, auquel cas on a $\beta_1 + \beta_2 + \beta_3 = d$. Par ailleurs, un tel terme dépend d'une des constantes uniquement si $\beta_3 \neq 0$. Son degré est donc donné par :

$$\mathrm{wt}\left((e\lambda_1 + \lambda_2 + \lambda_3) \bmod \left(2^{2m-1} - 1\right)\right)$$

et l'attaque serait réalisable à partir du moment où $\mathrm{wt}\left((e\lambda_1 + \lambda_2 + \lambda_3) \bmod \left(2^{2m-1} - 1\right)\right) < 2m - 1$.

Dans toute la suite de ce chapitre, on pose la convention qu'une somme d'exposants est toujours considérée modulo $2^{2m-1} - 1$, ce qui signifie typiquement que $e\lambda_1 + \lambda_2 + \lambda_3$ doit être compris comme $(e\lambda_1 + \lambda_2 + \lambda_3) \bmod \left(2^{2m-1} - 1\right)$.

3.8.1 Majoration brutale

La première idée à exploiter est celle de la majoration brutale des poids des λ_i. Pour cela, on va utiliser le corollaire 3.16. On obtient alors :

$$\begin{array}{rcl} \mathrm{wt}(e\lambda_1 + \lambda_2 + \lambda_3) & \leq & \underbrace{\mathrm{wt}(e\lambda_1)}_{\frac{(2m-1)-1}{2}+\mathrm{wt}(\lambda_1)} + \mathrm{wt}(\lambda_2) + \mathrm{wt}(\lambda_3) \\ & \leq & (m-1) + \beta_1 + 2\beta_2 + \beta_3. \end{array}$$

Soit :

$$\mathrm{wt}(e\lambda_1 + \lambda_2 + \lambda_3) \leq (m-1) + d + \beta_2 \tag{3.4}$$

comme $\beta_1 + \beta_2 + \beta_3 = d$.

Le terme considéré dépend d'une des constantes si $\beta_3 \geq 1$. On a donc $\beta_2 \leq d-1$. Dans ce cas, son degré vérifie

$$\mathrm{wt}(e\lambda_1 + \lambda_2 + \lambda_3) \leq (m-1) + 2d - 1 < 2m - 1 \text{ dès que } d < \frac{m+1}{2}.$$

Si on reconsidère l'attaque d'origine, le cas $d_1 = 3,\ m = 4$ ne vérifie pas cette inégalité : la majoration est donc trop grossière.

3.8.2 Étude au cas par cas

On cherche à affiner la majoration précédente en traitant séparément les valeurs élevées de β_2. Rappelons qu'on ne s'intéresse qu'aux termes faisant intervenir des constantes, c'est-à-dire aux cas où $\beta_3 \geq 1$.

Le cas $\beta_2 = d-1$

$\beta_1 = 0$ et $\beta_3 = 1$

On a alors :

$$\begin{aligned} \mathrm{wt}(e\lambda_1 + \lambda_2 + \lambda_3) &= \mathrm{wt}(\lambda_2 + \lambda_3) \\ &\leq 2\beta_2 + \beta_3 \\ &\leq 2(d-1) + 1 \leq 2(m-2) + 1 \\ &\leq 2m - 3 \\ &< 2m - 1. \end{aligned}$$

Le cas $\beta_2 = d-2$

On va placer ici un cas plus général qui consiste à considérer $\beta_2 = d-k$, $\beta_1 = 0$ et $\beta_3 = k$. Comme le cas similaire précédent conduit à un cas favorable pour l'attaque, tous ces cas qui conduisent à des monômes de degré inférieur seront favorables pour l'attaque.

$\beta_1 = 1$ et $\beta_3 = 1$

Le raisonnement que l'on utilise consiste à se demander si on peut avoir

$$\mathrm{wt}(e\lambda_1 + \lambda_2 + \lambda_3) = 2m - 1 \quad .$$

Comme $\mathrm{wt}(\lambda_1) = \mathrm{wt}(\lambda_3) = 1$, on a $\lambda_1 = 2^i$ et $\lambda_3 = 2^j$, donc $\mathrm{wt}(e\lambda_1 + \lambda_2 + \lambda_3) = \mathrm{wt}(2^i e + \lambda_2 + 2^j) = \mathrm{wt}(e + \lambda_2' + 2^k)$. Dans ce cas, cela revient donc à savoir si on peut obtenir $\mathrm{wt}(e + \lambda_2' + 2^k) = 2m - 1$. Or on a :

$$\begin{aligned} \mathrm{wt}(e + \lambda_2' + 2^k) &\leq \mathrm{wt}(e) + \mathrm{wt}(\lambda_2') + 1 \\ &\leq d + d - 2 + 1 = 2d - 1 \\ &\leq 2(m-1) - 1 = 2m - 3. \end{aligned}$$

On peut donc dire que $\beta_2 < d-2$, c'est-à-dire que $\beta_2 \leq d-3 \leq m-4$. En reprenant l'inégalité (3.4), on obtient alors : $\mathrm{wt}(e\lambda_1 + \lambda_2 + \lambda_3) \leq (m-1) + d + (d-3)$ d'où la condition sur d pour que $(m-1) + 2d - 3 < 2m - 1$:

$$d < \frac{m}{2} + \frac{3}{2}.$$

Dans ce cas, on a bien pour $m = 4$ et $d = 3$ l'inégalité vérifiée pour l'attaque d'origine.

Les autres cas

Les autres cas ne permettent pas d'obtenir des résultats généraux. Mais que peut-on déjà conclure de manière concrète de l'inégalité :

$$d < \frac{m}{2} + \frac{3}{2} \ ?$$

On peut regarder à quels degrés correspondent les chiffrements qui sont vulnérables pour cette configuration dans le tableau 3.8.2.

On désigne par

B_1 : la borne $d \leq m - 1$ impliquant :

$$\deg\left(T \circ S_{2m+1}(\mu^{L_{2m+1}} + c_{20}) + T \circ S_{2m+1}(\lambda^{L_{2m+1}} + c_{21})\right) < 2m - 1;$$

B_2 : la borne $d < 1 + \sqrt{2m - 2}$ impliquant

$$\deg\left(g(x)^e\right) < 2m - 1;$$

B_3 : la borne impliquant

$$d < \frac{m+3}{2}.$$

m	taille du bloc	B_1	B_2	B_3	degrés attaquables
2	32	1	2	2	$d = 1$ (sans intérêt)
3	48	2	3	2	$d \leq 2$
4	64	3	3	3	$d \leq 3$ (configuration d'origine)
5	80	4	3	3	$d \leq 3$
6	96	5	4	4	$d \leq 4$
10	160	9	5	6	$d \leq 5$

Notons que le premier cas non-résolu apparaît pour $m = 5$ qui correspond également à la valeur pour laquelle le terme $[g(x)^e]_{2m-1}$ commence à intervenir.

On peut par exemple rajouter d'après cette étude, que quelle que soit la taille du bloc $16m$, pour toute boîte S_{2m+1} de degré 2, tous les chiffrements sont vulnérables à cette attaque différentielle d'ordre $2m - 1$, pour toute boîte S_{2m-1} de degré 3, si $m \geq 4$ et de degré 4 si $m \geq 6$.

En conclusion, l'étude des propriétés des fonctions puissance presque courbes m'a permis d'expliquer les causes de la faiblesse rendant possible une attaque différentielle d'ordre 7 sur $M'1$ et d'étendre la propriété mise en évidence par Babbage et Frisch [BF00]. J'ai également généralisé l'attaque différentielle d'ordre supérieur sur $M'1$ en paramétrant l'algorithme par la taille de bloc du chiffrement.

Il est à noter que compte tenu de cette attaque sur $M'1$, la version modifiée, KASUMI, utilisée dans les mobiles de troisième génération s'est vu rajouter une boîte S_7 supplémentaire, ce qui, en augmentant le degré des bits de sortie, rend impraticable l'attaque précédente.

Par ailleurs, cette étude nous a permis de mettre en évidence le fait que la structure algébrique très forte des fonctions optimales vis-à-vis des attaques différentielles et linéaires pouvaient être à l'origine d'autres faiblesses. Nous allons voir que la vulnérabilité des fonctions presque courbes à la cryptanalyse différentielle d'ordre supérieur dépasse le simple cadre de MISTY1.

Chapitre 4

Cryptanalyse différentielle d'ordre supérieur des chiffrements de Feistel réduits à cinq tours

Comme on l'a vu lors de la cryptanalyse de MISTY1, l'utilisation des fonctions puissance presque courbes $x \mapsto x^s$ dans $\mathbf{F}_2^n$ introduit des faiblesses dans les systèmes les utilisant. Ceci est dû aux propriétés de ces fonctions qui limitent la valeur des degrés possibles lors du produit de leurs composantes booléennes. On va montrer ici de manière plus générale qu'un chiffrement itératif utilisant une fonction presque courbe comme fonction de confusion est vulnérable à une attaque différentielle de degré supérieur, et ceci pour des raisons similaires au cas des fonctions puissance, c'est-à-dire à cause de propriétés de divisibilité de leurs coefficients de Walsh.

4.1 Contexte

On s'intéresse à des chiffrements de Feistel de taille de bloc $2n$, réduits à cinq tours et on cherche à appliquer une attaque différentielle d'ordre supérieur en calculant des différentielles d'ordre moins élevé que le degré a priori de la fonction globale réalisée par le chiffrement à cinq tours.

Comme indiqué dans la partie 2.1.4, on ne prend en compte que $(r-3)$ passages dans la fonction de tour, c'est-à-dire qu'on soumet le chiffrement à une attaque à clair choisi. On considère donc que R_0 est une constante. Pour un chiffrement à cinq tours, on s'intéresse alors au degré de la fonction :

$$\begin{array}{rccc} H: & \mathbf{F}_2^n & \to & \mathbf{F}_2^n \\ & L_0 = x & \mapsto & R_3. \end{array}$$

Considérons des messages clairs de $2n$ bits, de la forme $(x \parallel c_0)$ où $x \in \mathbf{F}_2^n$ et c_0 est une constante fixée dans $\mathbf{F}_2^n$. Si on suit le chiffrement qui produit R_3, on a :

$$R_1 = \underbrace{L_0}_{=x} + \underbrace{F(R_0,k^{(1)})}_{=c_1}$$

$$R_2 = \underbrace{L_1}_{=R_0=c_0} + \underbrace{F(R_1,k^{(2)})}_{=F_2(x+c_1,\, k^{(2)})}$$

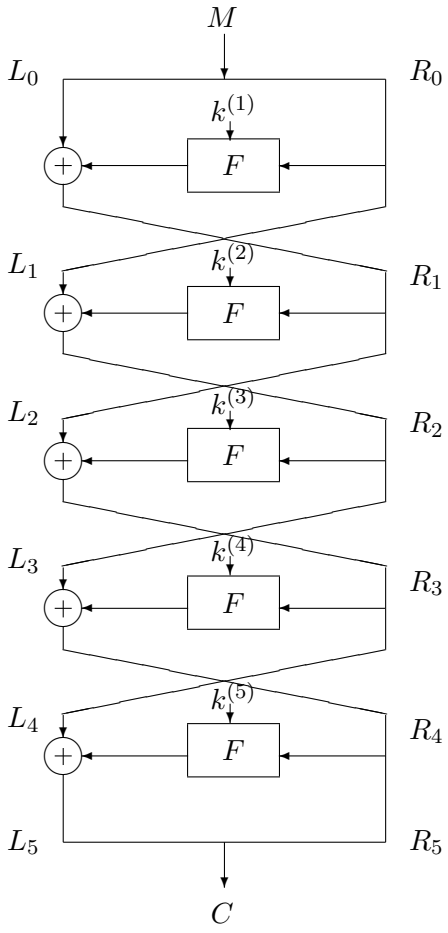


FIG. 4.1 – *Cinq tours d'un chiffrement de Feistel*

$$\begin{aligned} R_3 &= L_2 + F(R_2,k^{(3)}) \\ &= R_1 + F(c_0 + F(x + c_1,k^{(2)}),k^{(3)}) \\ &= x + c_1 + F(c_0 + F(x + c_1,k^{(2)}),k^{(3)}). \end{aligned}$$

L'expression que l'on considère alors est la suivante :

$$R_3 = H(x) = x + c_1 + F(c_0 + F(x + c_1,k^{(2)}),k^{(3)}). \tag{4.1}$$

Le degré a priori de cette expression est $\deg(F)^2$. Cette majoration ne permet donc pas de mener une attaque différentielle d'ordre supérieur dès que $\deg(F) \geq \sqrt{n}$, puisqu'on obtient alors $\deg(F)^2 \geq n$. Cependant, la majoration précédente ne prend en compte aucune des propriétés de la fonction F. Or on sait que les fonctions les plus indiquées pour qu'un chiffrement de Feistel résiste aux attaques différentielle et linéaire sont les fonctions presque courbes. Il convient donc de se pencher sur les propriétés de ces fonctions et en particulier sur leur incidence sur le degré du produit de leurs composantes booléennes. On va déterminer de manière

plus détaillée le degré maximum que l'expression 4.1 peut atteindre pour donner une borne supérieure à l'ordre de la différentielle applicable.

On considère donc les fonctions :

$$\begin{array}{rccl} F_k: & \mathbf{F}_2^n & \to & \mathbf{F}_2^n \\ & x & \mapsto & F(x,k) \quad \text{presque courbe pour tout } k \end{array}$$

$$\begin{array}{rccl} g: & \mathbf{F}_2^n & \to & \mathbf{F}_2^n \\ & x & \mapsto & c_0 + F_{k^{(2)}}(x+c_1) \end{array}$$

$$\begin{array}{rccl} G: & \mathbf{F}_2^n & \to & \mathbf{F}_2^n \\ & x & \mapsto & F_{k^{(3)}} \circ g(x). \end{array}$$

On a donc $H(x) = G(x) + x + c_1$. Il suffit alors de s'intéresser au degré de la fonction G.

Si F est une fonction presque courbe, alors g l'est aussi car elles ne diffèrent qu'à une constante près.

Pour évaluer le degré de la fonction G, on considère les composantes booléennes de F, c'est-à-dire les F_i telles que : $F(x) = (F_1(x),F_2(x),\cdots,F_n(x))$, où les $F_i : \mathbf{F}_2^n \to \mathbf{F}_2$ sont des fonctions booléennes. Puisque F_i est une fonction booléenne, elle possède une forme algébrique normale:

$$F_i(x) = \bigoplus_{u \in \mathbf{F}_2^n} a_u \left(\prod_{j=1}^{n} x_j^{u_j} \right) \quad \text{avec} \quad a_u = \bigoplus_{v \preceq u} F_i(v)$$

où $(v_1,\cdots,v_n) \preceq (u_1,\cdots,u_n)$ signifie que $\forall i, v_i \leq u_i$. Alors, la i-ème composante booléenne de $G = F \circ g$, s'écrit :

$$G_i(x) = F_i(g(x)) = \bigoplus_{u \in \mathbf{F}_2^n} a_u \left(\prod_{j=1}^{n} g_j(x)^{u_j} \right).$$

$F_i \circ g$ apparaît donc comme une somme de produits de composantes de g, le nombre maximum de composantes de g pouvant être multipliées est $\deg F$.

Afin d'évaluer le degré de la fonction G, il est nécessaire de s'intéresser au degré du produit de certaines fonctions booléennes.

4.2 Divisibilité du spectre de Walsh et degré de la composée de deux fonctions

Nous nous intéressons donc au cas d'une fonction composée $F' \circ F$ où F et F' sont deux applications de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$. Nous améliorons la borne triviale

$$\deg(F' \circ F) \leq \deg(F') \deg(F)$$

dans le cas où le spectre de Walsh de F est divisible par une grande puissance de 2. La situation se retrouve typiquement dans le cas où F est une fonction presque courbe (cf. proposition 2.7 page 58).

Définition 4.1 *Le spectre de Walsh d'une fonction F de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$ est dit* divisible par 2^ℓ *si toutes ses valeurs sont divisibles par 2^ℓ. En outre le spectre est dit* exactement divisible par 2^ℓ *si il contient au moins une valeur qui n'est pas divisible par $2^{\ell+1}$.*

La divisibilité des valeurs du spectre de Walsh d'un fonction F fournit une borne supérieure sur son degré. En effet, une conséquence directe de [Car94, Lemme 3], nous permet d'obtenir la borne suivante :

Proposition 4.2 *Soit F une application de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$. Si le spectre de Walsh de F est divisible par 2^ℓ, alors* $\deg(F) \leq n - \ell + 1$.

On peut exprimer la i-ème composante booléenne de $F' \circ F$ sous la forme

$$\big(F' \circ F\big)_i = f'(F_1(x), \ldots, F_n(x)),$$

où f' est la i-ème composante booléenne de F' et $(F_1, \ldots, F_n)$ sont les composantes booléennes de la fonction F. Lorsqu'on introduit ces composantes booléennes dans la forme algébrique normale de f', nous obtenons la forme suivante pour une composante booléenne de la fonction composée :

$$\sum_J \prod_{j \in J} F_j$$

où chaque produit contient au plus $\deg(f')$ composantes booléennes de F. Nous en déduisons que le degré de $F' \circ F$ ne peut être supérieur à celui du produit de $\deg(F')$ composantes booléennes de F.

Il nous faut donc étudier le spectre de Walsh du produit de fonctions booléennes. Pour ce faire nous avons besoin du lemme suivant :

Lemme 4.3 *Soient k fonctions booléennes à n variables $f_1, \ldots, f_k$ avec $k > 0$. On a alors :*

$$\mathcal{F}(\sum_{i=1}^{k} f_i) = 2^{n-1}\left[(-1)^k + 1\right] + \sum_{I \subset \{1,\ldots,k\}} (-2)^{|I|-1} \mathcal{F}(\prod_{i \in I} f_i)\ . \tag{4.2}$$

En outre, pour tout α non-nul de $\mathbf{F}_2^n$, on a :

$$\mathcal{F}(\sum_{i=1}^{k} f_i + \varphi_\alpha) = \sum_{I \subset \{1,\ldots,k\}} (-2)^{|I|-1} \mathcal{F}(\prod_{i \in I} f_i + \varphi_\alpha)\ .$$

Preuve :

– Nous démontrons d'abord la relation (4.2) par récurrence sur k. Elle est trivialement vérifiée pour $k = 1$. Pour $k = 2$, on a

$$\mathrm{wt}(f_1 + f_2) = \mathrm{wt}(f_1) + \mathrm{wt}(f_2) - 2\,\mathrm{wt}(f_1 f_2)\ ,$$

ce qui correspond à la relation (4.2). Supposons maintenant que la relation (4.2) est vérifiée pour tout $i \leq k$. Nous cherchons à montrer qu'elle est également satisfaite pour $(k+1)$ fonctions. Nous pouvons développer l'expression sous la forme suivante :

$$\mathcal{F}(\sum_{i=1}^{k+1} f_i) = \mathcal{F}(\sum_{i=1}^{k} f_i) + \mathcal{F}(f_{k+1}) - 2\mathcal{F}((\sum_{i=1}^{k} f_i) f_{k+1}) + 2^n\ .$$

En appliquant l'hypothèse de récurrence aux fonctions $f_1, \dots f_k$ puis aux fonctions $f_1 f_{k+1}, \dots f_k f_{k+1}$, on obtient :

$$\begin{aligned}\mathcal{F}(\sum_{i=1}^{k+1} f_i) &= 2^{n-1}\left[(-1)^k+1\right] + \sum_{I\subset\{1,\dots,k\}} (-2)^{|I|-1}\mathcal{F}(\prod_{i\in I} f_i) + \mathcal{F}(f_{k+1}) \\ &\quad -2^n\left[(-1)^k+1\right] + \sum_{I\subset\{1,\dots,k\}} (-2)^{|I|}\mathcal{F}(\prod_{i\in I} f_i f_{k+1}) + 2^n \\ &= 2^{n-1}[(-1)^{k+1}+1] + \sum_{I\subset\{1,\dots,k+1\}} (-2)^{|I|-1}\mathcal{F}(\prod_{i\in I} f_i)\ .\end{aligned}$$

– Considérons maintenant un élément non-nul α de $\mathbf{F}_2^n$. Pour toute fonction booléenne f, la relation (4.2) nous donne :

$$\mathcal{F}(f+\varphi_\alpha) = \mathcal{F}(f) + \mathcal{F}(\varphi_\alpha) - 2\mathcal{F}(f\varphi_\alpha) + 2^n = \mathcal{F}(f) - 2\mathcal{F}(f\varphi_\alpha) + 2^n\ .$$

Ainsi nous obtenons

$$\mathcal{F}(\sum_{i=1}^{k} f_i + \varphi_\alpha) = \mathcal{F}(\sum_{i=1}^{k} f_i) - 2\mathcal{F}(\sum_{i=1}^{k} f_i\varphi_\alpha) + 2^n\ .$$

Si nous appliquons maintenant la relation (4.2) aux fonctions $f_1, \dots, f_k, \varphi_\alpha$, nous pouvons écrire :

$$\begin{aligned}\mathcal{F}(\sum_{i=1}^{k} f_i + \varphi_\alpha) &= 2^{n-1}\left[(-1)^{k+1}+1\right] + \mathcal{F}(\varphi_\alpha) + \sum_{I\subset\{1,\dots,k\}} (-2)^{|I|-1}\mathcal{F}(\prod_{i\in I} f_i) \\ &\quad + \sum_{I\subset\{1,\dots,k\}} (-2)^{|I|}\mathcal{F}(\prod_{i\in I} f_i\varphi_\alpha) \\ &= 2^{n-1}\left[(-1)^{k+1}+1\right] + \sum_{I\subset\{1,\dots,k\}} (-2)^{|I|-1}\left[\mathcal{F}(\prod_{i\in I} f_i) - 2\mathcal{F}(\prod_{i\in I} f_i\varphi_\alpha)\right] \\ &= 2^{n-1}\left[(-1)^{k+1}+1\right] + \sum_{I\subset\{1,\dots,k\}} (-2)^{|I|-1}\left[\mathcal{F}(\prod_{i\in I} f_i + \varphi_\alpha) - 2^n\right] \\ &= 2^{n-1}\left[(-1)^{k+1}+1\right] + 2^{n-1}\left[\sum_{i=1}^{k}(-2)^i\binom{k}{i}\right] \\ &\quad + \sum_{I\subset\{1,\dots,k\}} (-2)^{|I|-1}\mathcal{F}(\prod_{i\in I} f_i + \varphi_\alpha) \\ &= \sum_{I\subset\{1,\dots,k\}} (-2)^{|I|-1}\mathcal{F}(\prod_{i\in I} f_i + \varphi_\alpha)\ .\end{aligned}$$

□

La relation précédente entre les coefficients de la transformée de Walsh de la somme de k fonctions booléennes et les coefficients de Walsh de leur produit, nous pouvons alors démontrer le théorème suivant :

Théorème 4.4 *Soient k fonctions booléennes à n variables $f_1,\ldots,f_k$ avec $k>0$. Supposons que pour tout sous-ensemble I de $\{1,\ldots,k\}$ nous ayons*

$$\forall \alpha \in \mathbf{F}_2^n,\ \mathcal{F}(\sum_{i\in I} f_i + \varphi_\alpha) \equiv 0 \bmod 2^\ell \ .$$

Alors, pour tout $I \subset \{1,\ldots,k\}$ de taille au plus ℓ, on a

$$\forall \alpha \in \mathbf{F}_2^n,\ \mathcal{F}(\prod_{i\in I} f_i + \varphi_\alpha) \equiv 0 \bmod 2^{\ell+1-|I|} \ . \tag{4.3}$$

Ainsi, nous obtenons une majoration du degré de la fonction produit de ℓ fonctions booléennes :

$$\deg(\prod_{i\in I} f_i) \leq n - \ell + |I| \ .$$

Preuve : Nous démontrons la relation (4.3) par récurrence sur la taille de I. Le résultat est trivialement vérifié pour $|I|=1$. Nous supposons maintenant que (4.3) est vérifiée pour tout I de taille $|I| \leq w$ et nous considérons un sous-ensemble $I \subset \{1,\ldots,k\}$ de taille $w+1$. D'après le lemme 4.3, pour tout $\alpha \in \mathbf{F}_2^n$ on a :

$$(-2)^w \mathcal{F}(\prod_{i\in I} f_i + \varphi_\alpha) \equiv \mathcal{F}(\sum_{i\in I} f_i + \varphi_\alpha) - \sum_{J\subset I, J\neq I} (-2)^{|J|-1} \mathcal{F}(\prod_{j\in J} f_j + \varphi_\alpha) \bmod 2^n \ .$$

Par hypothèse de récurrence, nous en déduisons que

$$(-2)^w \mathcal{F}(\prod_{i\in I} f_i + \varphi_\alpha) \equiv \mathcal{F}(\sum_{i\in I} f_i + \varphi_\alpha) \bmod 2^\ell \ .$$

Nous avons ainsi :

$$\mathcal{F}(\prod_{i\in I} f_i + \varphi_\alpha) \equiv 0 \bmod 2^{\ell-w} \ .$$

La borne supérieure sur le degré se déduit directement de la relation (4.3) et de la proposition 4.2. □

Ce résultat peut également se démontrer en utilisant certaines propriétés générales de la transformée de Fourier (cf. par exemple [Qui04, Th. 51]).

De l'application du théorème précédent aux composantes booléennes d'une application F de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$, nous déduisons le corollaire suivant :

Corollaire 4.5 *Soit F une application de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ telle que son spectre de Walsh soit divisible par 2^ℓ. Alors le degré du produit de tout ensemble de t composantes booléennes de F s'élève au plus à $n-\ell+t$.*

Ainsi, pour toute fonction F' de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$, on a

$$\deg(F' \circ F) \leq n - \ell + \deg(F') \ .$$

Plus particulièrement, lorsque F est une fonction presque courbe, on obtient :

$$\deg(F' \circ F) \leq \frac{n-1}{2} + \deg(F') \ .$$

Comme nous l'avons vu dans l'attaque de MISTY1, le résultat de ce corollaire est déjà connu dans le cas des fonctions puissance. Lorsqu'on identifie l'espace vectoriel $\mathbf{F}_2^n$ avec le corps fini à 2^n éléments $\mathbf{F}_{2^n}$, toute fonction F de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ peut être représentée par un unique polynôme de $\mathbf{F}_{2^n}[X]$, $F(X) = \sum_{u=0}^{2^n-1} a_u X^u$. Le degré de F (dans le sens du maximum des degré de ses composantes booléennes) est donné par $\deg(F) = \max_{u,a_u\neq 0} \mathrm{wt}(u)$, où $\mathrm{wt}(u)$ est le nombre de 1 dans l'écriture en base 2 de u, $u = \sum_{i=0}^{n-1} u_i 2^i$. Le cas des fonctions puissance comme nous l'avons vu est celui qui nous intéresse le plus dans la mesure où toutes les fonctions connues de non-linéarité presque optimale sont équivalentes à des fonctions puissance $x \mapsto x^s$ sur $\mathbf{F}_{2^n}$. Ainsi, si nous écrivons F' sous la forme d'un polynôme univarié $F'(X) = \sum_{u=0}^{2^n-1} a_u X^u$, on obtient lorsque $F : x \mapsto x^s$ l'expression de la composée des fonctions $F' \circ F(x) = \sum_{u=0}^{2^n-1} a_u X^{us \bmod (2^n-1)}$. Ainsi, $\deg(F' \circ F) \leq \max_{u,a_u\neq 0} \mathrm{wt}(us \bmod (2^n-1))$. Cette borne utilisée pour cryptanalyser MISTY1 est reliée à la divisibilité du spectre de Walsh de F par la proposition [CCD99, Coro. 2].

4.3 Majoration du degré de la fonction de chiffrement

Revenons maintenant à la cryptanalyse du chiffrement de Feistel étudié dans la partie 4.1. Il s'agissait de majorer le degré de la fonction G définie par $G(x) = F_{k^{(3)}}(c_0 + F_{k^{(2)}}(x + c_1))$. En appliquant le résultat sur le degré du produit de certaines fonctions booléennes à la fonction G, on obtient alors :

$$\deg(G) = \max_{1\leq i\leq n} \deg(F_i \circ g) \leq \frac{n-1}{2} + \deg(F).$$

La fonction

$$\begin{array}{rccc} H: & \mathbf{F}_2^n & \to & \mathbf{F}_2^n \\ & L_0 = x & \mapsto & R_3. \end{array}$$

vérifie donc $\deg(H) \leq \frac{n-1}{2} + \deg(F)$. Cette majoration est satisfaite quelles que soient les clés $k^{(1)}$, $k^{(2)}$ et $k^{(3)}$ utilisées et pour toute constante c_0. Notons que F étant une fonction presque courbe, on a $\deg(F) \leq \frac{n+1}{2}$.

Ce résultat est à comparer au calcul a priori qui donne à l'issue de trois tours : $\deg(G) \leq (\deg F)^2$. On constate donc que pour un faible degré de F, cette dernière majoration est préférable. Pour de grands degrés de F, la première majoration devient rapidement plus avantageuse.

Si on note d le degré de F, on en déduit que pour tout sous-espace vectoriel $V \subset \{0\} \times \mathbf{F}_2^n$, de dimension $\min\left(d^2+1, \frac{n+1}{2} + d\right)$, on a :

$$\sum_{x\in V} R_3(x \parallel c_0) = 0$$

pour toute valeur de c_0 dans $\mathbf{F}_2^n$. Cette attaque fournit une différentielle d'ordre $\frac{n+1}{2} + d$ dès que $d < \frac{n+1}{2}$. En effet, R_3 s'exprime en fonction du chiffré (L_5,R_5) et de la clé $k^{(5)}$ (cf. page 55) :

$$R_3 = L_5 + F(R_5, k^{(5)}).$$

Considérons l'ensemble E des messages chiffrés (L_5,R_5) correspondant aux textes clairs ($x \parallel c_0$), où x décrit un sous-espace vectoriel de $\mathbf{F}_2^n$ de dimension $t = \min(d^2+1, \frac{n+1}{2} + d)$ et c_0 est

une constante quelconque de $\mathbf{F}_2^n$. Alors la clé $k^{(5)}$ utilisée lors du dernier tour vérifie :

$$\sum_{(L_5,R_5)\in E} L_5 + F(R_5,k^{(5)}) = 0.$$

Il est alors possible de déterminer $k^{(5)}$ à partir de 2^t couples clairs-chiffrés. La complexité de l'attaque est de l'ordre de $2^{t+\ell}$ évaluations de F, où ℓ est le nombre de bits de $k^{(5)}$. Nous résumons ces résultats dans le tableau 4.1.

fonction F_k	divisibilité	Cas général	
		ordre de la différentielle	applicable
n impair $\mathcal{L}(F_k) = 2^{\frac{n+1}{2}}$	$2^{\frac{n+1}{2}}$	$\frac{n+1}{2} + \max\limits_{k\in\mathcal{K}} \deg(F_k)$	à l'exception du cas où $\deg(F_k) = \frac{n+1}{2}$
n pair	$2^{\frac{n}{2}+1}$	$\frac{n}{2} + \max\limits_{k\in\mathcal{K}} \deg(F_k)$	toujours
$\mathcal{L}(F_k) = 2^{\frac{n}{2}+1}$	$2^{\frac{n}{2}}$	$\frac{n}{2} + 1 + \max\limits_{k\in\mathcal{K}} \deg(F_k)$	à l'exception des cas où $\deg(F_k) \in \{\frac{n}{2}, \frac{n}{2}+1\}$

TAB. 4.1 – *Attaque différentielle d'ordre supérieur sur un chiffrement de Feistel à 5 tours utilisant une fonction de confusion F_k hautement non-linéaire : cas général*

4.4 Amélioration de l'attaque différentielle d'ordre supérieur

Il est possible d'améliorer l'attaque précédente dans le cas (valable pour la majorité des chiffrements de Feistel) où la fonction de tour F est paramétrée par ajout de la sous-clé $k^{(i)}$, la fonction chiffrante presque courbe f étant identique à chaque tour, comme sur la figure 4.2.

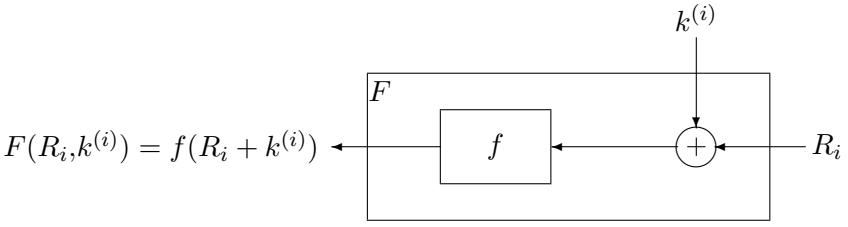


FIG. 4.2 – *Fonction de tour classique d'un chiffrement de Feistel*

Dans ce cas, cette amélioration permet de diviser par 2 la complexité de l'attaque ainsi que le nombre de couples clairs-chiffrés nécessaires. Elle permet en outre de prendre en compte le cas où la fonction itérée presque courbe f, est de degré maximal, c'est-à-dire $\deg(f) = \frac{n+1}{2}$.

En effet, la fonction G étudiée précédemment s'écrit :

$$\begin{array}{rccl} G: & \mathbf{F}_2^n & \to & \mathbf{F}_2^n \\ & x & \mapsto & f\left(k^{(3)} + c_0 + f(x + c_1 + k^{(2)})\right). \end{array}$$

Considérons alors la fonction G' définie par $G'(x) = f\left(k^{(3)} + c_0 + f(x)\right)$. D'après les résultats précédents, G' est de degré au plus $\left(\frac{n-1}{2} + d\right)$. Dans cette expression, les termes faisant intervenir les constantes c_0 ou $k^{(3)}$ sont le produit d'au plus $(d-1)$ composantes de f; ils sont donc de degré au plus $\frac{n-1}{2} + d - 1$. Ainsi, les termes de degré $\left(\frac{n-1}{2} + d\right)$ de G' ne dépendent ni de c_0, ni de $k^{(3)}$. On en déduit donc une différentielle d'ordre $\frac{n-1}{2} + d$ pour G' : pour tout sous-espace vectoriel V de dimension $\frac{n-1}{2} + d$, on a :

$$\forall a \in \mathbf{F}_2^n \ , \ D_V G'(a) = \sum_{v \in V} G'(a+v) = c$$

où la constante c ne dépend ni de c_0 ni des sous-clés. La fonction G correspondant à une translation de G', on obtient :

$$\begin{aligned} \forall a \in \mathbf{F}_2^n \quad \sum_{v \in V} G(a+v) &= \sum_{v \in V} G'(a + v + c_1 + k^{(2)}) \\ &= D_V G'(a + c_1 + k^{(2)}) = c. \end{aligned}$$

On peut calculer la constante c, qui ne dépend pas des sous-clés en prenant $k^{(1)} = k^{(2)} = k^{(3)} = 0$. On obtient ainsi une attaque nécessitant la connaissance de $2^{\frac{n-1}{2}+d}$ couples clairs-chiffrés, et dont la complexité correspond à $2^{n+\frac{n-1}{2}+d}$ évaluations de la fonction f.

fonction F	divisibilité	$F_k(x) = F(x+k)$	
		ordre de la différentielle	applicable
n impair	$2^{\frac{n+1}{2}}$	$\frac{n-1}{2} + \deg(F)$	toujours
n pair	$2^{\frac{n}{2}+1}$	$\frac{n}{2} - 1 + \deg(F)$	toujours
$\mathcal{L}(F) = 2^{\frac{n}{2}+1}$	$2^{\frac{n}{2}}$	$\frac{n}{2} + \deg(F)$	à l'exception du cas où $\deg(F) = \frac{n}{2} + 1$

TAB. 4.2 – *Attaque différentielle d'ordre supérieur sur un chiffrement de Feistel à 5 tours utilisant une fonction de confusion F_k hautement non-linéaire : cas où la clé est insérée par addition*

L'attaque a été implémentée et on donne ci-après un exemple des résultats obtenus. On a utilisé un DEC, processeur EV6 à 500Mhz. La fonction de tour f choisie est une fonction puissance presque courbe avec exposant de Kasami $s = 2^{2i} - 2^i + 1$ avec $i = 5$ soit $s = 241$ et donc de degré $\mathrm{wt}(s) = 5$. On prend comme paramètre n, la taille du bloc. On obtient alors les temps de calcul suivants :

m	ordre de la différentielle	temps de calcul
17	13	20 s.
19	14	3,15 mn
21	15	35 mn

TAB. 4.3 – *Temps de calcul sur un DEC, EV6 à 500 Mhz*

La cryptanalyse d'ordre 7 applicable à l'algorithme $M'1$ et la généralisation que nous avons établi au chapitre 3 nous ont conduit à l'élaboration d'une attaque différentielle d'ordre supérieur applicable à tout chiffrement de Feistel à 5 tours exceptés quelques rares cas de degré maximal pour la fonction de tour. Cette attaque nous permet de souligner le paradoxe lié à l'utilisation de fonctions optimales vis-à-vis d'une famille donnée d'attaques. En effet, la faiblesse des chiffrements précédents à la cryptanalyse différentielle d'ordre supérieur réside dans la résistance maximale qu'ils offrent à la cryptanalyse linéaire et différentielle par l'utilisation de fonctions de tour presque courbes. Ainsi les objets extrémaux semblent ne pas pouvoir fournir les meilleurs candidats pour des fonctions de tour pour les chiffrements itératifs par blocs. Dans notre cas, le critère de conception que nous pouvons en déduire est que le spectre de Walsh de la fonction de confusion ne doit pas être divisible par une grande puissance de 2.

L'utilisation des fonctions presque courbes, qui offrent pourtant une résistance optimale aux attaques différentielle et linéaire, semble donc devoir être évitée. Il paraît alors souhaitable de leur préférer des fonctions sous-optimales pour les critères classiques, mais qui ne possèdent pas cette propriété de divisibilité indésirable. Ainsi, la fonction inverse dans le corps fini à 2^n éléments, utilisée dans l'AES, offre une excellente résistance aux attaques différentielle et linéaire (dans le sens où sa non-linéarité atteint presque la valeur maximale), mais son spectre de Walsh est uniquement divisible par 4, qui est la plus faible divisibilité possible pour une permutation. Il s'agit d'ailleurs de l'unique permutation de la forme x^d possédant la divisibilité minimale [Hel76, Th. 4.7]. La structure algébrique de la fonction inverse peut néanmoins introduire d'autres faiblesses, en particulier vis-à-vis des attaques algébriques [CP02].

Cette attaque rappelle ainsi que l'existence de structures algébriques fortes permettant de donner des preuves de sécurité vis-à-vis d'attaques spécifiques sont des structures qui existent autant pour le concepteur que pour le cryptanalyste. Lorsque le premier les utilise pour démontrer la résistance d'un chiffrement, le second en dispose afin d'en déduire des propriétés pouvant affaiblir le chiffrement vis-à-vis d'autres attaques. Une tâche importante demeure donc la caractérisation de fonctions représentant le meilleur compromis au regard de tous les critères connus et ne possédant pas de fortes structures algébriques potentiellement affaiblissantes. Nous laissons le dernier mot à un autre proverbe cryptographique difficilement traduisible : *What is provably secure is probably not*[1].

1. Phrase due à Lars Knudsen.

Chapitre 5

Propriétés cryptographiques des fonctions booléennes symétriques

Les chapitres qui suivent portent sur l'étude du sous-ensemble des fonctions booléennes symétriques. Une fonction symétrique est une application dont la valeur ne dépend pas de l'ordre des composantes de son entrée. Pour une fonction booléenne, cette propriété signifie que sa valeur ne dépend que du poids de Hamming du vecteur binaire d'entrée. Objets algébriques importants, les fonctions symétriques connaissent de multiples et diverses applications, des réseaux de neurones à la cryptographie. Les fonctions booléennes symétriques ont été étudiées en premier lieu en théorie des circuits où les travaux portant sur leur complexité a donné lieu à de nombreux résultats. En effet, ces fonctions apparaissent naturellement lors de la réalisation de circuits pour des fonctions de seuil ou de tri. On sait en particulier que ces fonctions sont les seules dont on connaît une réalisation en un nombre de portes logiques qui est linéaire en le nombre de variables d'entrée [Weg87]. D'autre part, le fait que leur valeur ne dépende que du poids de leur vecteur d'entrée réduit considérablement la taille de leur représentation puisque le vecteur des valeurs permettant la caractérisation complète d'une fonction booléenne à n variables sans propriété particulière est de taille 2^n, alors que la caractérisation d'une fonction symétrique ne demande qu'un vecteur de taille $n+1$. En contrepartie de cette simplification il faut signaler le cardinal plus faible de l'ensemble considéré, qui, ajoutée à d'autres contraintes, telles que des propriétés cryptographiques à respecter va rapidement réduire le nombre de fonctions utilisables. Néanmoins, dans le cadre d'applications nécessitant un faible encombrement matériel ou logiciel, ces fonctions sont une bonne réponse car elles permettent de maintenir un nombre raisonnable de variables. Pour des systèmes de chiffrement à flot, parvenir à des valeurs acceptables pour des critères cryptographiques telles que la non-linéarité ou le degré algébrique ne peut se faire qu'avec un nombre suffisant de variables. Or une fonction de filtrage de plus de 15 variables, par exemple, ne peut être utilisée en pratique que si elle peut être représentée sous une forme compacte, c'est-à-dire soit par une table statique de petite taille, soit par un circuit électronique contenant peu de portes logiques, ce qui est le cas des fonctions symétriques.

D'un point de vue cryptographique, ces fonctions ont été introduites par Brüer [Brü84] qui présente la symétrie comme un critère permettant d'éviter qu'un motif particulier d'une entrée joue un rôle plus important au sein des entrées de même poids. Cette propriété a été jugée potentiellement trop contraignante par Mitchell [Mit90] car en conjonction avec d'autres critères cryptographiques, cette condition réduit considérablement le nombre de fonctions vérifiant toutes les propriétés requises. Le travail d'énumération de fonctions booléennes vérifiant plusieurs critères cryptographiques entrepris par Mitchell a particulièrement porté sur l'équilibre

et la résilience dans le cas des fonctions symétriques et a été poursuivi par Guo et Yang [YG95] . C'est ainsi que des familles infinies de fonctions symétriques équilibrées [vzGR97, SM03], résilientes [GHS93, vzGR97] et sans corrélation [SM03] ont été trouvées, apportant donc une réponse quant à la question de leur existence. Une conjecture demeure néanmoins sur la résilience maximum de ces fonctions qui semble ne jamais atteindre l'ordre 3 [vzGR97]. Les fonctions symétriques à n variables de non-linéarité maximale ont été en revanche entièrement caractérisées : lorsque n est pair, les fonctions courbes (de non-linéarité $2^{n-1} - 2^{\frac{n}{2}-1}$) sont les fonctions de degré 2 [Sav94]; lorsque n est impair, la non-linéarité maximale vaut $2^{n-1} - 2^{\frac{n-1}{2}}$ et est atteinte par des fonctions dont le spectre de Walsh est tri-valué, qui sont aussi dans ce cas les fonctions quadratiques [MS02]. Par ailleurs, l'apparition récente des attaques algébriques pose également la question de l'immunité algébrique des fonctions symétriques. Si ce critère n'a pas encore été étudié en détail pour cette famille de fonctions, nous savons néanmoins qu'il existe des fonctions d'immunité algébrique maximale, comme la fonction majorité par exemple [KMM05, DMS05].

Dans ce chapitre, je présente tout d'abord les définitions et propriétés générales concernant les fonctions booléennes symétriques. Je définis en particulier les vecteurs caractéristiques des fonctions symétriques dont je vais me servir de manière systématique dans le reste du document, que ce soit pour les valeurs, les coefficients de la forme algébrique normale, le spectre de Walsh ou le spectre d'auto-corrélation. Toutes ces notions, simplifiées dans le cas des fonctions symétriques m'ont permis de calculer les tableaux complets des caractéristiques de fonctions symétriques à n variables, auxquelles j'ai rajouté le calcul de l'immunité algébrique — laquelle valeur ne se simplifiant pas dans ce cas, nous ne pouvons calculer que jusqu'à des valeurs raisonnables de n (c'est-à-dire 16 ou 17 pour l'instant). Dans la partie suivante, je rappelle les résultats connus concernant les fonctions symétriques équilibrées. Vient alors le théorème structurel sur lequel repose une bonne partie des résultats nouveaux apportés par ce travail sur les fonctions symétriques, c'est-à-dire la périodicité du vecteur des valeurs simplifié lié au degré algébrique de la fonction considérée. Ce théorème permet d'améliorer pour les fonctions de degré faible la borne sur l'ordre de résilience maximal d'une fonction symétrique. Enfin, la dernière partie de ce chapitre est dévolue à la caractérisation des dérivées d'une fonction symétrique, caractérisation qui permet d'explorer différentes propriétés telles que le critère de propagation ou l'existence de structures linéaires.

5.1 Vecteurs caractéristiques d'une fonction symétrique

Cette première partie me permet de définir les fonctions symétriques, leurs représentations ainsi que tous les vecteurs qui les caractérisent. En effet, de nombreuses propriétés de fonctions booléennes se simplifient dans le cas des fonctions symétriques. C'est ainsi que nous utiliserons la notion de *simplifié* lorsque le vecteur considéré se rapportera à une fonction symétrique et sera constitué de la suite indexée par le poids des vecteurs de $\mathbf{F}_2^n$. Je m'attache également à relier le travail de J. von zur Gathen et J. R. Roche dans [vzGR97] avec les fonctions booléennes par leur forme numérique normale, ainsi que l'a fait Aline Gouget dans sa thèse [Gou04a]. Il s'agit en fait de montrer en quoi étudier le degré minimum atteignable par les polynômes à coefficients rationnels à valeurs dans {0,1} revient à s'intéresser à l'ordre de résilience maximum atteignable par les fonctions symétriques, résilience à laquelle nous nous intéresserons dans la partie 5.4. Enfin, je profite des simplifications ci-dessus mentionnées afin de donner quelques tableaux exhaustifs de fonctions symétriques et de leur propriétés.

5.1.1 Définitions

Une fonction booléenne à n variables est *symétrique* si elle est invariante par permutation des composantes de son vecteur d'entrée. Nous désignons par $\mathcal{S}ym_n$ l'ensemble des fonctions booléennes symétriques à n variables.

Définition 5.1 *Soit $f \in \mathcal{B}_n$. f est symétrique si et seulement si pour toute permutation σ de l'ensemble S_n des permutations de $\{1,\ldots,n\}$,*

$$f(x_1,x_2,\ldots,x_n) = f(x_{\sigma(1)},x_{\sigma(2)},\ldots,x_{\sigma(n)}).$$

Si on considère la partition de $\mathbf{F}_2^n$ définie par les classes d'équivalence engendrées par l'ensemble des permutations des vecteurs de $\mathbf{F}_2^n$, on constate que tous les éléments d'une même classe d'équivalence sont caractérisés par le même poids. On en déduit donc la caractérisation suivante des fonctions booléennes symétriques.

Proposition 5.2 *Soit $f \in \mathcal{B}_n$. f est symétrique si et seulement si il existe une fonction $v_f : \{0,\ldots,n\} \mapsto \mathbf{F}_2$ telle que :*

$$\forall\ x \in \mathbf{F}_2^n,\ f(x) = v_f(\mathrm{wt}(x)).$$

Ainsi, pour une fonction booléenne symétrique f à n variables, à la place du vecteur des valeurs ordinaire représenté, par $(f(P_0),f(P_1),\ldots,f(P_{2^n-1}))$, nous pouvons considérer ce que nous désignons par *vecteur des valeurs simplifié de f*. La nécessité d'un nom spécifique lorsqu'on considère le vecteur de la fonction v_f est apparue du fait de l'utilisation simultanée des deux notions, vecteur des valeurs classique et vecteur des valeurs simplifié dans certains contextes où la distinction est cruciale.

Définition 5.3 *Soit $f \in \mathcal{S}ym_n$, on appelle* vecteur des valeurs simplifié *de f qu'on note $v(f)$ le vecteur*

$$v(f) = (v_f(0),v_f(1),\ldots,v_f(n)),$$

où $v_f(i) = f(x)$ pour tout vecteur $x \in \mathbf{F}_2^n$ de poids i.

La propriété de symétrie permet également de simplifier l'écriture d'une fonction symétrique sous sa *forme algébrique normale*.

Proposition 5.4 *Soit $f \in \mathcal{B}_n$. f est symétrique si et seulement si sa forme algébrique normale peut s'écrire sous la forme :*

$$\forall\ (x_1,\ldots,x_n) \in \mathbf{F}_2^n,\quad f(x_1,\ldots,x_n) = \bigoplus_{i=0}^{n} \lambda_f(i) \bigoplus_{\substack{u\in\mathbf{F}_2^n \\ \mathrm{wt}(u)=i}} \left(\prod_{j=1}^{n} x_j^{u_j}\right),\quad \lambda_f(i) \in \mathbf{F}_2^n.$$

Nous notons $X_{i,\{1,\ldots,n\}}$, ou plus simplement $X_{i,n}$ lorsqu'il n'y a pas de risque de confusion, le polynôme symétrique élémentaire de degré i en les n variables $(x_1,\ldots,x_n)$:

$$X_{i,n}(x) = \bigoplus_{\substack{u\in\mathbf{F}_2^n \\ \mathrm{wt}(u)=i}} \left(\prod_{j=1}^{n} x_j^{u_j}\right).$$

On remarquera que $X_{i,n}$ est un polynôme homogène, c'est-à-dire dont tous les monômes ont même degré. Cette expression nous permet de noter plus commodément la forme algébrique normale d'une fonction symétrique par :

$$\forall\ (x_1,\ldots,x_n) \in \mathbf{F}_2^n, \quad f(x_1,\ldots,x_n) = \bigoplus_{i=0}^{n} \lambda_f(i) X_{i,n}(x), \quad \lambda_f(i) \in \mathbf{F}_2^n.$$

Cette écriture signifie tout simplement que toute fonction symétrique, qui dans sa forme algébrique normale est un polynôme symétrique sur $\mathbf{F}_2[x_1,\ldots,x_n]/(x_1^2 - x_1,\ldots,x_n^2 - x_n)$, est une combinaison linéaire de polynômes symétriques élémentaires, $\{X_i, 0 \leq i \leq n\}$ base de l'espace des polynômes symétriques.

De la même manière que pour le vecteur des valeurs nous pouvons remplacer les valeurs de la forme algébrique normale ordinaire, $(c_f(P_0),c_f(P_1),\ldots,c_f(P_{2^n-1}))$ par un vecteur de taille n que nous appellerons par analogie *vecteur simplifié de l'ANF* de f.

Définition 5.5 *Soit $f \in \mathcal{S}ym_n$, on appelle* vecteur simplifié de l'ANF *de f qu'on note $\lambda(f)$ le vecteur*

$$\lambda(f) = (\lambda_f(0),\lambda_f(1),\ldots,\lambda_f(n)).$$

Exemple : Considérons la fonction $f \in \mathcal{S}ym_3$ définie par :

$$f(x) = f(x_1,x_2,x_3) = (x_1 \oplus x_2 \oplus x_3) \oplus x_1x_2x_3.$$

Le tableau suivant représente la table de vérité de f ainsi que les coefficients de sa forme algébrique normale.

$x = (x_1,x_2,x_3)$	$\mathrm{wt}(x)$	$f(x)$	$c_f(x)$
(0, 0, 0)	0	0	0
(1, 0, 0), (0, 1, 0), (0, 0, 1)	1	1	1
(1, 1, 0), (1, 0, 1), (0, 1, 1)	2	0	0
(1, 1, 1)	3	0	1

Nous pouvons en déduire que $\lambda(f) = (0,1,0,1)$ et que $v(f) = (0,1,0,0)$. ◇

La relation entre les valeurs d'une fonction et les coefficients de sa forme algébrique normale induit également dans le cas des fonctions symétriques une relation entre le vecteur des valeurs simplifié et le vecteur simplifié de l'ANF :

Proposition 5.6 *Soit $f \in \mathcal{S}ym_n$. Alors $v(f)$ et $\lambda(f)$ vérifient:*

$$\forall\ i \in \{0,\ldots,n\},\ v_f(i) = \bigoplus_{k \preceq i} \lambda_f(k) \quad et \quad \lambda_f(i) = \bigoplus_{k \preceq i} v_f(k)$$

Preuve : Soit $f(x_1,\ldots,x_n) = \bigoplus_{i=0}^{n} \lambda_f(i) X_{i,n}(x)$. Pour un vecteur $x \in \mathbf{F}_2^n$ de poids k, $X_{i,n}(x)$ contient $\binom{i}{k}$ monômes non-nuls. On a alors :

$$v_f(i) = \bigoplus_{k=0}^{i} \binom{i}{k} \lambda_f(k).$$

Le théorème de Lucas (voir par exemple [Com74, p. 79]) donne l'expression des coefficients binomiaux modulo un nombre premier p. Soient a et b deux entiers dont les représentations p-adiques sont $a = \sum_{i=0}^{e} a_i p^i$ et $b = \sum_{i=0}^{e} b_i p^i$. On a alors :

$$\binom{a}{b} \equiv \prod_{i=0}^{e} \binom{a_i}{b_i} \pmod{p}.$$

Pour $p = 2$, on obtient :

$$\binom{a}{b} \equiv 1 \pmod{2} \quad \text{si et seulement si } \mathrm{supp}(b) \subseteq \mathrm{supp}(a),$$

c'est-à-dire $b \preceq a$ ce qui signifie que $\forall\ i,\ b_i \leq a_i$. On obtient donc :

$$v_f(i) = \bigoplus_{k=0}^{i} \binom{i}{k} \lambda_f(k) = \bigoplus_{k \preceq i} \lambda_f(k).$$

Réciproquement, on peut calculer la valeur des coefficients de la forme algébrique normale de f à partir de son vecteur des valeurs simplifié. Le coefficient en $u \in \mathbf{F}_2^n$ de la forme algébrique normale de f vaut :

$$c_f(u) = \bigoplus_{x \preceq u} f(x),$$

ce que la propriété de symétrie permet d'écrire sous la forme :

$$c_f(u) = \bigoplus_{k=0}^{\mathrm{wt}(u)} \binom{\mathrm{wt}(u)}{k} v_f(k).$$

On remarque d'une part que les valeurs $c_f(u)$, $u \in \mathbf{F}_2^n$ ne dépendent que du poids de u, et d'autre part qu'on peut appliquer le théorème de Lucas. Soit u tel que $\mathrm{wt}(u) = i$, on obtient alors :

$$\lambda_f(i) = \bigoplus_{k=0}^{i} \binom{i}{k} v_f(k) = \bigoplus_{k \preceq i} v_f(k).$$

□

La symétrie des fonctions de $\mathcal{S}ym_n$ se retrouve naturellement dans l'expression de leurs coefficients de Walsh.

Proposition 5.7 *[Sav94, DW97] Soit $f \in \mathcal{B}_n$. f est une fonction booléenne symétrique si et seulement si sa transformée de Walsh est une fonction symétrique à valeurs dans $\mathbf{Z}$. On peut alors représenter le spectre de Walsh de f sous la forme d'un vecteur de taille $n+1$, que nous appellerons* spectre de Walsh simplifié *:*

$$\mathsf{F}(f) = (\mathsf{F}_f(0), \mathsf{F}_f(1), \ldots, \mathsf{F}_f(n)), \quad \mathsf{F}_f(i) \in \mathbf{Z}.$$

En outre, l'expression des coefficients de Walsh de f devient alors :

$$\forall i,\ 0 \leq i \leq n, \quad \mathsf{F}_f(i) = \sum_{w=0}^{n} (-1)^{v_f(w)} P_w(i,n),$$

où P_w est le polynôme de Krawtchouk binaire de degré w, c'est-à-dire le coefficient de x^w dans la forme développée de $(1-x)^i(1+x)^{(n-i)}$,

$$P_w(i,n) = \sum_{k=0}^{w} \binom{i}{k}\binom{n-i}{w-k}(-1)^k.$$

Preuve : Soit a un élément de $\mathbf{F}_2^n$ de poids i. On a alors :

$$\begin{aligned}
\mathcal{F}(f+\varphi_a) &= \sum_{x\in\mathbf{F}_2^n} (-1)^{f(x)\oplus a\cdot x} \\
&= \sum_{w=0}^{n} \sum_{\substack{x\in\mathbf{F}_2^n \\ \mathrm{wt}(x)=w}} (-1)^{v_f(w)\oplus a\cdot x} \\
&= \sum_{w=0}^{n} (-1)^{v_f(w)} \sum_{\substack{x\in\mathbf{F}_2^n \\ \mathrm{wt}(x)=w}} (-1)^{a\cdot x} \\
&= \sum_{w=0}^{n} (-1)^{v_f(w)} \sum_{\substack{x\in\mathbf{F}_2^n \\ \mathrm{wt}(x)=w}} (-1)^{|\mathrm{supp}(a)\cap\mathrm{supp}(x)|}.
\end{aligned}$$

Il faut donc déterminer le nombre de x de $\mathbf{F}_2^n$ de poids w tels que $|\mathrm{supp}(a)\cap\mathrm{supp}(x)| = k$, pour un k donné.

Le vecteur a étant fixé de poids i, pour un x de poids w, cela revient à chercher pour tout $k \le \min(i,w)$, le nombre de façons de choisir k positions pour les 1 de x parmi les 1 de a, les $w-k$ positions restantes étant libres. Ce nombre est $\binom{i}{k}\binom{n-i}{w-k}$. On obtient donc :

$$\begin{aligned}
\sum_{\substack{x\in\mathbf{F}_2^n \\ \mathrm{wt}(x)=w}} (-1)^{|\mathrm{supp}(a)\cap\mathrm{supp}(x)|} &= \sum_{k=0}^{w} \binom{i}{k}\binom{n-i}{w-k}(-1)^k \\
&= P_w(i,n)
\end{aligned}$$

où $P_w(i,n)$ est le polynôme de de Krawtchouk de degré w et de paramètres i et n. On pose en effet, comme d'usage, que $\binom{i}{k} = 0$ lorsque $k > i$. Finalement, on obtient :

$$\mathcal{F}(f+\varphi_a) = \sum_{w=0}^{n} (-1)^{v_f(w)} P_w(i,n)$$

valeur qui ne dépend que de $\mathrm{wt}(a) = i$.

Réciproquement, en utilisant la transformée de Walsh inverse, on voit que si une fonction booléenne f a ses coefficients de Walsh $\mathcal{F}(f+\varphi_a)$ qui ne dépendent que du poids de a, alors f est une fonction symétrique. □

Proposition 5.8 *Soit $f \in \mathcal{S}ym_n$. Alors, pour tout $a \in \mathbf{F}_2^n$, les coefficients $\mathcal{F}(D_a f)$ de l'auto-corrélation de f (coefficients de Walsh en 0 de la dérivée de f relativement à a), ne dépendent que du poids de a :*

$$\mathcal{F}(D_a f) = \mathsf{Ac}_f(\mathrm{wt}(a)).$$

On peut alors représenter ces coefficients sous la forme d'un vecteur de taille $n+1$:

$$\mathsf{Ac}(f) = (\mathsf{Ac}_f(0), \ldots, \mathsf{Ac}_f(n)) \quad \mathsf{Ac}_f(i) \in \mathbf{Z}$$

Preuve : Soit a un élément de $\mathbf{F}_2^n$ de poids i. On a alors :

$$\begin{aligned}\mathcal{F}(D_a f) &= \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x) \oplus f(x+a)} \\ &= \sum_{w=0}^{n} (-1)^{v_f(w)} \sum_{\substack{x \in \mathbf{F}_2^n \\ \mathrm{wt}(x)=w}} (-1)^{f(x+a)}.\end{aligned}$$

On a $\mathrm{wt}(x+a) = \mathrm{wt}(x) + \mathrm{wt}(a) - 2\,|\mathrm{supp}(a) \cap \mathrm{supp}(x)|$.

Il faut donc déterminer le nombre de x de $\mathbf{F}_2^n$ de poids w tels que $|\mathrm{supp}(a) \cap \mathrm{supp}(x)| = k$, pour un k donné, calcul effectué dans la démonstration de la proposition 5.7. On a donc :

$$\sum_{\substack{x \in \mathbf{F}_2^n \\ \mathrm{wt}(x)=w}} (-1)^{f(x+a)} = \sum_{k=0}^{w} \binom{i}{k} \binom{n-i}{w-k} (-1)^{v_f(w+i-2k)}.$$

Finalement, on obtient :

$$\mathcal{F}(D_a f) = \sum_{w=0}^{n} (-1)^{v_f(w)} \sum_{k=0}^{w} \binom{i}{k} \binom{n-i}{w-k} (-1)^{v_f(w+i-2k)}$$

valeur qui ne dépend que de $\mathrm{wt}(a) = i$. □

Ainsi, ces vecteurs simplifiés permettent de représenter les fonctions booléennes symétriques de façon compacte : 2^{n+1} fonctions plutôt que 2^{2^n} dans le cas général et des vecteurs caractéristiques de taille $(n+1)$ à la place de 2^n. Nous pouvons donc donner leur liste complète pour des valeurs raisonnables de n (c'est-à-dire pour lesquelles il est encore possible de ne pas avoir de tableaux de plusieurs centaines de pages). Nous en présentons deux pour exemple, pour $n = 4$ (cf. tableau 5.1) et $n = 5$ (cf. tableau 5.2), fonctions de terme constant nul. Dans ces 2 tableaux, la 5$^{\mathrm{e}}$ colonne contient une croix si la fonction est équilibrée. La 6$^{\mathrm{e}}$ donne la valeur de la non-linéarité, la 7$^{\mathrm{e}}$ l'ordre de résilience, la 8$^{\mathrm{e}}$ le degré de propagation et la 9$^{\mathrm{e}}$ l'immunité algébrique.

TAB. 5.1 – *Propriétés des fonctions symétriques sur* $\mathbf{F}_2^n$ *pour* $n = 4$

ANF	valeurs	spectre de Walsh	spectre d'auto-corrélation	équ.	NL	rés.	prop.	AI
0100 0	0101 0	0, 0, 0, 0, 16	16, −16, 16,−16, 16	×	0	3	0	1
0010 0	0011 0	−4, 4, 4, −4, −4	16, 0, 0, 0, 0		6	0	4	2
0110 0	0110 0	−4, −4, 4, 4, −4	16, 0, 0, 0, 0		6	0	4	2
0001 0	0001 0	8, 4, 0, −4, 8	16, 0, 8, 0, 0		4	0	1	1
0101 0	0100 0	8, −4, 0, 4, 8	16, 0, 8, 0, 0		4	0	1	1
0011 0	0010 0	4, 0, 4, 0, −12	16, −8, 8, −8, 16		2	0	0	1
0111 0	0111 0	-12, 0, 4, 0, 4	16, 8, 8, 8, 16		2	0	0	1
0000 1	0000 1	14, 2, −2, 2, −2	16, 12, 12, 12, 12		1	0	0	1
0100 1	0101 1	−2, 2, −2, 2, 14	16, −12, 12, −12, 12		1	0	0	1
0010 1	0011 1	−6, 6, 2, −2, −6	16, 4, 4, −4, −4		5	0	0	2
0110 1	0110 1	−6, −2, 2, 6, −6	16, −4, 4, 4, −4		5	0	0	2
0001 1	0001 1	6, 6, −2, −2, 6	16, 4, 4, −4, −4		5	0	0	2
0101 1	0100 1	6, −2, −2, 6, 6	16, −4, 4, 4, −4		5	0	0	2
0011 1	0010 1	2, 2, 2, 2, −14	16, −12, 12, −12, 12		1	0	0	1
0111 1	0111 1	−14, 2, 2, 2, 2	16, 12, 12, 12, 12		1	0	0	1

TAB. 5.2 – *Propriétés des fonctions symétriques sur* $\mathbf{F}_2^n$ *pour* $n = 5$

ANF	valeurs	spectre de Walsh	spectre d'auto-corrélation	équ.	NL	rés.	prop.	AI
0100 00	0101 01	0, 0, 0, 0, 0, 32	32, −32, 32, −32, 32, −32	×	0	4	0	1
0010 00	0011 00	−8, 0, 8, 0, −8, 0	32, 0, 0, 0, 0, 32		12	0	4	2
0110 00	0110 01	0, −8, 0, 8, 0, −8	32, 0, 0, 0, 0, −32	×	12	0	4	2
0001 00	0001 00	12, 4, 4, −4, −4, 20	32, −8, 16, −8, 16, −8		6	0	0	1
0101 00	0100 01	20, −4, −4, 4, 4, 12	32, 8, 16, 8, 16, 8		6	0	0	1
0011 00	0010 00	12, −4, 4, 4, −4, −20	32, −8, 16, −8, 16, −8		6	0	0	1
0111 00	0111 01	−20, −4, 4, 4, −4, 12	32, 8, 16, 8, 16, 8		6	0	0	1
0000 10	0000 11	20, 8, −4, 0, 4, −8	32, 16, 16, 8, 8, 8		6	0	0	2
0100 10	0101 10	−8, 4, 0, −4, 8, 20	32, −16, 16, −8, 8, −8		6	0	0	1
0010 10	0011 11	−20, 8, 4, 0, −4, −8	32, 16, 16, 8, 8, 8		6	0	0	2
0110 10	0110 10	−8, −4, 0, 4, 8, −20	32, −16, 16, −8, 8, −8		6	0	0	2
0001 10	0001 11	0, 12, 0, −4, 0, 12	32, 8, 8, −8, −8, −32	×	10	0	0	3
0101 10	0100 10	12, 0, −4, 0, 12, 0	32, −8, 8, 8, −8, 32		10	0	0	2
0011 10	0010 11	0, 4, 0, 4, 0, −28	32, −24, 24, −24, 24, −32	×	2	0	0	2
0111 10	0111 10	−28, 0, 4, 0, 4, 0	32, 24, 24, 24, 24, 32		2	0	0	1
0000 01	0000 01	30, 2, −2, 2, −2, 2	32, 28, 28, 28, 28, 28		1	0	0	1
0100 01	0101 00	2, −2, 2, −2, 2, 30	32, −28, 28, −28, 28, −28		1	0	0	1
0010 01	0011 01	−10, 2, 6, 2, −10, 2	32, −4, 4, 4, −4, 28		11	0	0	2
0110 01	0110 00	2, −10, 2, 6, 2, −10	32, 4, 4, −4, −4, −28		11	0	0	2
0001 01	0001 01	10, 6, 2, −2, −6, 22	32, −12, 20, −12, 12, −12		5	0	0	1
0101 01	0100 00	22, −6, −2, 2, 6, 10	32, 12, 20, 12, 12, 12		5	0	0	1
0011 01	0010 01	10, −2, 2, 6, −6, −18	32, −12, 12, −4, 12, −12		7	0	0	2
0111 01	0111 00	−18, −6, 6, 2, −2, 10	32, 12, 12, 4, 12, 12		7	0	0	1
0000 11	0000 10	22, 6, −2, −2, 6, −10	32, 12, 20, 12, 12, 12		5	0	0	1
0100 11	0101 11	−10, 6, −2, −2, 6, 22	32, −12, 20, −12, 12, −12		5	0	0	1
0010 11	0011 10	−18, 6, 6, −2, −2, −10	32, 12, 12, 4, 12, 12		7	0	0	2
0110 11	0110 11	−10, −2, −2, 6, 6, −18	32, −12, 12, −4, 12, −12		7	0	0	2
0001 11	0001 10	2, 10, 2, −6, 2, 10	32, 4, 4, −4, −4, −28		11	0	0	2
0101 11	0100 11	10, 2, −6, 2, 10, 2	32, −4, 4, 4, −4, 28		11	0	0	2
0011 11	0010 10	2, 2, 2, 2, 2, −30	32, −28, 28, −28, 28, −28		1	0	0	1
0111 11	0111 11	−30, 2, 2, 2, 2, 2	32, 28, 28, 28, 28, 28		1	0	0	1

5.1.2 Forme numérique normale d'une fonction symétrique

Nous nous intéressons ici à une autre représentation des fonctions booléennes qui permet de mettre en lien les travaux de von zur Gathen et Roche [vzGR97] avec les fonctions booléennes symétriques. Nous allons dans ce qui suit définir la *forme numérique normale* d'une fonction booléenne et montrer en quoi celle d'une fonction symétrique est reliée à la notion de *gap* d'un vecteur binaire. L'article [vzGR97] étudie les polynômes $p \in \mathbf{R}[x]$ qui, restreints à l'ensemble $\{0,\ldots,n\}$, ne prennent que deux valeurs entières. Plus exactement et sans perte de généralité ils ont limité leur étude à celle des polynômes à valeurs dans {0,1} de degré au plus n. Dans ce cas, les polynômes considérés appartiennent à $\mathbf{Q}[x]$. En fait un tel polynôme à une variable, que nous notons p_f, interpole les valeurs d'une fonction booléenne symétrique f à n variables pour les valeurs du poids des vecteurs d'entrée. Le problème étudié est celui du degré minimal que peut atteindre un tel polynôme, ce qui conduit à définir la notion de *gap* d'un vecteur binaire de taille $n+1$ par $\gamma(v(f)) = n - \deg(p_f)$, où $v(f) = (v_f(0),\ldots,v_f(n)) = (p_f(0),\ldots,p_f(n))$. Nous montrons qu'étudier ce problème revient à étudier de manière détournée la résilience de la fonction $f+\varphi_{\mathbf{1}}$. Nous nous référerons donc à cet article fréquemment en ce qui concerne les propriétés d'équilibre et de résilience des fonctions booléennes symétriques, dans la mesure où ces travaux sont ceux qui prouvent et recensent le plus de familles infinies de fonctions équilibrées ou résilientes. L'article propose par ailleurs une conjecture encore sans démonstration à ce jour sur l'ordre de résilience maximal d'une fonction booléenne symétrique.

Conjecture 5.9 *[vzGR97] Soit $f \in \mathcal{S}ym_n$. Alors f est au plus* 3*-résiliente.*

Résultats généraux sur la NNF d'une fonction booléenne

Plusieurs représentations polynomiales existent pour une même fonction booléenne. Outre l'omniprésente forme algébrique normale, citons également la *forme numérique normale*, puisqu'elle va nous servir à relier les travaux de von zur Gathen et Roche dans [vzGR97] avec les fonctions booléennes symétriques. Cette représentation a été introduite par C. Carlet et P. Guillot [CG99] dans le cadre des fonctions booléennes appliquées à la cryptographie et aux codes correcteurs d'erreurs. Elle permet de représenter toute fonction booléenne sous la forme d'un polynôme multivarié à coefficients dans $\mathbf{Z}$ et à valeurs dans {0,1}.

Définition 5.10 *Soit f une fonction booléenne à n variables. La* forme numérique normale *(NNF pour* Numerical Normal Form*) de f est l'unique polynôme de $\mathbf{Z}[x_1,\ldots,x_n]/(x_1^2 - x_1,\ldots,x_n^2 - x_n)$ représentant f :*

$$\forall\ x \in \mathbf{F}_2^n, \quad f(x) = \sum_{u \in \mathbf{F}_2^n} \mu_u\ x^u, \quad \mu_u \in \mathbf{Z}, \text{ avec } x^u = x_1^{u_1} x_2^{u_2} \cdots x_n^{u_n}.$$

Le degré numérique *de f, noté $d_{num}(f)$ est alors le degré de sa forme numérique normale.*

Les coefficients de la forme algébrique normale correspondent au reste modulo 2 des coefficients de la forme numérique normale qui se déduisent des valeurs de f par la formule suivante [CG99, Prop. 2] :

$$\mu_u = (-1)^{\mathrm{wt}(u)} \sum_{\substack{a \in \mathbf{F}_2^n \\ a \preceq u}} (-1)^{\mathrm{wt}(a)} f(a).$$

Proposition 5.11 *[CG99]* **(Lien entre la NNF et la transformée de Walsh)** *Soit f une fonction booléenne à n variables et μ_u, $u \in \mathbf{F}_2^n$ les coefficients de sa NNF. Alors,*

$$\forall a \in \mathbf{F}_2^n, \quad \mathcal{F}(f+\varphi_a) = 2^n\delta_0(a) + (-1)^{\mathrm{wt}(a)+1} \sum_{\substack{u\in\mathbf{F}_2^n \\ a\preceq u}} 2^{n-\mathrm{wt}(u)+1}\mu_u$$

et

$$\forall u \in \mathbf{F}_2^n, \quad \mu_u = \frac{1}{2}\delta_0(u) - 2^{-n-1}(-2)^{\mathrm{wt}(u)} \sum_{\substack{a\in\mathbf{F}_2^n \\ u\preceq a}} \mathcal{F}(f+\varphi_a),$$

où $\delta_x(y) = 1$ si $x = y$ et $\delta_x(y) = 0$ sinon.

Dans les formules précédentes, la sommation porte sur l'ensemble des vecteurs dont le support contient le support d'un vecteur donné. Cette configuration étant peu maniable, on préférera les formules suivantes, qui donnent les coefficients de Walsh de la fonction $f+\varphi_{\mathbf{1}}$ en fonction de la NNF de f, où $\mathbf{1}$ est le vecteur tout-à-un de $\mathbf{F}_2^n$.

Corollaire 5.12 *Soit f une fonction booléenne à n variables et μ_u, $u \in \mathbf{F}_2^n$ les coefficients de sa NNF. Alors,*

$$\forall a \in \mathbf{F}_2^n, \quad \mathcal{F}((f+\varphi_{\mathbf{1}})+\varphi_a) = 2^n\delta_{\mathbf{1}}(a) + (-1)^{n-\mathrm{wt}(a)+1} \sum_{\substack{u\in\mathbf{F}_2^n \\ \bar{u}\preceq a}} 2^{n-\mathrm{wt}(u)+1}\mu_u \tag{5.1}$$

et

$$\forall u \in \mathbf{F}_2^n, \quad \mu_u = \frac{1}{2}\delta_0(u) - 2^{-n-1}(-2)^{\mathrm{wt}(u)} \sum_{\substack{a\in\mathbf{F}_2^n \\ a\preceq \bar{u}}} \mathcal{F}((f+\varphi_{\mathbf{1}})+\varphi_a). \tag{5.2}$$

Preuve : On utilise le fait que, pour tout a, $\mathcal{F}((f+\varphi_{\mathbf{1}})+\varphi_a) = \mathcal{F}(f+\varphi_{\bar{a}})$. On déduit donc de la proposition précédente que

$$\mathcal{F}((f+\varphi_{\mathbf{1}})+\varphi_a) = 2^n\delta_0(\bar{a}) + (-1)^{n-\mathrm{wt}(a)+1} \sum_{\substack{u\in\mathbf{F}_2^n \\ \bar{a}\preceq u}} 2^{n-\mathrm{wt}(u)+1}\mu_u.$$

Or $\bar{a} \preceq u$ si et seulement si le complémentaire du support de a est inclus dans le support de u. Ceci équivaut à dire que le complémentaire du support de u est inclus dans le support de a, *i.e.*, $\bar{u} \preceq a$. On a donc

$$\mathcal{F}((f+\varphi_{\mathbf{1}})+\varphi_a) = 2^n\delta_{\mathbf{1}}(a) + (-1)^{n-\mathrm{wt}(a)+1} \sum_{\substack{u\in\mathbf{F}_2^n \\ \bar{u}\preceq a}} 2^{n-\mathrm{wt}(u)+1}\mu_u.$$

De façon similaire, on obtient, pour tout $u \in \mathbf{F}_2^n$,

$$\begin{aligned}
\mu_u &= \frac{1}{2}\delta_0(u) - 2^{-n-1}(-2)^{\mathrm{wt}(u)} \sum_{\substack{a\in\mathbf{F}_2^n \\ u\preceq a}} \mathcal{F}(f+\varphi_a) \\
&= \frac{1}{2}\delta_0(u) - 2^{-n-1}(-2)^{\mathrm{wt}(u)} \sum_{\substack{a\in\mathbf{F}_2^n \\ u\preceq a}} \mathcal{F}((f+\varphi_{\mathbf{1}})+\varphi_{\bar{a}}) \\
&= \frac{1}{2}\delta_0(u) - 2^{-n-1}(-2)^{\mathrm{wt}(u)} \sum_{\substack{a\in\mathbf{F}_2^n \\ u\preceq \bar{a}}} \mathcal{F}((f+\varphi_{\mathbf{1}})+\varphi_{a}) \\
&= \frac{1}{2}\delta_0(u) - 2^{-n-1}(-2)^{\mathrm{wt}(u)} \sum_{\substack{a\in\mathbf{F}_2^n \\ a\preceq \bar{u}}} \mathcal{F}((f+\varphi_{\mathbf{1}})+\varphi_{a})
\end{aligned}$$

où la dernière égalité est obtenue en utilisant le fait que $u \preceq \bar{a}$ si et seulement si $a \preceq \bar{u}$. □

Ce corollaire permet de faire le lien entre les coefficients de la NNF de f et l'ordre de résilience de la fonction $(f + \varphi_{\mathbf{1}})$.

Proposition 5.13 *[CG01, Prop. 1] Soit f une fonction booléenne à n variables et μ_u, $u \in \mathbf{F}_2^n$ les coefficients de sa NNF. Alors, la fonction $(f + \varphi_{\mathbf{1}})$ est t-résiliente si et seulement si*

$$\forall u \in \mathbf{F}_2^n,\ \mathrm{wt}(u) \geq n - t,\ \mu_u = 0.$$

Cela signifie en d'autres termes que la fonction $f+\varphi_{\mathbf{1}}$ est t-résiliente si et seulement si le degré numérique de f est inférieur ou égal à $n - t - 1$.

Preuve : La fonction $(f + \varphi_{\mathbf{1}})$ est t-résiliente si et seulement si

$$\forall a \in \mathbf{F}_2^n,\ \mathrm{wt}(a) \leq t,\ \mathcal{F}((f + \varphi_{\mathbf{1}}) + \varphi_a) = 0.$$

La formule (5.2) montre que si $(f + \varphi_{\mathbf{1}})$ est t-résiliente, alors $\mu_u = 0$ pour tout u tel que $\mathrm{wt}(u) \geq n - t$. Réciproquement, on déduit de la formule (5.1) que $(f + \varphi_{\mathbf{1}})$ est t-résiliente dès que $\mu_u = 0$ pour tout u de poids supérieur ou égal à $n - t$. □

Application aux fonctions symétriques

Proposition 5.14 (NNF d'une fonction symétrique)
Soit f une fonction symétrique à n variables et $(v_0, \ldots, v_n)$ son vecteur des valeurs simplifié. Alors, les coefficients μ_u, $u \in \mathbf{F}_2^n$ de la NNF de f ne dépendent que du poids de u. Ils sont donnés par

$$\mu_u = (-1)^{\mathrm{wt}(u)} \sum_{w=0}^{\mathrm{wt}(u)} (-1)^w \binom{\mathrm{wt}(u)}{w} v_w.$$

Preuve : Soit $u \in \mathbf{F}_2^n$. On a :

$$\begin{aligned}
\mu_u &= (-1)^{\mathrm{wt}(u)} \sum_{\substack{a \in \mathbf{F}_2^n \\ a \preceq u}} (-1)^{\mathrm{wt}(a)} f(a) \\
&= (-1)^{\mathrm{wt}(u)} \sum_{w=0}^{\mathrm{wt}(u)} \sum_{\substack{a \in \mathbf{F}_2^n \\ a \preceq u \\ \mathrm{wt}(a)=w}} (-1)^w v_w \\
&= (-1)^{\mathrm{wt}(u)} \sum_{w=0}^{\mathrm{wt}(u)} (-1)^w v_w \#\{a \in \mathbf{F}_2^n,\ a \preceq u \text{ et } \mathrm{wt}(a) = w\}.
\end{aligned}$$

Le nombre de vecteurs $a \in \mathbf{F}_2^n$ de poids w tels que $a \preceq u$ correspond au nombre de manières de choisir w positions dans le support de u, *i.e.*, $\binom{\mathrm{wt}(u)}{w}$. On a donc

$$\mu_u = (-1)^{\mathrm{wt}(u)} \sum_{w=0}^{\mathrm{wt}(u)} (-1)^w \binom{\mathrm{wt}(u)}{w} v_w.$$

□

En combinant la proposition 5.13 avec l'expression des coefficients de la NNF d'une fonction symétrique, on déduit immédiatement :

Corollaire 5.15 *Soit f une fonction symétrique à n variables et $(v_0, \ldots, v_n)$ son vecteur des valeurs simplifié. Alors, la fonction $(f + \varphi_{\mathbf{1}})$ est t-résiliente si et seulement si*

$$\forall m,\ n - t \leq m \leq n,\ \sum_{w=0}^{m} (-1)^w \binom{m}{w} v_w = 0.$$

Remarque 5.16 *La fonction $g = f + \varphi_{\mathbf{1}}$ est symétrique si et seulement si f est symétrique. Dans ce cas, ces deux fonctions sont liées par les propriétés suivantes :*

- *le vecteur des valeurs de g est donné par*

$$\forall i,\ 0 \leq i \leq n,\ v_g(i) = v_f(i) \oplus (i \bmod 2).$$

- *le spectre de Walsh de g est donné par*

$$\forall i,\ 0 \leq i \leq n,\ F_g(i) = F_f(n - i),$$

où $F_f(i)$ est la valeur des coefficients de Walsh de f en un vecteur de poids i. Cette dernière formule est simplement déduite du fait que $\mathcal{F}(g + \varphi_a) = \mathcal{F}(f + \varphi_{\bar{a}})$.

La caractérisation du gap $\gamma(v(f))$ d'un vecteur binaire $v(f)$ de taille $n+1$ fournie par le théorème 2.2 de [vzGR97] peut se formuler de la manière suivante.

Proposition 5.17 *Soit f une fonction symétrique à n variables et $v(f) \in \mathbf{F}_2^{n+1}$ son vecteur des valeurs. Alors, la fonction $(f + \varphi_{\mathbf{1}})$ est t-résiliente si et seulement si le gap du vecteur $v(f)$ est égal à $t + 1$.*

Les résultats obtenus par von zur Gathen et Roche portent sur un polynôme univarié p_f interpolant une fonction symétrique f telle que $f + \varphi_{\mathbf{1}}$ est $\gamma(v(f)) - 1$-résiliente. Nous nous attacherons par la suite à transposer les caractéristiques des polynômes p_f en caractéristiques pour la fonction $f + \varphi_{\mathbf{1}}$.

5.2 Les fonctions symétriques équilibrées

Une grande part de l'intérêt porté aux fonctions booléennes symétriques s'est investi dans l'étude des propriétés d'équilibre et de résilience de ces fonctions. Nous allons ici nous intéresser d'abord au cas des fonctions équilibrées qui présente quelques propriétés remarquables et qui nous servira de base à l'étude de la résilience. Nous mettons d'ores et déjà de côté les fonctions symétriques affines, évidemment équilibrées et qui correspondent aux fonctions f de vecteur des valeurs simplifié $v_f(i+1) = v_f(i) \oplus 1$, pour tout i, $0 \leq i \leq n-1$.

5.2.1 Les fonctions équilibrées triviales

Une première famille de fonctions équilibrées de manière quasi-évidente est celle que nous appelons *triviale*. Soit f une fonction booléenne symétrique à n variables, son poids $\mathrm{wt}(f)$ vaut alors :

$$\mathrm{wt}(f) = \sum_{i=0}^{n} \binom{n}{i} v_f(i).$$

Nous étudions plutôt la quantité équivalente $\mathcal{F}(f)$ qui s'annule lorsque f est équilibrée.

$$\mathcal{F}(f) = 2^n - 2\,\mathrm{wt}(f) = \sum_{i=0}^{n} \binom{n}{i} (-1)^{v_f(i)}.$$

En utilisant la propriété de symétrie des coefficients binomiaux, $\binom{n}{i} = \binom{n}{n-i}$, on obtient alors l'expression du poids de f :

$$\begin{aligned}
\mathcal{F}(f) &= \sum_{i=0}^{\frac{n-1}{2}} \binom{n}{i} \left((-1)^{v_f(i)} + (-1)^{v_f(n-i)} \right) , \quad \text{si } n \text{ est impair,} \\
&= \sum_{i=0}^{\frac{n}{2}-1} \binom{n}{i} \left((-1)^{v_f(i)} + (-1)^{v_f(n-i)} \right) + \binom{n}{\frac{n}{2}} (-1)^{v_f(\frac{n}{2})} , \quad \text{si } n \text{ est pair.}
\end{aligned}$$

Dans le cas où n est impair, on voit immédiatement qu'une condition suffisante pour que f soit équilibrée, c'est-à-dire pour que $\mathcal{F}(f) = 0$, est :

$$\forall\ 0 \leq i \leq n,\ v_f(i) = v_f(n-i) \oplus 1.$$

Définition 5.18 *Soient n un entier impair et $f \in \mathcal{S}ym_n$. La fonction f est une fonction* équilibrée triviale *si et seulement si :*

$$\forall\ 0 \leq i \leq n,\ v_f(i) = v_f(n-i) \oplus 1.$$

Les fonctions équilibrées triviales correspondent exactement aux fonctions symétriques f qui vérifient $D_{\mathbf{1}} f = 1$. Ces fonctions n'existent pas lorsque n est pair car pour tout vecteur u de poids $\mathrm{wt}(u) = n/2$, $D_{\mathbf{1}} f(u) = f(u) \oplus f(u+\mathbf{1}) = v_f(n/2) \oplus v_f(n/2) = 0$. Ce motif remarquable de vecteur des valeurs est aussi nommé motif anti-palindrome [BPP01], [SM03]. Les fonctions équilibrées triviales correspondent également au cas impair des partitions triviales définies par Mitchell dans [Mit90, Th. 3.6.5] , les partitions triviales dans le cas pair correspondant aux fonctions affines. Une partition de l'ensemble des n coefficients binomiaux produisant une

fonction symétrique équilibrée telle que définie dans [Mit90] est en fait équivalente au motif du vecteur des valeurs simplifié d'une fonction symétrique équilibrée.

Exemple : Une fonction symétrique équilibrée triviale à 5 variables

Soit $f \in \mathcal{S}ym_5$ avec $v(f) = (0,0,1,0,1,1)$. On a alors :

$$\begin{aligned}\lambda(f) &= (0,0,1,1,1,0)\\ \mathsf{F}(f) &= (0,4,0,4,0,-28)\\ \mathsf{Ac}(f) &= (32,-24,24,-24,24,-32)\end{aligned}$$

◇

Les fonctions équilibrées triviales sont caractérisées par les propriétés équivalentes suivantes :

Proposition 5.19 *Soient n un entier impair et $f \in \mathcal{S}ym_n$. Les propriétés suivantes sont équivalentes :*

- **(i)** *Pour tout i, $0 \leq i \leq n$, $v_f(i) = v_f(n-i) \oplus 1$;*
- **(ii)** *La dérivée de f relativement au vecteur tout-à-un, $\mathbf{1}$, est la fonction constante 1 ;*
- **(iii)** *Pour tout i pair tel que $0 \leq i \leq n$, $\mathsf{F}_f(i) = 0$;*
- **(iv)** *Pour tout $a \in \mathbf{F}_2^n$, $D_{a+\mathbf{1}}f = D_a f + 1$ ce qui implique en particulier que pour tout i, $0 \leq i \leq n$, $\mathsf{Ac}_f(n-i) = -\mathsf{Ac}_f(i)$.*

Preuve : Montrons que les propriétés **(ii)** et **(iii)** sont équivalentes. Soit $H_{\mathbf{1}}$ l'hyperplan $\{0,\mathbf{1}\}^{\perp}$ qui correspond à l'ensemble des vecteurs de poids pair. D'après [CCCF01, Lemma V.2], on a l'égalité suivante :

$$\sum_{a \in H_{\mathbf{1}}} \mathcal{F}^2(f + \varphi_a) = 2^{n-1} \left(\mathcal{F}(D_0 f) + \mathcal{F}(D_{\mathbf{1}} f)\right).$$

Ceci implique que $\mathcal{F}(D_{\mathbf{1}}f) = -2^n$ si et seulement si $\forall\, a \in H_{\mathbf{1}}$, $\mathcal{F}(f + \varphi_a) = 0$.

Montrons maintenant que les propriétés **(ii)** et **(iv)** sont équivalentes. Nous utilisons la relation simple suivante sur la dérivée d'une fonction booléenne f relativement à la somme de deux vecteurs a et b :

$$D_{a+b}f = D_a f + D_b f + D_a D_b f.$$

Si on applique cette égalité dans le cas où $b = \mathbf{1}$, on obtient :

$$\begin{aligned}D_{a+\mathbf{1}}f &= D_a f + D_{\mathbf{1}} f + D_a D_{\mathbf{1}} f\\ &= D_a f + 1 + 0.\end{aligned}$$

La réciproque vient du fait que $D_{a+\mathbf{1}}f = D_a f + 1$ ce qui conduit pour $a = 0$ à $D_{\mathbf{1}}f = 1$. □

5.2.2 Fonctions symétriques équilibrées pour $n = p - 1$, p premier

Les résultats de simulation pour un nombre de variables inférieur à 128, montrent que pour ces valeurs, les fonctions équilibrées triviales forment un sous-ensemble très important des fonctions équilibrées. En fait, une recherche exhaustive jusqu'à $n = 128$ effectuée dans [vzGR97] montre que pour n impair, toutes les fonctions équilibrées sont des fonctions équilibrées triviales, sauf pour $n \in \{13,29,31,33,35,41,47,61,63,73,97,103\}$. De même, les fonctions symétriques pour n pair et inférieur à 128 qui ne sont pas affines n'existent que pour $n = 6t+2$

pour certaines valeurs de t ou pour $n \in \{24{,}34{,}48{,}54\}$. En outre, le théorème suivant établit qu'il n'existe pas de fonction symétrique équilibrée pour $n = p - 1$ avec p premier.

Proposition 5.20 *[vzGR97, Th. 2.1] Soient p un entier premier et $f \in \mathcal{S}ym_{p-1}$. Si f est équilibrée, alors f est de degré 1.*

Preuve : Si f est équilibrée, cela signifie que :

$$\sum_{i=0}^{p-1} \binom{p-1}{i} (-1)^{v_f(i)} = 0.$$

Or pour tout $i < p$, on a :

$$\begin{aligned}
\binom{p-1}{i} &= \frac{(p-1)!}{(p-1-i)!i!} \\
&= \frac{1}{i!} \prod_{j=0}^{i-1} (p-1-j) \\
&= \frac{1}{i!} \prod_{j=0}^{i-1} -(j+1) \pmod p \\
&= \frac{1}{i!} \prod_{j=1}^{i} (-j) \pmod p \\
&= \frac{1}{i!} (-1)^i i! \pmod p \\
&= (-1)^i \pmod p
\end{aligned}$$

Donc :

$$\sum_{i=0}^{p-1} \binom{p-1}{i} (-1)^{v_f(i)} = 0 \Rightarrow \sum_{i=0}^{p-1} (-1)^{v_f(i)+i} = 0 \pmod p$$

Or, cette somme ne contient que p termes à valeurs dans $\{\pm 1\}$. Elle s'annule donc modulo p uniquement si elle vaut $\pm p$, c'est-à-dire s'il existe une constante $\varepsilon \in \mathbf{F}_2$ telle que $\forall\ i, 0 \leq i \leq p-1, v_f(i) \oplus i = \varepsilon$. Ce qui signifie que :

$$\forall\ i, 0 \leq i \leq p-1, v_f(i) = i \oplus \varepsilon.$$

Cette égalité définit une fonction affine. □

5.2.3 Les constructions de fonctions équilibrées non-triviales connues

L'idée générale de toutes les constructions de fonctions symétriques résilientes ou sans corrélation telles que présentées dans [GHS93], [vzGR97], [SM03], repose sur l'exploitation de propriétés des coefficients binomiaux qui permettent de résoudre des équations diophantiennes surdéterminées (ce qui conduit donc à chaque fois à des familles infinies de fonctions). La

première propriété exploitée est la symétrie des coefficients binomiaux. Elle permet d'écrire le poids d'une fonction $f \in \mathcal{S}ym_n$ sous la forme :

$$\begin{aligned}\mathrm{wt}(f) &= \sum_{i=0}^{\frac{n-1}{2}} \binom{n}{i} \left(v_f(i) + v_f(n-i)\right) \ , \quad \text{si } n \text{ est impair,} \\ &= \sum_{i=0}^{\frac{n}{2}-1} \binom{n}{i} \left(v_f(i) + v_f(n-i)\right) + \binom{n}{\frac{n}{2}} v_f\left(\frac{n}{2}\right) \ , \quad \text{si } n \text{ est pair.}\end{aligned}$$

Cette symétrie déjà mise en évidence par les fonctions équilibrées triviales conduit naturellement à représenter le vecteur des valeurs simplifié $v(f)$ sous forme dite repliée lorsqu'on s'intéresse au caractère équilibré de f.

Définition 5.21 *Soit n un entier et $f \in \mathcal{S}ym_n$. On appelle* vecteur des valeurs replié de f, *noté $vr(f)$, le vecteur de $\{0,1,2\}^{\lfloor \frac{n-1}{2} \rfloor + 1}$ défini par :*

$$vr_f(i) = v_f(i) + v_f(n-i), \quad 0 \leq i \leq \left\lfloor \frac{n-1}{2} \right\rfloor + 1.$$

À tout vecteur vr de $\{0,1,2\}^{\lfloor \frac{n-1}{2} \rfloor + 1}$ correspond exactement $2^{nb(1)}$ fonctions symétriques si n est impair et $2^{nb(1)+1}$ fonctions si n est pair ou $nb(1)$ désigne le nombre de coefficients égaux à 1 dans vr.

Deux vecteurs des valeurs repliés particuliers apparaissent alors immédiatement :

- le vecteur $vr = (1,1,\ldots,1)$ quand n est impair, qui est le vecteur des valeurs replié de toutes les fonctions équilibrées triviales ;
- le vecteur $vr = (0,2,0,2,\ldots)$ quand n est pair qui est le vecteur des valeurs replié de la fonction linéaire $\varphi_{\mathbf{1}}$.

Ces deux vecteurs repliés correspondant à des fonctions équilibrées jouent un rôle important. En effet, la fonction $f \in \mathcal{S}ym_n$ est équilibrée si et seulement si son vecteur des valeurs replié vérifie l'équation :

$$\begin{aligned}\sum_{i=0}^{\frac{n-1}{2}} \binom{n}{i} vr_f(i) &= 2^{n-1} \ , \quad \text{si } n \text{ est impair,} \\ \sum_{i=0}^{\frac{n}{2}-1} \binom{n}{i} vr_f(i) + \binom{n}{\frac{n}{2}} v_f\left(\frac{n}{2}\right) &= 2^{n-1} \ , \quad \text{si } n \text{ est pair.}\end{aligned}$$

De façon équivalente, on peut remplacer le terme 2^{n-1} par l'expression donnant le poids de la fonction affine $\varphi = \varphi_{\mathbf{1}} + \varepsilon$ en fonction de son vecteur des valeurs replié, *i.e.*,

$$\begin{aligned}&v_\varphi(i) + v_\varphi(n-i) = 1 \ , \quad \text{si } n \text{ est impair} \\ &v_\varphi(i) + v_\varphi(n-i) = 2\left(\varepsilon + (i \bmod 2)\right) \ , \quad \text{si } n \text{ est pair, où } \varepsilon = \lambda_\varphi(0),\end{aligned}$$

et on a donc :

$$2^{n-1} = \sum_{i=0}^{\frac{n-1}{2}} \binom{n}{i}, \quad \text{si } n \text{ est impair,}$$

$$2^{n-1} = \sum_{i=0}^{\frac{n}{2}-1} \binom{n}{i} 2\left(\varepsilon \oplus (i \bmod 2)\right) + \binom{n}{\frac{n}{2}} \left(\varepsilon \oplus \frac{n}{2} \bmod 2\right), \quad \text{si } n \text{ est pair.}$$

Finalement, on cherche à résoudre l'équation $\mathrm{wt}(f) - \mathrm{wt}(\varphi) = 0$, c'est-à-dire :

$$\sum_{i=0}^{\frac{n-1}{2}} \binom{n}{i} \left(vr_f(i) - 1\right) = 0, \text{ si } n \text{ est impair,}$$

$$\sum_{i=0}^{\frac{n}{2}-1} \binom{n}{i} \left(vr_f(i) - 2\left(\varepsilon \oplus (i \bmod 2)\right)\right) + \binom{n}{\frac{n}{2}} \left(v_f\left(\frac{n}{2}\right) - \left(\varepsilon \oplus \frac{n}{2} \bmod 2\right)\right) = 0, \text{ si } n \text{ est pair.}$$

Une hypothèse simplificatrice permet de considérer, sans perte de généralité, que dans le cas pair $v_f\left(\frac{n}{2}\right) = \varepsilon \oplus \frac{n}{2} \bmod 2$, ce qui revient alors à considérer que $\varphi = \varphi_{\mathbf{1}} \oplus v_f\left(\frac{n}{2}\right) \oplus \frac{n}{2} \bmod 2$. Soient, pour tout i, $0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor$, les coefficients :

$$vr_f(i) - 1 \quad \in \{-1,0,1\} \text{ si } n \text{ est impair,}$$

$$vr_f(i) - 2(i \bmod 2) \quad \in \begin{cases} \{-2,-1,0\} & \text{si } n \text{ est pair et } i \text{ impair,} \\ \{0,1,2\} & \text{si } n \text{ est pair et } i \text{ pair,} \end{cases}$$

Ces coefficients mesurent l'écart entre le vecteur replié de f et celui d'une fonction affine. Ils correspondent, au signe près aux coefficients $\alpha(i)$ des *folded vectors* introduits dans [vzGR97], c'est-à-dire qu'on a :

$$\alpha(i) = (-1)^i (vr_f(i) - 1) \quad \in \{-1,0,1\} \text{ si } n \text{ est impair,}$$

$$\alpha(i) = (-1)^i \big(vr_f(i) - 2(i \bmod 2)\big) \quad \in \begin{cases} \{-2,-1,0\} & \text{si } n \text{ est pair et } i \text{ impair,} \\ \{0,1,2\} & \text{si } n \text{ est pair et } i \text{ pair.} \end{cases}$$

Considérons le vecteur $\big(\alpha(0),\alpha(1)\ldots,\alpha(\lfloor \frac{n-1}{2} \rfloor)\big)$, le *folded vector* de [vzGR97]. La recherche exhaustive de fonctions symétriques équilibrées est envisageable et représentable sous cette forme jusqu'à $n = 128$ [vzGR97] et nous avons transposé le tableau de von zur Gathen et Roche dans le tableau 5.6. Une première étape dans une tentative de recherche systématique de familles infinies de fonctions équilibrées réside dans l'annulation de tous les coefficients $\alpha(i)$ sauf quelques coefficients consécutifs parmi ceux définis ci-dessus. Ainsi, nous obtenons les différentes familles infinies que nous allons détailler.

Le cas le plus simple à considérer est celui où deux coefficients consécutifs $(\alpha(k),\alpha(k+1))$ sont non-nuls. Dans ce cas il faut résoudre l'équation :

$$\binom{n}{k} \alpha(k) - \binom{n}{k+1} \alpha(k+1) = 0,$$

c'est-à-dire $(k+1)\alpha(k) - (n-k)\alpha(k+1) = 0$. Lorsque n est impair, il n'y a pas de solution à cette équation si $k \leq \frac{n-3}{2}$. Lorsque n est pair, il faut que les coefficients soient de même signe

et pour que la condition n pair soit respectée, la seule solution est $(\alpha(k),\alpha(k+1)) = (2,1)$. Ceci conduit à la famille infinie de fonctions symétriques équilibrées suivante.

Proposition 5.22 *[vzGR97, Th. 3.6] Soient n un entier pair, k un entier tel que $0 \leq k \leq \frac{n-4}{2}$ et une fonction $f \in \mathcal{S}ym_n$ de vecteur des valeurs simplifié le vecteur $v(f)$ décrit ci-dessous ou son vecteur complémentaire :*

$$v(f) = \big((v_{\varphi_1}(i))_{0\leq i\leq k-1}, \underbrace{1}_{v_f(k)}, \underbrace{\varepsilon}_{v_f(k+1)}, (v_{\varphi_1}(i))_{k+2\leq i\leq n-k-2}, \underbrace{\varepsilon\oplus 1}_{v_f(n-k-1)}, \underbrace{1}_{v_f(n-k)}, (v_{\varphi_1}(i))_{n-k+1\leq i\leq n}\big),$$

où $\varepsilon \in \{0,1\}$. Alors f est équilibrée si et seulement si

$$n = 6t+2 \text{ et } k = 2t \text{ avec } t \geq 1.$$

La proposition se démontre en résolvant l'équation $2(k+1) = n-k$. Dans [SM03], une caractérisation similaire est démontrée. Les premiers exemples de telles fonctions sont obtenus pour $n = 8$ et $n = 14$. Les fonctions correspondantes ont les caractéristiques suivantes :

TAB. 5.3 – *Fonctions symétriques équilibrées non-affine de terme constant nul pour $n = 6t+2$ et $k = 2t$ avec $t \geq 1$.*

n	ANF	valeurs	spectre de Walsh	spectre d'auto-corrélation	NL	deg
8	0110 0001 0	0110 0111 0	0, 28, −24, −12, 16, 12, −8, −28, 32	256, 24, 24, −8, 0, −8, −16, 24, 32	112	7
8	0111 0101 0	0111 0011 0	0, −28, −24, 12, 16, −12, −8, 28, 32	256, 24, 24, −8, 0, −8, −16, 24, 32	112	7
14	0100 1111 0110 100	0101 1101 0011 010	0, −1144, −440, 176, 80, −40, 24, 0, −96, 40, 232, −176, −528, 1144, 8376	16384, −6088, 7848, −4328, 5408, −4256, 4448, −4288, 3888, −4256, 3536, −4328, 3888, −6088, 8376	4004	12
14	0100 1010 0011 100	0101 1001 0111 010	0, 1144, −440, −176, 80, 40, 24, 0, −96, −40, 232, 176, −528, −1144, 8376	16384, −6088, 7848, −4328, 5408, −4256, 4448, −4288, 3888, −4256, 3536, −4328, 3888, −6088, 8376	4004	12

Il faut également y ajouter les fonctions avec un terme constant non-nul.

L'étape suivante consiste à considérer trois coefficients consécutifs $(\alpha(k-1),\alpha(k),\alpha(k+1))$ non-nuls. Dans ce cas, von zur Gathen et Roche ont démontré que les deux seules solutions possibles sont celles que nous présentons ci-après. Le premier résultat s'énonce comme suit :

Proposition 5.23 *[vzGR97, Th. 3.8] Soient n un entier pair, k un entier tel que $1 \leq k \leq \frac{n-4}{2}$ et une fonction $f \in \mathcal{S}ym_n$ de vecteur des valeurs simplifié le vecteur $v(f)$ décrit ci-dessous ou son vecteur complémentaire :*

$$v(f) = \big((v_{\varphi_1}(i))_{0\leq i\leq k-2}, \underbrace{\varepsilon_1}_{v_f(k-1)}, \underbrace{k\oplus 1}_{v_f(k)}, \underbrace{\varepsilon_2}_{v_f(k+1)}, (v_{\varphi_1}(i))_{k+2\leq i\leq n-k-2}, \underbrace{\varepsilon_2\oplus 1}_{v_f(n-k-1)}, \underbrace{k\oplus 1}_{v_f(n-k)}, \underbrace{\varepsilon_1\oplus 1}_{v_f(n-k+1)}, (v_{\varphi_1}(i))_{n-k+2\leq i\leq n}\big),$$

où $\varepsilon_1, \varepsilon_2 \in \{0,1\}$. *Alors* f *est équilibrée si et seulement si*

$$n = 4t^2 - 2 \text{ et } k = 2t^2 - t - 1, \text{ avec } t \geq 2.$$

Le premier exemple de telles fonctions est obtenu pour $n = 14$.

TAB. 5.4 – *Fonctions symétriques équilibrées non-affines à* 14 *variables de terme constant nul pour* $n = 4t^2 - 2$ *et* $k = 2t^2 - t - 1$ *avec* $t \geq 2$.

n	ANF	valeurs	spectre de Walsh	spectre d'auto-corrélation	NL	deg
14	0100 1010 1010 000	0101 1001 1001 010	0, 0, 176, −176, −128, 128, 112, −112, −128, 128, 176, −176, 0, 0, 368	16384, −368, 368, 72, −72, 0, 0, −32, 32, 0, 0, 72, −72, −368, 368	8008	10
14	0100 0110 0110 000	0101 0011 0011 010	0, 0, 176, 176, −128, −128, 112, 112, −128, −128, 176, 176, 0, 0, 368	16384, −368, 368, 72, −72, 0, 0, −32, 32, 0, 0, 72, −72, −368, 368	8008	10
14	0100 0101 1001 110	0101 0001 1011 010	0, 1716, 176, −220, −128, 52, 112, −28, −128, 52, 176, −220, 0, 1716, 368	16384, −368, 4328, 72, 2088, 0, 672, −32, −640, 0, −2160, 72, −4032, −368, 368	7334	13
14	0100 1001 0101 110	0101 1011 0001 010	0, −1716, 176, 220, −128, −52, 112, 28, −128, −52, 176, 220, 0, −1716, 368	16384, −368, 4328, 72, 2088, 0, 672, −32, −640, 0, −2160, 72, −4032, −368, 368	7334	13

Le second résultat avec trois coefficients consécutifs non-nuls fait intervenir la suite de Fibonacci définie par $F_0 = 0$, $F_1 = 1$ et $F_i = F_{i-1} + F_{i-2}$ pour tout $i \geq 2$. On peut alors énoncer la proposition suivante :

Proposition 5.24 *[vzGR97, Th. 3.9] Soient* n *un entier pair,* k *un entier tel que* $1 \leq k \leq \frac{n-3}{2}$ *et une fonction* $f \in \mathcal{S}ym_n$ *de vecteur des valeurs simplifié le vecteur suivant ou son complémentaire :*

$$v(f) = \big(V_1, \underbrace{k \oplus 1}_{v_f(k-1)}, \underbrace{k \oplus 1}_{v_f(k)}, \underbrace{k \bmod 2}_{v_f(k+1)}, V_2, V_2 + \mathbf{1}, \underbrace{k \bmod 2}_{v_f(n-k-1)}, \underbrace{k \oplus 1}_{v_f(n-k)}, \underbrace{k \oplus 1}_{v_f(n-k+1)}, V_1 + \mathbf{1}\big),$$

où $V_1 \in \mathbf{F}_2^{k-1}$ *et* $V_2 \in \mathbf{F}_2^{\frac{n-1}{2}-k-1}$. *Alors* f *est équilibrée si et seulement si*

$$n = F_{2i+2}F_{2i+3} - 1 \text{ et } k = F_{2i}F_{2i+3}, \text{ avec } i \geq 2 \text{ tel que } i \not\equiv 1 \bmod 3.$$

Les premiers exemples de fonctions obtenus grâce à ce résultat apparaissent pour $n = 103$.

On considère maintenant quatre coefficients successifs $(\alpha(k-2), \alpha(k-1), \alpha(k), \alpha(k+1))$ non-nuls. On obtient ainsi la caractérisation suivante :

Proposition 5.25 *[vzGR97, Th. 3.10] Soient* n *un entier pair,* k *un entier tel que* $2 \leq k \leq \frac{n-3}{2}$ *et une fonction* $f \in \mathcal{S}ym_n$ *de vecteur des valeurs simplifié le vecteur* $v(f)$ *décrit ci-dessous ou son vecteur complémentaire :*

$$v(f) = \big(V_1, \underbrace{k \oplus 1}_{v_f(k-2)}, \underbrace{k \bmod 2}_{v_f(k-1)}, \underbrace{k \bmod 2}_{v_f(k)}, \underbrace{k \oplus 1}_{v_f(k+1)}, V_2,$$
$$V_2 + \mathbf{1}, \underbrace{k \oplus 1}_{v_f(n-k-1)}, \underbrace{k \bmod 2}_{v_f(n-k)}, \underbrace{k \bmod 2}_{v_f(n-k+1)}, \underbrace{k \oplus 1}_{v_f(n-k+2)}, V_1 + \mathbf{1}\big),$$

où $V_1 \in \mathbf{F}_2^{k-2}$ *et* $V_2 \in \mathbf{F}_2^{\frac{n-1}{2}-k-1}$. *Alors* f *est équilibrée si et seulement si*

$$n = 4t^2 - 3 \text{ et } k = 2t^2 - t - 1 \text{ avec } t \geq 2.$$

Les premiers exemples de telles fonctions sont obtenus pour $n = 13$, dans ce cas, nous obtenons les 8 fonctions de terme constant nul que nous détaillons dans le tableau ci-après auxquelles il faut ajouter les fonctions de terme constant non-nul.

TAB. 5.5 – *Fonctions symétriques équilibrées non-affines à* 13 *variables de terme constant nul pour* $n = 4t^2 - 3$ *et* $k = 2t^2 - t - 1$ *avec* $t \geq 2$.

n	ANF	valeurs	spectre de Walsh	spectre d'auto-corrélation	NL	deg
13	0110 1100 1100 00	0110 1100 1100 01	0, −128, −176, −48, 128, 0, −112, 16, 128, 0, −176, −48, 0, −128, 0	8192, −176, 264, 48, −24, 16, −16, −16, 16, −48, 24, 176, −264, 7824, 8192	4008	9
13	0101 1111 1110 00	0100 1100 1101 01	0, 88, −176, 24, 128, −8, −112, −8, 128, 24, −176, 88, 0, 184, 0	8192, −272, 272, 24, −24, 16, −16, −16, 16, −48, 48, 168, −168, 7824, 8192	4004	10
13	0011 1001 1001 10	0010 1100 1100 11	0, −84, −176, −20, 128, 12, −112, 12, 128, −20, −176, −84, 0, −180, 0	8192, −264, 264, 24, −24, 16, −16, −16, 16, −48, 48, 176, −176, 7824, 8192	4006	12
13	0000 1010 1011 10	0000 1100 1101 11	0, 132, −176, 52, 128, 4, −112, −12, 128, 4, −176, 52, 0, 132, 0	8192, −168, 272, 48, −24, 16, −16, −16, 16, −48, 24, 168, −272, 7824, 8192	4008	12
13	0111 0101 0100 00	0111 0011 0010 01	0, −128, 176, −48, −128, 0, 112, 16, −128, 0, 176, −48, 0, −128, 0	8192, −176, 264, 48, −24, 16, −16, −16, 16, −48, 24, 176, −264, 7824, 8192	4008	9
13	0100 0110 0110 00	0101 0011 0011 01	0, 88, 176, 24, −128, −8, 112, −8, −128, 24, 176, 88, 0, 184, 0	8192, −272, 272, 24, −24, 16, −16, −16, 16, −48, 48, 168, −168, 7824, 8192	4004	10
13	0010 0000 0001 10	0011 0011 0010 11	0, −84, 176, −20, −128, 12, 112, 12, −128, −20, 176, −84, 0, −180, 0	8192, −264, 264, 24, −24, 16, −16, −16, 16, −48, 48, 176, −176, 7824, 8192	4006	12
13	0001 0011 0011 10	0001 0011 0011 11	0, 132, 176, 52, −128, 4, 112, −12, −128, 4, 176, 52, 0, 132, 0	8192, −168, 272, 48, −24, 16, −16, −16, 16, −48, 24, 168, −272, 7824, 8192	4008	12

Les résultats suivants concernent des fonctions équilibrées de construction plus complexe.

Proposition 5.26 *[vzGR97, Th. 4.3] Soient* n *un entier pair,* k *un entier tel que* $1 \leq k \leq \frac{n-3}{2}$ *et*

$$n = F_{2i+2}F_{2i+3} \text{ et } k = F_{2i}F_{2i+3}, \text{ avec } i \geq 2 \text{ tel que } i \not\equiv 1 \bmod 3,$$

et une fonction $f \in \mathcal{S}ym_n$ *dont le vecteur des valeurs simplifié est le vecteur* $v(f)$ *défini ci-après ou le vecteur complémentaire :*

$$v(f) = \begin{cases} \big((0)^*|_{k-1} \underbrace{0}_{k-1}, \underbrace{0}_{k}, \underbrace{1}_{k+1}, (10)^*|_{n-2k-3} \underbrace{0}_{n-k+1}, \underbrace{0}_{n-k}, \underbrace{1}_{n-k-1}, (1)^*|_{k-1}\big) & \text{si } k \text{ est impair,} \\ \big((10)^*|_{k-1} \underbrace{1}_{k-1}, \underbrace{1}_{k}, \underbrace{1}_{k+1}, (01)^*|_{n-2k-3} \underbrace{1}_{n-k+1}, \underbrace{1}_{n-k}, \underbrace{0}_{n-k-1}, (01)^*|_{k-1}\big) & \text{si } k \text{ est pair.} \end{cases}$$

Alors f *est équilibrée.*

En fait la fonction f définie ci-dessus est 1-résiliente comme nous le verrons à la proposition 5.44 page 134.

Proposition 5.27 *[vzGR97, Th. 4.4(i)] Soit les entiers $r \geq 2$, pair, n impair et k tels que*

$$n = \frac{4+3\sqrt{2}}{8}\left(3+2\sqrt{2}\right)^r + \frac{4-3\sqrt{2}}{8}\left(3-2\sqrt{2}\right)^r,$$
$$k = \frac{2+\sqrt{2}}{8}\left(3+2\sqrt{2}\right)^r + \frac{2-\sqrt{2}}{8}\left(3-\sqrt{2}\right)^r - \frac{1}{2},$$

et la fonction $f \in \mathcal{S}ym_n$ de vecteur des valeurs simplifié le vecteur $v(f)$ ci-dessous ou son complémentaire :

$$v(f) = \big((v_{\varphi_1}(i)\oplus 1)_{0\leq i\leq k-1}, \underbrace{k \bmod 2}_{v_f(k)}, \underbrace{k\oplus 1}_{v_f(k+1)}, V,$$
$$V+\mathbf{1}, \underbrace{k\oplus 1}_{v_f(n-k-1)}, \underbrace{k \bmod 2}_{v_f(n-k)}, (v_{\varphi_1}(i))_{n-k+1\leq i\leq n}\big),$$

où $V \in \mathbf{F}_2^{\frac{n-1}{2}-k}$. Alors f est équilibrée.

Proposition 5.28 *[vzGR97, Th. 4.4(ii)] Soit les entiers $u \geq 4$, $u \not\equiv 0 \bmod 3$, n pair, k et m tels que*

$$n = 4u^2-2, \quad k = \frac{4u^2-4}{3}, \quad m = 2u^2-u-1,$$

et la fonction $f \in \mathcal{S}ym_n$ de vecteur des valeurs simplifié le vecteur $v(f)$ ci-dessous ou son complémentaire :

$$v(f) = \big((v_{\varphi_1}(i))_{0\leq i\leq k-1}, \underbrace{k \bmod 2}_{v_f(k)}, \underbrace{\varepsilon_1}_{v_f(k+1)}, (v_{\varphi_1}(i))_{k+2\leq i\leq m-2},$$
$$\underbrace{\varepsilon_2}_{v_f(m-1)}, \underbrace{m \bmod 2}_{v_f(m)}, \underbrace{\varepsilon_3}_{v_f(m+1)}, (v_{\varphi_1}(i))_{m+2\leq i\leq n-m-2}, \underbrace{\varepsilon_3\oplus 1}_{v_f(n-m-1)}, \underbrace{m \bmod 2}_{v_f(m)}, \underbrace{\varepsilon_2\oplus 1}_{v_f(n-m+1)},$$
$$(v_{\varphi_1}(i))_{n-m+2\leq i\leq n-k-2}, \underbrace{\varepsilon_1\oplus 1}_{v_f(n-k-1)}, \underbrace{k \bmod 2}_{v_f(n-k)}, (v_{\varphi_1}(i))_{n-k+1\leq i\leq n}\big),$$

Alors f est équilibrée.

La complexité de la dernière proposition ainsi que le tableau que nous reprenons dans la section suivante nous oblige à introduire la notion de *vecteur replié*, similaire au *folded vector* de von zur Gathen et Roche. Dans notre cas il désigne bien le vecteur des valeurs replié, c'est-à-dire le vecteur dont le i-ème coefficient vaut $v_f(i)+v_f(n-i)$ pour $0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor$.

Proposition 5.29 *[vzGR97, Th. 4.4(iii)] Soit les entiers $r \geq 2$, $r \equiv 2 \bmod 4$, n impair et k tels que*

$$n = \frac{4+3\sqrt{2}}{8}\left(3+2\sqrt{2}\right)^r + \frac{4-3\sqrt{2}}{8}\left(3-2\sqrt{2}\right)^r,$$
$$k = \frac{2+\sqrt{2}}{8}\left(3+2\sqrt{2}\right)^r + \frac{2-\sqrt{2}}{8}\left(3-2\sqrt{2}\right)^r - \frac{1}{2},$$

et la fonction $f \in \mathcal{S}ym_n$ *de vecteur replié*

$$v_f(i) + v_f(n-i) = \begin{cases} i \oplus 1 & si\ i \in \{0,\ldots,k-1\} \setminus \{\ell,\ell+1\} \\ 1 & si\ i \in \{\ell+1\} \cup \{k+2,\ldots,\frac{n-1}{2}\} \\ i \bmod 2 & si\ i \in \{\ell,k,k+1\} \end{cases}.$$

Alors $n \equiv 5 \bmod 6$ *et* f *est équilibrée.*

5.2.4 Tableau des fonctions équilibrées connues pour $n \leq 128$

Le tableau 5.6 suivant est la représentation avec des vecteurs repliés du tableau de von zur Gathen et Roche dans [vzGR97]. Cette liste comprend les vecteurs repliés des fonctions symétriques booléennes à n variables pour $2 \leq n \leq 128$ à l'exception des fonctions équilibrées triviales et des fonctions équilibrées pour n de la forme $6t+2$. Nous faisons référence aux théorèmes invoqués et aux propositions auxquelles les vecteurs correspondent. Le nombre de fonctions distinctes représentées par chaque solution replié vaut $2^{nb(1)+1}$ où $nb(1)$ est le nombre de 1 dans le vecteur replié en tenant compte des vecteurs des valeurs complémentaires. Pour les valeurs paires de n, la valeur d'indice $\frac{n}{2}$ a été remplacée par ε_0 afin de matérialiser la valeur pivot.

5.3 Les motifs réguliers du vecteur des valeurs simplifié

Nous avons vu dans la partie précédente que certains motifs réguliers du vecteur des valeurs simplifié d'une fonction symétrique correspondaient à des fonctions équilibrées, le cas *régulier* étant celui des fonctions affines et plus généralement des fonctions équilibrées triviales quand n est impair. Dans cette partie nous allons étudier en détail un autre type de motif régulier du vecteur des valeurs simplifié d'une fonction symétrique : les motifs périodiques. Notion définie de manière assez naturelle, nous verrons qu'elle est reliée au degré de la fonction symétrique représentée. Cette propriété permet de simplifier l'expression du poids d'une fonction symétrique et de déterminer les fonctions symétriques équilibrées pour de petits degrés.

5.3.1 Périodicité du vecteur des valeurs simplifié

Soient $u_0,\ldots,u_{T-1}$, T éléments de $\mathbf{F}_2$ et la suite binaire infinie, $u = (u_n)_{n\in\mathbf{N}}$ définie par $\forall\ n \in \mathbf{N}$, $u_{n+T} = u_n$. La suite u est périodique de période T et sa première période est $u_0,\ldots,u_{T-1}$. On note cette suite $u = (u_0,\ldots,u_{T-1})^*$.

On cherche à étudier les vecteurs des valeurs simplifiés de fonctions de $\mathcal{S}ym_n$, constitués par la répétition périodique de motifs de taille $T \leq n$. L'idée d'étudier les fonctions symétriques dont le vecteur des valeurs simplifié est périodique a pour origine les résultats portant sur la non-linéarité maximale des fonctions booléennes symétriques. En effet, Savicky pour un nombre de variables pair puis Maitra et Sarkar pour un nombre de variables impair ont mis en évidence le fait que les vecteurs des valeurs simplifié des fonctions symétriques de degré 2 correspondent à certains motifs périodiques de période 4.

Proposition 5.30 *[Sav94] Soit* f *une fonction booléenne symétrique à* n *variables,* n *pair, alors les propriétés suivantes sont équivalentes :*

(i) *La fonction* f *est courbe ;*

TAB. 5.6 – *Vecteurs des valeurs repliés, $(v_i + v_{n-i})$, des fonctions symétriques équilibrées (non-triviales) pour $n \leq 128$*

n	vecteur des valeurs replié	prop.	# fonctions
13	1110 220	5.25	2^4
14	0202 101ε_0	5.23	2^3
24	2021 0021 2102 ε_0		2^4
	0222 0121 0121 ε_0		2^5
29	1111 1112 0122 011		2^{11}
31	1112 0112 2120 2002		2^7
	1110 2112 2010 1012		2^9
33	1111 1111 1111 2002 1	5.25	2^{14}
34	0202 0202 0202 0121 0ε_0	5.23	2^3
	0202 0220 0121 1012 0ε_0		2^5
	0202 0220 0100 2012 0ε_0		2^3
	0202 0220 0121 1211 0ε_0		2^6
	0202 0220 0100 2211 0ε_0		2^4
35	2020 2020 2020 2002 11	5.27	2^3
	2020 2020 2022 1002 11	5.29	2^4
38	0200 0100 2202 1100 021ε_0		2^5
41	1111 1010 1110 2202 1201 1		2^{13}
	1111 1010 1110 2012 1201 1		2^{14}
	1111 1202 2210 1210 0211 1		2^{12}
	1111 1202 2210 1020 0211 1		2^{11}
44	0202 0010 0222 0010 0111 21ε_0		2^7
47	1111 1111 1110 1121 2201 1111		2^{20}
48	2021 0011 1201 1011 0101 1202 ε_0		2^{12}
54	0202 0202 0202 0210 0102 1012 020ε_0		2^5
61	1111 1111 1111 1111 1111 1111 1022 011	5.25	2^{28}
62	0202 0202 0202 0202 0202 0202 0210 120ε_0	5.23	2^3
	0202 0202 0202 0202 0202 2102 0210 120ε_0	5.28	2^4
	0202 2002 0211 0012 2211 0122 1012 110ε_0		2^{11}
	0202 2002 0211 0012 2211 0122 1012 110ε_0		2^{11}
	0202 2002 0211 0012 2211 2022 1020 210ε_0		2^8
	0202 2002 0211 0012 2211 0122 1020 210ε_0		2^9
	0202 2002 0201 2121 0022 2210 0022 110ε_0		2^7
63	0202 0021 0121 2020 2120 0020 0220 0211		2^7
73	1111 1111 1111 1110 1000 1101 2101 2201 1111 1		2^{28}
	1111 1111 1111 1110 1211 2222 0120 2201 1111 1		2^{25}
74	0202 0202 0202 0202 2100 0020 1111 2102 0202 0ε_0		2^7
	0202 0202 0202 0202 1200 0121 1021 1012 0202 0ε_0		2^8
	0202 0202 0202 0202 1121 2021 1021 1012 0202 0ε_0		2^9
97	$(1)^*\vert_{30}$ 1111 1111 1111 2002 111	5.25	2^{46}
98	$(02)^*\vert_{30}$ 0202 0202 0202 0121 020ε_0	5.23	2^{20}
	$(02)^*\vert_{30}$ 0221 0202 0202 0121 020ε_0	5.28	2^3
103	$(1)^*\vert_{30}$ 1111 1111 2201 1111 1111 11	5.24	2^{50}
104	$(1)^*\vert_{30}$ 1111 1111 1012 0202 0202 02ε_0	5.44	2^{41}

(ii) *Pour tout* $k \in \{0,\ldots,n-2\}$ *on a* $v_f(k+2) = v_f(k) \oplus 1$ *;*
(iii) *La fonction* f *est quadratique.*

Proposition 5.31 *[MS02, Th. 5] Soient* $n \geq 3$ *un entier impair et* $f \in \mathcal{S}ym_n$ *alors les propriétés suivantes sont équivalentes :*

(i) *la non-linéarité de* f *vaut* $2^{n-1} - 2^{\frac{n-1}{2}}$ *;*
(ii) $v(f)$ *est un vecteur constitué de* $(n+1)$ *éléments contigus de* $(0011)^*$*;*
(iii) *l'ensemble des valeurs du spectre de Walsh de* f *est* $\{0, \pm 2^{\frac{n+1}{2}}\}$*;*
(iv) *le degré de* f *est 2.*

Le fait que $v(f)$ soit un vecteur constitué de $(n+1)$ éléments contigus de $(0011)^*$, signifie que $v(f)$ est un vecteur constitué des $(n+1)$ premières valeurs de $(0011)^*$, $(0110)^*$, $(1100)^*$ ou $(1001)^*$. Nous utiliserons donc la définition suivante :

Définition 5.32 *Un vecteur* $v = (v_0,\ldots,v_{n-1})$ *est* la troncature à n (éléments) de la suite u *s'il est composé des* n *premières valeurs de la suite* u*.*

En outre, v *est* périodique *de période* T *s'il est la troncature à* n *de la suite périodique* $(v_0,\ldots,v_{T-1})^*$*. Nous noterons alors* $v = (v_0,\ldots,v_{T-1})^*|_n$*.*

Nous pouvons maintenant généraliser le lien entre le degré d'une fonction booléenne symétrique et la périodicité de son vecteur des valeurs simplifié.

Théorème 5.33 *Soit* $f \in \mathcal{S}ym_n$ *de vecteur des valeurs simplifié* $v(f) = (v_0,\ldots,v_n)$ *et de vecteur simplifié de l'ANF* $\lambda(f) = (\lambda_0,\ldots,\lambda_n)$*.*

Alors $v(f)$ *est périodique de période* 2^t*, avec* $2^t < n$*, si et seulement si* $\deg(f) \leq 2^t - 1$*.*

En outre, $(v_0,\ldots,v_{2^t-1})$ *est le vecteur des valeurs simplifié de la fonction booléenne symétrique à* (2^t-1) *variables dont le vecteur simplifié de l'ANF est* $(\lambda_0,\ldots,\lambda_{2^t-1})$*.*

Preuve : Montrons que si $v(f)$ est périodique de période 2^t alors $\deg(f) \leq 2^t - 1$. Pour cela, calculons λ_i, pour $i \leq n$ et montrons que les coefficients sont nuls pour $i \geq 2^t$.

On peut décomposer tout entier $i \leq n$ sous la forme $i = i_1 2^t + i_0$ avec $i_0 \leq 2^t - 1$. En appliquant la proposition 5.6, on obtient :

$$\begin{aligned}
\lambda_{i_1 2^t + i_0} &= \bigoplus_{k \preceq (i_1 2^t + i_0)} v_k \\
&= \bigoplus_{k_1 \preceq i_1} \bigoplus_{k_0 \preceq i_0} v_{k_1 2^t + k_0} \\
&= \bigoplus_{k_1 \preceq i_1} \bigoplus_{k_0 \preceq i_0} v_{k_0} \\
&= 2^{\mathrm{wt}(i_1)} \bigoplus_{k_0 \preceq i_0} v_{k_0},
\end{aligned}$$

ce qui signifie que $\lambda_{i_1 2^t + i_0} = 0$ dès que $i_1 \neq 0$, c'est-à-dire dès que $i \geq 2^t - 1$. Cela signifie que, $\deg(f) < 2^t$. On remarque également que les 2^t premiers coefficients du vecteur de l'ANF simplifié, $(\lambda_0,\ldots,\lambda_{2^t-1})$, forment exactement le vecteur simplifié de l'ANF de la fonction booléenne symétrique à (2^t-1) variables dont le vecteur des valeurs simplifié est $(v_0,\ldots,v_{2^t-1})$.

On démontre la réciproque par des calculs similaires. Soit $f \in \mathcal{S}ym_n$ de degré $\deg(f) \leq 2^t - 1$. En utilisant à nouveau la proposition 5.6 ainsi que la décomposition de i, on obtient :

$$\begin{aligned} v_{i_1 2^t + i_0} &= \bigoplus_{k \preceq (i_1 2^t + i_0)} \lambda_k \\ &= \bigoplus_{k \preceq i_0} \lambda_k \\ &= v_{i_0}. \end{aligned}$$

On obtient bien ainsi la périodicité du vecteur des valeurs simplifié de f. Les 2^t premiers termes forment alors le vecteur des valeurs simplifié de la fonction booléenne à $(2^t - 1)$ variables dont le vecteur de l'ANF simplifié est $(\lambda_0, \ldots, \lambda_{2^t-1})$.

□

Ce résultat s'adapte au cas particulier des fonctions de degré exactement 2^t.

Proposition 5.34 *Soit $f \in \mathcal{S}ym_n$. Alors $\deg(f) = 2^t$ si et seulement si $v(f)$ est périodique de période 2^{t+1} et est la troncature à $(n+1)$ éléments de $(v_0, \ldots, v_{2^t-1}, v_0 \oplus 1, \ldots, v_{2^t-1} \oplus 1)^*$.*

Preuve : La preuve est similaire à celle de la proposition précédente. Dans ce cas, on utilise le fait que pour tout $i > 2^t$, $\lambda_i = 0$ et $\lambda_{2^t} = 1$. On obtient alors :

$$\begin{aligned} v_{i_1 2^t + i_0} &= \bigoplus_{k \preceq (i_1 2^t + i_0)} \lambda_k \\ &= \bigoplus_{k_1 \preceq i_1} \bigoplus_{k_0 \preceq i_0} \lambda_{k_1 2^t + k_0} \\ &= v_{i_0} \oplus i_1 \end{aligned}$$

Réciproquement, si $v(f) = (v_0, \ldots, v_{2^t-1}, v_0 \oplus 1, \ldots, v_{2^t-1} \oplus 1)^*|_{n+1}$, on a pour tout $i < 2^t$:

$$\begin{aligned} \lambda_{2^t + i} &= \bigoplus_{k \preceq 2^t + i} v_k \\ &= \bigoplus_{k_1 \preceq 1} \bigoplus_{k_0 \preceq i} v_{k_1 2^t + k_0} \\ &= \bigoplus_{k_0 \preceq i} (v_{k_0} \oplus v_{2^t + k_0}) \\ &= \bigoplus_{k_0 \preceq i} 1 = 2^{\mathrm{wt}(i)} \pmod 2 \end{aligned}$$

ce qui vaut 0 sauf si $i = 0$. □

Cette propriété permet encore de réduire la taille de la représentation d'une fonction booléenne symétrique à n variables. En effet, la propriété de symétrie elle-même permet d'abord de réduire la taille de la représentation du vecteur des valeurs de 2^n pour une fonction booléenne quelconque à $(n+1)$ pour une fonction symétrique. Ensuite, la périodicité du vecteur

des valeurs simplifié d'une fonction symétrique permet de réduire la représentation de la fonction à un vecteur des valeurs de taille $2^{\lfloor \log_2 d \rfloor + 1}$, où d est le degré de la fonction. Il est alors nécessaire de préciser la valeur de n, ce qui induit $(\lfloor \log_2 n \rfloor + 1)$ bits supplémentaires.

Exemple : Tableau des vecteurs des valeurs simplifiés des fonctions booléennes symétriques à n variables de degré inférieur à 3 et de terme constant nul

$\deg(f)$	$\lambda(f)$	$v(f)$
1	$(0,1,0^*)\|_{n+1}$	$(0,1)^*\|_{n+1}$
2	$(0,0,1,0^*)\|_{n+1}$	$(0,0,1,1)^*\|_{n+1}$
2	$(0,1,1,0^*)\|_{n+1}$	$(0,1,1,0)^*\|_{n+1}$
3	$(0,0,0,1,0^*)\|_{n+1}$	$(0,0,0,1)^*\|_{n+1}$
3	$(0,1,0,1,0^*)\|_{n+1}$	$(0,1,0,0)^*\|_{n+1}$
3	$(0,0,1,1,0^*)\|_{n+1}$	$(0,0,1,0)^*\|_{n+1}$
3	$(0,1,1,1,0^*)\|_{n+1}$	$(0,1,1,1)^*\|_{n+1}$

◇

Comme nous le verrons par la suite,la propriété de périodicité a également de substantielles conséquences pour le calcul du poids des fonctions symétriques et l'étude de la résilience.

5.3.2 Les fonctions de seuil et les fonctions exactes

Certaines fonctions symétriques très usitées dans la plupart des applications (pas uniquement cryptographiques) présentent d'autres motifs réguliers. Ainsi, la fonction *seuil*, notée T_s^n, est la fonction qui vaut 1 si et seulement si le poids de Hamming de son vecteur d'entrée est supérieur ou égal à s :

$$\begin{array}{rccl} T_s^n : & \mathbf{F}_2^n & \to & \mathbf{F}_2 \\ & x & \mapsto & \begin{cases} 0 \text{ si } \mathrm{wt}(x) < s \ ; \\ 1 \text{ si } \mathrm{wt}(x) \geq s. \end{cases} \end{array}$$

Un exemple incontournable de cette famille de fonctions est la *fonction majorité* qui correspond au cas où le seuil vaut $\lfloor \frac{n}{2} \rfloor + 1$. Une caractérisation de la forme algébrique normale de cette fonction a été donnée dans [Fil01].

De même, la fonction *exactement* s, E_s^n, est la fonction symétrique qui vaut 1 uniquement lorsque le poids du vecteur d'entrée est égal à s, *i.e.,*

$$\begin{array}{rccl} E_s^n : & \mathbf{F}_2^n & \to & \mathbf{F}_2^n \\ & x & \mapsto & \begin{cases} 0 & \text{si } \mathrm{wt}(x) \neq s \\ 1 & \text{si } \mathrm{wt}(x) = s \end{cases} \end{array}$$

Ces deux familles de fonctions symétriques interviennent dans de nombreux circuits. Leur implémentation a fait l'objet de plusieurs travaux [Weg87, BPP01] ayant pour but ultime la conception d'un circuit permettant de les représenter de façon optimale. Outre les implémentations efficaces connues, on recherche en effet des bornes inférieures sur le nombre de portes requises pour représenter la fonction afin de déterminer si les meilleurs circuits connus sont optimaux. Ainsi, Boyar, Peralta et Pochuev [BPP01] ont récemment amélioré les bornes inférieures sur le nombre de portes ET nécessaires en minorant le degré de ces fonctions.

Ici, nous sommes en mesure d'améliorer ces bornes et surtout de détailler la forme générale du vecteur simplifié de l'ANF de ces fonctions en exploitant le fait que leur vecteur des valeurs simplifié est constant à partir d'un certain poids.

La fonction qui associe le vecteur simplifié des valeurs au vecteur simplifié de l'ANF d'une fonction symétrique est une involution. On peut donc échanger simplement les rôles de ces deux vecteurs dans les résultats précédents. On peut en particulier utiliser la propriété de périodicité énoncée dans le théorème 5.33 et caractériser l'ANF des fonctions symétriques dont le vecteur simplifié des valeurs s'annule à partir d'un seuil.

Théorème 5.35 *Soit n et t deux entiers tels que $n \geq 2^t - 1$. Soit $f \in \mathcal{S}ym_n$ de vecteur des valeurs simplifié $v_f = (v_0, \ldots, v_n)$ et de vecteur simplifié de l'ANF $\lambda_f = (\lambda_0, \ldots, \lambda_n)$. Alors, $v_i = 0$ pour tout $i \geq 2^t$ si et seulement si λ_f est périodique de période 2^t. De plus, $(\lambda_0, \ldots, \lambda_{2^t-1})$ est le vecteur simplifié de l'ANF de la fonction booléenne symétrique à $(2^t - 1)$ variables dont le vecteur simplifié des valeurs est $(v_0, \ldots, v_{2^t-1})$.*

On applique directement ce théorème dans le cas des fonctions seuil T_s^n.

Proposition 5.36 *Soit n et s deux entiers tels que $s \leq 2^t$ pour un entier $t \leq \log_2 n$. Alors, le vecteur simplifié de l'ANF de la fonction seuil à s à n variables, T_s^n est la troncature à $(n+1)$ valeurs de la suite*

$$\underbrace{\lambda(T_s^{2^t-1})}_{2^t} \left(\underbrace{\lambda(1 \oplus T_s^{2^t-1})}_{2^t} \right)^* .$$

En particulier,

$$\deg(T_s^n) \geq \begin{cases} n - (n \mod 2^t) + s & \text{si } s \leq (n \mod 2^t) \\ n - (n \mod 2^t) & \text{sinon} \end{cases} .$$

Preuve : Le théorème précédent appliqué à la fonction $(1 \oplus T_s^n)$, dont le vecteur des valeurs vérifie $v_{1 \oplus T_s^n}(i) = 0$ pour tout $i \geq s$, implique que le vecteur simplifié de l'ANF de $(1 \oplus T_s^n)$ est la troncature à $n+1$ éléments de

$$(\lambda(1 \oplus T_s^{2^t-1}))^* .$$

On en déduit le vecteur simplifié de l'ANF de T_s^n en annulant le coefficient constant. La borne inférieure sur le degré de la fonction est obtenue directement en remarquant que chaque période de longueur 2^t (exceptée la première) du vecteur simplifié de l'ANF de T_s^n contient un 1 en position 0 et en position s. En effet, la position 0 de $\lambda(1 \oplus T_s^{2^t-1})$ vaut 1 car il s'agit de la valeur en 0 de la fonction $(1 \oplus T_s^n)$. La position s de $\lambda(1 \oplus T_s^{2^t-1})$ est celle de $\lambda(T_s^{2^t-1})$; elle est donnée par

$$\lambda_{T_s^{2^t-1}}(s) = \bigoplus_{i \preceq s} v_{T_s^{2^t-1}}(i) = v_{T_s^{2^t-1}}(s) = 1 .$$

□

Cette borne générale sur le degré des fonctions améliore la borne donnée dans [BPP01, Coro. 4] qui établissait

$$\deg(T_s^n) \geq \max\{s, 2^{\lfloor \log_2 n \rfloor}\} .$$

Par exemple, pour $n = 78$ et $s = 5$, on trouve

$$\deg(T_5^{78}) \geq 72 + 5 = 77$$

au lieu de

$$\deg(T_5^{78}) \geq 64 \ .$$

Ce théorème facilite grandement le calcul du vecteur simplifié de l'ANF des fonctions seuil car, pour un seuil s donné, il permet de se ramener à une fonction ayant un nombre réduit et fixe de variables. De plus, ce nombre étant de la forme $2^t - 1$, le calcul de l'ANF de la fonction seuil à partir de son vecteur des valeurs est beaucoup plus aisé.

On obtient bien entendu des résultats similaires pour les fonctions exactes.

Proposition 5.37 *Soit n et s deux entiers tels que $s < 2^t$ pour un entier $t \leq \log_2 n$. Alors, le vecteur simplifié de l'ANF de la fonction exactement s à n variables, E_s^n, est la troncature à $(n+1)$ valeurs de la suite*

$$\left(\lambda(E_s^{2^t-1})\right)^* \ .$$

En particulier,

$$\deg(E_s^n) \geq \begin{cases} n - (n \mod 2^t) + s & si\ s \leq (n \mod 2^t) < 2^t - 1 \\ n - \left((n+1) \mod 2^t\right) & sinon \end{cases} \ .$$

Preuve : La forme du vecteur simplifié de l'ANF de E_s^n se déduit directement du théorème 5.35. La borne sur le degré est obtenue en constatant que chaque période de longueur 2^t de $\lambda(E_s^n)$ contient un 1 en position s et en dernière position. En effet, on a

$$\lambda_{E_s^{2^t-1}}(s) = \bigoplus_{i \preceq s} v_{E_s^{2^t-1}}(i) = v_{E_s^{2^t-1}}(s) = 1$$

et

$$\lambda_{E_s^{2^t-1}}(2^t - 1) = \bigoplus_{i \preceq 2^t-1} v_{E_s^{2^t-1}}(i) = v_{E_s^{2^t-1}}(s) = 1 \ .$$

□

L'ANF des fonctions exactement s, pour un nombre quelconque de variables, est donc entièrement déterminée par celle de $E_s^{2^t-1}$ où $t = \lfloor \log_2(s+1) \rfloor$. Par définition du vecteur des valeurs de cette fonction, on en déduit que, pour tout entier i de t bits, $\lambda_{E_s^{2^t-1}}(i) = 1$ si et seulement si $s \preceq i$. Ces résultats simplifient considérablement la détermination de l'ANF des fonctions seuil et des fonctions exactement. On peut notamment en déduire la forme exacte de l'ANF de ces fonctions pour les valeurs particulières de s et ce quelque soit le nombre de variables de la fonction. Ces résultats font l'objet d'un travail en cours.

5.4 Résilience des fonctions symétriques

La résilience d'une fonction symétrique, c'est-à-dire le fait qu'elle reste équilibrée quand un nombre donné de ses composantes d'entrée est fixé, fait l'objet d'une conjecture énoncée par von zur Gathen et Roche dans [vzGR97], qui avance qu'une fonction booléenne symétrique non-affine est au plus 2-résiliente.

Les études portant sur la propriété de résilience des fonctions symétriques montrent qu'il existe des familles infinies de fonctions respectivement 1 et 2-résilientes [GHS93], [vzGR97] et

sans-corrélation d'ordre 3 [SM03]. Une borne générale sur l'ordre de résilience d'une fonction booléenne symétrique est donnée par la proposition suivante.

Proposition 5.38 *[vzGR97] Soient $n \geq 2$ et f une fonction booléenne symétrique t-résiliente à n variables avec* $\deg f > 1$, *alors*

$$t \leq \frac{n-3}{2}.$$

Cette borne appelée *borne triviale* ne le devient que si on se ramène à l'étude du polynôme univarié p_f de degré au plus n interpolateur des valeurs de $v(f)$ pour f non-constante. On sait qu'alors soit p_f soit $p_f - 1$ a au moins $\frac{n+1}{2}$ zéros dans $\{0, \ldots, n\}$. Ainsi, p_f étant non-constant, on a $\deg(p_f) = \deg(p_f - 1) \geq \frac{n+1}{2}$. Ainsi $\gamma(v(f)) \leq \frac{n-1}{2}$ pour $n \geq 2$, ce qui signifie qu'une fonction symétrique f à n variables, $n \geq 2$ peut être au plus $\frac{n-3}{2}$ -résiliente.

Plus précisément, dans [vzGR97], le résultat est amélioré en prenant en compte la proposition 5.20. En effet, toute borne supérieure sur l'intervalle séparant deux nombres premiers consécutifs donne une borne supérieure sur l'ordre de résilience maximal atteignable par une fonction booléenne symétrique. À l'aide d'un théorème de Mozzochi [Moz86] sur la différence entre deux nombres premiers consécutifs, on obtient alors qu'une fonction booléenne symétrique t-résiliente est telle que $t = \mathcal{O}(n^{0,548})$.

L'étude de résilience d'une fonction booléenne est reliée de manière évidente à celle de ses restrictions. Nous montrons que les restrictions de fonctions symétriques sont symétriques, ce qui nous permet de détailler la structure des familles infinies de fonctions résilientes connues que nous présentons dans la partie suivante. Nous combinons enfin les propriétés des restrictions des fonctions symétriques avec la propriété de périodicité du vecteur des valeurs simplifié afin de déduire une nouvelle borne sur l'ordre de résilience d'une fonction booléenne symétrique. L'étude sur les restrictions des fonctions symétriques nous permettra par ailleurs d'obtenir des résultats sur les fonctions de haute non-linéarité.

5.4.1 Restrictions d'une fonction symétrique

Nous rappelons que pour tout $f \in \mathcal{B}_n$ et tout sous-espace vectoriel $V \subset \mathbf{F}_2^n$, la *restriction de f à un translaté $a + V$ de V* est la fonction

$$\begin{array}{rccc} f_{a+V} : & V & \to & \mathbf{F}_2 \\ & x & \mapsto & f(a+x) \end{array}$$

La fonction f_{a+V} peut être identifiée de manière immédiate à une fonction booléenne à $\dim(V)$ variables.

Nous nous intéressons au cas particulier où V est un sous-espace engendré par k vecteurs de la base canonique de $\mathbf{F}_2^n$. Nous désignons par $\overline{V}$ son sous-espace supplémentaire.

Soit $f \in \mathcal{S}ym_n$. La propriété de symétrie nous permet de ramener l'étude du cas où V est un sous-espace engendré par k vecteurs de la base canonique de $\mathbf{F}_2^n$ au cas où les vecteurs sont les k premiers de la base : $V = \langle e_1, \ldots, e_k \rangle$. En outre, les restrictions de f à V et à tous les translatés $a + V$, $a \in \overline{V}$ sont des fonctions symétriques à k variables. En effet,

$$a \in \overline{V}, \quad \forall\, x \in V, \quad f_{a+V}(x) = f(a+x) = v_f(\mathrm{wt}(a) + \mathrm{wt}(x))$$

car a et x sont par définition à supports disjoints. Ainsi $f_{a+V}(x)$, de toute évidence, ne dépend que du poids de x lorsque a est fixé et la fonction f_{a+V} ne dépend que de $\mathrm{wt}(a)$. Ainsi, on

peut déduire les expressions du vecteur simplifié de l'ANF et du vecteur des valeurs simplifié de f_{a+V} à partir de ceux de f.

Exemple : Considérons la fonction à 5 variables f dont le vecteur simplifié de l'ANF est donné par : $\lambda(f) = (0,1,1,1,0,0)$. Le calcul de son vecteur des valeurs simplifié nous donne $v(f) = (0,1,1,1,0,1)$. Calculons maintenant le vecteur des valeurs des restrictions de f aux translatés du sous-espace vectoriel $V = \langle e_1, \ldots, e_4 \rangle$. Nous obtenons deux fonctions f_V et f_{1+V} dont le vecteur des valeurs simplifié est donné par :

$$v(f_V) = (0,1,1,1,0)$$
$$v(f_{1+V}) = (1,1,1,0,1).$$

De manière plus générale, si on considère $f \in \mathcal{S}ym_n$ de vecteur des valeurs simplifié $v(f) = (v_0, \ldots, v_n)$ alors les vecteurs des valeurs simplifiés des restrictions $f_{\mathrm{wt}(a)} = f_{a+V}$ de f aux translatés du sous-espace $V = \langle e_1, \ldots, e_k \rangle$ sont donnés par :

$$\begin{array}{lcccccccc} v(f_0) & = & (v_0, & v_1, & \ldots & \ldots & & v_k) & \\ v(f_1) & = & & (v_1, & v_2, & \ldots & \ldots & v_{k+1}) & \\ \vdots & & & & & & & & \\ v(f_{n-k}) & = & & & & (v_{n-k}, & \ldots & \ldots & v_n) \end{array}$$

◇

Cette propriété est détaillée dans la proposition suivante.

Proposition 5.39 *Soient $f \in \mathcal{S}ym_n$ et $V = \langle e_1, \ldots e_k \rangle$ avec $k \leq n$. Alors, pour tout $a \in \overline{V} = \langle e_{k+1}, \ldots, e_n \rangle$, la restriction de f à $a + V$ est une fonction symétrique à k variables qui ne dépend que de* $\mathrm{wt}(a)$. *En outre, les expressions du vecteur simplifié de l'ANF et du vecteur des valeurs simplifié de f_{a+V} sont données par, $\forall\ i, 0 \leq i \leq k$,*

$$v_{f_{a+V}}(i) = v_f(i + \mathrm{wt}(a))$$
$$\lambda_{f_{a+V}}(i) = \bigoplus_{j \preceq \mathrm{wt}(a)} \lambda_f(i+j).$$

Preuve : Soit

$$f(x) = \bigoplus_{u \in \mathbf{F}_2^n} c_f(u) \prod_{i=1}^{n} x_i^{u_i},\ x \in \mathbf{F}_2^n$$

l'expression de la forme algébrique normale de f. Alors pour tout $x \in V$,

$$\begin{aligned} f_{a+V}(x) &= \bigoplus_{u \in \mathbf{F}_2^k} \bigoplus_{v \in \mathbf{F}_2^{n-k}} c_f(u\|v) \prod_{i=1}^{k} x_i^{u_i} \prod_{j=1}^{n-k} a_j^{v_j} \\ &= \bigoplus_{u \in \mathbf{F}_2^k} \prod_{i=1}^{k} x_i^{u_i} \bigoplus_{v \in \mathbf{F}_2^{n-k}} c_f(u\|v) \prod_{j=1}^{n-k} a_j^{v_j}, \end{aligned}$$

où le vecteur $u\|v$, $u \in \mathbf{F}_2^k$, $v \in \mathbf{F}_2^{n-k}$, de taille n est la concaténation du vecteur u et du vecteur v. Par ailleurs,

$$\prod_{j=1}^{n-k} a_j^{v_j} = 1 \text{ si et seulement si } \mathrm{supp}(v) \subseteq \mathrm{supp}(a),$$

i.e., $v \preceq a$. Nous obtenons ainsi :

$$f_{a+V}(x) = \bigoplus_{u \in \mathbf{F}_2^k} \prod_{i=1}^{k} x_i^{u_i} \bigoplus_{v \preceq a} c_f(u\|v).$$

Du fait que f est symétrique, $c_f(u\|v) = \lambda_f(\mathrm{wt}(u) + \mathrm{wt}(v))$. D'où on tire que,

$$\begin{aligned} \bigoplus_{v \preceq a} c_f(u\|v) &= \bigoplus_{j=0}^{\mathrm{wt}(a)} \binom{\mathrm{wt}(a)}{j} \lambda(\mathrm{wt}(u) + j) \\ &= \bigoplus_{j \preceq \mathrm{wt}(a)} \lambda_f(\mathrm{wt}(u) + j). \end{aligned}$$

L'expression de la forme algébrique normale de f_{a+V} s'en déduit finalement :

$$f_{a+V}(x) = \bigoplus_{i=0}^{k} \left(\bigoplus_{j \preceq \mathrm{wt}(a)} \lambda_f(j+i) \right) X_{i,k}(x).$$

□

Nous pouvons déduire de cette expression le corollaire immédiat suivant (qui n'est pas vrai en général pour une fonction booléenne quelconque) sur le degré des restrictions d'une fonction booléenne symétrique.

Corollaire 5.40 *Soient $f \in \mathcal{S}ym_n$ une fonction de degré d et $V = \langle e_1, \ldots e_k \rangle$ avec $d \leq k \leq n$. Alors les restrictions de f à tous les translatés $a+V$, $a \in \overline{V} = \langle e_{k+1}, \ldots, e_n \rangle$, sont de degré d.*

5.4.2 Élements de construction de fonctions symétriques résilientes

Par définition, f est une fonction booléenne t-résiliente si et seulement si toutes ses restrictions à des sous-espaces affines de dimension $n-k$ sont $(t-k)$-résilientes. Dans le cas des fonctions symétriques, la formule de calcul du triangle de Pascal $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ montre qu'on peut se limiter aux restrictions aux sous-espaces vectoriels.

Proposition 5.41 *Soit $f \in \mathcal{S}ym_n$. Alors f est t-résiliente si et seulement si pour $1 \leq k \leq t$, $f_{\langle e_1, \ldots, e_{n-k} \rangle}$ est $(t-k)$-résiliente.*

Par ailleurs, l'étude précédente concernant les restrictions de fonctions symétriques nous a inspiré une méthode immédiate pour construire des fonctions symétriques 1-résilientes d'un nombre de variables pair. Elle consisterait à combiner deux restrictions équilibrées triviales. Nous pouvons démontrer que cette construction produit toujours des fonctions affines.

Proposition 5.42 *Soient n un entier pair et $f \in \mathcal{S}ym_n$. Soit $H = \langle e_1, \ldots, e_{n-1} \rangle$. Si les restrictions de f à H et à $e_n + H$ sont équilibrées triviales, alors* $\deg(f) = 1$.

Preuve : Supposons que f_H et f_{e_n+H} sont équilibrées triviales. Cela signifie pour f_H que pour tout i, $0 \leq i \leq n-1$,$v_f(i) = v_f(n-1-i) \oplus 1$ et pour f_{e_n+H} que pour tout j, $1 \leq j \leq n$, $v_f(j) = v_f(n-j+1) \oplus 1$. Nous obtenons alors le système suivant :

$$\begin{array}{lcll} v_f(i) & = & v_f(n-1-i) \oplus 1, & \forall i, 0 \leq i \leq n-1 \\ v_f(n-1-i) & = & v_f(i+2) \oplus 1, & \forall i, 0 \leq i \leq n-2. \end{array}$$

Ainsi, pour tout i, $0 \leq i \leq n-2$, $v_f(i) = v_f(i+2)$. D'après le théorème 5.33, $\deg(f) \leq 1$ car $v(f)$ est périodique de période 2. Comme f ne peut pas être une fonction constante puisque ses restrictions sont équilibrées, f est donc de degré 1. □

Comme nous allons le voir par la suite, toutes les constructions connues de fonctions symétriques 2-résilientes (qui couvrent par ailleurs toutes les fonctions 2-résilientes jusqu'à 128 variables) sont équilibrées triviales. On déduit alors directement de la proposition précédente qu'il n'existe pas de fonction symétrique 3-résiliente de degré supérieur à 1 jusqu'à 128 variables. La question de savoir si la propriété demeure pour un nombre de variables quelconque reste encore un problème ouvert.

5.4.3 Constructions connues de fonctions booléennes symétriques résilientes

Nous reportons maintenant les familles infinies de fonctions booléennes symétriques résilientes connues. Elles recoupent naturellement en partie les fonctions équilibrées connues. Les familles de fonctions résilientes connues sont des fonctions 2-résilientes à n variables et de leurs restrictions 1-résilientes à $(n-1)$ variables. À la lumière de la proposition 5.41, les fonctions 2-résilientes connues correspondent à des constructions connues de fonctions équilibrées à n, $n-1$ et $n-2$ variables qui conduisent à des vecteurs des valeurs compatibles.

Proposition 5.43 *[GHS93][vzGR97] Soient les entiers $t \geq 2$, $n = 4t^2 - 1$ et $k = 2t^2 - t - 1$ et la fonction $f \in \mathcal{S}ym_n$ de vecteur des valeurs simplifié*

$$v(f) = (v_{\varphi_1}(i))_{0\leq i\leq k-1}, \underbrace{k \oplus 1}_{v_f(k)}, \underbrace{k \bmod 2}_{v_f(k+1)}, (v_{\varphi_1}(i))_{k+2\leq i\leq n-k-2},$$
$$\underbrace{k \oplus 1}_{v_f(n-k-1)}, \underbrace{k \bmod 2}_{v_f(n-k)}, (v_{\varphi_1}(i))_{n-k+1\leq i\leq n}),$$

ou son complémentaire. Alors f est 2-résiliente.

En effet, f est équilibrée triviale. Sa restriction à l'espace $\langle e_1, \ldots, e_{n-1}\rangle$ est équilibrée d'après la proposition 5.23 et sa restriction à l'espace $\langle e_1, \ldots, e_{n-2}\rangle$ est équilibrée d'après la proposition 5.25. Le premier exemple de telles fonctions apparaît pour $n = 15$ (k vaut alors 5). Le vecteur des valeurs de la fonction de terme constant nul est (0,1,0,1,0,0,1,1,0,0,1,1,0,1,0,1). Nous obtenons ainsi la chaîne de fonctions suivantes (dont nous donnons les vecteurs des valeurs simplifiés) par les différentes restrictions de la fonction 2-résiliente où $H = \langle e_1, \ldots, e_{14}\rangle$ et $V = \langle e_1, \ldots, e_{13}\rangle$, nous désignons par f_{i+V} la restriction de f à $a + V$ où $\mathrm{wt}(a) = i$.

n		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
15	f	(0,	1,	0,	1,	0,	0,	1,	1,	0,	0,	1,	1,	0,	1,	0,	1)	équ. triviale
14	f_H	(0,	1,	0,	1,	0,	0,	1,	1,	0,	0,	1,	1,	0,	1,	0)		Prop. 5.23
14	f_{1+H}	(1,	0,	1,	0,	0,	1,	1,	0,	0,	1,	1,	0,	1,	0,	1)		Prop. 5.23
13	f_V	(0,	1,	0,	1,	0,	0,	1,	1,	0,	0,	1,	1,	0,	1)			Prop. 5.25
13	f_{1+V}	(1,	0,	1,	0,	0,	1,	1,	0,	0,	1,	1,	0,	1,	0)			équ. triviale
13	f_{2+V}	(0,	1,	0,	0,	1,	1,	0,	0,	1,	1,	0,	1,	0,	1)			Prop. 5.25

Nous donnons également le tableau des caractéristiques des fonctions mentionnées.

Nous constatons que la fonction est équilibrée triviale mais que ses restrictions appartiennent aux deux familles mentionnées dans la partie des fonctions équilibrées. Les triplets

TAB. 5.7 – *Caractéristiques des restrictions de la fonction booléenne symétrique 2-résiliente à 15 variables de terme constant nul.*

n	ANF	valeurs	spectre de Walsh	spectre d'auto-corrélation	NL	rés.
15	0100 0110 0110 0000	0101 0011 0011 0101	0, 0, 0, 352, 0, −256, 0, 224, 0, −256, 0, 352, 0, 0, 0, 736, 0	32768, −736, 736, 144, −144, 0, 0, −64, 64, 0, 0, 144, −144, −736, 736, −32768, 32768	16016	2
14	0100 0110 0110 000	0101 0011 0011 010	0, 0, 176, 176, −128, −128, 112, 112, −128, −128, 176, 176, 0, 0, 368	16384, −368, 368, 72, −72, 0, 0, −32, 32, 0, 0, 72, −72, −368, 368	8008	1
14	1100 1010 1010 000	1010 0110 0110 101	0, 0, −176, 176, 128, −128, −112, 112, 128, −128, −176, 176, 0, 0, −368	16384, −368, 368, 72, −72, 0, 0, −32, 32, 0, 0, 72, −72, −368, 368	8008	1
13	0100 0110 0110 00	0101 0011 0011 01	0, 88, 176, 24, −128, −8, 112, −8, −128, 24, 176, 88, 0, 184, 0	8192, −272, 272, 24, −24, 16, −16, −16, 16, −48, 48, 168, −168, 7824, 8192	4004	0
13	1100 1010 1010 00	1010 0110 0110 10	0, -88, 0, 152, 0, -120, 0, 120, 0, -152, 0, 88, 0, -184	8192, -96, 96, 48, -48, -16, 16, -16, 16, 48, -48, -96, 96, -8192	4004	0
13	0101 1111 1110 00	0100 1100 1101 01	0, 88, −176, 24, 128, −8, −112, −8, 128, 24, −176, 88, 0, 184, 0	8192, −272, 272, 24, −24, 16, −16, −16, 16, −48, 48, 168, −168, 7824, 8192	4004	0

suivants de valeurs de n où apparaît cette succession de fonctions sont (33,34,35), (61,62,63), (97,98,99). Ce sont en fait, avec la famille que nous décrivons ci-après, les seules fonctions 2-résilientes que les simulations exhaustives de von zur Gathen et Roche ont recensées jusqu'à $n = 128$.

Proposition 5.44 *[vzGR97] Soient les entiers $i \geq 2$ tel que $i \not\equiv 1 \bmod 3$, $n = F_{2i+2}F_{2i+3}+1$, $k = F_{2i}F_{2i+3}$ et la fonction $f \in \mathcal{S}ym_n$ dont le vecteur des valeurs simplifié est donné par le vecteur suivant ou son complémentaire :*

$$v(f) = \begin{cases} \big((0)^*|_k 01 (10)^*|_{n-2k-3} 01 (1)^*|_k\big), & \text{si } k \text{ est impair,} \\ \big((10)^*|_k 10 (01)^*|_{n-2k-3} 10 (01)^*|_k\big), & \text{si } k \text{ est pair.} \end{cases}$$

Alors f est 2-résiliente.

On vérifie que les fonctions ainsi définies sont équilibrées triviales. Le premier triplet de valeurs de n pour lequel apparaît la suite de fonctions 2-résilientes, 1-résilientes et équilibrées obtenues à partir de ce résultat et des restrictions de la fonction ainsi obtenue correspond à (103,104,105). Avec les trois fonctions recensées précédemment, ce sont les seules fonctions 2-résilientes pour n inférieur ou égal à 128.

Par ailleurs, comme nous l'avons montré à la proposition 5.42, l'existence d'une fonction symétrique 3-résiliente non-affine impliquerait nécessairement l'existence d'une fonction symétrique 2-résiliente non-affine qui n'appartienne pas aux familles décrites par les propositions 5.43 et 5.44.

5.4.4 Ordre de résilience et motifs réguliers du vecteur des valeurs simplifié

Nous nous attachons dans cette partie à utiliser les propriétés des restrictions d'une fonction symétrique avec la périodicité du vecteur des valeurs simplifié pour démontrer de nouveaux résultats concernant l'ordre de résilience des fonctions symétriques.

Proposition 5.45 *Soit $f \in \mathcal{S}ym_n$ une fonction t-résiliente dont le vecteur des valeurs simplifié est ultimement périodique de période T et de pré-période n_0, c'est-à-dire pour lequel $v_f(i) = v_f(i+T)$ pour tout $i \geq n_0$ — avec $t \geq n_0 + T - 1$. Alors, pour tout $m \geq 0$, il existe une fonction symétrique à $n+m$ variables de degré au moins $\deg(f)$ qui est $(t+m)$-résiliente.*

Preuve : Le vecteur des valeurs de f est de la forme :

$$v(f) = (v_f(0), \ldots, v_f(n_0-1), [v_f(n_0), \ldots, v_f(n_0+T-1)]^*)|_{n+1}.$$

D'après la proposition 5.39, f est t-résiliente si et seulement si les $(t+1)$ fonctions $f_i \in \mathcal{S}ym_{n-t}$, pour $0 \leq i \leq t$, dont le vecteur des valeurs simplifié est

$$v(f_i) = (v_f(i), v_f(i+1), \ldots, v_f(n-t+i))$$

sont équilibrées. De plus d'après le corollaire 5.40, nous savons que toutes les fonctions ont le même degré que f car la dimension du sous-espace auquel on restreint est $(n-t)$ et on a $\deg(f) < n-t$ d'après l'inégalité de Siegenthaler [Sie84]. Comme f est ultimement périodique de période T et que $t \geq n_0 + T - 1$, on a :

$$v(f_{i+T}) = v(f_i), \ \forall i \geq n_0.$$

Cela signifie qu'à partir de n_0+T, il y a une répétition du vecteur des valeurs des restrictions. Nous allons alors construire à partir de f une fonction dépendant d'un nombre supérieur de variables et ce grâce à ses restrictions. Pour cela on définit pour tout $k > t$, les fonctions de vecteur des valeurs simplifié

$$v(f_k) = v(f_{n_0+(k-n_0) \bmod T}).$$

Toutes ces fonctions sont équilibrées et de degré égal à $\deg(f)$ car $n_0 \leq n_0+(k-n_0) \bmod T \leq t$.

Pour tout $m \geq 0$, définissons alors $g_m \in \mathcal{S}ym_{n+m}$ telle que

$$v(g_m) = (v_f(0), \ldots, v_f(n_0-1), [v_f(n_0), \ldots, v_f(n_0+T-1)]^*)|_{n+m+1}.$$

Toutes les restrictions de g_m aux translatés de $\langle e_1, \ldots, e_{n-t} \rangle$ correspondent aux fonctions de vecteur des valeurs simplifié $v(f_k)$, $0 \leq k \leq m+t$ que nous venons de définir. Elles sont équilibrées, ce qui signifie que g_m est $(t+m)$-résiliente. De plus, le degré des restrictions étant égal à $\deg(f)$, on a $\deg(g_m) \geq \deg(f)$.

□

Cette proposition assez surprenante repose essentiellement sur la condition $t \geq n_0 + T - 1$ qui impose qu'il existe au moins une répétition parmi les $t+1$ vecteurs des valeurs simplifiés des restrictions de f à des sous-espaces de dimension $n-t$. Elle implique donc qu'on puisse alors construire indéfiniment des fonctions résilientes d'un nombre de variables supérieur grâce à la répétition du motif périodique des restrictions équilibrées. À la lumière de la proposition 5.20 cette propriété conduit clairement à une contradiction.

Corollaire 5.46 *Soit f une fonction symétrique dont le vecteur des valeurs simplifié est ultimement périodique, c'est-à-dire pour lequel $v_f(i) = v_f(i+T)$ pour tout $i \geq n_0$. Si f est t-résiliente avec $t \geq n_0 + T - 1$, alors $\deg(f) = 1$.*

Preuve : Soit p un nombre premier strictement plus grand que n. D'après la proposition précédente, il existe une fonction g à $p-1$ variables de degré $\deg(g) \geq \deg(f)$ qui est équilibrée (puisqu'elle est $t+(p-1-n)$-résiliente). D'après la proposition 5.20, $\deg(g) = 1 \geq \deg(f)$. □

Ce résultat s'applique donc dans le cas des fonctions symétriques de vecteur des valeurs simplifié périodique. Il montre en particulier que l'ordre de résilience d'une fonction symétrique est limité par son degré.

Théorème 5.47 *Soit f une fonction booléenne symétrique telle que $\deg(f) \leq 2^\ell$. Si f est $(2^\ell-1)$-résiliente, alors $\deg(f) = 1$.*

Preuve : Soit $f \in \mathcal{S}ym_n$. D'après le théorème 5.33, si $\deg(f) \leq 2^\ell - 1$, alors $v(f)$ est périodique de période 2^ℓ. Le résultat découle donc immédiatement du corollaire précédent.

Si $\deg(f) = 2^\ell$, alors d'après la proposition 5.34, $v(f) = (v_0,\ldots,v_{2^\ell-1},v_0 \oplus 1,\ldots,v_{2^\ell} \oplus 1)^*|_n$ pour $v_0,\ldots,v_{2^\ell-1} \in \mathbf{F}_2$. f est $(2^\ell-1)$-résiliente si et seulement si toutes les fonctions symétriques de vecteur des valeurs simplifié

$$v(f_i) = (v_f(i),v_f(i+1),\ldots,v_f(n-2^\ell+1+i))$$

pour $0 \leq i < 2^\ell$ sont équilibrées. Les fonctions qu'on obtient en complémentant les vecteurs précédents, *i.e.*,

$$v(f_{i+2^\ell}) = (v_f(i) \oplus 1,v_f(i+1) \oplus 1,\ldots,v_f(n-2^\ell+1+i) \oplus 1)$$

sont, de manière évidente, équilibrées pour tout $0 \leq i < 2^\ell$. Les mêmes arguments que ceux utilisés dans la démonstration de la proposition 5.45 peuvent alors servir pour montrer que pour tout $m \geq 0$, la fonction à $(n+m)$ variables dont le vecteur des valeurs simplifié est la troncature à $(n+m+1)$ éléments de $(v_0,\ldots,v_{2^\ell-1},v_0 \oplus 1,\ldots,v_{2^\ell} \oplus 1)^*$ est $(2^\ell-1+m)$-résiliente et que son degré est au moins $\deg(f)$. Pour $m = p-n-1$ où p est un nombre premier strictement supérieur à n, on obtient $\deg(f) = 1$. □

Exemple : Considérons une fonction symétrique non-affine à 161 variables (afin d'avoir un nombre de variables supérieur à 128). La borne explicite de von zur Gathen et Roche (proposition 5.38) nous dit qu'une fonction booléenne symétrique à 161 variables est au plus $\frac{n-3}{2} = 79$-résiliente. La borne de Siegenthaler nous dit qu'une fonction booléenne à 161 variables de degré $d \leq 32$ est au plus $n-d-1 = 128$-résiliente. La nouvelle borne que nous venons d'établir nous permet de dire qu'une telle fonction symétrique est au plus 30-résiliente. ◇

La proposition 5.45 permet également de démontrer que toutes les restrictions d'une fonction symétrique résiliente sont distinctes. De manière plus précise, on peut montrer le résultat suivant.

Proposition 5.48 *S'il existe une fonction symétrique à n variables t-résiliente de degré $d \neq 1$, alors il existe $(k+1)$ fonctions symétriques équilibrées distinctes à $(n-k)$ variables de degré d pour $0 \leq k \leq t$.*

Preuve : Soit k tel que $0 \leq k \leq t$. Il nous faut démontrer que les $(k+1)$ fonctions symétriques f_i à $(n-k)$ variables, définies par les vecteurs des valeurs simplifiés

$$v(f_i) = (v_f(i),v_f(i+1),\ldots,v_f(n-k+i)),\ 0 \leq i \leq k$$

sont distinctes.

Supposons que $f_{j_1} = f_{j_2}$ pour un couple (j_1,j_2) tel que $0 \leq j_1 < j_2 \leq k$. Alors, les vecteur des valeurs simplifiés $v(f_{j_1+1})$ et $v(f_{j_2+1})$ sont soit égaux soit différents par leur dernière composante, *i.e.*,

$$v(f_{j_1+1}) \oplus v(f_{j_2+1}) = (0,0\ldots,0,1).$$

Dans le dernier cas, $f_{j_1+1}+f_{j_2+1}$ est de degré $(n-k)$ car son poids de Hamming est impair. Or, $\deg(f) \leq n-t \leq n-k$ car f est t-résiliente, ce qui implique que $\deg(f_{j_1+1}) = \deg(f_{j_2+1}) = \deg(f)$ (d'après le corollaire 5.40), ce qui conduit à une contradiction. D'où on déduit que $f_{j_1+1} = f_{j_2+1}$ et par induction que $f_{j_1+m} = f_{j_2+m}$ pour tout $m \leq t - j_2$. Cela signifie donc que le vecteur des valeurs simplifié de f est ultimement périodique de période $j_2 - j_1$ et de pré-période j_1 : pour tout i tel que $j_1 \leq i \leq t - j_2 + j_1$,

$$v_f(i + j_2 - j_1) = v_f(i).$$

On déduit du corollaire 5.46 que f est de degré 1. □

Ce résultat semble en partie montrer que l'ordre de résilience des fonctions symétriques est limité par le faible nombre de fonctions symétriques équilibrées (en particulier pour un nombre pair de variables). En fait von zur Gathen et Roche ont montré que pour $n \geq 2$ la probabilité pour qu'une fonction booléenne symétrique soit équilibrée est inférieure à $2^{-\frac{n}{3}}$ [vzGR97, Th.].

5.5 Dérivées des fonctions symétriques

Cette partie est dédiée à l'étude du critère de propagation des fonctions symétriques. Les caractéristiques de propagation d'une fonction booléenne sont déterminées par les propriétés de ses dérivées.

5.5.1 Propriétés générales des dérivées

Une première propriété remarquable des dérivées d'une fonction symétrique réside dans leur équivalence linéaire lorsque les vecteurs relativement auxquels on considère les dérivées ont le même poids de Hamming.

Proposition 5.49 *Soient $f \in \mathcal{S}ym_n$ et $a, b \in \mathbf{F}_2^n$ tels que $\mathrm{wt}(a) = \mathrm{wt}(b)$. Alors $D_a f$ et $D_b f$ sont linéairement équivalentes, c'est-à-dire qu'il existe une permutation linéaire σ de $\mathbf{F}_2^n$ telle que $D_a f = D_b f \circ \sigma$.*

Preuve : Soient a, $b \in \mathbf{F}_2^n$ tels que $\mathrm{wt}(a) = \mathrm{wt}(b)$. Il existe alors une permutation π de $\{1,\ldots,n\}$ telle que $(b_1,\ldots,b_n) = (a_{\pi(1)},\ldots,a_{\pi(n)})$. Cette permutation des indices induit donc une permutation linéaire σ de $\mathbf{F}_2^n$ telle que $b = \sigma(a)$. La fonction f étant symétrique, on a alors :

$$\begin{aligned} D_b f(\sigma(x)) &= f(\sigma(x)) \oplus f(\sigma(x)+b) \\ &= f(x) \oplus f(\sigma(x)+\sigma(a)) \\ &= f(x) \oplus f(\sigma(x+a)) \\ &= f(x) \oplus f(x+a) \\ &= D_a f(x). \end{aligned}$$

□

Les propriétés des dérivées que nous considérons étant pour la plupart invariantes par composition avec une application linéaire — comme le poids ou le degré — nous étudierons seulement les dérivées des fonctions symétriques relativement au vecteur ε_k de poids k pour $1 \leq k \leq n$, que nous définissons comme la somme des k derniers vecteurs de la base canonique de $\mathbf{F}_2^n$, $\varepsilon_k = e_{n-k+1} + \cdots + e_n$.

Les dérivées d'une fonction symétrique ne sont pas symétriques en général. Cependant, on peut les décomposer en fonctions symétriques.

Proposition 5.50 *Soient $f \in \mathcal{S}ym_n$ et k un entier tel que $1 \leq k \leq n-1$. Considérons le sous-espace $V = \langle e_1, \ldots, e_{n-k} \rangle$ et le vecteur $\varepsilon_k = e_{n-k+1} + \cdots + e_n$. Alors, les restrictions de $D_{\varepsilon_k} f$ à tous les sous-espaces affines $b + V$, $b \in \langle e_{n-k+1}, \ldots, e_n \rangle$, notées g_b :*

$$\begin{array}{rccc} g_b : & V & \to & \mathbf{F}_2 \\ & x & \mapsto & D_{\varepsilon_k} f(x+b) \end{array}$$

sont des fonctions symétriques à $(n-k)$ variables qui ne dépendent que de $\mathrm{wt}(b)$, ainsi nous les désignerons plus volontiers par $g_{\mathrm{wt}(b)}$. En outre, leur vecteur des valeurs simplifié et leur vecteur simplifié de l'ANF sont donnés par : pour tout $0 \leq i \leq n-k$

$$\begin{aligned} v_{g_b}(i) &= v_f(i + \mathrm{wt}(b)) \oplus v_f(i + k - \mathrm{wt}(b)) \\ \lambda_{g_b}(i) &= \bigoplus_{j \preceq k - \mathrm{wt}(b)} \lambda_f(i+j) \oplus \bigoplus_{j \preceq \mathrm{wt}(b)} \lambda_f(i+j). \end{aligned}$$

Nous remarquons que pour tout k tel que $1 \leq k \leq n-1$, $g_{\mathrm{wt}(b)} = g_{k-\mathrm{wt}(b)}$. D'autre part, lorsque k est pair, $g_{\frac{k}{2}} = 0$.

Preuve : Soit $b \in \overline{V} = \langle e_{n-k+1}, \ldots, e_n \rangle$. Alors, pour tout $y = x + b$ avec $x \in V$, on a

$$\begin{aligned} \mathrm{wt}(y) &= \mathrm{wt}(x) + \mathrm{wt}(b) \\ \mathrm{wt}(y + \varepsilon_k) &= \mathrm{wt}(x) + \mathrm{wt}(b + \varepsilon_k) \\ &= \mathrm{wt}(x) + k - \mathrm{wt}(b). \end{aligned}$$

Ainsi, pour tout $x \in V$,

$$\begin{aligned} D_{\varepsilon_k} f(x+b) &= f(x+b) \oplus f(x + \varepsilon_k + b) \\ &= v_f(\mathrm{wt}(x) + \mathrm{wt}(b)) \oplus v_f(\mathrm{wt}(x) + k - \mathrm{wt}(b)) \end{aligned}$$

ce qui prouve que g_b est symétrique et ne dépend que de $\mathrm{wt}(b)$.

Calculons maintenant les coefficients du vecteur de l'ANF simplifié de g_b. Décomposons pour ce faire la forme algébrique normale de f selon $(V, \overline{V})$:

$$f(x+y) = \bigoplus_{u \in \mathbf{F}_2^{n-k}} \bigoplus_{v \in \mathbf{F}_2^k} c_f(u \| v) \prod_{i=1}^{n-k} x_i^{u_i} \prod_{j=n-k+1}^{n} y_j^{v_j}, \quad (x,y) \in (V, \overline{V}).$$

Alors, pour tout $b \in \overline{V}$ et $x \in V$, on a :

$$\begin{aligned} g_b(x) &= D_{\varepsilon_k} f(x+b) \\ &= \bigoplus_{u \in \mathbf{F}_2^{n-k}} \bigoplus_{v \in \mathbf{F}_2^k} c_f(u \| v) \prod_{i=1}^{n-k} x_i^{u_i} \left(\prod_{i=1}^{k} (b_i \oplus 1)^{v_i} \oplus \prod_{i=1}^{k} b_i^{v_i} \right). \end{aligned}$$

Or, $\prod_{i=1}^{k} b_i^{v_i} = 1$ si et seulement si $\mathrm{supp}(v) \subseteq \mathrm{supp}(b)$, *i.e.*, $v \preceq b$. On obtient alors :

$$g_b(x) = \bigoplus_{u \in \mathbf{F}_2^{n-k}} \prod_{i=1}^{n-k} x_i^{u_i} \left(\bigoplus_{v \preceq \bar{b}} c_f(u\|v) \oplus \bigoplus_{v \preceq b} c_f(u\|v) \right),$$

où $\bar{b} = b + \varepsilon_k$. La fonction f étant symétrique, on a $c_f(u\|v) = \lambda_f(\mathrm{wt}(u) + \mathrm{wt}(v))$. Par conséquent,

$$\begin{aligned}
&\bigoplus_{v \preceq \bar{b}} c_f(u\|v) \oplus \bigoplus_{v \preceq b} c_f(u\|v) \\
&\qquad = \left(\bigoplus_{i=0}^{k-\mathrm{wt}(b)} \lambda_f(\mathrm{wt}(u)+i) \binom{k-\mathrm{wt}(b)}{i} \right) \oplus \left(\bigoplus_{i=0}^{\mathrm{wt}(b)} \lambda_f(\mathrm{wt}(u)+i) \binom{\mathrm{wt}(b)}{i} \right) \\
&\qquad = \bigoplus_{i \preceq k-\mathrm{wt}(b)} \lambda_f(\mathrm{wt}(u)+i) \oplus \bigoplus_{i \preceq \mathrm{wt}(b)} \lambda_f(\mathrm{wt}(u)+i).
\end{aligned}$$

Ainsi, les coefficients de la forme algébrique normale de la fonction g_b à $(n-k)$ variables sont donnés pour tout $0 \leq j \leq n-k$ par :

$$\lambda_{g_b}(j) = \bigoplus_{i \preceq k-\mathrm{wt}(b)} \lambda_f(j+i) \oplus \bigoplus_{i \preceq \mathrm{wt}(b)} \lambda_f(j+i).$$

□

Nous illustrons le résultat précédent en l'appliquant au cas des dérivées d'une fonction symétrique relativement à un vecteur de poids 1 ou de poids 2.

Corollaire 5.51 *Soit $f \in \mathcal{S}ym_n$. Alors la forme algébrique normale de $D_{e_n} f$ et de $D_{e_{n-1}+e_n} f$ est :*

$$\begin{aligned}
D_{e_n} f(x) &= \bigoplus_{i=0}^{n-1} \lambda_f(i+1) X_{i,n-1}(x) \\
D_{e_{n-1}+e_n} f(x) &= (x_{n-1} \oplus x_n \oplus 1) \bigoplus_{i=0}^{n-2} \lambda_f(i+2) X_{i,n-2}(x).
\end{aligned}$$

Preuve : Lorsque $k = 1$, nous savons d'après la proposition précédente que les restrictions de $D_{e_n} f$ à $H = \langle e_1, \ldots, e_{n-1} \rangle$ et à $e_n + H$, qu'on note g_0 et g_1 sont égales. Leur vecteur simplifié de l'ANF est donné pour tout $0 \leq i \leq n-1$ par :

$$\begin{aligned}
\lambda_{g_0}(i) &= \lambda_{g_1}(i) \\
&= \bigoplus_{j \preceq 1} \lambda_f(i+j) \oplus \bigoplus_{j \preceq 0} \lambda_f(i+j) \\
&= \lambda_f(i+1).
\end{aligned}$$

On en déduit donc que :

$$\begin{aligned} D_{e_n} f(x) &= (x_n \oplus 1) g_0(x_1, \ldots, x_{n-1}) \oplus x_n g_1(x_1, \ldots, x_{n-1}) \\ &= g_0(x_1, \ldots, x_{n-1}) \\ &= \bigoplus_{i=0}^{n-1} \lambda_f(i+1) X_{i,n-1}(x). \end{aligned}$$

Lorsque $k = 2$ et $V = \langle e_1, \ldots, e_{n-2} \rangle$, on note g_0,g_1 et g_2 les restrictions de $D_{e_{n-1}+e_n} f$ respectivement à V, à $e_n + V$ (ou de manière équivalente à $e_{n-1} + V$) et à $e_{n-1} + e_n + V$. D'après la proposition précédente, on a pour tout i, $0 \leq i \leq n-2$:

$$\begin{aligned} \lambda_{g_0}(i) &= \lambda_{g_2}(i) \\ &= \bigoplus_{j \preceq 2} \lambda_f(i+j) \oplus \bigoplus_{j \preceq 0} \lambda_f(i+j) \\ &= \lambda_f(i+2) \end{aligned}$$

et

$$\lambda_{g_1}(i) = \bigoplus_{j \preceq 1} \lambda_f(i+j) \oplus \bigoplus_{j \preceq 1} \lambda_f(i+j) = 0.$$

Ainsi,

$$D_{e_{n-1}+e_n} f(x) = (x_{n-1} \oplus x_n \oplus 1) \bigoplus_{i=0}^{n-2} \lambda_f(i+2) X_{i,n-2}(x).$$

□

La formule précédente souligne le fait que $\deg(D_{e_{n-1}+e_n} f) = \deg(f) - 1$. Nous en déduisons notamment le corollaire suivant qui généralise un résultat de Dawson et Wu [DW97] qui montre que la seule structure linéaire possible pour une fonction symétrique de degré strictement supérieur à 1 est le vecteur tout-à-un. De plus amples conditions nécessaires sur l'existence de structures linéaires sont développées dans la partie 5.5.3.

Corollaire 5.52 *Soit $f \in \mathcal{S}ym_n$. Alors pour tout $a \in \mathbf{F}_2^n \setminus \{0,\mathbf{1}\}$, $\deg(D_a f) = \deg(f) - 1$.*

Preuve : Pour tout $a \in \mathbf{F}_2^n$ tel que $a \neq 0,\mathbf{1}$, il existe $b \in \mathbf{F}_2^n$ tel que $\mathrm{wt}(b) = \mathrm{wt}(a)$ et $\mathrm{wt}(a+b) = 2$. En effet, il suffit de choisir pour b un vecteur comportant $\mathrm{wt}(a) - 1$ composantes dans le support de a et une composante à l'extérieur. On obtient alors :

$$D_a f + D_b f = D_{a+b} f + D_a D_b f.$$

Il est évident que $D_a f$ et $D_b f$ sont de degré au plus $\deg(f) - 1$ et $\deg(D_a D_b f) \leq \deg(f) - 2$. En outre, nous savons d'après le corollaire précédent que $D_{a+b} f$ est de degré exactement $\deg(f) - 1$ car $\mathrm{wt}(a+b) = 2$. On en déduit donc que $D_a f + D_b f$ est de degré $\deg(f) - 1$, ce qui implique que $\deg(D_a f) = \deg(f) - 1$. □

Exemple : Soit f une fonction booléenne symétrique à 7 variables dont le vecteur simplifié de l'ANF est donné par $\lambda(f) = (0,1,1,0,1,0,0,0)$. Le vecteur des valeurs simplifié vaut alors $v(f) = (0,1,1,0,1,0,0,1)$. Nous allons déterminer les dérivées de f relativement à $\varepsilon_k = e_{n-k+1} + \cdots + e_n$ pour $1 \leq k \leq 6$.

◇

		g_0	g_1	g_2	g_3	g_4	g_5	g_6
$D_{\varepsilon_1}f$	v	1011 101						
	λ	1101 000						
$D_{\varepsilon_2}f$	v	1100 11	0	g_0				
	λ	1010 00						
$D_{\varepsilon_3}f$	v	0010 0	0111 0	g_1	g_0			
	λ	0011 0	0111 0					
$D_{\varepsilon_4}f$	v	1111	0100	0	g_1	g_0		
	λ	1000	0101					
$D_{\varepsilon_5}f$	v	010	111	111	g_2	g_1	g_0	
	λ	010	100	100				
$D_{\varepsilon_6}f$	v	0	1	0	0	g_2	g_1	g_0
	λ							

TAB. 5.8 – *Vecteurs simplifiés des valeurs et de l'ANF des restrictions des dérivées de la fonctions symétrique f à 7 variables de vecteur simplifié de l'ANF $\lambda(f) = (0,1,1,0,1,0,0,0)$.*

5.5.2 Fonctions symétriques vérifiant le critère de propagation

Certaines applications cryptographiques nécessitent que la différence entre les sorties de la fonction booléenne utilisée soit uniformément distribuée pour des différences entre les vecteur d'entrée de poids faible. Cette propriété appelée *critère de propagation* [PLL+91], est particulièrement importante lorsque la fonction est utilisée dans une fonction de hachage ou un chiffrement par bloc.

Définition 5.53 *[PLL+91] Une fonction $f \in \mathcal{B}_n$ satisfait le* critère de propagation de degré k *($PC(k)$) si $\mathcal{F}(D_a f) = 0$ pour tout $a \in \mathbf{F}_2^n$ tel que $1 \leq \mathrm{wt}(a) \leq k$.*

On sait que les fonctions de $\mathcal{B}_n$ qui vérifient $PC(n)$ sont les fonctions courbes [MS90]. Les fonctions symétriques qui satisfont $PC(n)$ sont donc les fonctions quadratiques dont le nombre de variables est pair [Sav94]. Nous obtenons ici une caractérisation similaire des fonctions symétriques qui satisfont $PC(2)$. La proposition 5.50 et le corollaire 5.51 nous permettent d'obtenir directement le théorème suivant qui a aussi été démontré par Aline Gouget [Gou04b] .

Théorème 5.54 *Les fonctions booléennes symétriques à n variables qui satisfont le critère de propagation d'ordre 2 sont les fonctions quadratiques. En outre, elles satisfont $PC(n)$ si n est pair et $PC(n-1)$ si n est impair.*

Preuve : Soit $V = \langle e_1, \ldots, e_{n-2} \rangle$. On note g_0,g_1 et g_2 les restrictions de $D_{e_{n-1}+e_n}f$ respectivement à V, à $e_n + V$ (ou de manière équivalente à $e_{n-1} + V$) et à $e_{n-1} + e_n + V$. Le poids de $D_{e_{n-1}+e_n}f$ est donné par

$$\mathrm{wt}(D_{e_{n-1}+e_n}f) = \mathrm{wt}(g_0) + 2\,\mathrm{wt}(g_1) + \mathrm{wt}(g_2).$$

Or, d'après le corollaire 5.51 nous savons que $g_0 = g_2$ et $g_1 = 0$. Par conséquent, $D_{e_{n-1}+e_n}f$ est équilibrée si et seulement si $g_0 = g_2 = 1$. La proposition 5.50 donne le vecteur simplifié de l'ANF de g_0 et g_2 :

$$\lambda_{g_0}(i) = \lambda_{g_2}(i) = \lambda_f(i+2).$$

Elle implique que g_0 est la fonction constante égale à 1 si et seulement si $\lambda_f(2) = 1$ et pour tout i, $i \geq 3$, $\lambda_f(i) = 0$, *i.e.*, $\deg(f) = 2$.

En outre, toutes les dérivées relativement à $a \in \mathbf{F}_2^n \setminus \{0,\mathbf{1}\}$ des fonctions symétrique quadratiques sont équilibrées car elles sont de degré 1 (Corollaire 5.52). Aussi, les fonctions quadratiques sont exactement les fonctions satisfaisant $PC(n-1)$. Il nous faut donc examiner le cas de $D_{\mathbf{1}}f$ avec n pair pour déterminer complètement l'ordre du critère de propagation (lorsque n est impair, les fonctions ne peuvent pas satisfaire $PC(n)$). Par définition, $v_{D_{\mathbf{1}}f}(i) = v_f(i) \oplus v_f(n-i)$. Nous savons par ailleurs que le vecteur des valeurs simplifié d'une fonction booléenne symétrique quadratique est périodique de période 4 et qu'on a pour tout $0 \leq i \leq n-2$, $v_f(i+2) = v_f(i) \oplus 1$. On en déduit donc que pour tout k tel que $i + 2k \leq n-2$, $v_f(i+2k) = v_f(i) \oplus k \bmod 2$. Ainsi si on choisit $2k = n - 2i$ — ce qu'on ne peut faire que si n est pair — on obtient $v_f(n-i) = v_f(i) \oplus \left(\frac{n}{2} - i\right) \bmod 2$. Finalement, on en déduit que

$$\begin{aligned} v_{D_{\mathbf{1}}f}(i) &= v_f(i) \oplus v_f(n-i) \\ &= \left(\frac{n}{2} - i\right) \bmod 2 \\ &= i \bmod 2 \oplus \frac{n}{2} \bmod 2. \end{aligned}$$

Ainsi, lorsque f est quadratique, n pair, alors $D_{\mathbf{1}}f$ est affine, donc équilibrée. Dans ce cas f vérifie donc $PC(n)$. Nous retrouvons ainsi le résultat de Savicky [Sav94] : les fonctions booléennes symétriques courbes sont les fonctions quadratiques.

□

Ainsi, le seul problème ouvert reste la caractérisation des fonctions symétriques qui satisfont $PC(1)$. Une fraction importante de ces fonctions sont celles dont les dérivées relativement à tout vecteur de poids 1 ont des restrictions équilibrées triviales (dans le sens de la proposition 5.50). Ces fonctions peuvent être caractérisées comme suit :

Proposition 5.55 *Soit $f \in \mathcal{S}ym_n$, n pair. Alors, les assertions suivantes sont équivalentes:*

(i) *f satisfait $PC(1)$ et les restrictions de $D_{e_n}f$ à $\langle e_1, \ldots, e_{n-1}\rangle$ sont des fonctions équilibrées triviales;*

(ii) *$D_{\mathbf{1}}f = \varphi_{\mathbf{1}} + \varepsilon$, $\varepsilon \in \mathbf{F}_2^n$;*

(iii) *il existe $\varepsilon \in \mathbf{F}_2$ tel que $\forall\, i$, $0 \leq i \leq n$, $\mathsf{F}_f(n-i) = (-1)^{i+\varepsilon}\mathsf{F}_f(i)$.*

En outre, si f vérifie une des propriétés ci-dessus, alors $D_a f$ est équilibrée pour tout $a \in \mathbf{F}_2^n$ dont le poids de Hamming est impair.

Preuve : Tout d'abord, montrons que **(i)** et **(ii)** sont équivalents. Dans ce qui suit, on note g la fonction symétrique à $(n-1)$ variables qui correspond à la restriction de $D_{e_n}f$ à $\langle e_1, \ldots, e_{n-1}\rangle$ (ou de manière équivalente à $e_n + \langle e_1, \ldots, e_{n-1}\rangle$). D'après la proposition 5.39, on a :

$$v_g(i) = v_f(i) \oplus v_f(i+1), \ \forall i, 0 \leq i \leq n-1.$$

Aussi, g est équilibrée triviale si et seulement si pour tout i, $0 \leq i \leq n-1$, $v_g(i) = v_g(n-1-i) \oplus 1$, ce qui veut dire que pour tout i, $0 \leq i \leq n-1$,

$$v_f(i) \oplus v_f(i+1) = v_f(n-i-1) \oplus v_f(n-i) \oplus 1. \tag{5.3}$$

De manière évidente,$D_{\mathbf{1}}f$ est une fonction symétrique à n variables car elle vérifie :

$$D_{\mathbf{1}}f(x) = f(x) \oplus f(x+\mathbf{1}) = v_f(\mathrm{wt}(x)) \oplus v_f(n - \mathrm{wt}(x)).$$

Ainsi, l'équation (5.3) est équivalente à

$$v_{D_{\mathbf{1}}f}(i) = v_{D_{\mathbf{1}}f}(i+1) \oplus 1, \ \forall i, 0 \leq i \leq n-1,$$

ce qui signifie que $D_{\mathbf{1}}f$ est de degré 1 (Th. 5.33).

Prouvons maintenant que **(ii)** et **(iii)** sont équivalents. Pour ce faire, calculons $\mathcal{F}(f + D_{\mathbf{1}}f + \varphi_a)$, $a \in \mathbf{F}_2^n$:

$$\begin{aligned}\mathcal{F}(f + D_{\mathbf{1}}f + \varphi_a) &= \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x+\mathbf{1}) \oplus a \cdot x} \\ &= (-1)^{\mathrm{wt}(a)} \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x) \oplus a \cdot x} \\ &= (-1)^{\mathrm{wt}(a)} \mathcal{F}(f + \varphi_a).\end{aligned}$$

Or, $D_{\mathbf{1}}f = \varphi_{\mathbf{1}} + \varepsilon, \ \varepsilon \in \mathbf{F}_2$, si et seulement si pour tout $a \in \mathbf{F}_2^n$,

$$\mathcal{F}(f + D_{\mathbf{1}}f + \varphi_a) = \mathcal{F}(f + \varphi_{\overline{a}} + \varepsilon) = (-1)^{\varepsilon} \mathcal{F}(f + \varphi_{\overline{a}}).$$

Ainsi, $D_{\mathbf{1}}f = \varphi_{\mathbf{1}} + \varepsilon$ si et seulement si $\mathcal{F}(f + \varphi_{\overline{a}}) = (-1)^{\mathrm{wt}(a)+\varepsilon} \mathcal{F}(f + \varphi_a)$ pour tout $a \in \mathbf{F}_2^n$.

Montrons finalement que **(iii)** implique que $\forall a \in \mathbf{F}_2^n$, $\mathrm{wt}(a)$ impair, $\mathcal{F}(D_a f) = 0$. Soit H l'hyperplan composé des vecteurs de n bits de poids pair. Nous déduisons de [CCCF01, Théorème V.1] que

$$\mathcal{F}^2(f + \varphi_a) + \mathcal{F}^2(f + \varphi_{\overline{a}}) = 2 \sum_{e \in H} (-1)^{a \cdot e} \mathcal{F}(D_e f).$$

Par ailleurs, l'assertion **(iii)** implique que

$$\mathcal{F}^2(f + \varphi_a) + \mathcal{F}^2(f + \varphi_{\overline{a}}) = 2\mathcal{F}^2(f + \varphi_a).$$

Or,

$$\mathcal{F}^2(f + \varphi_a) = \sum_{e \in \mathbf{F}_2^n} (-1)^{a \cdot e} \mathcal{F}(D_e f).$$

Aussi, en combinant les deux égalités, nous obtenons que pour tout $a \in \mathbf{F}_2^n$,

$$\sum_{e \in \overline{H}} (-1)^{a \cdot e} \mathcal{F}(D_e f) = 0,$$

où $\overline{H} = \mathbf{F}_2^n \setminus H$. Nous déduisons que pour tout $u \in \overline{H}$

$$\begin{aligned}\sum_{a \in \mathbf{F}_2^n} (-1)^{a \cdot u} \sum_{e \in \overline{H}} (-1)^{a \cdot e} \mathcal{F}(D_e f) &= \sum_{e \in \overline{H}} \mathcal{F}(D_e f) \sum_{a \in \mathbf{F}_2^n} (-1)^{a \cdot (u+e)} \\ &= 2^n \mathcal{F}(D_u f) = 0.\end{aligned}$$

□

5.5.3 Dérivée relativement au vecteur tout-à-un

Nous portons finalement notre attention sur le dernier cas de dérivées non traité par la proposition 5.50, la dérivée relativement au vecteur tout-à-un. L'intérêt principal de ce cas est qu'il permet de déterminer complètement si une fonction symétrique a ou non une structure linéaire. Nous donnons ici l'expression de la forme algébrique normale de $D_{\mathbf{1}}f$ en fonction du vecteur simplifié de l'ANF de f. Ceci nécessite un résultat liminaire qui met en jeu les dérivées d'ordre supérieur de f. Pour tout sous-espace $V \subset \mathbf{F}_2^n$ de dimension k, nous notons $D_V^{(k)}f$ la dérivée d'ordre k de $f \in \mathcal{B}_n$ relativement à V, *i.e.* la fonction à n variables,

$$D_V^{(k)}f = D_{a_1}D_{a_2}\dots D_{a_k}f$$

où $(a_1,\dots,a_k)$ est une base quelconque de V.

Lemme 5.56 *La dérivée de $f \in \mathcal{B}_n$ relativement au vecteur tout-à-un est:*

$$D_{\mathbf{1}}f = \sum_{k=1}^{n} \sum_{1\le i_1<\dots<i_k\le n} D^{(k)}_{\langle e_{i_1},\dots,e_{i_k}\rangle}f.$$

Preuve : L'égalité est vérifiée pour $n = 1$. Nous la prouvons ensuite par récurrence sur n. Écrivons f sous la forme

$$f(x_1,\dots,x_n) = x_n f_1(x_1,\dots,x_{n-1}) \oplus f_2(x_1,\dots,x_{n-1}),$$

où $f_1,f_2 \in \mathcal{B}_{n-1}$. Nous obtenons alors :

$$\begin{aligned} D_{\mathbf{1}_n}f(x_1,\dots,x_n) &= x_n f_1(x_1,\dots,x_{n-1}) \oplus f_2(x_1,\dots,x_{n-1}) \\ &\quad \oplus (x_n\oplus 1)f_1(x_1\oplus 1,\dots,x_{n-1}\oplus 1) \oplus f_2(x_1\oplus 1,\dots,x_{n-1}\oplus 1) \\ &= x_n D_{\mathbf{1}_{n-1}}f_1(x_1,\dots,x_{n-1}) \oplus D_{\mathbf{1}_{n-1}}f_2(x_1,\dots,x_{n-1}) \\ &\quad \oplus f_1(x_1\oplus 1,\dots,x_{n-1}\oplus 1). \end{aligned}$$

Par ailleurs, évaluons l'expression

$$A_n(f) = \sum_{k=1}^{n} \sum_{1\le i_1<\dots<i_k\le n} D^{(k)}_{\langle e_{i_1},\dots,e_{i_k}\rangle}f.$$

$$\begin{aligned} A_n(f) &= \sum_{k=1}^{n} \sum_{1\le i_1<\dots<i_k\le n} D^{(k)}_{\langle e_{i_1},\dots,e_{i_k}\rangle}(x_n f_1 + f_2) \\ &= \sum_{k=1}^{n-1} \sum_{1\le i_1<\dots<i_k\le n-1} D^{(k)}_{\langle e_{i_1},\dots,e_{i_k}\rangle}(x_n f_1 + f_2) \\ &\quad + \sum_{k=1}^{n-1} \sum_{1\le i_1<\dots<i_k\le n-1} D_{e_n}D^{(k)}_{\langle e_{i_1},\dots,e_{i_k}\rangle}(x_n f_1 + f_2) + D_{e_n}(x_n f_1 + f_2) \\ &= x_n A_{n-1}(f_1) + A_{n-1}(f_2) + A_{n-1}(f_1) + f_1, \end{aligned}$$

puisque $D_{e_n}f_2 = 0$ car f_2 ne dépend pas de x_n.

En appliquant l'hypothèse de récurrence à f_1 et f_2, on obtient alors:

$$A_n(f) = x_n D_{\mathbf{1}_{n-1}} f_1 + D_{\mathbf{1}_{n-1}} f_2 + D_{\mathbf{1}_{n-1}} f_1 + f_1 = D_{\mathbf{1}_n} f.$$

□

Nous déduisons ainsi la forme algébrique normale de $D_{\mathbf{1}} f$.

Proposition 5.57 *Soit $f \in \mathcal{S}ym_n$. La dérivée de f relativement au vecteur tout-à-un est une fonction booléenne symétrique à n variables dont le vecteur des valeurs simplifié et le vecteur simplifié de l'ANF sont donnés par : pour tout i, $0 \leq i \leq n$,*

$$v_{D_{\mathbf{1}} f}(i) = v_f(i) \oplus v_f(n-i)$$
$$\lambda_{D_{\mathbf{1}} f}(i) = \bigoplus_{\substack{k \preceq n-i \\ k \neq 0}} \lambda_f(i+k).$$

Preuve : La relation entre le vecteur des valeurs simplifié de f et de $D_{\mathbf{1}} f$ provient directement de la définition. Nous déterminons à présent le vecteur simplifié de l'ANF de $D_{\mathbf{1}} f$. D'après le lemme précédent, nous avons :

$$D_{\mathbf{1}} f = \sum_{k=1}^{n} \sum_{1 \leq i_1 < \cdots < i_k \leq n} D^{(k)}_{\langle e_{i_1}, \ldots, e_{i_k} \rangle} f.$$

La dérivée d'ordre k de f, $D^{(k)}_{\langle e_{i_1}, \ldots, e_{i_k} \rangle} f$ ne dépend que des $(n-k)$ variables dont les indices appartiennent à $\{1, \ldots, n\} \setminus \{i_1, \ldots, i_k\} = \overline{\{i_1, \ldots, i_k\}}$. Plus précisément, nous déduisons du corollaire 5.51 page 139 que

$$D^{(k)}_{\langle e_{i_1}, \ldots, e_{i_k} \rangle} f = \bigoplus_{i=0}^{n-k} \lambda_f(i+k) X_{i, \overline{\{i_1, \ldots, i_k\}}}(x).$$

Aussi,

$$\left(\sum_{1 \leq i_1 < \cdots < i_k \leq n} D^{(k)}_{\langle e_{i_1}, \ldots, e_{i_k} \rangle} f \right)(x) = \bigoplus_{1 \leq i_1 < \cdots < i_k \leq n} \bigoplus_{i=0}^{n-k} \lambda_f(i+k) X_{i, \overline{\{i_1, \ldots, i_k\}}}(x)$$
$$= \bigoplus_{i=0}^{n-k} \lambda_f(i+k) \bigoplus_{1 \leq i_1 < \cdots < i_k \leq n} X_{i, \overline{\{i_1, \ldots, i_k\}}}(x)$$

Nous devons maintenant simplifier la somme :

$$\bigoplus_{1 \leq i_1 < \cdots < i_k \leq n} X_{i, \overline{\{i_1, \ldots, i_k\}}}(x).$$

Elle est composée de $\binom{n}{n-k}$ polynômes symétriques élémentaires de degré i qui peuvent avoir des variables en commun. Pour tout sous-ensemble fixé de i indices, le produit des i variables

correspondantes apparaît dans exactement $\binom{n-i}{n-k-i}$ termes $X_{i,\{\dots\}}$. Aussi, la somme consiste en tous les produits de i variables, chacun d'eux étant répété $\binom{n-i}{n-k-i}$ fois, *i.e.*,

$$\bigoplus_{1\le i_1<\cdots<i_k\le n} X_{i,\overline{\{i_1,\dots,i_k\}}}(x) = \binom{n-i}{n-k-i} X_{i,n}(x).$$

Nous obtenons ainsi :

$$\left(\sum_{1\le i_1<\cdots<i_k\le n} D^{(k)}_{\langle e_{i_1},\dots,e_{i_k}\rangle} f\right)(x) = \bigoplus_{i=0}^{n-k} \lambda_f(i+k)\left(\binom{n-i}{n-k-i} \bmod 2\right) X_{i,n}(x).$$

Ce qui nous donne pour l'expression de la dérivée :

$$\begin{aligned} D_{\mathbf{1}} f(x) &= \bigoplus_{k=1}^{n}\left(\bigoplus_{i=0}^{n-k} \lambda_f(i+k)\left(\binom{n-i}{k} \bmod 2\right) X_{i,n}(x)\right) \\ &= \bigoplus_{i=0}^{n-1}\left(\bigoplus_{k=1}^{n-i} \lambda_f(i+k)\left(\binom{n-i}{k} \bmod 2\right)\right) X_{i,n}(x) \\ &= \bigoplus_{i=0}^{n-1} \bigoplus_{\substack{k\preceq n-i\\ \overline{k}\neq 0}} \lambda_f(i+k) X_{i,n}(x). \end{aligned}$$

□

Nous en déduisons immédiatement le résultat suivant sur le degré de $D_{\mathbf{1}} f$.

Proposition 5.58 *Soit $f \in Sym_n$ de degré d. Alors le degré de $D_{\mathbf{1}} f$ est donné par :*

- $\deg(D_{\mathbf{1}} f) = d-1$ *si et seulement si* $(n-d)$ *est pair ;*
- *si* $(n-d)$ *est impair alors*
  - $\deg(D_{\mathbf{1}} f) = d-2$ *si* $\lambda_f(d-1) = \frac{n-d-1}{2} \bmod 2$,
  - $\deg(D_{\mathbf{1}} f) \le \deg(f) - 4$ *sinon.*

Preuve : Notons μ le vecteur simplifié de l'ANF de $D_{\mathbf{1}} f$. L'expression des coefficients est :

$$\mu(i) = \bigoplus_{\substack{k\preceq n-i\\ \overline{k}\neq 0}} \lambda_f(i+k).$$

On a alors :

$$\mu(d-1) = \bigoplus_{\substack{k\preceq n-d+1\\ k\neq 0}} \lambda_f(d-1+k).$$

$\mu(d-1)$ vaut 0 si $1 \not\preceq n-d+1$, c'est-à-dire si $n-d$ est impair et $\mu(d-1) = \lambda_f(d)$ si et seulement si $1 \preceq n-d+1$ c'est-à-dire si $n-d$ est pair. Pour essayer de déterminer le degré de la dérivée dans le cas où $n-d$ est impair, nous devons calculer $\mu(d-2)$:

$$\mu(d-2) = \bigoplus_{\substack{k\preceq n-d+2\\ k\neq 0}} \lambda_f(d-2+k).$$

Un argument similaire au précédent nous permet de dire que :

– si $n-d \equiv 1 \bmod 4$ alors $\mu(d-2) = \lambda_f(d) \oplus \lambda_f(d-1)$ et $\mu(d-3) = 0$. Aussi,

$$\deg(D_{\mathbf{1}}f) \begin{cases} = d-2 & \text{si } \lambda_f(d-1) = 0 \\ \leq d-4 & \text{si } \lambda_f(d-1) = 1 \end{cases};$$

– si $n-d \equiv 3 \bmod 4$ alors $\mu(d-2) = \lambda_f(d-1)$ et $\mu(d-3) = \lambda_f(d-1)$. Aussi,

$$\deg(D_{\mathbf{1}}f) \begin{cases} = d-2 & \text{si } \lambda_f(d-1) = 1 \\ \leq d-4 & \text{si } \lambda_f(d-1) = 0 \end{cases}.$$

□

Une conséquence immédiate de ce résultat est que $f \in \mathcal{S}ym_n$, $\deg(f) \neq 1$, ne peut pas avoir de structure linéaire si $n - \deg(f)$ est pair. Plus précisément on obtient le corollaire suivant.

Corollaire 5.59 *Soit $f \in \mathcal{S}ym_n$, $\deg(f) \neq 2$. Si f possède une structure linéaire non-triviale, alors $n - \deg(f)$ est impair et*

$$\lambda_f(\deg(f)-1) \equiv \frac{n-\deg(f)+1}{2} \bmod 2.$$

On remarque en particulier que les fonctions équilibrées triviales étudiées à la partie 5.2.1 sont de degré d pair et que le coefficient des monômes de degré $d-1$ est entièrement déterminé par les valeurs de n et d.

Chapitre 6

Fonctions symétriques de degré faible

Dans ce chapitre, je m'attache en particulier à l'étude des fonctions booléennes symétriques de petit degré. En effet, j'ai montré dans le paragraphe 5.3 de la page 123 que les fonctions symétriques de petit degré sont caractérisées par une petite période du vecteur des valeurs simplifié. Cette propriété les rend donc particulièrement intéressante en vue de leur implémentation logicielle ou matérielle. Il est possible de tirer parti de cette petite période pour simplifier le calcul des coefficients de Walsh d'une fonction booléenne symétrique en utilisant une propriété de séries extraites. Ainsi j'ai pu caractériser entièrement les fonctions booléennes symétriques quadratiques et cubiques dont le vecteur des valeurs est périodique de période 4. Bien que la technique utilisée se complique un peu pour les degrés supérieurs, il est néanmoins possible de déterminer toutes les fonctions équilibrées à n variables de degré au plus 7 et d'étudier en détails certaines propriétés pour les degrés inférieurs à 8. Il s'agit des premiers résultats (non-triviaux) sur le poids des fonctions symétriques qui portent sur des familles de fonctions quel que soit leur nombre de variables.

6.1 Expression des coefficients de Walsh utilisant la périodicité

Dans cette partie, je détermine les valeurs du coefficients de Walsh en utilisant la périodicité pour simplifier son expression dans le cas des fonctions booléennes symétriques.

Grâce à la périodicité du vecteur des valeurs, nous pouvons modifier l'écriture de l'expression des coefficients de Walsh d'une fonction booléenne symétrique f à n variables et de degré inférieur à $2^\ell - 1$. Nous exprimons $\mathsf{F}_f(j)$ pour $0 \leq j \leq n$:

$$\mathsf{F}_f(j) = \sum_{w=0}^{n} (-1)^{v_f(w)} P_w(j,n)$$

où P_w est le polynôme de Krawtchouk binaire de degré w, c'est-à-dire le coefficient de x^w dans la forme développée du polynôme $K_{j,n}(x) = (1-x)^j(1+x)^{(n-j)}$,

$$P_w(j,n) = \sum_{k=0}^{w} \binom{j}{k}\binom{n-j}{w-k}(-1)^k.$$

Ainsi, pour une fonction f de degré $\deg(f) \leq 2^\ell - 1$, nous pouvons écrire:

$$\begin{aligned}\mathsf{F}_f(j) &= \sum_{w=0}^{n} (-1)^{v_f(w)} P_w(j,n) \\ &= \sum_{w=0}^{2^\ell-1} (-1)^{v_f(w)} \sum_{\substack{0\leq k\leq n \\ k\equiv w \bmod 2^\ell}} P_k(j,n)\end{aligned}$$

Nous utilisons ici une formule appelée *series multisection* [Com74]. Soit ϕ une fonction que nous écrivons sous forme de série formelle,

$$\phi(x) = \sum_{j=0}^{\infty} \phi_j x^j,$$

alors la valeur de la somme

$$S_\phi(j_0,T)(x) = \sum_{j=0}^{\infty} \phi_{j_0+jT} x^{j_0+jT} = \sum_{\substack{0\leq k\leq n \\ k\equiv j_0 \bmod T}} \phi_k x^k$$

est donnée par la formule

$$S_\phi(j_0,T)(x) = \frac{1}{T} \sum_{t=0}^{T-1} e^{-\frac{2i\pi}{T} j_0 t} \phi(e^{\frac{2i\pi}{T} t} x), \quad \text{avec } i^2 = -1.$$

Nous nous intéressons à la valeur de $S_{K_{j,n}}(w,2^\ell)(1)$ dont nous abrégeons la notation pour plus de commodité :

$$S_{K_{j,n}}(w,2^\ell) = \sum_{\substack{0\leq k\leq n \\ k\equiv w \bmod 2^\ell}} P_k(j,n) = \frac{1}{2^\ell} \sum_{t=0}^{2^\ell-1} e^{-\frac{2i\pi}{2^\ell} wt} K_{j,n}(e^{\frac{2i\pi}{2^\ell} t}),$$

car nous calculons :

$$\mathsf{F}_f(j) = \sum_{w=0}^{2^\ell-1} (-1)^{v_w} S_{K_{j,n}}(w,2^\ell).$$

Le développement de la formule dans le cas des polynômes de Krawtchouk nous donne :

- quand j est pair :

$$S_{K_{j,n}}(w,T) = \frac{1}{T} \sum_{t=0}^{T-1} (-1)^{\frac{j}{2}} \cos\left(t(n-2w)\frac{\pi}{T}\right) \left(2\cos\left(t\frac{\pi}{T}\right)\right)^{n-j} \left(2\sin\left(t\frac{\pi}{T}\right)\right)^{j},$$

- quand j est impair :

$$S_{K_{j,n}}(w,T) = \frac{1}{T} \sum_{t=0}^{T-1} (-1)^{\frac{j-1}{2}} \sin\left(t(n-2w)\frac{\pi}{T}\right) \left(2\cos\left(t\frac{\pi}{T}\right)\right)^{n-j} \left(2\sin\left(t\frac{\pi}{T}\right)\right)^{j}.$$

Dans le cas où $j = 0$, on retrouve naturellement le cas des coefficients binomiaux (voir par exemple [Com74, p 84]).

$$S_{B_n}(w,T) = \sum_{\substack{0\le t\le n\\ t\equiv w \bmod T}} \binom{n}{t}.$$

Nous obtenons dans ce cas-là, en notant $B_n(x) = (1+x)^n$:

$$S_{B_n}(w,T) = \frac{1}{T}\sum_{t=0}^{T-1} \cos\left(t(n-2w)\frac{\pi}{T}\right)\left(2\cos\left(t\frac{\pi}{T}\right)\right)^n.$$

Le cas où $j = n$, correspond au cas des coefficients binomiaux aux signes alternés.

$$S_{A_n}(w,T) = \sum_{\substack{0\le t\le n\\ t\equiv w \bmod T}} \binom{n}{t}(-1)^t$$

En notant $A_n(x) = (1-x)^n$, nous obtenons :

$$S_{A_n}(w,T) = \begin{cases} \dfrac{1}{T}\displaystyle\sum_{t=0}^{T-1}(-1)^{\frac{n}{2}} \cos\left(t(n-2w)\frac{\pi}{T}\right)\left(2\sin\left(t\frac{\pi}{T}\right)\right)^n & \text{si } n \text{ est pair,}\\ \dfrac{1}{T}\displaystyle\sum_{t=0}^{T-1}(-1)^{\frac{n-1}{2}} \sin\left(t(n-2w)\frac{\pi}{T}\right)\left(2\sin\left(t\frac{\pi}{T}\right)\right)^n & \text{si } n \text{ est impair.}\end{cases}$$

Dans le cas où $T = 2^\ell$, on cherche à simplifier l'expression de la somme en utilisant la parité des fonctions trigonométriques. Nous allons d'abord traiter le cas pair, le cas impair étant très similaire, nous n'en donnerons que le résultat. Le cas $j = 0$ se distingue à cause de la valeur de $\left(2\sin\left(t\frac{\pi}{T}\right)\right)^j$ lorsque $t = 0$, de même pour le cas $j = n$ à cause de la valeur de $\left(2\cos\left(t\frac{\pi}{2^\ell}\right)\right)^{n-j}$ lorsque $t = 2^{\ell-1}$. Pour $0 \le j \le n$, j pair, on a :

$$\begin{aligned} 2^\ell S_{K_{j,n}}(w,2^\ell) = {} & 2^{n-j}0^j \\ & + \sum_{t=1}^{2^{\ell-1}-1} (-1)^{\frac{j}{2}} \cos\left(t(n-2w)\frac{\pi}{2^\ell}\right)\left(2\cos\left(t\frac{\pi}{2^\ell}\right)\right)^{n-j}\left(2\sin\left(t\frac{\pi}{2^\ell}\right)\right)^j \\ & + (-1)^{\frac{j}{2}} \cos\left((n-2w)\frac{\pi}{2}\right) 2^j 0^{n-j} \\ & + \sum_{t=2^{\ell-1}+1}^{2^\ell-1} (-1)^{\frac{j}{2}} \cos\left(t(n-2w)\frac{\pi}{2^\ell}\right)\left(-2\cos\left((2^\ell-t)\frac{\pi}{2^\ell}\right)\right)^{n-j}\left(2\sin\left((2^\ell-t)\frac{\pi}{2^\ell}\right)\right)^j. \end{aligned}$$

Il faut simplifier la dernière partie de la sommation :

$$\begin{aligned} &\sum_{t=2^{\ell-1}+1}^{2^\ell-1} (-1)^{\frac{j}{2}} \cos\left(t(n-2w)\frac{\pi}{2^\ell}\right)\left(-2\cos\left((2^\ell-t)\frac{\pi}{2^\ell}\right)\right)^{n-j}\left(2\sin\left((2^\ell-t)\frac{\pi}{2^\ell}\right)\right)^j = \\ &\qquad \sum_{k=1}^{2^{\ell-1}-1} (-1)^{\frac{j}{2}} \cos\left((2^\ell-k)(n-2w)\frac{\pi}{2^\ell}\right)\left(-2\cos\left(k\frac{\pi}{2^\ell}\right)\right)^{n-j}\left(2\sin\left(k\frac{\pi}{2^\ell}\right)\right)^j. \end{aligned}$$

On se sert alors du fait que

$$\begin{aligned}\cos\left((2^\ell-k)(n-2w)\frac{\pi}{2^\ell}\right) &= \cos\left((n-2w)\pi\right)\cos\left(k(n-2w)\frac{\pi}{2^\ell}\right)\\ &= (-1)^n\cos\left(k(n-2w)\frac{\pi}{2^\ell}\right),\end{aligned}$$

pour obtenir dans le cas où $j=0$:

$$S_{B_n}(w,T) = 2^{n-\ell} + 2^{1-\ell}\sum_{t=1}^{2^{\ell-1}-1}\cos\left(t(n-2w)\frac{\pi}{2^\ell}\right)\left(2\cos\left(t\frac{\pi}{2^\ell}\right)\right)^n, \tag{6.1}$$

dans le cas où $j=n$, si n est pair :

$$S_{A_n}(w,T) = (-1)^{\frac{j}{2}}\cos\left((n-2w)\frac{\pi}{2}\right)2^{n-\ell}+2^{1-\ell}\sum_{t=1}^{2^{\ell-1}-1}(-1)^{\frac{n}{2}}\cos\left(t(n-2w)\frac{\pi}{2^\ell}\right)\left(2\sin\left(t\frac{\pi}{2^\ell}\right)\right)^n \tag{6.2}$$

et pour j pair, $j\neq 0$ et $j\neq n$:

$$S_{K_{j,n}}(w,2^\ell) = 2^{1-\ell}\sum_{t=1}^{2^{\ell-1}-1}(-1)^{\frac{j}{2}}\cos\left(t(n-2w)\frac{\pi}{2^\ell}\right)\left(2\cos\left(t\frac{\pi}{2^\ell}\right)\right)^{n-j}\left(2\sin\left(t\frac{\pi}{2^\ell}\right)\right)^j. \tag{6.3}$$

Dans le cas où j est impair, $j\neq n$, des calculs similaires nous donnent le résultat suivant :

$$S_{K_{j,n}}(w,2^\ell) = 2^{1-\ell}\sum_{t=1}^{2^{\ell-1}-1}(-1)^{\frac{j-1}{2}}\sin\left(t(n-2w)\frac{\pi}{2^\ell}\right)\left(2\cos\left(t\frac{\pi}{2^\ell}\right)\right)^{n-j}\left(2\sin\left(t\frac{\pi}{2^\ell}\right)\right)^j. \tag{6.4}$$

et si n est impair :

$$S_{A_n}(w,T) = (-1)^{\frac{j}{2}}\sin\left((n-2w)\frac{\pi}{2}\right)2^{n-\ell}+2^{1-\ell}\sum_{t=1}^{2^{\ell-1}-1}(-1)^{\frac{n-1}{2}}\sin\left(t(n-2w)\frac{\pi}{2^\ell}\right)\left(2\sin\left(t\frac{\pi}{2^\ell}\right)\right)^n \tag{6.5}$$

6.2 Étude de la période 4

6.2.1 Calculs préliminaires

Lorsque $j\neq 0$ et $j\neq n$, les valeurs de $S_{K_{j,n}}(w,4)$ dépendant de $n-2w \bmod 4$, sont données par :

$$S_{K_{j,n}}(w,4) = \begin{cases}(-1)^{\frac{j}{2}}(-1)^{\frac{n-2w}{4}}2^{\frac{n}{2}-1} & \text{si } n-2w\equiv 0 \bmod 4\\ (-1)^{\frac{j}{2}}(-1)^{\frac{n-2w-1}{4}}2^{\frac{n-3}{2}} & \text{si } n-2w\equiv 1 \bmod 4\\ 0 & \text{si } n-2w\equiv 2 \bmod 4\\ (-1)^{\frac{j}{2}}(-1)^{\frac{n-2w+1}{4}}2^{\frac{n-3}{2}} & \text{si } n-2w\equiv 3 \bmod 4\end{cases}, \text{ et } j \text{ pair.} \tag{6.6}$$

$$S_{K_{j,n}}(w,4) = \begin{cases}0 & \text{si } n-2w\equiv 0 \bmod 4\\ (-1)^{\frac{j-1}{2}}(-1)^{\frac{n-2w-1}{4}}2^{\frac{n-3}{2}} & \text{si } n-2w\equiv 1 \bmod 4\\ (-1)^{\frac{j-1}{2}}(-1)^{\frac{n-2w-2}{4}}2^{\frac{n}{2}-1} & \text{si } n-2w\equiv 2 \bmod 4\\ (-1)^{\frac{j-1}{2}}(-1)^{\frac{n-2w-3}{4}}2^{\frac{n-3}{2}} & \text{si } n-2w\equiv 3 \bmod 4\end{cases}, \text{ et } j \text{ impair.} \tag{6.7}$$

Lorsque $j = 0$, on a :

$$S_{B_n}(w,4) = \begin{cases} 2^{n-2} + (-1)^{\frac{n-2w}{4}} 2^{\frac{n}{2}-1} & \text{si } n-2w \equiv 0 \bmod 4 \\ 2^{n-2} + (-1)^{\frac{n-2w-1}{4}} 2^{\frac{n-3}{2}} & \text{si } n-2w \equiv 1 \bmod 4 \\ 2^{n-2} & \text{si } n-2w \equiv 2 \bmod 4 \\ 2^{n-2} + (-1)^{\frac{n-2w+1}{4}} 2^{\frac{n-3}{2}} & \text{si } n-2w \equiv 3 \bmod 4 \end{cases}. \tag{6.8}$$

Lorsque $j = n$ avec n pair, on a :

$$S_{A_n}(w,4) = \begin{cases} (-1)^{\frac{n}{2}} 2^{n-2} + (-1)^{\frac{n}{2}} (-1)^{\frac{n-2w}{4}} 2^{\frac{n}{2}-1} & \text{si } n-2w \equiv 0 \bmod 4 \\ (-1)^{\frac{n}{2}+1} 2^{n-2} & \text{si } n-2w \equiv 2 \bmod 4 \end{cases}. \tag{6.9}$$

Enfin, lorsque $j = n$ avec n impair, on a :

$$S_{A_n}(w,4) = \begin{cases} (-1)^{\frac{n-1}{2}} 2^{n-2} + (-1)^{\frac{n-1}{2}} (-1)^{\frac{n-2w-1}{4}} 2^{\frac{n-3}{2}} & \text{si } n-2w \equiv 1 \bmod 4 \\ (-1)^{\frac{n+1}{2}} 2^{n-2} + (-1)^{\frac{n-1}{2}} (-1)^{\frac{n-2w-3}{4}} 2^{\frac{n-3}{2}} & \text{si } n-2w \equiv 3 \bmod 4 \end{cases}. \tag{6.10}$$

6.2.2 Les fonctions symétriques quadratiques

Nous sommes maintenant en mesure de fournir une description exhaustive des fonctions booléennes symétriques quadratiques à n variables. Nous savons déjà que les fonctions symétriques à n variables sont courbes lorsque n est pair [Sav94]et que leur spectre de Walsh est tri-valué et prend les valeurs $\{0, \pm 2^{\frac{n+1}{2}}\}$ si n est impair [MS02]. Cependant, nous pouvons améliorer ces résultats grâce aux calculs précédents et en déduire la caractérisation complète de toutes les fonctions symétriques quadratiques.

Proposition 6.1 *Soit $f \in \mathcal{S}ym_n$ une fonction de degré 2, $n \geq 3$, et de vecteur simplifié de l'ANF $\lambda(f) = (0,\lambda,1,0,\ldots,0)$, $\lambda \in \mathbf{F}_2$. Alors ses caractéristiques sont données dans le tableau 6.1.*

Le tableau 6.1 nous donne, en plus du spectre des fonctions quadratiques qui était bien connu, les valeurs exactes de chaque coefficient de Walsh, y compris leur signe. Cette information est connue pour très peu de familles infinies de fonctions booléennes.

De la même manière, si nous considérons la fonction quadratique g de vecteur simplifié de l'ANF $(1,\lambda,1,0,\ldots,0)$, i.e., *$g = f + 1$, nous déduisons immédiatement ses caractéristiques du tableau 6.1 par*

$$\mathcal{F}(g + \varphi_a) = -\mathcal{F}(f + \varphi_a),\ a \in \mathbf{F}_2^n,$$

c'est-à-dire $\mathsf{F}_g(j) = -\mathsf{F}_f(j)$, $j = 0,\ldots,n$ et

$$\mathcal{F}(D_a g) = \mathcal{F}(D_a f),\ a \in \mathbf{F}_2^n,$$

c'est-à-dire $\mathsf{Ac}_g(j) = \mathsf{Ac}_f(j)$, $j = 0,\ldots,n$.

Preuve : Soit $f \in \mathcal{S}ym_n$ de vecteur simplifié de l'ANF $(0,\lambda,1,0,\ldots,0)$. D'après le théorème 5.33, nous savons que son vecteur des valeurs simplifié $v(f)$ est la troncature à n éléments de $(0,\lambda,1,\lambda \oplus 1)^*$. Nous pouvons donc écrire :

$$\begin{aligned} \mathsf{F}_f(j) &= S_{K_{j,n}}(0,4) + (-1)^{\lambda} S_{K_{j,n}}(1,4) - S_{K_{j,n}}(2,4) + (-1)^{\lambda+1} S_{K_{j,n}}(3,4) \\ &= S_{K_{j,n}}(0,4) - S_{K_{j,n}}(2,4) + (-1)^{\lambda} \left(S_{K_{j,n}}(1,4) - S_{K_{j,n}}(3,4) \right) \end{aligned} \tag{6.11}$$

TAB. 6.1 – *Caractéristiques des fonctions booléennes symétriques quadratiques de vecteur simplifié de l'ANF* $(0,\lambda,1,0,\ldots,0)$

	$n \equiv 0 \bmod 4$	$n \equiv 2 \bmod 4$	$n \equiv 1 \bmod 4$	$n \equiv 3 \bmod 4$
$\mathsf{F}_f(j)$, $j \equiv 0 \bmod 4$	$(-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	$(-1)^{\lambda}(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}((-1)^{\lambda}+1)$	$(-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}((-1)^{\lambda+1}+1)$
$\mathsf{F}_f(j)$, $j \equiv 1 \bmod 4$	$(-1)^{\lambda+1}(-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	$(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}((-1)^{\lambda+1}+1)$	$(-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}((-1)^{\lambda+1}-1)$
$\mathsf{F}_f(j)$, $j \equiv 2 \bmod 4$	$(-1)^{\frac{n}{4}+1} 2^{\frac{n}{2}}$	$(-1)^{\lambda+1}(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}((-1)^{\lambda+1}-1)$	$(-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}((-1)^{\lambda}-1)$
$\mathsf{F}_f(j)$, $j \equiv 3 \bmod 4$	$(-1)^{\lambda}(-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	$(-1)^{\frac{n-2}{4}+1} 2^{\frac{n}{2}}$	$(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}((-1)^{\lambda}-1)$	$(-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}((-1)^{\lambda}+1)$
$\mathsf{Ac}_f(j)$,$j \neq 0,n$	0	0	0	0
$\mathsf{Ac}_f(n)$	0	0	$(-1)^{\lambda} 2^n$	$(-1)^{\lambda+1} 2^n$

Ainsi, les calculs préliminaires 6.2.1 nous permettent d'obtenir le tableau 6.1.

Il nous faut maintenant déterminer le spectre d'auto-corrélation de f. Quand n est pair, f est courbe ce qui signifie que $\mathcal{F}(D_a f) = \mathsf{Ac}_f(\mathrm{wt}(a)) = 0$ pour tout $a \in \mathbf{F}_2^n$. Quand n est impair, $D_a f$ est de degré 1 (ce qui implique que la fonction est équilibrée) pour tout $a \neq \mathbf{1}$(Corollaire 5.52). En outre, nous savons que $\deg(D_{\mathbf{1}} f) < 1$ (proposition 5.58). Aussi $D_{\mathbf{1}} f$ est constante et sa valeur est donnée par la proposition 5.57 :

$$\lambda_{D_{\mathbf{1}} f}(0) = \bigoplus_{k \preceq n, k \neq 0} \lambda_f(k) = \begin{cases} \lambda & \text{if } n \equiv 1 \bmod 4 \\ \lambda \oplus 1 & \text{if } n \equiv 3 \bmod 4 \end{cases}.$$

□

6.2.3 Les fonctions symétriques cubiques

Nous déterminons maintenant, de manière similaire, les coefficients de Walsh et le spectre d'auto-corrélation des fonctions symétriques cubiques, quel que soit leur nombre de variables.

Proposition 6.2 *Soit $f \in \mathcal{S}ym_n$ une fonction de degré 3, $n \geq 3$, et de vecteur simplifié de l'ANF $\lambda(f) = (0,\lambda_1,\lambda_2,1,0,\ldots,0)$, λ_1, $\lambda_2 \in \mathbf{F}_2$. Alors son spectre de Walsh est donné par les tableaux 6.2, 6.3, 6.4, 6.5, suivant les valeurs de λ_1 et λ_2.*

De la même manière, si nous considérons la fonction cubique g de vecteur simplifié de l'ANF $(1,\lambda_1,\lambda_2,1,0,\ldots,0)$, i.e., *$g = f + 1$, nous déduisons immédiatement ses caractéristiques des tableaux 6.2, 6.3, 6.4, 6.5 par*

$$\mathcal{F}(g + \varphi_a) = -\mathcal{F}(f + \varphi_a),\ a \in \mathbf{F}_2^n,$$

c'est-à-dire $\mathsf{F}_g(j) = -\mathsf{F}_f(j),\ j = 0,\ldots,n.$

On remarque en particulier la valeur de la non-linéarité $2^{n-1} - \frac{\mathcal{L}(f)}{2}$:

$$\mathcal{L}(f) = \max\left(|\mathsf{F}_f(0)|, |\mathsf{F}_f(n)|\right) .$$

Preuve : Soit $f \in \mathcal{S}ym_n$ une fonction de degré 3 de vecteur simplifié de l'ANF $(0,\lambda_1,\lambda_2,1,0,\ldots,0)$. D'après le théorème 5.33, nous déduisons que le vecteur des valeurs simplifié $v(f)$ est périodique de période 4, troncature à n éléments de $(0,\lambda_1,\lambda_2,\lambda_1 \oplus \lambda_2 \oplus 1)^*$. Ainsi,

$$\mathsf{F}_f(j) = S_{K_{j,n}}(0,4) + (-1)^{\lambda_1} S_{K_{j,n}}(1,4) + (-1)^{\lambda_2} S_{K_{j,n}}(2,4) + (-1)^{\lambda_1+\lambda_2+1} S_{K_{j,n}}(3,4).$$

Nous déduisons les résultats des calculs préliminaires 6.2.1.

Lorsque $j = 0$, les valeurs de $\mathsf{F}_f(0)$ sont données par :

$$\mathsf{F}_f(0) = \left(1 + (-1)^{\lambda_1} + (-1)^{\lambda_2} + (-1)^{\lambda_1+\lambda_2+1}\right) 2^{n-2} + \begin{cases} (-1)^{\frac{n}{4}}(1 + (-1)^{\lambda_2+1}) 2^{\frac{n}{2}-1} & n \equiv 0 \bmod 4 \\ (-1)^{\frac{n-1}{4}}(1 + (-1)^{\lambda_1} + (-1)^{\lambda_2+1} + (-1)^{\lambda_1+\lambda_2}) 2^{\frac{n-3}{2}} & n \equiv 1 \bmod 4 \\ (-1)^{\frac{n-2}{4}}(-1)^{\lambda_1}(1 + (-1)^{\lambda_2}) 2^{\frac{n}{2}-1} & n \equiv 2 \bmod 4 \\ (-1)^{\frac{n+1}{4}}(1 + (-1)^{\lambda_1+1} + (-1)^{\lambda_2+1} + (-1)^{\lambda_1+\lambda_2+1}) 2^{\frac{n-3}{2}} & n \equiv 3 \bmod 4 \end{cases}.$$

TAB. 6.2 – *Caractéristiques de la fonction booléenne symétrique cubique de vecteur simplifié de l'ANF* (0,0,0,1,0...,0)

		$\lambda(f) = (0,0,0,1,0\ldots,0)$			
		$n \equiv 0 \bmod 4$	$n \equiv 2 \bmod 4$	$n \equiv 1 \bmod 4$	$n \equiv 3 \bmod 4$
$\mathsf{F}_f(0)$		2^{n-1}	$2^{n-1} + (-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$2^{n-1} + (-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$2^{n-1} - (-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}$
$\mathsf{F}_f(j)$	$j \equiv 0 \bmod 4$	0	$(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$(-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}$
$j \neq 0$	$j \equiv 2 \bmod 4$	0	$-(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$-(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$-(-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}$
$j \neq n$	$j \equiv 1 \bmod 4$	$(-1)^{\frac{n}{4}-1} 2^{\frac{n}{2}}$	0	$-(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$(-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}}$
	$j \equiv 3 \bmod 4$	$-(-1)^{\frac{n}{4}-1} 2^{\frac{n}{2}}$	0	$(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$-(-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}}$
$\mathsf{F}_f(n)$		2^{n-1}	$2^{n-1} + (-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$2^{n-1} - (-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$2^{n-1} + (-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}}$

TAB. 6.3 – *Caractéristiques de la fonction booléenne symétrique cubique de vecteur simplifié de l'ANF* (0,1,0,1,0...,0)

		$\lambda(f) = (0,1,0,1,0\ldots,0)$			
		$n \equiv 0 \bmod 4$	$n \equiv 2 \bmod 4$	$n \equiv 1 \bmod 4$	$n \equiv 3 \bmod 4$
$\mathsf{F}_f(0)$		2^{n-1}	$2^{n-1} - (-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$2^{n-1} - (-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$2^{n-1} + (-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}$
$\mathsf{F}_f(j)$	$j \equiv 0 \bmod 4$	0	$-(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$-(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$(-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}$
$j \neq 0$	$j \equiv 2 \bmod 4$	0	$(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$-(-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}$
$j \neq n$	$j \equiv 1 \bmod 4$	$-(-1)^{\frac{n}{4}-1} 2^{\frac{n}{2}}$	0	$(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$-(-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}}$
	$j \equiv 3 \bmod 4$	$(-1)^{\frac{n}{4}-1} 2^{\frac{n}{2}}$	0	$-(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$(-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}}$
$\mathsf{F}_f(n)$		2^{n-1}	$2^{n-1} - (-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$2^{n-1} + (-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$2^{n-1} - (-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}}$

TAB. 6.4 – *Caractéristiques de la fonction booléenne symétrique cubique de vecteur simplifié de l'ANF* (0,0,1,1,0 . . . ,0)

		$\lambda(f) = (0,0,1,1,0\ldots,0)$			
		$n \equiv 0 \bmod 4$	$n \equiv 2 \bmod 4$	$n \equiv 1 \bmod 4$	$n \equiv 3 \bmod 4$
$\mathsf{F}_f(0)$		$2^{n-1} + (-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	2^{n-1}	$2^{n-1} + (-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$2^{n-1} + (-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}$
$\mathsf{F}_f(j)$	$j \equiv 0 \bmod 4$	$(-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	0	$(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$(-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}$
$j \neq 0$	$j \equiv 2 \bmod 4$	$-(-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	0	$-(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$-(-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}$
$j \neq n$	$j \equiv 1 \bmod 4$	0	$(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$(-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}}$
	$j \equiv 3 \bmod 4$	0	$-(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$-(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$-(-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}}$
$\mathsf{F}_f(n)$		$-2^{n-1} + (-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	-2^{n-1}	$-2^{n-1} + (-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$-2^{n-1} + (-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}}$

TAB. 6.5 – *Caractéristiques de la fonction booléenne symétrique cubique de vecteur simplifié de l'ANF* (0,1,1,1,0...,0)

		$\lambda(f) = (0,1,1,1,0\ldots,0)$			
		$n \equiv 0 \bmod 4$	$n \equiv 2 \bmod 4$	$n \equiv 1 \bmod 4$	$n \equiv 3 \bmod 4$
$\mathsf{F}_f(0)$		$-2^{n-1} + (-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	-2^{n-1}	$-2^{n-1} + (-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$-2^{n-1} + (-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}$
$\mathsf{F}_f(j)$	$j \equiv 0 \bmod 4$	$(-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	0	$(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$(-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}$
$j \neq 0$	$j \equiv 2 \bmod 4$	$-(-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	0	$-(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$-(-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}}$
$j \neq n$	$j \equiv 1 \bmod 4$	0	$(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$(-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}}$
	$j \equiv 3 \bmod 4$	0	$-(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$-(-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$-(-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}}$
$\mathsf{F}_f(n)$		$2^{n-1} + (-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	2^{n-1}	$2^{n-1} + (-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}}$	$2^{n-1} + (-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}}$

Lorsque j est pair, les valeurs de $\mathsf{F}_f(j)$ pour $j \neq 0$ et $j \neq n$ sont données par :

$$\mathsf{F}_f(j) = \begin{cases} (-1)^{\frac{j}{2}}(-1)^{\frac{n}{4}}(1+(-1)^{\lambda_2+1})2^{\frac{n}{2}-1} & n \equiv 0 \bmod 4 \\ (-1)^{\frac{j}{2}}(-1)^{\frac{n-1}{4}}(1+(-1)^{\lambda_2+1}+(-1)^{\lambda_1}+(-1)^{\lambda_1+\lambda_2})2^{\frac{n-3}{2}} & n \equiv 1 \bmod 4 \\ (-1)^{\frac{j}{2}}(-1)^{\frac{n-2}{4}}(-1)^{\lambda_1}(1+(-1)^{\lambda_2})2^{\frac{n}{2}-1} & n \equiv 2 \bmod 4 \\ (-1)^{\frac{j}{2}}(-1)^{\frac{n+1}{4}}(1+(-1)^{\lambda_2+1}+(-1)^{\lambda_1+1}+(-1)^{\lambda_1+\lambda_2+1})2^{\frac{n-3}{2}} & n \equiv 3 \bmod 4 \end{cases}.$$

Lorsque j est impair, les valeurs de $\mathsf{F}_f(j)$ pour $j \neq n$ sont données par :

$$\mathsf{F}_f(j) = \begin{cases} (-1)^{\frac{j-1}{2}}(-1)^{\frac{n}{4}-1}(-1)^{\lambda_1}(1+(-1)^{\lambda_2})2^{\frac{n}{2}-1} & n \equiv 0 \bmod 4 \\ (-1)^{\frac{j-1}{2}}(-1)^{\frac{n-1}{4}}(1+(-1)^{\lambda_2+1}+(-1)^{\lambda_1+1}+(-1)^{\lambda_1+\lambda_2+1})2^{\frac{n-3}{2}} & n \equiv 1 \bmod 4 \\ (-1)^{\frac{j-1}{2}}(-1)^{\frac{n-2}{4}}(1+(-1)^{\lambda_2+1})2^{\frac{n}{2}-1} & n \equiv 2 \bmod 4 \\ (-1)^{\frac{j-1}{2}}(-1)^{\frac{n-3}{4}}(1+(-1)^{\lambda_2+1}+(-1)^{\lambda_1}+(-1)^{\lambda_1+\lambda_2})2^{\frac{n-3}{2}} & n \equiv 3 \bmod 4 \end{cases}.$$

Lorsque $j = n$, n pair, les valeurs de $\mathsf{F}_f(n)$ sont données par :

$$\mathsf{F}_f(n) = \begin{cases} 2^{n-2}\left(1+(-1)^{\lambda_2}+(-1)^{\lambda_1+1}+(-1)^{\lambda_1+\lambda_2}\right) & \\ \quad +(-1)^{\frac{n}{4}}2^{\frac{n}{2}-1}\left(1+(-1)^{\lambda_2+1}\right) & n \equiv 0 \bmod 4 \\ 2^{n-2}\left(1+(-1)^{\lambda_2}+(-1)^{\lambda_1+1}+(-1)^{\lambda_1+\lambda_2}\right) & \\ \quad -(-1)^{\frac{n-2}{4}}2^{\frac{n}{2}-1}\left((-1)^{\lambda_1}+(-1)^{\lambda_1+\lambda_2}\right) & n \equiv 2 \bmod 4 \end{cases}.$$

Lorsque $j = n$, n impair, les valeurs de $\mathsf{F}_f(n)$ sont données par :

$$\mathsf{F}_f(n) = \begin{cases} 2^{n-2}\left(1+(-1)^{\lambda_2}+(-1)^{\lambda_1+1}+(-1)^{\lambda_1+\lambda_2}\right) & \\ \quad +(-1)^{\frac{n-1}{4}}2^{\frac{n-3}{2}}\left(1+(-1)^{\lambda_2+1}+(-1)^{\lambda_1+1}+(-1)^{\lambda_1+\lambda_2+1}\right) & n \equiv 1 \bmod 4 \\ 2^{n-2}\left(1+(-1)^{\lambda_2}+(-1)^{\lambda_1+1}+(-1)^{\lambda_1+\lambda_2}\right) & \\ \quad -(-1)^{\frac{n-3}{4}}2^{\frac{n-3}{2}}\left(1+(-1)^{\lambda_1}+(-1)^{\lambda_2+1}+(-1)^{\lambda_1+\lambda_2}\right) & n \equiv 3 \bmod 4 \end{cases}.$$

□

L'étude portant sur les fonctions symétriques quadratiques nous permet de déterminer le spectre d'auto-corrélation des fonctions cubiques.

Proposition 6.3 *Soit $f \in \mathcal{S}ym_n$ de degré 3 et de vecteur simplifié de l'ANF*

$$\lambda(f) = (0,\lambda_1,\lambda_2,1,0,\ldots,0).$$

Alors son spectre d'auto-corrélation est donné par le tableau 6.6.

Preuve : D'après la proposition 5.49, nous pouvons nous limiter aux cas des dérivées relativement au vecteur $\varepsilon_k = e_{n-k+1} + \cdots + e_n$ pour $0 \leq k \leq n$. Nous utilisons les mêmes notations que dans la proposition 5.50. Soit $V = \langle e_1,\ldots,e_{n-k}\rangle$ et $\overline{V} = \langle e_{n-k+1},\ldots,e_n\rangle$. Nous savons alors que toutes les restrictions g_b, $b \in \overline{V}$ de $D_{\varepsilon_k}f$ aux sous-espaces affines $b+V$ sont des

TAB. 6.6 – *Spectre d'auto-corrélation de f, fonction symétrique cubique de vecteur simplifié de l'ANF $\lambda(f) = (0,\lambda_1,\lambda_2,1,0,\ldots,0)$*

$\mathsf{Ac}_f(j)$	$n \equiv 0 \bmod 4$	$n \equiv 2 \bmod 4$	$n \equiv 1 \bmod 4$	$n \equiv 3 \bmod 4$
$j \equiv 0 \bmod 4,\ j \neq 0,\ n$	2^{n-1}	2^{n-1}	2^{n-1}	2^{n-1}
$j \equiv 1 \bmod 4,\ j \neq n$	$(-1)^{\frac{n}{4}+\lambda_1} 2^{\frac{n}{2}} (1-(-1)^{\lambda_2})$	$\times(-1)^{\frac{n-2}{4}+\lambda_1} 2^{\frac{n}{2}} (1+(-1)^{\lambda_2})$	$(-1)^{\frac{n-1}{4}+\lambda_1} 2^{\frac{n+1}{2}}$	$(-1)^{\lambda_1+\lambda_2}(-1)^{\frac{n-3}{4}} 2^{\frac{n+1}{2}}$
$j \equiv 2 \bmod 4,\ j \neq n$	2^{n-1}	2^{n-1}	2^{n-1}	2^{n-1}
$j \equiv 3 \bmod 4,\ j \neq n$	$(-1)^{\frac{n}{4}+\lambda_1} 2^{\frac{n}{2}} (1-(-1)^{\lambda_2})$	$(-1)^{\frac{n-2}{4}+\lambda_1} 2^{\frac{n}{2}} (1+(-1)^{\lambda_2})$	$(-1)^{\frac{n-1}{4}+\lambda_1} 2^{\frac{n+1}{2}}$	$(-1)^{\lambda_1+\lambda_2}(-1)^{\frac{n-3}{4}} 2^{\frac{n+1}{2}}$
$j = n$	$(1-(-1)^{\lambda_2})2^{n-1}$	$(1+(-1)^{\lambda_2})2^{n-1}$	$(-1)^{\frac{n-1}{4}+\lambda_1} 2^{\frac{n+1}{2}}$	$(-1)^{\lambda_1+\lambda_2}(-1)^{\frac{n-3}{4}} 2^{\frac{n+1}{2}}$

fonctions symétriques et ne dépendent que de $\mathrm{wt}(b)$. Leur vecteur des valeurs simplifié est donné par : (cf. Proposition 5.50):

$$\lambda_{g_b}(i) = \bigoplus_{j \preceq k-\mathrm{wt}(b)} \lambda_f(i+j) \oplus \bigoplus_{j \preceq \mathrm{wt}(b)} \lambda_f(i+j).$$

Puisque pour tout (i,j) tels que $i+j>3$, $\lambda_f(i+j)=0$, k et $\mathrm{wt}(b)$ peuvent n'être considérés que modulo 4 : seuls leurs 2 derniers bits interviennent dans la formule précédente. On obtient alors la forme algébrique normale suivante pour g_b, résumée dans le tableau 6.7.

TAB. 6.7 – *Dérivées relativement à $\varepsilon_k = e_{n-k+1} + \cdots + e_n$ de la fonction symétrique cubique f de vecteur simplifié de l'ANF $\lambda(f) = (0,\lambda_1,\lambda_2,1,0,\ldots,0)$: ANF des restrictions de D_{ε_k} à $b + \langle e_1 + \cdots + e_{n-k}\rangle$, $b \in \langle e_{n-k+1} + \cdots + e_n\rangle$.*

	$wt(b) \pmod 4$			
$k \pmod 4$	0	1	2	3
0	$(0,0,0,\ldots)$	$(\lambda_2 \oplus 1,1,0,\ldots)$	$(0,0,0,\ldots)$	$(\lambda_2 \oplus 1,1,0,\ldots)$
1	$(\lambda_1,\lambda_2,1,0,\ldots)$	$(\lambda_1,\lambda_2,1,0,\ldots)$	$(\lambda_1 \oplus 1,\lambda_2,1,0,\ldots)$	$(\lambda_1 \oplus 1,\lambda_2,1,0,\ldots)$
2	$(\lambda_2,1,0,\ldots)$	$(0,0,0,\ldots)$	$(\lambda_2,1,0,\ldots)$	$(0,0,0,\ldots)$
3	$(\lambda_1 \oplus \lambda_2 \oplus 1,$ $\lambda_2 \oplus 1,1,0,\ldots)$	$(\lambda_1 \oplus \lambda_2,$ $\lambda_2 \oplus 1,1,0,\ldots)$	$(\lambda_1 \oplus \lambda_2,$ $\lambda_2 \oplus 1,1,0,\ldots)$	$(\lambda_1 \oplus \lambda_2 \oplus 1,$ $\lambda_2 \oplus 1,1,0,\ldots)$

On en déduit que pour $\varepsilon_k = e_{n-k+1} + \cdots + e_n$,

$$\begin{aligned}\mathcal{F}(D_{\varepsilon_k} f) &= \mathsf{Ac}_f(k) \\ &= \sum_{b \in \overline{V}} \mathcal{F}(g_b) \\ &= \sum_{\beta=0}^{k} \binom{k}{\beta} \mathcal{F}(g_{\varepsilon_\beta}) \\ &= \sum_{i=0}^{4} \mathcal{F}(g_{\varepsilon_i}) S_{B_n}(i,4).\end{aligned}$$

Lorsque k est pair, toutes les restrictions g_b sont des fonctions constantes ou linéaires. Nous pouvons donc en déduire que :

$$\mathcal{F}(D_{\varepsilon_k} f) = \begin{cases} 2^{n-k}(S_{B_n}(0,4) + S_{B_n}(2,4)) = 2^{n-1} & \text{si } k \equiv 0 \bmod 4 \\ 2^{n-k}(S_{B_n}(1,4) + S_{B_n}(3,4)) = 2^{n-1} & \text{si } k \equiv 2 \bmod 4 \end{cases}$$

Lorsque k est impair, on obtient :

– si $k \equiv 1 \bmod 4$

$$\begin{aligned}\mathcal{F}(D_{\varepsilon_k} f) &= \mathcal{F}(g_0)\,(S_{B_n}(0,4) + S_{B_n}(1,4) - S_{B_n}(2,4) - S_{B_n}(3,4)) \\ &= \mathcal{F}(g_0)(-1)^{\frac{k-1}{4}} 2^{\frac{k+1}{2}},\end{aligned}$$

– si $k \equiv 3 \bmod 4$

$$\begin{aligned}\mathcal{F}(D_{\varepsilon_k} f) &= \mathcal{F}(g_0)\left(S_{B_n}(0,4) - S_{B_n}(1,4) - S_{B_n}(2,4) + S_{B_n}(3,4)\right) \\ &= \mathcal{F}(g_0)(-1)^{\frac{k+1}{4}} 2^{\frac{k+1}{2}}.\end{aligned}$$

La valeur de $\mathcal{F}(g_0)$ peut se déduire de la proposition 6.1 car g_0 est une fonction symétrique quadratique à $(n-k)$ variables.

Enfin, la proposition 5.57 nous permet de calculer le poids de $D_{\mathbf{1}} f$: pour tout $0 \leq i \leq 2$:

$$\lambda_{D_{\mathbf{1}} f}(i) = \bigoplus_{\substack{k \preceq n-i \\ k \neq 0}} \lambda_f(k+i).$$

Nous obtenons alors les valeurs suivantes pour le vecteur simplifié de l'ANF de $D_{\mathbf{1}} f$:

$$\lambda(D_{\mathbf{1}} f) = \begin{cases} (0,\lambda_2 \oplus 1,0,\ldots,0) & \text{si } n \equiv 0 \bmod 4 \\ (\lambda_2,\lambda_2,0,\ldots,0) & \text{si } n \equiv 2 \bmod 4 \\ (\lambda_1,0,1,0,\ldots,0) & \text{si } n \equiv 1 \bmod 4 \\ (\lambda_1 \oplus \lambda_2 \oplus 1,1,1,0,\ldots,0) & \text{si } n \equiv 3 \bmod 4 \end{cases}.$$

Les poids correspondants sont obtenus grâce à la proposition 6.1. □

6.3 Poids des fonctions symétriques de degré inférieur à 8

Nous étudions dans cette partie les fonctions symétriques de degré au plus 7. Nous voulons principalement déterminer quelles sont les fonctions symétriques équilibrées de degré au plus 7 pour tout nombre de variables. Lorsque $\deg(f) \leq 7$, le vecteur des valeurs simplifié de f est la troncature de $(v_0,v_1,\ldots,v_7)^*$ avec $(v_0,\ldots,v_7) \in \mathbf{F}_2^8$. De ce fait,

$$\mathcal{F}(f) = \sum_{i=0}^{7} (-1)^{v_i} S_{B_n}(i,4),$$

où $S_{B_n}(i,4)$ est donné par la formule (6.1). Ainsi,

$$\mathcal{F}(f) = 2^{n-3} \sum_{i=0}^{7} (-1)^{v_i} + \frac{1}{4} \sum_{j=1}^{3} \left(2\cos\left(j\frac{\pi}{8}\right)\right)^n \sum_{i=0}^{7} (-1)^{v_i} \cos\left(j(n-2i)\frac{\pi}{8}\right).$$

Nous savons que

$$\cos\left(\frac{\pi}{8}\right) = \sqrt{\frac{1}{2} + \frac{1}{2\sqrt{2}}} \quad \text{et} \quad \cos\left(\frac{3\pi}{8}\right) = \sqrt{\frac{1}{2} - \frac{1}{2\sqrt{2}}}.$$

Tout d'abord nous étudions le cas des fonctions symétriques de degré au plus 7 qui dé-

pendent d'un nombre pair de variables $n = 2t$. Nous obtenons alors

$$\mathcal{F}(f) = 2^{2t-3}\sum_{i=0}^{7}(-1)^{v_i} + 2^{t-2}\sum_{i=0}^{7}(-1)^{v_i}\cos\left((t-i)\frac{\pi}{2}\right) + \frac{1}{4}\left(2+\sqrt{2}\right)^t\sum_{i=0}^{7}(-1)^{v_i}\cos\left((t-i)\frac{\pi}{4}\right) + \frac{1}{4}\left(2-\sqrt{2}\right)^t\sum_{i=0}^{7}(-1)^{v_i}\cos\left((t-i)\frac{3\pi}{4}\right).$$

Comme les angles doivent être considérées modulo 2π, les valeurs des cosinus respectifs ne dépendent que de $(t-i) \bmod 8$. Nous pouvons combiner cette propriété avec la périodicité de $v(f)$, de période 8 ce qui nous permet d'écrire :

$$\mathcal{F}(f) = 2^{2t-3}\sum_{i=0}^{7}(-1)^{v_i} + 2^{t-2}\sum_{i=0}^{7}(-1)^{v_{t-i}}\cos\left(i\frac{\pi}{2}\right) + \frac{1}{4}\left(2+\sqrt{2}\right)^t\sum_{i=0}^{7}(-1)^{v_{t-i}}\cos\left(i\frac{\pi}{4}\right) + \frac{1}{4}\left(2-\sqrt{2}\right)^t\sum_{i=0}^{7}(-1)^{v_{t-i}}\cos\left(i\frac{3\pi}{4}\right)$$

où les indices de v_{t-i} doivent être considérés modulo 8. Nous obtenons alors :

$$\begin{aligned}\mathcal{F}(f) = {} & 2^{2t-3}\sum_{i=0}^{7}(-1)^{v_i} + 2^{t-2}\left((-1)^{v_t} - (-1)^{v_{t-2}} + (-1)^{v_{t-4}} - (-1)^{v_{t-6}}\right) \\ & + \frac{1}{4\sqrt{2}}\left(\left(2+\sqrt{2}\right)^t - \left(2-\sqrt{2}\right)^t\right)\left((-1)^{v_{t-1}} - (-1)^{v_{t-3}} - (-1)^{v_{t-5}} + (-1)^{v_{t-7}}\right) \\ & + \frac{1}{4}\left(\left(2+\sqrt{2}\right)^t + \left(2-\sqrt{2}\right)^t\right)\left((-1)^{v_t} - (-1)^{v_{t-4}}\right)\end{aligned}$$

Dans ce qui suit, nous utiliserons les notations, $\forall k \geq 0$:

$$D_k^+ = \frac{1}{2}\left((2+\sqrt{2})^k + (2-\sqrt{2})^k\right), \qquad D_k^- = \frac{1}{2\sqrt{2}}\left((2+\sqrt{2})^k - (2-\sqrt{2})^k\right), \tag{6.12}$$

ainsi que

$$\forall (a,b,c,d) \in \mathbf{F}_2^4, \quad \begin{cases} A(a,b,c,d) & = (-1)^a + (-1)^b + (-1)^c + (-1)^d \\ B(a,b,c,d) & = (-1)^a + (-1)^b + (-1)^{c+1} + (-1)^{d+1} \end{cases}. \tag{6.13}$$

L'expression de $\mathcal{F}(f)$ devient alors :

$$\begin{aligned}\mathcal{F}(f) = {} & 2^{2t-3}\left(A(v_t,v_{t-4},v_{t-2},v_{t-6}) + A(v_{t-1},v_{t-7},v_{t-3},v_{t-5})\right) + 2^{t-2}B(v_t,v_{t-4},v_{t-2},v_{t-6}) \\ & + \frac{1}{2}D_t^-\,B(v_{t-1},v_{t-7},v_{t-3},v_{t-5}) + \frac{1}{2}D_t^+\left((-1)^{v_t} - (-1)^{v_{t-4}}\right).\end{aligned} \tag{6.14}$$

Nous cherchons maintenant à déterminer quelles sont les fonctions booléennes symétriques équilibrées à $2t$ variables de degré au plus 7. Pour ce faire, nous utilisons le lemme suivant :

Lemme 6.4 *On a pour tout* $n \geq 3$,

$$2^{n-1+\lceil \frac{n}{2} \rceil} < D_n^+$$

et pour tout $n \geq 14$,

$$2^{n-1+\lfloor \frac{n}{2} \rfloor} < D_n^- < 2^{2n-5} + 2^{2n-6}.$$

Preuve : Par définition, on a :

$$\begin{aligned} D_n^+ &= \frac{1}{2}\left((2+\sqrt{2})^n + (2-\sqrt{2})^n\right) \\ &= \sum_{\substack{0 \leq i \leq n \\ i \text{ pair}}} \binom{n}{i} 2^{n-\frac{i}{2}}. \end{aligned}$$

Ce qui signifie que

$$D_n^+ > 2^{n-\lfloor \frac{n}{2} \rfloor} \sum_{\substack{0 \leq i \leq n \\ i \text{ pair}}} \binom{n}{i} \geq 2^{n-1} \cdot 2^{\lceil \frac{n}{2} \rceil}, \quad \text{pour tout } n \geq 2.$$

Un calcul similaire pour D_n^- nous donne

$$\begin{aligned} D_n^- &= \frac{1}{2\sqrt{2}}\left((2+\sqrt{2})^n - (2-\sqrt{2})^n\right) \\ &= \sum_{\substack{0 \leq i \leq n \\ i \text{ impair}}} \binom{n}{i} 2^{n-\frac{i+1}{2}}. \end{aligned}$$

Ainsi,

$$D_n^- > 2^{n-\lceil \frac{n}{2} \rceil} \sum_{\substack{0 \leq i \leq n \\ i \text{ impair}}} \binom{n}{i} \geq 2^{n-1} \cdot 2^{\lfloor \frac{n}{2} \rfloor}, \quad \text{pour tout } n \geq 3.$$

En outre,

$$\begin{aligned} D_n^- &= n \cdot 2^{n-1} + \binom{n}{3} 2^{n-2} + \binom{n}{5} 2^{n-3} + \sum_{\substack{7 \leq i \leq n \\ i \text{ impair}}} \binom{n}{i} 2^{n-\frac{i+1}{2}} \\ &\leq n \cdot 2^{n-1} + \binom{n}{3} 2^{n-2} + \binom{n}{5} 2^{n-3} + 2^{n-4}\left(2^{n-1} - n - \binom{n}{3} - \binom{n}{5}\right) \\ &\leq 2^{2n-5} + 2^{n-4}\left(7n + 3\binom{n}{3} + \binom{n}{5}\right). \end{aligned}$$

Comme pour tout $n \geq 14$ nous avons $7n + 3\binom{n}{3} + \binom{n}{5} < 2^{n-2}$, nous obtenons :

$$D_n^- < 2^{2n-5} + 2^{2n-6} \quad \text{pour tout } n \geq 14.$$

□

Théorème 6.5 *Pour tout n pair, $n \geq 2$, il n'existe pas de fonction booléenne symétrique équilibrée à n variables de degré inférieur ou égal à 7 à l'exception des fonctions de degré 1 et des fonctions à 8 variables dont le vecteur simplifié de l'ANF est donné par :*

$$\lambda(f) = (\varepsilon,1,1,0,0,0,0,1,0) \qquad et \qquad \lambda(f) = (\varepsilon,1,1,1,0,1,0,1,0).$$

Preuve : Pour toute fonction booléenne symétrique f à $n = 2t$ variables et telle que $\deg(f) \leq 7$, nous pouvons écrire grâce à (6.14):

$$\mathcal{F}(f) = 2^{2t-3}(A_1 + A_2) + 2^{t-2}B_1 + \frac{1}{2}D_t^- B_2 + \frac{1}{2}D_t^+ \left((-1)^{v_t} + (-1)^{v_{t-4}+1}\right),$$

où

$$\begin{cases} A_1 = A(v_t,v_{t-4},v_{t-2},v_{t-6}) \\ B_1 = B(v_t,v_{t-4},v_{t-2},v_{t-6}) \end{cases}, \quad \text{et} \quad \begin{cases} A_2 = A(v_{t-1},v_{t-7},v_{t-3},v_{t-5}) \\ B_2 = B(v_{t-1},v_{t-7},v_{t-3},v_{t-5}) \end{cases}.$$

Il nous faut distinguer plusieurs cas :

- si $A_1 + A_2 \neq 0$, alors $|A_1 + A_2| \geq 2$. Il s'ensuit que

  soit $\quad \mathcal{F}(f) \geq 2^{2t-2} - 2^t - D_t^+ - 2D_t^- \quad$ soit $\quad \mathcal{F}(f) \leq -2^{2t-2} + 2^t + D_t^+ + 2D_t^-$.

  En prenant en compte le fait que pour tout $k > 0$, $D_k^+ + 2D_k^- = D_{k+1}^-$, il en découle que

  $$|\mathcal{F}(f)| \geq 2^{2t-2} - 2^t - D_{t+1}^- > 0 \text{ pour tout } t \geq 14,$$

  la dernière inégalité étant déduite du lemme 6.4;
- si $A_1 + A_2 = 0$ et $B_1 \neq 0$, alors

  $$|\mathcal{F}(f)| \geq \frac{1}{2}\left|D_t^+ \left((-1)^{v_t} + (-1)^{v_{t-4}+1}\right) + D_t^- B_2\right| - 2^t.$$

  Il s'ensuit que lorsque $v_t = v_{t-4}$ et que $B_2 \neq 0$,

  $$\mathcal{F}(f) \geq D_t^- - 2^t > 0.$$

  Lorsque $v_t = v_{t-4}$ et $B_2 = 0$, on obtient de manière immédiate :

  $$|\mathcal{F}(f)| = 2^{t-2}|B_1| > 0.$$

  Lorsque $v_t \neq v_{t-4}$, on a

  $$\left|D_t^+ + \frac{1}{2}D_t^- B_2\right| \geq \left|D_t^+ - 2D_t^-\right| = 2D_{t-1}^- > 2^t \ ;$$

- si $A_1 + A_2 = 0$ et $B_1 = 0$, alors

  $$\mathcal{F}(f) = \frac{1}{2}D_t^+ \left((-1)^{v_t} + (-1)^{v_{t-4}+1}\right) + \frac{1}{2}D_t^- B_2.$$

  Si $v_t \neq v_{t-4}$, alors

  $$|\mathcal{F}(f)| \geq \left|D_t^+ - 2D_t^-\right| = 2D_{t-1}^- > 0.$$

  Si $v_t = v_{t-4}$, on obtient alors $\mathcal{F}(f) = \frac{1}{2}D_t^- B_2$. Or, la configuration $v_t = v_{t-4}$, $B_1 = B_2 = 0$ et $A_1 + A_2 = 0$ apparaît si et seulement si $v_i = \varepsilon$ lorsque i est pair et $v_i = \varepsilon \oplus 1$ lorsque i est impair. Cela signifie que $v(f)$ est la troncature de $(0,1)^*$ ou de $(1,0)^*$, *i.e.*, que f est de degré 1.

Ainsi nous avons démontré que pour tout n entier tel que $n \geq 28$, une fonction $f \in \mathcal{S}ym_n$ telle que $\deg(f) \leq 7$ est équilibrée si et seulement si $\deg(f) = 1$. Le calcul de toutes les valeurs possibles de $\mathcal{F}(f)$ pour les fonctions booléennes symétriques à n variables, n pair et inférieur à 28 permet de vérifier que les seules fonctions symétriques équilibrées sont celles de degré 1 et les fonctions à 8 variables définies par les vecteurs simplifiés de l'ANF suivant :

$$\lambda(f) = (\varepsilon,1,1,0,0,0,0,1,0) \qquad \text{et} \qquad \lambda(f) = (\varepsilon,1,1,1,0,1,0,1,0).$$

□

Nous nous attachons maintenant à l'étude des fonctions symétriques de degré inférieur à 7 qui dépendent d'un nombre impair de variables, $n = 2t + 1$. L'expression de $\mathcal{F}(f)$ devient alors :

$$\begin{aligned}
\mathcal{F}(f) &= 2^{2t-2}\sum_{i=0}^{7}(-1)^{v_i} + 2^{t-2}\sqrt{2}\sum_{i=0}^{7}(-1)^{v_i}\cos\left((t-i)\frac{\pi}{2}+\frac{\pi}{4}\right)\\
&\quad + \frac{1}{4}\left(2+\sqrt{2}\right)^{t+\frac{1}{2}}\sum_{i=0}^{7}(-1)^{v_i}\cos\left((t-i)\frac{\pi}{4}+\frac{\pi}{8}\right)\\
&\quad + \frac{1}{4}\left(2-\sqrt{2}\right)^{t+\frac{1}{2}}\sum_{i=0}^{7}(-1)^{v_i}\cos\left((t-i)\frac{3\pi}{4}+\frac{3\pi}{8}\right)\\
&= 2^{2t-2}\sum_{i=0}^{7}(-1)^{v_i} + 2^{t-2}\sqrt{2}\sum_{i=0}^{7}(-1)^{v_{t-i}}\cos\left(i\frac{\pi}{2}+\frac{\pi}{4}\right)\\
&\quad + \frac{1}{4}\left(2+\sqrt{2}\right)^{t+\frac{1}{2}}\sum_{i=0}^{7}(-1)^{v_{t-i}}\cos\left(i\frac{\pi}{4}+\frac{\pi}{8}\right)\\
&\quad + \frac{1}{4}\left(2-\sqrt{2}\right)^{t+\frac{1}{2}}\sum_{i=0}^{7}(-1)^{v_{t-i}}\cos\left(i\frac{3\pi}{4}+\frac{3\pi}{8}\right),
\end{aligned}$$

la dernière égalité étant obtenu grâce au fait que l'on peut considérer $(t-i)$ modulo 8 et que le vecteur des valeurs simplifié de f est périodique de période 8. En développant la somme précédente on obtient :

$$\begin{aligned}
\mathcal{F}(f) = 2^{2t-2}\sum_{i=0}^{7}(-1)^{v_i} &+ 2^{t-2}\left((-1)^{v_t} - (-1)^{v_{t-1}} - (-1)^{v_{t-2}} + (-1)^{v_{t-3}}\right.\\
&\left.+(-1)^{v_{t-4}} - (-1)^{v_{t-5}} - (-1)^{v_{t-6}} + (-1)^{v_{t-7}}\right)\\
+\frac{1}{4}\left(\left(2+\sqrt{2}\right)^{t+\frac{1}{2}}\cos\frac{\pi}{8} + \left(2-\sqrt{2}\right)^{t+\frac{1}{2}}\cos\frac{3\pi}{8}\right)&\left((-1)^{v_t} - (-1)^{v_{t-3}} - (-1)^{v_{t-4}} + (-1)^{v_{t-7}}\right)\\
+\frac{1}{4}\left(\left(2+\sqrt{2}\right)^{t+\frac{1}{2}}\cos\frac{3\pi}{8} - \left(2-\sqrt{2}\right)^{t+\frac{1}{2}}\cos\frac{\pi}{8}\right)&\left((-1)^{v_{t-1}} - (-1)^{v_{t-2}} - (-1)^{v_{t-5}} + (-1)^{v_{t-6}}\right)
\end{aligned}$$

En utilisant les notations définies par (6.12), on obtient :

$$\begin{aligned}
\left(2+\sqrt{2}\right)^{t+\frac{1}{2}}\cos\frac{\pi}{8} + \left(2-\sqrt{2}\right)^{t+\frac{1}{2}}\cos\frac{3\pi}{8} &= \frac{1}{2}\left(\left(2+\sqrt{2}\right)^{t+1} + \left(2-\sqrt{2}\right)^{t+1}\right)\\
&= D^{+}_{t+1}\\
&= 2D^{+}_{t} + 2D^{-}_{t}
\end{aligned}$$

et

$$\left(2+\sqrt{2}\right)^{t+\frac{1}{2}}\cos\frac{3\pi}{8}-\left(2-\sqrt{2}\right)^{t+\frac{1}{2}}\cos\frac{\pi}{8}=\frac{1}{\sqrt{2}}\left(\left(2+\sqrt{2}\right)^{t}-\left(2-\sqrt{2}\right)^{t}\right)$$
$$=2D_t^-.$$

Les notations (6.13), nous permettent d'écrire :

$$\begin{aligned}\mathcal{F}(f)=&\,2^{2t-2}\left(A(v_t,v_{t-7},v_{t-3},v_{t-4})+A(v_{t-1},v_{t-6},v_{t-2},v_{t-5})\right)\\&+2^{t-2}\left(A(v_t,v_{t-7},v_{t-3},v_{t-4})-A(v_{t-1},v_{t-6},v_{t-2},v_{t-5})\right)\\&+\frac{1}{2}D_t^+B(v_t,v_{t-7},v_{t-3},v_{t-4})+\frac{1}{2}D_t^-\left(B(v_t,v_{t-7},v_{t-3},v_{t-4})+B(v_{t-1},v_{t-6},v_{t-2},v_{t-5})\right).\end{aligned}$$

Nous cherchons à déterminer les fonctions pour lesquelles cette quantité s'annule.

Théorème 6.6 *Pour tout n impair tel que $n\geq 3$, les seules fonctions booléennes symétriques équilibrées à n variables de degré inférieur ou égal à 7 sont des fonctions équilibrées triviales.*

Preuve : Pour tout $f\in\mathcal{S}ym_n$ avec n impair, $n=2t+1$ et $\deg(f)\leq 7$, on a

$$\mathcal{F}(f)=2^{2t-2}(A_1+A_2)+2^{t-2}(A_1-A_2)+\frac{1}{2}D_t^+B_1+\frac{1}{2}D_t^-(B_1+B_2),$$

où

$$\begin{cases}A_1=A(v_t,v_{t-7},v_{t-3},v_{t-4})\\B_1=B(v_t,v_{t-7},v_{t-3},v_{t-4})\end{cases}\quad\text{et}\quad\begin{cases}A_2=A(v_{t-1},v_{t-6},v_{t-2},v_{t-5})\\B_2=B(v_{t-1},v_{t-6},v_{t-2},v_{t-5})\end{cases}.$$

Il nous faut distinguer plusieurs cas :

- si $A_1+A_2\neq 0$, alors $|A_1+A_2|\geq 2$. Il s'ensuit que

  $$\text{soit}\quad\mathcal{F}(f)\geq 2^{2t-1}-2^{t+1}-2D_t^+-4D_t^-\quad\text{soit}\quad\mathcal{F}(f)\leq -2^{2t-1}+2^{t+1}+2D_t^++4D_t^-$$

  En prenant en compte le fait que pour tout $k>0$, $D_k^++2D_k^-=D_{k+1}^-$, on en déduit que

  $$|\mathcal{F}(f)|\geq 2^{2t-1}-2^{t+1}-2D_{t+1}^->0\quad\text{pour tout } t\geq 13,$$

  la dernière inégalité étant déduite du lemme 6.4;
- si $A_1+A_2=0$ et $A_1\neq 0$, on a

  $$|\mathcal{F}(f)|\geq\left|\frac{1}{2}D_t^+B_1+\frac{1}{2}D_t^-(B_1+B_2)\right|-2^t.$$

  D'après le lemme 6.4, il s'ensuit que lorsque $B_1=0$ et $B_2\neq 0$,

  $$|\mathcal{F}(f)|\geq D_t^--2^t>0\quad\text{pour tout } t\geq 4.$$

  Lorsque $B_1=0$ et $B_2=0$, on obtient de manière immédiate

  $$|\mathcal{F}(f)|\geq 2^t>0.$$

  Lorsque $B_1\neq 0$,

  $$\begin{aligned}|\mathcal{F}(f)|&\geq(D_t^++D_t^-)-2D_t^--2^t\\&\geq D_{t-1}^+-2^t>0\quad\text{pour tout } t\geq 3\ ;\end{aligned}$$

– si $A_1 = A_2 = 0$, on a :

$$|\mathcal{F}(f)| = \left|\frac{1}{2}B_1(D_t^+ + D_t^-) + \frac{1}{2}B_2 D_t^-\right|.$$

Si $B_1 \neq 0$, on en déduit que

$$|\mathcal{F}(f)| \geq (D_t^+ + D_t^-) - 2D_t^- = D_{t-1}^+ > 0.$$

Si $B_1 = 0$, on obtient

$$|\mathcal{F}(f)| = \frac{1}{2}|B_2| D_t^-$$

qui s'annule si et seulement si $B_2 = 0$.

Nous avons ainsi démontré que pour tout n impair tel que $n \geq 27$, une fonction $f \in \mathcal{S}ym_n$ telle que $\deg(f) \leq 7$ est équilibrée si et seulement si $A_1 = A_2 = B_1 = B_2 = 0$. Cette situation se produit si et seulement si

$$\begin{cases} v_t \oplus v_{t-7} = 1 \\ v_{t-3} \oplus v_{t-4} = 1 \\ v_{t-1} \oplus v_{t-6} = 1 \\ v_{t-2} \oplus v_{t-5} = 1 \end{cases} \text{, avec} \quad t = \frac{n-1}{2}.$$

Cette condition revient exactement à dire que $v_i \oplus v_{2t+1-i} = 1$ pour tout $0 \leq i \leq 2t+1$. Le calcul de toutes les valeurs possibles de $\mathcal{F}(f)$ pour les fonctions symétriques à n variables avec n impair et $3 \leq n < 27$ nous permet de vérifier que les seules fonctions booléennes symétriques équilibrées dans ce cas sont équilibrées triviales. □

Les deux théorèmes précédents qui donnent toutes les fonctions booléennes équilibrées de degré au plus 7, nous permettent de déterminer toutes les fonctions symétriques de degré au plus 8 qui soit vérifient $PC(1)$ soit sont 1-résilientes. Les fonctions qui vérifient $PC(1)$ s'obtiennent en combinant les théorèmes 6.5 et 6.6, ainsi que les propositions 5.51 et 5.55.

Corollaire 6.7 *Soit $f \in \mathcal{S}ym_n$, $n \geq 3$, telle que* $\deg(f) \leq 8$. *Alors f vérifie $PC(1)$ si et seulement si elle vérifie une des conditions suivantes :*

- $\deg(f) = 2$;
- $D_{\mathbf{1}}f$ *est de degré* 1;
- f *est une fonction à* 9 *variables de degré* 8 *définie par un des* 8 *vecteurs simplifiés de l'ANF suivant :*

$$\lambda(f) = (\varepsilon_0,\varepsilon_1,1,1,0,0,0,0,1,0) \quad ou \quad \lambda(f) = (\varepsilon_0,\varepsilon_1,1,1,1,0,1,0,1,0) \quad avec \quad \varepsilon_0,\varepsilon_1 \in \mathbf{F}_2.$$

Corollaire 6.8 *Soit $f \in \mathcal{S}ym_n$, $n \geq 3$ telle que* $\deg(f) \leq 7$. *Alors f est* 1*-résiliente si et seulement si elle est de degré* 1.

Preuve : Par définition, f est 1-résiliente si et seulement si elle est équilibrée et que les deux fonctions symétriques à $(n-1)$ variables f_H et f_{e_n+H} qui correspondent aux restrictions de f à $H = \langle e_1,\ldots,e_{n-1}\rangle$ et à $e_n + H$ sont équilibrées.

Si n est pair, f_H et f_{e_n+H} sont équilibrées si et seulement si elles sont équilibrées triviales (Th. 6.6). Dans ce cas, la proposition 5.42 implique que f est de degré 1.

Si n est impair et $\deg(f) \neq 1$, alors soit f_H soit f_{e_n+H} est une des 8 fonctions à 8 variables de degré 7 définies par le théorème 6.5. D'après le corollaire 5.40 il s'ensuit que f est une fonction symétrique à 9 variables de degré 7. Elle ne peut pas être équilibrée triviale car $\deg(D_{\mathbf{1}}f) = \deg(f) - 1 = 6$ (Prop 5.58). Ce qui conduit à une contradiction d'après le théorème 6.6. □

Chapitre 7

Non-linéarité des fonctions symétriques

Je m'attache dans ce chapitre à caractériser les fonctions symétriques de non-linéarité élevée, plus précisément celles dont la non-linéarité est supérieure à $2^{n-1} - 2^{\lfloor \frac{n-1}{2} \rfloor} - 2^t$, avec $0 \leq t < \lfloor \frac{n-1}{2} \rfloor$. En effet, nous savons que les fonctions symétriques sont de non-linéarité maximale si et seulement si elles sont quadratiques. Leur non-linéarité vaut alors $2^{n-1} - 2^{\frac{n}{2}-1}$ lorsque n est pair [Sav94], dans ce cas les fonctions sont *courbes* et $2^{n-1} - 2^{\frac{n-1}{2}}$ lorsque n est impair [MS02]. Les fonctions de non-linéarité maximale sont ainsi des fonctions de petit degré algébrique ce qui les rend impropres pour un système de chiffrement. Néanmoins, des fonctions alliant une non-linéarité légèrement plus faible et un degré algébrique élevé pourraient convenir à cet usage et il est donc important de déterminer si de telles fonctions existent et de les caractériser. J'étudie dans ce chapitre les cas de non-linéarité sous-optimale, en particulier, je relie cette propriété à la périodicité du vecteur des valeurs simplifié. Je mets notamment en évidence une famille infinie de fonctions symétriques équilibrées de degré algébrique élevé et de non-linéarité presque optimale.

7.1 Fonctions symétriques de non-linéarité maximale

Nous rappelons dans cette partie les résultats et leur démonstration concernant les fonctions de non-linéarité maximale. Pour une fonction booléenne f à n variables, nous utiliserons fréquemment la notation $\mathcal{L}(f)$ que nous rappelons :

$$\begin{aligned}\mathcal{L}(f) &= \max_{a \in \mathbf{F}_2^n} | \mathcal{F}(f + \varphi_a) | \\ &= \max_{a \in \mathbf{F}_2^n} | 2^n - 2\,\mathrm{wt}(f + \varphi_a) |\end{aligned}$$

Proposition 7.1 *[Sav94] Soit f une fonction booléenne symétrique à n variables, n pair, alors les propriétés suivantes sont équivalentes :*

- **(i)** *La fonction f est courbe ;*
- **(ii)** *Pour tout $k \in \{0, \ldots, n-2\}$ on a $v_f(k+2) = v_f(k) \oplus 1$;*
- **(ii)** *La fonction f est quadratique.*

Nous avons déjà démontré cette proposition en étudiant le critère de propagation des fonctions symétriques. Nous avons en particulier utilisé l'équivalence : f est une fonction courbe si et seulement si elle vérifie $PC(n)$.

Nous aurons besoin dans la suite du chapitre du résultat intermédiaire suivant.

Lemme 7.2 *Soit $f \in \mathcal{S}ym_n$ et le sous-espace vectoriel $V = \langle e_1, \dots e_{n-2} \rangle$. Considérons la fonction f_{e_n+V} restriction de f au sous-espace affine $e_n + V$. Alors $\mathcal{L}(f) \geq 2\mathcal{L}(f_{e_n+V})$.*

Preuve : Nous allons utiliser des propriétés de décomposition de fonctions booléennes. Soit $f_{\overline{H}}$ la restriction de f au sous-espace affine $\overline{H} = \{x \in \mathbf{F}_2^n, x_{n-1} + x_n = 1\}$. Le corollaire V.3 de [CCCF01] nous permet de dire que $\mathcal{L}(f) \geq \mathcal{L}(f_{\overline{H}})$. En outre, en utilisant les notations de la proposition 5.39 page 131, $f_{\overline{H}}$ peut s'exprimer sous la forme

$$f_{\overline{H}}(x_1, \dots, x_{n-1}) = (1 + x_{n-1}) f_{e_n+V}(x_1, \dots, x_{n-2}) + x_{n-1} f_{e_{n-1}+V}(x_1, \dots, x_{n-2})$$

où $V = \langle e_1, \dots e_{n-2} \rangle$. Or d'après la proposition 5.39 page 131 nous savons que $f_{e_n+V} = f_{e_{n-1}+V}$ et que cette fonction est une fonction symétrique à $(n-2)$ variables. Soit $g = f_{e_n+V} = f_{e_{n-1}+V}$, nous avons alors $f_{\overline{H}}(x_1, \dots, x_{n-1}) = g(x_1, \dots, x_{n-2})$ ce qui implique que $\mathcal{L}(f_{\overline{H}}) = 2\mathcal{L}(g)$. □

Ce lemme nous permet alors de donner une nouvelle preuve du résultat qui fait l'objet de l'article [MS02] de Maitra et Sarkar . Notons qu'une démonstration alternative reposant sur la notion de normalité est donnée par Claude Carlet dans [Car04].

Proposition 7.3 *[MS02, Th. 5] Soient $n \geq 3$ un entier impair et $f \in \mathcal{S}ym_n$ alors les propriétés suivantes sont équivalentes :*

- **(o)** *la non linéarité de f est supérieure ou égale à $2^{n-1} - 2^{\frac{n-1}{2}}$;*
- **(i)** *la non-linéarité de f vaut $2^{n-1} - 2^{\frac{n-1}{2}}$;*
- **(ii)** *$v(f)$ est un vecteur constitué de $(n+1)$ éléments contigus de $(0011)^*$;*
- **(iii)** *le degré de f est 2 ;*
- **(iv)** *l'ensemble des valeurs du spectre de Walsh de f est $\{0, \pm 2^{\frac{n+1}{2}}\}$.*

Preuve : Le théorème 5.33 page 125 portant sur la périodicité du vecteur des valeurs simplifié d'une fonction booléenne symétrique nous donne l'équivalence entre **(ii)** et **(iii)**. L'assertion **(iii)** implique **(iv)** de manière immédiate d'après la proposition 6.1 page 153.

On a par ailleurs que **(iv)** implique **(i)** et que **(i)** implique **(o)** de manière évidente.

Il nous reste donc à montrer que **(o)** implique **(iii)**. Nous le démontrons par récurrence sur n. Pour $n = 3$, on constate que l'hypothèse est vérifiée. Supposons maintenant qu'elle

n	ANF	valeurs	spectre de Walsh	spectre d'auto-corrélation	équ.	NL	rés.	propagation
3	0100	0101	0, 0, 0, 8	8, −8, 8, −8	×	0	2	0
3	0010	0011	0, 4, 0, −4	8, 0, 0, −8	×	2	0	2
3	0110	0110	−4, 0, 4, 0	8, 0, 0, 8		2	0	2
3	0001	0001	6, 2, −2, 2	8, 4, 4, 4		1	0	0
3	0101	0100	2, −2, 2, 6	8, −4, 4, −4		1	0	0
3	0011	0010	2, 2, 2, −6	8, −4, 4, −4		1	0	0
3	0111	0111	−6, 2, 2, 2	8, 4, 4, 4		1	0	0

est vérifiée jusqu'au rang $n-2$. Considérons la décomposition de f relativement au sous-espace $V = \langle e_1, \dots e_{n-2} \rangle$ et à ses translatés, $f = (f_0, f_1, f_1, f_2)$ où les f_i sont des fonctions symétriques quadratiques à $n-2$ variables. D'après le lemme 7.2, nous avons que $\mathcal{L}(f_1) \leq 2^{\frac{n-1}{2}}$ si $\mathcal{L}(f) \leq 2^{\frac{n+1}{2}}$. Or f_1 est une fonction booléenne à $n-2$ variables, $n-2$ impair, donc $\mathcal{L}(f_1) \geq 2^{\frac{n-1}{2}}$ par hypothèse de récurrence. On a donc $\mathcal{L}(f_1) = 2^{\frac{n-1}{2}}$. On peut alors exprimer f sous la forme :

$$f = (x_{n-1} + 1)(x_n + 1) f_0 + x_{n-1}(x_n + 1) f_1 + x_n(x_{n-1} + 1) f_1 + x_{n-1} x_n f_2,$$

ce qui nous donne

$$f = f_0 + x_{n-1}(f_0 + f_1) + x_n(f_0 + f_1) + x_{n-1}x_n(f_0 + f_2).$$

Supposons que $deg(f) = k$ avec $k > 2$. Alors $deg(f_0) = k$, et nous devons aussi avoir $deg(f_0 + f_1) = k - 1$ c'est-à-dire $deg(f_1) = k$, ce qui conduit à une contradiction. Ainsi nous pouvons conclure que que $deg(f) = 2$. □

Le résultat précédent implique en particulier que la plus grande non-linéarité possible pour une fonction symétrique à n variables, n impair, est exactement $2^{n-1} - 2^{\frac{n-1}{2}}$. Ceci n'est pas le cas en général pour les fonctions booléennes quelconques sauf pour $n \leq 7$. Pour tout n impair $n \geq 15$, on connaît des fonctions dont la non-linéarité dépasse cette valeur [PW83]. Pour $9 \leq n \leq 13$, le problème reste ouvert.

7.2 Fonctions symétriques de non-linéarité élevée

Nous cherchons maintenant à caractériser les fonctions de non-linéarité élevée. Nous rattachons ainsi la non-linéarité à la périodicité d'une partie du vecteur des valeurs simplifié d'une fonction symétrique.

Théorème 7.4 *Soit f une fonction booléenne symétrique à n variables. Si*

$$\mathcal{L}(f) < 2^{\lfloor \frac{n+1}{2} \rfloor} + 2^{t+1}$$

pour un entier t, $0 \leq t < \lfloor \frac{n+1}{2} \rfloor$, alors

$$v_f(i+2) = v_f(i) \oplus 1, \text{ pour tout } t \leq i \leq n-2-t \ ,$$

ou de manière équivalente, $f = q + h$ avec q fonction symétrique quadratique et h fonction symétrique à n variables telle que $v_h(i) = 0$ pour tout $t \leq i \leq n - t$.

Preuve : Par récurrence sur t.

- Démonstration pour $t = 0$. Supposons que $\mathcal{L}(f) < 2^{\lfloor \frac{n+1}{2} \rfloor} + 2$. $\mathcal{L}(f)$ étant un entier pair, alors $\mathcal{L}(f) = 2^{\lfloor \frac{n+1}{2} \rfloor}$. Dans ce cas, nous savons d'après [Sav94, MS02] que f est quadratique et l'expression de $v(f)$ se déduit directement de la proposition 5.34.
- Récurrence : l'énoncé étant vérifié pour $t-1$, on cherche à démontrer que cela implique sa validité pour t. Supposons maintenant que $\mathcal{L}(f) < 2^{\lfloor \frac{n+1}{2} \rfloor} + 2^{t+1}$. Soit $g = f_{e_n+V} = f_{e_{n-1}+V}$. D'après le lemme 7.2, la fonction g est une fonction symétrique à $(n-2)$ variables qui vérifie alors

  $$\mathcal{L}(g) < 2^{\lfloor \frac{(n-2)+1}{2} \rfloor} + 2^t \ .$$

  Par hypothèse de récurrence, nous en déduisons que $v_g(i+2) = v_g(i) \oplus 1$ pour tout i tel que $t - 1 \leq i \leq (n-2) - 1 - t$. Or le vecteur des valeurs simplifié de g et celui de f sont liés par la relation $v_g(i) = v_f(i+1)$ pour tout $0 \leq i \leq n-2$ (cf. Prop. 5.39). Il s'ensuit que

  $$v_f(i+2) = v_g(i+1) = v_g(i-1) \oplus 1 = v_f(i) \oplus 1 \quad \text{pour tout } t \leq i \leq n-2-t \ .$$

□

Ce résultat peut en fait être également démontré à l'aide d'un théorème prouvé par Claude Carlet [Car04, Theorème 5]. De ce résultat nous pouvons déduire un corollaire immédiat qui porte sur une condition nécessaire pour qu'un vecteur des valeurs simplifié soit celui d'une fonction symétrique f telle que

$$\mathcal{L}(f) < 2^{\frac{n}{2}+1} \text{ si } n \text{ est pair,}$$
$$\mathcal{L}(f) < 2^{\frac{n+1}{2}} + 2^{\frac{n-1}{2}} \text{ si } n \text{ est impair.}$$

Corollaire 7.5 *Soit f une fonction booléenne symétrique à n variables.*
- *Si n est pair et $v_f(\frac{n}{2}-1) = v_f(\frac{n}{2}+1)$, alors $\mathcal{L}(f) \geq 2^{\frac{n}{2}+1}$;*
- *Si n est impair et $v_f(\frac{n+1}{2}) = v_f(\frac{n-3}{2})$ ou $v_f(\frac{n+3}{2}) = v_f(\frac{n-1}{2})$, alors $\mathcal{L}(f) \geq 2^{\frac{n+1}{2}} + 2^{\frac{n-1}{2}}$.*

7.3 Fonctions symétriques de non-linéarité $\mathcal{N}_{\max} - 1$ et $\mathcal{N}_{\max} - 2$

Nous cherchons maintenant à identifier complètement les fonctions symétriques dont la non-linéarité est très proche de la non-linéarité optimale. Nous allons pour ce faire utiliser le lemme suivant qui permet de calculer le spectre de Walsh d'une telle fonction à partir de celui d'une fonction quadratique.

Lemme 7.6 *Soit $f \in \mathcal{S}ym_n$ telle que*

$$\mathcal{L}(f) < 2^{\lfloor \frac{n+1}{2} \rfloor} + 2^{t+1}$$

pour un entier t, $0 \leq t < \lfloor \frac{n+1}{2} \rfloor$. Il existe alors une fonction quadratique $q \in \mathcal{S}ym_n$ telle que, pour tout j, $0 \leq j \leq n$,

$$\mathsf{F}_f(j) = \mathsf{F}_q(j) - 2\sum_{i=0}^{t-1} \left(h_i(-1)^{v_q(i)} + h_{n-i}(-1)^{v_q(n-i)\oplus j} \right) P_i(j,n)$$

où $h_i = v_f(i) \oplus v_q(i)$, $i \in \{0,\ldots,t-1\} \cup \{n-t+1,\ldots,n\}$, et $P_i(j,n)$ est le polynôme de Krawtchouk de degré i défini à la proposition 5.7.

Preuve : Le théorème 7.4 implique que $f = q + h$ où q est une fonction quadratique et h est une fonction symétrique telle que $v_h(i) = 0$ pour tout $t \leq i \leq n-t$. Soit $h_i = v_h(i)$. D'après la proposition 5.7, les coefficients de Walsh de f sont donnés par :

$$\mathsf{F}_f(j) = \sum_{i=0}^{n} (-1)^{v_f(i)} P_i(j,n) \ , \text{ pour tout } j,\ 0 \leq j \leq n \ .$$

En prenant en compte le fait que $(-1)^{v_f(i)} = (-1)^{v_q(i)}(-1)^{h_i} = (-1)^{v_q(i)}(1-2h_i)$, on peut écrire que

$$\mathsf{F}_f(j) = \mathsf{F}_q(j) - 2\left(\sum_{i=0}^{t-1} (-1)^{v_q(i)} h_i P_i(j,n) + \sum_{i=n-t}^{t} (-1)^{v_q(i)} h_i P_i(j,n) \right)$$

et le résultat s'ensuit lorsqu'on ajoute le fait que $P_{n-i}(j,n) = (-1)^j P_i(j,n)$ pour tout entier j. □

Proposition 7.7 *Les fonctions booléennes symétriques f à n variables telles que*

$$\mathcal{L}(f) = 2^{\lfloor \frac{n+1}{2} \rfloor} + 2$$

sont les 8 fonctions de degré n définies par les vecteurs simplifiés de l'ANF suivants :

$$\lambda_f = (a,b,1,0,\ldots,0,1) \quad et \quad \lambda_f = (a,b,0,1,\ldots,1,1), \quad a,b \in \mathbf{F}_2$$

Preuve : D'après le théorème 7.4, $f = q + h$ où q est une fonction symétrique quadratique et h est une fonction symétrique de vecteur des valeurs simplifié $v(h) = (h_0,0,\ldots,0,h_n)$. Ainsi le vecteur simplifié de l'ANF de h vaut

$$\lambda_h = (h_0,h_0,\ldots,h_0,h_0 \oplus h_n) \ .$$

On sait par ailleurs que $\mathcal{L}(f)$ n'est pas divisible par 4 si et seulement si f est de degré n (car dans ce cas son poids de Hamming est impair). Nous devons donc avoir $h_0 \oplus h_n = 1$. Lorsque $h_0 = 1$ et $h_n = 0$, le lemme précédent nous conduit à :

$$\mathsf{F}_f(j) = \mathsf{F}_g(j) - 2(-1)^{v_q(0)} \ .$$

De la proposition 6.1 nous tirons que pour toute fonction f telle que $\lambda_f = (a,b,0,1,\ldots,1,1)$, l'ensemble $\{|\, \mathsf{F}_f(j)\,|\,, 0 \le j \le n\}$ vaut $\{2^{\frac{n}{2}}-2, 2^{\frac{n}{2}}+2\}$ lorsque n est pair et $\{2^{\frac{n+1}{2}}-2, 2^{\frac{n+1}{2}}+2\}$ lorsque n est impair.

De manière similaire, lorsque $h_0 = 0$ et $h_n = 1$, nous avons :

$$\mathsf{F}_f(j) = \mathsf{F}_q(j) - 2(-1)^{v_q(n) \oplus j} \ .$$

Ainsi, pour toute fonction f telle que $\lambda_f = (a,b,1,0,\ldots,0,1)$, l'ensemble $\{|\, \mathsf{F}_f(j)\,|\,, 0 \le j \le n\}$ vaut $\{2^{\frac{n}{2}} - 2, 2^{\frac{n}{2}} + 2\}$ si n est pair et $\{2^{\frac{n+1}{2}} - 2, 2^{\frac{n+1}{2}} + 2\}$ si n est impair. □

Proposition 7.8 *Les fonctions booléennes symétriques f à n variables telles que*

$$\mathcal{L}(f) = 2^{\lfloor \frac{n+1}{2} \rfloor} + 4$$

sont les 4 fonctions de degré $(n-1)$ définies par les vecteurs simplifiés de l'ANF suivants :

$$\lambda_f = (a,b,0,1,\ldots,1,0) \ , \quad a,b \in \mathbf{F}_2 \ .$$

Preuve : D'après le théorème 7.4, $f = q + h$ où q est une fonction symétrique quadratique et h est une fonction symétrique telle que $v(h) = (h_0,h_1,0,\ldots,0,h_{n-1},h_n)$. Le lemme précédent nous permet d'écrire pour tout j, $0 \le j \le n$:

$$\mathsf{F}_f(j) = \mathsf{F}_g(j) - 2\left(h_0(-1)^{v_q(0)} + h_n(-1)^{v_q(n)+j}\right) - 2(n-2j)\left(h_1(-1)^{v_q(1)} + h_{n-1}(-1)^{v_q(n-1)+j}\right).$$

On voit immédiatement que

$$\left| h_0(-1)^{v_q(0)} + h_n(-1)^{v_q(n)+j} \right| \leq 2.$$

On en déduit donc que :

$$| \mathsf{F}_f(j) | \geq \left| \mathsf{F}_g(j) - 2(n-2j) \left(h_1(-1)^{v_q(1)} + h_{n-1}(-1)^{v_q(n-1)+j} \right) \right| - 4.$$

Si $(h_1,h_{n-1}) \neq (0,0)$ et $j \equiv v_q(1) \oplus v_q(n-1) \bmod 2$, on a :

$$h_1(-1)^{v_q(1)} + h_{n-1}(-1)^{v_q(n-1)+j} \in \{(-1)^{v_q(1)}, 2(-1)^{v_q(1)}\}.$$

Cependant, la proposition 6.1 nous montre que lorsque $j \equiv v_q(1) \oplus v_q(n-1) \bmod 2$, alors $\mathsf{F}_q(j)$ peut prendre les deux valeurs $\pm 2^{\lfloor \frac{n+1}{2} \rfloor}$. Ainsi il existe toujours une valeur de $j \leq 3$ telle que

$$\mathsf{F}_q(j) = (-1)^{v_q(1)+1} 2^{\lfloor \frac{n+1}{2} \rfloor}$$

ce qui implique que

$$\left| \mathsf{F}_q(j) - 2(n-2j) \left(h_1(-1)^{v_q(1)} + h_{n-1}(-1)^{v_q(n-1)+j} \right) \right| \geq 2^{\lfloor \frac{n+1}{2} \rfloor} + 2(n-2j) \ .$$

Cela signifie que quel que soit le choix pour (h_0,h_n), il existe toujours une valeur de $j \leq 3$ telle que

$$\begin{aligned} | \mathsf{F}_f(j) | &\geq 2^{\lfloor \frac{n+1}{2} \rfloor} + 2(n-2j) - 4 \\ &\geq 2^{\lfloor \frac{n+1}{2} \rfloor} + 2n - 16 \\ &> 2^{\lfloor \frac{n+1}{2} \rfloor} + 4 \end{aligned}$$

pour $n > 10$. On en déduit donc que h_1 et h_{n-1} doivent être simultanément nuls pour que $\mathcal{L}(f) = 2^{\lfloor \frac{n+1}{2} \rfloor} + 4$ lorsque $n > 10$. Le calcul pour les valeurs de $n \leq 10$ nous confirme que cette condition est aussi valide dans ces cas. Ainsi, le vecteur de l'ANF simplifié de h vaut :

$$\lambda_h = (h_0,h_0,\ldots,h_0,h_0 \oplus h_n)$$

et il faut que $h_0 = h_n = 1$, car dans le cas contraire on aurait $\mathcal{L}(f) = 2^{\lfloor \frac{n+1}{2} \rfloor} + 2$. Pour une telle fonction h, on a alors :

$$\mathsf{F}_f(j) = \mathsf{F}_q(j) - 2\left((-1)^{v_q(0)} + (-1)^{v_q(n)+j} \right) \ .$$

On vérifie enfin que les fonctions ainsi définies atteignent bien la non-linéarité attendue. Considérons les fonctions symétriques quadratiques de vecteur des valeurs simplifié $\lambda(q) = (0,\lambda,1,0,\ldots,0)$, $\lambda \in \mathbf{F}_2$, alors $v_q(0) = 0$ et $v_q(n)$ dépend de la valeur de $n \bmod 4$:

$$v_q(n) = \begin{cases} 0 & \text{si } \ n \equiv 0 \bmod 4 \\ 1 & \text{si } \ n \equiv 2 \bmod 4 \\ \lambda & \text{si } \ n \equiv 1 \bmod 4 \\ \lambda \oplus 1 & \text{si } \ n \equiv 3 \bmod 4. \end{cases}$$

TAB. 7.1 – *Coefficients de Walsh des fonctions $f \in \mathcal{S}ym_n$ de terme constant non-nul et de non-linéarité $2^{n-1} - 2^{\lfloor \frac{n-1}{2} \rfloor} - 2$: $\lambda(f) = (1,\bar{\lambda},0,1,\ldots,1,0)$.*

$\mathsf{F}_f(j)$	$n \equiv 0 \bmod 4$	$n \equiv 2 \bmod 4$	$n \equiv 1 \bmod 4$	$n \equiv 3 \bmod 4$
$j \equiv 0 \bmod 4$	$(-1)^{\frac{n}{4}} 2^{\frac{n}{2}} - 4$	$(-1)^{\lambda}(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$((-1)^{\lambda}+1)\left((-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}} - 2\right)$	$((-1)^{\lambda+1}+1)\left((-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}} - 2\right)$
$j \equiv 1 \bmod 4$	$(-1)^{\lambda+1}(-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	$(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}} - 4$	$((-1)^{\lambda+1}+1)\left((-1)^{\frac{n-1}{4}} 2^{\frac{n-1}{2}} - 2\right)$	$((-1)^{\lambda}+1)\left((-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}} - 2\right)$
$j \equiv 2 \bmod 4$	$(-1)^{\frac{n}{4}+1} 2^{\frac{n}{2}} - 4$	$(-1)^{\lambda+1}(-1)^{\frac{n-2}{4}} 2^{\frac{n}{2}}$	$((-1)^{\lambda}+1)\left((-1)^{\frac{n+3}{4}} 2^{\frac{n-1}{2}} - 2\right)$	$((-1)^{\lambda+1}+1)\left((-1)^{\frac{n-3}{4}} 2^{\frac{n-1}{2}} - 2\right)$
$j \equiv 3 \bmod 4$	$(-1)^{\lambda}(-1)^{\frac{n}{4}} 2^{\frac{n}{2}}$	$(-1)^{\frac{n-2}{4}+1} 2^{\frac{n}{2}} - 4$	$((-1)^{\lambda+1}+1)\left((-1)^{\frac{n+3}{4}} 2^{\frac{n-1}{2}} - 2\right)$	$((-1)^{\lambda}+1)\left((-1)^{\frac{n+1}{4}} 2^{\frac{n-1}{2}} - 2\right)$

Nous pouvons calculer les valeurs de $\mathsf{F}_q(j) - 2\left(1 + (-1)^{v_q(n)+j}\right)$ que nous présentons dans le tableau 7.1.

Dans le cas où le terme constant vaut 1, alors $v_q(0) = 1$ et on déduit la valeur des coefficients de Walsh de manière immédiate puisqu'elle vaut $-\mathcal{F}(f + \varphi_\alpha)$. Aussi pour toute fonction f telle que $\lambda_f = (a,b,0,1,\ldots,1,0)$, l'ensemble $\{|\,\mathsf{F}_f(j)\,|\,,\ 0 \leq j \leq n\}$ vaut $\{2^{\frac{n}{2}},2^{\frac{n}{2}} - 4,2^{\frac{n}{2}} + 4\}$ lorsque n est pair et $\{2^{\frac{n+1}{2}} + 4,2^{\frac{n+1}{2}} - 4,0\}$ lorsque n est impair □

Comme les fonctions de non-linéarité $2^{\lfloor\frac{n+1}{2}\rfloor} + 4$ correspondent à la somme d'une fonction équilibrée triviale et d'une fonction quadratique, on voit en particulier à la lecture du tableau 7.1 que pour n impair, il existe toujours une fonction de cette forme qui soit équilibrée triviale. Cette famille infinie de fonctions semblent donc particulièrement bien adaptées à une utilisation en cryptographie puisqu'elles sont équilibrées et qu'elles possèdent un degré et une non-linéarité presque optimaux. Leur caractère symétrique garantit par ailleurs un faible coût d'implémentation notamment matérielle.

Conclusion et perspectives

Dans cette thèse j'ai abordé la question des critères de sécurité des algorithmes de chiffrement à clé secrète. La première partie de mes travaux a porté sur la cryptanalyse des chiffrements itératifs par blocs et en particulier la généralisation de la cryptanalyse différentielle d'ordre supérieur, alors que la deuxième partie a été dévolue à l'étude des propriétés cryptographiques de la famille des fonctions booléennes symétriques. Ces deux axes abordent la cryptographie symétrique sous des angles différents, l'un du point de vue du cryptanalyste, l'autre de celui du concepteur. Ils soulèvent cependant tous deux le problème de l'utilisation de fonctions fortement structurées dans un système de chiffrement.

Les fonctions booléennes symétriques en cryptographie

Les fonctions booléennes symétriques font l'objet de nombreuses études dans d'autres domaines que la cryptographie car elles interviennent de manière intensive et naturelle dans la conception de circuits. Elles possèdent en effet la bonne propriété d'être représentables de manière peu coûteuse sous forme logicielle ou matérielle. J'ai utilisé cette propriété pour calculer de manière exhaustive et rapide les caractéristiques de toutes les fonctions booléennes symétriques jusqu'à 24 variables ; confrontée ensuite au problème du dépouillement des données, j'ai dû me restreindre à des fonctions particulières pour un nombre de variables plus élevé.

L'étude que j'ai menée sur cette famille de fonctions a mis à jour une propriété de périodicité de leur vecteur des valeurs simplifié (vecteur qui représente les valeurs prises par la fonction pour les différents poids du vecteur d'entrée) liée au degré algébrique. Cette propriété combinée à la caractérisation des restrictions d'une fonction symétrique m'a permis d'améliorer les bornes existantes sur l'ordre de résilience maximal atteignable par une fonction booléenne symétrique. Cette nouvelle borne ne dépend plus du nombre de variables de la fonction mais uniquement de son degré algébrique contrairement aux bornes précédemment connues. Néanmoins, la conjecture de von zur Gathen et Roche [vzGR97] qui affirme qu'une fonction symétrique est au plus 2-résiliente demeure un problème ouvert.

J'ai ensuite étudié les dérivées d'une fonction symétrique. Celles-ci ne sont en général pas symétriques mais leurs restrictions à certains sous-espaces le sont. J'ai donc utilisé cette propriété pour caractériser les fonctions symétriques vérifiant le critère de propagation. J'ai aussi pu déterminer les conditions d'existence d'une structure linéaire pour ces fonctions. La propriété de périodicité a également un impact sur le calcul des caractéristiques (poids, spectre de Walsh) d'une fonction symétrique, en particulier pour une petite période, ce qui correspond à des degrés algébriques faibles. J'ai ainsi pu donner l'expression complète du spectre de Walsh et d'auto-corrélation des fonctions symétriques de degré 2 et 3. Bien que la technique utilisée se complique un peu pour les degrés supérieurs, il est néanmoins possible de déterminer toutes les fonctions équilibrées à n variables de degré au plus 7 et d'étudier en détail certaines propriétés pour les degrés inférieurs à 8. Il s'agit des premiers résultats (non-triviaux) sur le poids des

fonctions symétriques qui portent sur des familles de fonctions quel que soit leur nombre de variables.

Enfin, le dernier chapitre porte sur l'étude de la non-linéarité. Les fonctions booléennes symétriques de non-linéarité maximale sont connues et ont été entièrement caractérisées, ce sont les fonctions quadratiques [Sav94, MS02]. Leur faible degré, qui les rend inutilisables dans la plupart des applications cryptographiques, m'a donc amenée à m'intéresser aux fonctions sous-optimales. Les propriétés de périodicité et des restrictions d'une fonction symétrique m'ont ainsi permis de caractériser les fonctions de non-linéarité élevée, plus précisément strictement supérieure à $2^{n-1} - 2^{\lfloor \frac{n-1}{2} \rfloor} - 2^t$, avec $0 \leq t < \lfloor \frac{n-1}{2} \rfloor$. Dans cette famille de fonctions, j'ai notamment pu montrer que les fonctions de non-linéarité sous-optimale $2^{n-1} - 2^{\lfloor \frac{n-1}{2} \rfloor} - 2$ sont exactement 4 fonctions de degré $(n-1)$ dont le vecteur des valeurs simplifié se déduit de celui des fonctions quadratiques ; cela signifie en particulier que pour n impair, deux de ces fonctions sont équilibrées. Ainsi ces fonctions équilibrées de non-linéarité et de degré algébrique presque maximaux pourraient être particulièrement indiquées pour être utilisées en tant que fonction de filtrage dans un chiffrement à flot. Néanmoins la structure de telles fonctions, notamment leur faible distance aux fonctions quadratiques, pourrait être une source de faiblesse pour le chiffrement ainsi conçu.

Il est important de souligner le fait que tous les critères cryptographiques, combinés aux contraintes d'implémentation, sont difficiles à vérifier simultanément. Ainsi, l'ajout supplémentaire de la structure symétrique réduit encore le choix des fonctions optimales disponibles pour le concepteur de systèmes de chiffrement. De manière générale les fonctions symétriques extrêmement structurées font craindre l'existence d'attaques tirant parti de leurs propriétés. La question de savoir si le caractère symétrique peut être exploité dans une cryptanalyse, ou s'il est possible d'utiliser des fonctions symétriques en cryptographie demeure donc un problème ouvert. Il reste par ailleurs à étudier la propriété cryptographique d'immunité algébrique pour les fonctions symétriques. Enfin, en dehors du domaine de la cryptologie, les propriétés que j'ai mises en évidence dans mes travaux doivent pouvoir faire l'objet d'adaptation dans d'autres cadres d'utilisation afin d'exploiter de manière optimale les propriétés de périodicité des vecteurs des valeurs simplifié et de l'ANF comme je l'ai fait pour les fonctions seuils et exactes afin de concevoir des circuits efficaces.

Cryptanalyses différentielles d'ordre supérieur

Dans la première partie de mes travaux, je me suis intéressée à la sécurité des chiffrements itératifs par blocs vis-à-vis des attaques statistiques et en particulier de celles de la famille de la cryptanalyse différentielle. Je me suis attachée à modéliser les cryptanalyses sur le dernier tour et j'ai utilisé cette modélisation pour décrire les cryptanalyses classiques que sont les attaques linéaire et différentielle ; j'ai ainsi détaillé les généralisations de l'attaque différentielle que sont les cryptanalyses par différentielle impossible, par différentielle tronquée et par différentielle d'ordre supérieur. La cryptanalyse différentielle d'ordre supérieur s'applique en particulier dès que, pour toutes les clés $\mathbf{k}$, le chiffrement réduit $G_{\mathbf{k}}$ possède une structure linéaire d'ordre supérieur, c'est-à-dire dès qu'il existe un sous-espace vectoriel V pour lequel $D_V G_{\mathbf{k}}$, la dérivée du chiffrement réduit, est constante.

Une telle différentielle d'ordre 7 existe pour $M'1$, version de MISTY1 sans fonction linéaire et a été utilisée par Tanaka et al. [THK99] pour élaborer une attaque différentielle d'ordre supérieur sur $M'1$. Babbage et Frisch ont montré dans [BF00] que cette propriété est main-

tenue en remplaçant une des boîtes S, la boîte S_7 par n'importe quelle fonction puissance presque courbe de même degré. J'ai montré qu'une propriété plus générale liée à la structure même des fonctions de substitution utilisées, les fonctions presque courbes, est à l'origine de cette faiblesse. Plus précisément, le théorème de McEliece fournit une majoration du degré de la composée de deux fonctions puissance presque courbes qui explique l'origine de l'attaque et sa validité lorsqu'on substitue un exposant par un autre de même poids. J'ai ainsi pu analyser plus généralement une famille de chiffrements de même structure que MISTY1, paramétrée par la taille de bloc $16m$, pour cette attaque.

De manière plus générale, j'ai montré que les fonctions dont le spectre de Walsh est divisible par une grande puissance de 2 sont vulnérables au même type d'attaque différentielle d'ordre supérieur car le degré de leur composée admet une borne supérieure qui croît bien plus lentement que la borne triviale. J'ai par exemple pu analyser les chiffrements de Feistel à 5 tours et montrer que l'utilisation de fonctions presque courbes, fonctions dont le spectre de Walsh est divisible par $2^{\frac{n+1}{2}}$, comme fonction de tour les rend vulnérables à la cryptanalyse différentielle d'ordre supérieur. De tels résultats soulèvent naturellement les questions suivantes. Est-il possible de généraliser la technique appliquée sur 5 tours à un nombre de tours plus élevé ? Il faudrait dans ce cas évaluer la divisibilité de la composée de r fonctions, $r > 2$, dont le spectre de Walsh est divisible par une grande puissance de 2, afin de suivre précisément l'évolution du degré de la fonction de chiffrement à chaque tour. Une meilleure approximation du degré de la fonction de chiffrement réduite ne peut en effet qu'améliorer l'attaque en diminuant l'ordre de la différentielle utilisable. D'autre part, la complexité en nombre de couples clairs-chiffrés s'élevant à $2^{\dim(V)}$ pour une cryptanalyse s'appuyant sur une dérivée relativement à un sous-espace vectoriel V, il est légitime de se demander si on peut trouver une structure linéaire d'ordre plus faible que le degré de la fonction de chiffrement réduite. Une autre piste pour réduire cette complexité repose sur la prise en compte d'un plus petit nombre de composantes booléennes de la dérivée d'ordre supérieur, ce qui revient à considérer des dérivées d'ordre supérieur tronquées.

Un axe de recherche plus général consiste à tenter d'analyser, par des propriétés similaires de divisibilité des coefficients de Walsh, d'autres cas particuliers de la cryptanalyse différentielle d'ordre supérieur, notamment les attaques intégrales (ou *Square attacks*) [KW02, DKR97] qui s'appliquent lorsque la fonction de confusion est constituée de plusieurs boîtes S juxtaposées, comme dans le DES ou l'AES. Ce type de construction est en effet très répandu dans la mesure où, pour des raisons de performance, ces fonctions sont soit mises en table soit implémentées par un circuit ; elles ne peuvent donc être définies que sur un petit nombre de variables (une boîte S de l'AES s'applique sur un octet). Dans le cas de la concaténation de k boîtes S définies sur n bits, on considère la fonction de confusion F définie par :

$$\begin{array}{rccl} F: & (\mathbf{F}_2^n)^k & \to & (\mathbf{F}_2^n)^k \\ & (x^1,\ldots,x^k) & \mapsto & (S(x^1),\ldots,S(x^k)). \end{array}$$

Les coefficients de Walsh de la fonction F s'écrivent alors :

$$\begin{aligned} \mathcal{F}(\varphi_b \circ F + \varphi_a) &= \sum_{x \in \left(\mathbf{F}_2^n\right)^k} (-1)^{\varphi_b \circ F(x) + \varphi_a(x)} \\ &= \sum_{j=1}^{k} \sum_{x^j \in \mathbf{F}_2^n} (-1)^{\sum_{j=1}^{k} b^j \cdot S(x^j) + \sum_{j=1}^{k} a^j \cdot x^j}, \end{aligned}$$

ce qui permet de les exprimer en fonction de ceux de S :

$$\begin{aligned}\mathcal{F}(\varphi_b \circ F + \varphi_a) &= \prod_{j=1}^{k} \sum_{x^j \in \mathbf{F}_2^n} (-1)^{b^j \cdot S(x^j) + a^j \cdot x^j} \\ &= \prod_{j=1}^{k} \mathcal{F}(\varphi_{b^j} \circ S + \varphi_{a^j}).\end{aligned}$$

Ainsi, $\mathcal{L}(F) = \mathcal{L}(S)^k$ ce qui augmente donc la divisibilité du spectre de Walsh de la fonction de confusion. Dans le cas de l'AES où $k = 16$ et la boîte S correspond à la fonction inverse $x \mapsto x^{-1}$ dans le corps fini à 2^8 éléments, la divisibilité faible, égale à 4, de la boîte S devient ainsi pour F égale à 2^{32}. On peut alors se demander si les outils d'analyse que j'ai développés peuvent également englober les attaques de la famille des *Square attacks* sur les chiffrements conçus sur le modèle de l'AES.

Perspectives générales

Le constat qui s'impose à l'issue de ces travaux souligne le rôle paradoxal joué par les fonctions fortement structurées en cryptographie. Les cryptanalyses et les contraintes d'implémentation tendent en effet à imposer leur utilisation en tant qu'objets optimaux au regard des critères requis.

Ainsi, les premières attaques génériques sur les systèmes de chiffrements ont révélé la nécessité pour les fonctions entrant dans leur conception de vérifier des propriétés algébriques fortes. Pour les chiffrement conçus suite à ces résultats, résister au mieux aux attaques connues et pouvoir être utilisés dans des environnements imposés a sélectionné certaines fonctions optimales au regard de l'état des connaissances du moment, qui se sont souvent révélées être des fonctions extrémales possédant de fortes structures algébriques. De telles fonctions offrent donc des propriétés remarquables tant pour le concepteur que pour le cryptanalyste dont la tâche revient alors à exploiter ces structures en vue de déterminer des propriétés discriminantes pour les chiffrements ainsi conçus. Dans cette optique, les fonctions presque courbes garantissant une résistance maximale aux cryptanalyses linéaire et différentielle ont, par définition, toujours un spectre de Walsh divisible par une grande puissance de 2. C'est cette propriété qui, comme je l'ai montré, rend les systèmes qui les utilisent vulnérables à des attaques différentielles d'ordre supérieur. De la même manière, la fonction puissance inverse utilisée dans l'AES qui correspond au cas le plus favorable pour contrer l'attaque différentielle d'ordre supérieur que j'ai présentée (divisibilité 4), semble être le choix le moins adapté face aux attaques algébriques.

La question que soulève logiquement une telle constatation est celle de la détermination de fonctions sous-optimales. Le concepteur a la tâche de présenter un système qui s'appuie sur des preuves ou au moins de fortes présomptions de sécurité pour pouvoir prétendre proposer sérieusement son utilisation. De telles preuves peuvent difficilement s'établir sans imposer de fortes structures aux fonctions entrant dans la conception de ces systèmes. Ces structures sont alors autant de faiblesses potentielles... Un compromis entre les critères existants est une tâche qui se révèle déjà délicate et qui réduit souvent comme peau de chagrin la liste des fonctions potentiellement utilisables. La recherche de fonctions optimales doit maintenant se diriger vers un compromis équilibré entre les critères conduisant à des propriétés sur lesquelles pourront

s'appuyer les preuves des concepteurs et une certaine forme d'indétermination qui doit laisser le cryptanalyste incapable d'attaquer le système. Gageons qu'un tel équilibre ne se stabilisera pas avant de longues années.

Bibliographie

[3GP] « Specification of the 3GPP confidentiality and integrity algorithms; Document 2: Kasumi algorithm specification ». `http://www.3gpp.org/ftp/Specs/html-info/35201.htm`.

[AB05] F. Arnault et T.B. Berger. « F-FCSR: Design of a new class of stream ciphers ». In *Fast Software Encryption - FSE 2005*, volume 3557 de *Lecture Notes in Computer Science*, pages 83–97. Springer-Verlag, 2005.

[Ano94] Anonymous. « RC4 Source Code ». `http://cypherpunks.venona.com/date/1994/09/msg00428.html`, septembre 1994.

[Arm04] F. Armknecht. « Improving fast algebraic attacks ». In *Fast Software Encryption - FSE 2004*, volume 3017 de *Lecture Notes in Computer Science*, pages 65–82. Springer-Verlag, 2004.

[BBC+05a] C. Berbain, O. Billet, A. Canteaut, N. Courtois, B. Debraize, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin et H. Sibert. « Decim: a new stream cipher for hardware applications ». `http://www.ecrypt.eu.org/stream/`, 2005. Réponse à : Call for Stream Cipher Primitives, Network of Excellence in Cryptology ECRYPT.

[BBC+05b] C. Berbain, O. Billet, A. Canteaut, N. Courtois, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin et H. Sibert. « Sosemanuk: a fast oriented software-oriented stream cipher ». `http://www.ecrypt.eu.org/stream/`, 2005. Réponse à : Call for Stream Cipher Primitives, Network of Excellence in Cryptology ECRYPT.

[BBS99] E. Biham, A. Biryukov et A. Shamir. « Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials ». In *Advances in Cryptology - EUROCRYPT'99*, volume 1592 de *Lecture Notes in Computer Science*, pages 12–23. Springer-Verlag, 1999.

[BCP05] L. Budaghyan, C. Carlet et A. Pott. « New constructions of Almost Bent and Almost Perfect Nonlinear polynomials ». In *Workshop on Coding and Cryptography - WCC 2005*, pages 306–315, Bergen, Norvège, 2005.

[BCQ04] A. Biryukov, C. De Cannière et M. Quisquater. « On multiple linear approximations ». In *Advances in Cryptology - CRYPTO 2004*, volume 3152 de *Lecture Notes in Computer Science*, pages 1–22. Springer-Verlag, 2004.

[Ber68] E.R. Berlekamp. *Algebraic Coding Theory*. McGraw-Hill, 1968.

[BF00] S. Babbage et L. Frisch. « On MISTY1 Higher Order Differential Cryptanalysis ». In *Proceedings of ICISC 2000*, volume 2015 de *Lecture Notes in Computer Science*, pages 22–36. Springer-Verlag, 2000.

[BGW99] M. Briceno, I. Goldberg et D. Wagner. « A pedagogical implementation of A5/1 ». `http://www.gsm-security.net/papers/a51.shtml`, 1999.

[BLM+05] A. Braeken, J. Lano, N. Mentens, B. Preneel et I. Verbauwhede. « SFINKS: a synchronous stream cipher for restricted hardware environments ». `http://www.ecrypt.eu.org/stream/`, 2005. Réponse à : Call for Stream Cipher Primitives, Network of Excellence in Cryptology ECRYPT.

[BM84] M. Blum et S. Micali. « How to Generate Cryptographically Strong Sequences Of Pseudo-Random Bits ». *SIAM Journal on Computing*, 13, 1984.

[BPP01] J. Boyar, R. Peralta et D. Pochuev. « Concrete conjunctive complexity of symmetric functions ». Rapport Technique YALEU/DCS/TR1219, Yale University, 2001.

[Brü84] J.-O. Brüer. « On pseudorandom sequences as crypto generators ». In *Proc. of the 1984 International Zürich Seminar on Digital Communications*, pages 157–161. IEEE, 1984.

[BS91] E. Biham et A. Shamir. « Differential Cryptanalysis of DES-like cryptosystems ». *Journal of Cryptology*, 4(1):3–72, 1991.

[BS93] E. Biham et A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer-Verlag, 1993.

[BS05] E. Biham et J. Seberry. « **Py** (Roo): a fast and secure stream cipher using rolling arrays ». `http://www.ecrypt.eu.org/stream/`, 2005. Réponse à : Call for Stream Cipher Primitives, Network of Excellence in Cryptology ECRYPT.

[Can97] A. Canteaut. « Differential cryptanalysis of Feistel ciphers and differentially uniform mappings ». In *Selected Areas on Cryptography, SAC'97*, pages 172–184, Ottawa, Canada, 1997.

[Car94] C. Carlet. « Two new classes of bent functions ». In *Advances in Cryptology - EUROCRYPT'93*, volume 765 de *Lecture Notes in Computer Science*, pages 77–101. Springer-Verlag, 1994.

[Car04] C. Carlet. « On the degree, nonlinearity, algebraic thickness, and nonnormality of Boolean functions, with developments on symmetric functions ». *IEEE Transactions on Information Theory*, 50(9):2178–2185, 2004.

[CCCF01] A. Canteaut, C. Carlet, P. Charpin et C. Fontaine. « On cryptographic properties of the cosets of $R(1,m)$ ». *IEEE Transactions on Information Theory*, 47(4):1494–1513, 2001.

[CCD99] A. Canteaut, P. Charpin et H. Dobbertin. « A new characterization of almost bent functions ». In *Fast Software Encryption - FSE'99*, volume 1636 de *Lecture Notes in Computer Science*, pages 186–200. Springer-Verlag, 1999.

[CCD00a] A. Canteaut, P. Charpin et H. Dobbertin. « Binary m-sequences with three-valued crosscorrelation: a proof of Welch's conjecture ». *IEEE Transactions on Information Theory*, 46:4–8, 2000.

[CCD00b] A. Canteaut, P. Charpin et H. Dobbertin. « Weight divisibility of cyclic codes, highly nonlinear functions of $GF(2^m)$, and crosscorrelation of maximum-length sequences ». *SIAM Journal of Discrete Mathematics*, 13(1):105–138, 2000.

[CCV02] A. Canteaut, P. Charpin et M. Videau. « Cryptanalysis of block ciphers and weight divisibility of some binary codes ». In M. Blaum, P.G. Farrell et H.C.A. Van Tilborg, éditeurs, *Information, Coding and Mathematics (Workshop honoring Bob McEliece on his 60th birthday)*, volume 687 de *Kluwer Internat. Ser. Engrg. Comput. Sci.*, pages 75–97. Kluwer, 2002.

[CCZ98] C. CARLET, P. CHARPIN et V. ZINOVIEV. « Codes, Bent Functions and Permutations Suitable for DES-like Cryptosystems ». *Designs, Codes and Cryptography*, 15(2):125–156, 1998.

[CG99] C. CARLET et P. GUILLOT. « A New Representation of Boolean Functions ». In *Proceedings of Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 13th International Symposium, AAECC-13*, volume 1719 de *Lecture Notes in Computer Science*, pages 94–103. Springer-Verlag, 1999.

[CG01] C. CARLET et P. GUILLOT. « *Codes and Association Schemes* », volume 56 de *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Chapitre Bent, resilient functions and the numerical normal form, pages 87–96. American Mathematical Society, 2001.

[Cha98] P. CHARPIN. « *Handbook of Coding Theory* », volume II, Chapitre Open problems on cyclic codes, pages 1765–1853. Elsevier, 1998.

[CJM02] P. CHOSE, A. JOUX et M. MITTON. « Fast Correlation Attacks: An Algorithmic Point of View ». In *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 de *Lecture Notes in Computer Science*, pages 209–221. Springer-Verlag, 2002.

[CJS00] V. CHEPYSHOV, T. JOHANSSON et B. SMEETS. « A simple algorithm for fast correlation attacks on stream ciphers ». In *Fast Software Encryption - FSE 2000*, volume 1978, pages 181–195. Springer-Verlag, 2000.

[CKM94] D. COPPERSMITH, H. KRAWCZYK et Y. MANSOUR. « The shrinking generator ». In *Advances in Cryptology - CRYPTO'93*, volume 773 de *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[CM03] N. COURTOIS et W. MEIER. « Algebraic attacks on stream ciphers with linear feedback ». In *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 de *Lecture Notes in Computer Science*, pages 345–359, 2003.

[Com74] L. COMTET. *Advanced Combinatorics*. D. Reidel Publishing Compagny, 1974.

[Cou03] N. COURTOIS. « Fast algebraic attacks on stream ciphers with linear feedback ». In *Advances in Cryptology - CRYPTO 2003*, volume 2729 de *Lecture Notes in Computer Science*, pages 176–194. Springer-Verlag, 2003.

[Cou04] N.T. COURTOIS. « Feistel Schemes and Bi-linear Cryptanalysis ». In *Advances in Cryptology - CRYPTO 2004*, volume 3152 de *Lecture Notes in Computer Science*, pages 23–40. Springer-Verlag, 2004.

[CP02] N.T. COURTOIS et J. PIEPRZYK. « Cryptanalysis of Block Ciphers with Overdefined Systems of Equations ». In *Advances in Cryptology - ASIACRYPT 2002*, volume 2502 de *Lecture Notes in Computer Science*, pages 267–287. Springer-Verlag, 2002.

[Cry01] CRYPTREC. « Evaluation of Cryptographic Techniques ». `http://www.ipa.go.jp/security/enc/CRYPTREC/index-e.html`, 2001.

[CT00] A. CANTEAUT et M. TRABBIA. « Improved fast correlation attacks using parity-check equations of weight 4 and 5 ». In *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 de *Lecture Notes in Computer Science*, pages 573–588. Springer-Verlag, 2000.

[CTZ97] P. CHARPIN, A. TIETÄVÄINEN et V. ZINOVIEV. « On binary cyclic codes with minimum distance $d = 3$ ». *Problems of Information Transmission*, 33(4):287–296, 1997.

[CV95] F. Chabaud et S. Vaudenay. « Links between differential and linear cryptanalysis ». In *Advances in Cryptology - EUROCRYPT'94*, volume 950 de *Lecture Notes in Computer Science*, pages 356–365. Springer-Verlag, 1995.

[CV02a] A. Canteaut et M. Videau. « Degree of composition of highly nonlinear functions and applications to higher order cryptanalysis ». In *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 de *Lecture Notes in Computer Science*, pages 518–533. Springer-Verlag, 2002.

[CV02b] A. Canteaut et M. Videau. « Higher order differential attacks on iterated block ciphers using almost bent round functions ». In *Proc. 2002 IEEE International Symposium on Information Theory*, page 209. IEEE, 2002.

[CV02c] A. Canteaut et M. Videau. « Weakness of Block Ciphers Using Highly Nonlinear Confusion Functions. ». Rapport Technique RR-4367, INRIA, 2002. `http://www.inria.fr/rrrt/rr-4367.html`.

[CV05] A. Canteaut et M. Videau. « Symmetric Boolean functions ». *IEEE Transactions on Information Theory*, 51(8):2791–2811, 2005.

[DCG+00] E. Dawson, A. Clark, J.Dj. Golic, W. Millan, L. Penna et L. Simpson. « The LILI-128 keystream generator ». In *Proceedings of the first NESSIE Workshop*, 2000.

[DGM04] D.K. Dalai, K.C. Gupta et S. Maitra. « Results on algebraic immunity for cryptographically significant Boolean functions ». In *Progress in Cryptology - Indocrypt 2004*, volume 1880 de *Lecture Notes in Computer Science*, pages 92–106. Springer-Verlag, 2004.

[DH76] W. Diffie et M. Hellman. « New directions in cryptography ». *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[Dil74] J.F. Dillon. « *Elementary Hadamard Difference sets* ». Thèse de doctorat, University of Maryland, 1974.

[DKR97] J. Daemen, L.R. Knudsen et V. Rijmen. « The block cipher Square ». In *Fast Software Encryption - FSE'97*, volume 1267 de *Lecture Notes in Computer Science*, pages 149–165. Springer-Verlag, 1997.

[DMS05] D.K. Dalai, S. Maitra et S. Sarkar. « Basic Theory in Construction of Boolean Functions with Maximum Possible Annihilator Immunity ». Rapport Technique, Cryptology ePrint Archive, 2005. `http://eprint.iacr.org/2005/229`.

[Dob98a] H. Dobbertin. « Almost Perfect Nonlinear power functions on $GF(2^n)$: the Niho case ». *Information and Computation*, 1998.

[Dob98b] H. Dobbertin. « Almost Perfect Nonlinear power functions on $GF(2^n)$: the Welch case ». *IEEE Transactions on Information Theory*, 1998.

[Dob98c] H. Dobbertin. « One-to-one highly nonlinear power functions on $GF(2^n)$ ». *Applicable Algebra in Engineering, Communication and Computing*, 9(2):139–152, 1998.

[DR99] J. Daemen et V. Rijmen. « AES proposal: the Rijndael block cipher ». Available at `http://csrc.nist.gov/encryption/aes/rijndael/`, 1999.

[DR02] J. Daemen et V. Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer-Verlag, 2002.

[DW97] E. Dawson et C.K. Wu. « On The Linear Structure of Symmetric Boolean Functions ». *Australasian Journal of Combinatorics*, 16:239–243, 1997.

[ECR05] ECRYPT. « ECRYPT Stream Cipher Project ». `http://www.ecrypt.eu.org/stream/`, 2005.

[EJ02] P. Ekdahl et T. Johansson. « A new version of the stream cipher SNOW ». In *Selected Areas in Cryptography – SAC 2002*, volume 2295 de *Lecture Notes in Computer Science*, pages 47–61. Springer-Verlag, 2002.

[FA03] J.C. Faugère et G. Ars. « An Algebraic Cryptanalysis of Nonlinear Filter Generators using Gröbner bases ». Rapport Technique RR-4739, INRIA, 2003. `http://www.inria.fr/rrrt/rr-4739.html`.

[Fil01] E. Filiol. « *Techniques de reconstruction en cryptologie et théorie des codes* ». Thèse de doctorat, École polytechnique, 2001.

[FIP99] « FIPS 46-3. Data Encryption Standard ». Federal Information Processing Standards Publication 46-3, 1999. U.S. Department of Commerce/N.I.S.T.

[FIP01] « FIPS 197. Advanced Encryption Standard ». Federal Information Processing Standards Publication 197, 2001. U.S. Department of Commerce/N.I.S.T.

[GCD00] J.Dj. Golić, A. Clark et E. Dawson. « Generalized inversion attack on nonlinear filter generators ». *IEEE Transactions on Computers*, 49(10):1100–1109, 2000.

[GGK05] B. Gammel, R. Göttfert et O. Kniffler. « The Achterbahn Stream Cipher ». `http://www.ecrypt.eu.org/stream/`, 2005. Réponse à : Call for Stream Cipher Primitives, Network of Excellence in Cryptology ECRYPT.

[GHS93] K. Gopalakrishnan, D.G. Hoffman et D.R. Stinson. « A Note on a Conjecture Concerning Symmetric Resilient Functions ». *Information Processing Letters*, 47(3):139–143, 1993.

[Gil97] H. Gilbert. « *Cryptanalyse statistique des algorithmes de chiffrement et sécurité des schémas d'authentification* ». Thèse de doctorat, Université de Paris-Sud, 1997.

[Gol96] J.Dj. Golic. « On the Security of Nonlinear Filter Generators ». In *Fast Software Encryption - FSE'96*, volume 1039 de *Lecture Notes in Computer Science*, pages 173–188. Springer-Verlag, 1996.

[Gou04a] A. Gouget. « *Etude des propriétés cryptographiques des fonctions booléennes et algorithme de confusion pour le chiffrement symétrique* ». Thèse de doctorat, Université de Caen Basse-Normandie, 2004.

[Gou04b] A. Gouget. « On the propagation criterion of Boolean functions ». In C. Xing K. Feng, H. Niederreiter, éditeur, *Coding, Cryptography and Combinatorics*, volume 23 de *Progr. Comput. Sci. Appl. Logic*. Birkhäuser Verlag, Basel, 2004.

[Har96] C. Harpes. *Cryptanalysis of iterated block ciphers*, volume 7 de *ETH Series in Information Processing*. Hartung-Gorre Verlag, Konstanz, 1996.

[Hel76] T. Helleseth. « Some results about the cross-correlation function between two maximal linear sequences ». *Discrete Mathematics*, 16:209–232, 1976.

[Hel80] M.E. Hellman. « A cryptanalytic time-memory tradeoff ». *IEEE Transactions on Information Theory*, 44(9):1131–1139, 1980.

[HJM05] M. Hell, T. Johansson et W. Meier. « Grain: A Stream Cipher for Constrained Environments ». `http://www.ecrypt.eu.org/stream/`, 2005. Réponse à : Call for Stream Cipher Primitives, Network of Excellence in Cryptology ECRYPT.

[HK98] T. Helleseth et P. Vijay Kumar. « *Handbook of Coding Theory* », volume II, Chapitre Sequences with low correlation, pages 1765–1853. Elsevier, 1998.

[HKM95] C. Harpes, G. Kramer et J.L. Massey. « A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma ». In *Advances in Cryptology - EUROCRYPT'95*, volume 921 de *Lecture Notes in Computer Science*, pages 24–38. Springer-Verlag, 1995.

[HS05] J. Hong et P. Sarkar. « Rediscovery of Time Memory Tradeoffs ». Rapport Technique 2005/090, Cryptology ePrint Archive, 2005. `http://eprint.iacr.org/2005/090`.

[HX01] H. Hollman et Q. Xiang. « A proof of the Welch and Niho conjectures on crosscorrelations of binary m-sequences ». *Finite Fields and Their Applications*, 7(2):253–286, 2001.

[JJ99a] T. Johansson et F. Jönsson. « Fast correlation attacks based on turbo code techniques ». In *Advances in Cryptology - CRYPTO'99*, volume 1666 de *Lecture Notes in Computer Science*, pages 181–197. Springer-Verlag, 1999.

[JJ99b] T. Johansson et F. Jönsson. « Improved fast correlation attack on stream ciphers via convolutional codes ». In *Advances in Cryptology - EUROCRYPT'99*, volume 1592 de *Lecture Notes in Computer Science*, pages 347–362. Springer-Verlag, 1999.

[JJ00] T. Johansson et F. Jönsson. « Fast correlation attacks through reconstruction of linear polynomials ». In *Advances in Cryptology - CRYPTO'00*, volume 1880 de *Lecture Notes in Computer Science*, pages 300–315. Springer-Verlag, 2000.

[JK97] T. Jakobsen et L.R. Knudsen. « The interpolation attack on block ciphers ». In *Fast Software Encryption - FSE'97*, volume 1267 de *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[Jun05] P. Junod. « *Statistical Cryptanalysis of Block Ciphers* ». Thèse de doctorat, École Polytechnique Fédérale de Lausanne, 2005.

[Kas71] T. Kasami. « The weight enumerators for several classes of subcodes of the second order binary Reed-Muller codes ». *Information and Control*, 18:369–394, 1971.

[Kel05] L. Keliher. « Refined analysis of bounds related to linear and differential cryptanalysis for the AES ». In *AES 2004*, volume 3373 de *Lecture Notes in Computer Science*, pages 42–57. Springer-Verlag, 2005.

[KMM05] J.D. Key, T.P. McDonough et V.C. Mavron. « Information sets and partial permutation decoding for codes from finite geometries ». *Finite Fields and Their Applications*, 2005.

[Knu95] L.R. Knudsen. « Truncated and higher order differentials ». In *Fast Software Encryption - FSE'94*, volume 1008 de *Lecture Notes in Computer Science*, pages 196–211. Springer-Verlag, 1995.

[KR94] B.S. Kaliski et M.J.B. Robshaw. « Linear cryptanalysis using multiple approximations ». In *Advances in Cryptology - CRYPTO '94*, volume 839 de *Lecture Notes in Computer Science*, pages 26–39. Springer-Verlag, 1994.

[KR96] L.R. Knudsen et M.J.B. Robshaw. « Non-linear Approximations in Linear Cryptanalysis ». In *Advances in Cryptology - EUROCRYPT '96*, volume 1070 de *Lecture Notes in Computer Science*, pages 224–236. Springer-Verlag, 1996.

[KRW99] L.R. Knudsen, M.J.B. Robshaw et D. Wagner. « Truncated Differentials and Skipjack ». In *Advances in Cryptology - CRYPTO'99*, volume 1666 de *Lecture Notes in Computer Science*, pages 165–180. Springer-Verlag, 1999.

[KS02] A. Klimov et A. Shamir. « A New Class of Invertible Mappings ». In *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 de *Lecture Notes in Computer Science*, pages 470–483, 2002.

[Kuk99] Z. Kukorelly. *On the validity of certain hypotheses used in linear cryptanalysis*, volume 13 de *ETH Series in Information Processing*. Hartung-Gorre Verlag, Konstanz, 1999.

[KW02] L.R. Knudsen et D. Wagner. « Integral cryptanalysis ». In *Fast Software Encryption - FSE 2002*, volume 2365 de *Lecture Notes in Computer Science*, pages 112–127. Springer-Verlag, 2002.

[Lai94] X. Lai. « Higher order derivatives and differential cryptanalysis ». In *Proc. "Symposium on Communication, Coding and Cryptography", in honor of J. L. Massey on the occasion of his 60'th birthday*, 1994.

[Lev04] S. Leveiller. « *Quelques algorithmes de cryptanalyse du registre filtré* ». Thèse de doctorat, École Nationale Supérieure des Télécommunications, 2004.

[LMM91] X. Lai, J.L. Massey et S. Murphy. « Markov ciphers and differential cryptanalysis ». In *Advances in Cryptology - EUROCRYPT'91*, volume 547 de *Lecture Notes in Computer Science*, pages 17–38. Springer-Verlag, 1991.

[Mas69] J.L. Massey. « Shift-register synthesis and BCH decoding ». *IEEE Transactions on Information Theory*, 15:122–127, 1969.

[Mat93] M. Matsui. « Linear cryptanalysis method for DES cipher ». In *Advances in Cryptology - EUROCRYPT'93*, volume 765 de *Lecture Notes in Computer Science*, pages 386–397. Springer-Verlag, 1993.

[Mat94] M. Matsui. « The first experimental cryptanalysis of the Data Encryption Standard ». In *Advances in Cryptology - CRYPTO'94*, volume 839 de *Lecture Notes in Computer Science*, pages 1–11. Springer-Verlag, 1994.

[Mat97] M. Matsui. « New Block Encryption Algorithm MISTY ». In *Fast Software Encryption - FSE'97*, volume 1267 de *Lecture Notes in Computer Science*, pages 54–68. Springer-Verlag, 1997.

[McE72] R.J. McEliece. « Weight congruence for p-ary cyclic codes ». *Discrete Mathematics*, 3:177–192, 1972.

[MH05] H. Molland et T. Helleseth. « A linear weakness in T-functions ». In *2005 IEEE International Symposium on Information Theory*. IEEE, 2005.

[Min02] M. Minier. « *Preuves d'analyse et de sécurité en cryptologie à clé secrète* ». Thèse de doctorat, Université de Limoges, 2002.

[Mit90] C.J. Mitchell. « Enumerating Boolean functions of cryptographic significance ». *Journal of Cryptology*, 2(3):155–170, 1990.

[Moz86] C.J. Mozzochi. « On the difference between consecutive primes ». *Journal of Number Theory*, 24:181–187, 1986.

[MPC04] W. Meier, E. Pasalic et C. Carlet. « Algebraic attacks and decomposition of Boolean functions ». In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 de *Lecture Notes in Computer Science*, pages 474–491. Springer-Verlag, 2004.

[MS88] W. Meier et O. Staffelbach. « Fast correlation attacks on stream ciphers ». In *Advances in Cryptology - EUROCRYPT'88*, volume 330 de *Lecture Notes in Computer Science*. Springer-Verlag, 1988.

[MS89] W. Meier et O. Staffelbach. « Fast correlation attacks on certain stream ciphers ». *Journal of Cryptology*, pages 159–176, 1989.

[MS90] W. Meier et O. Staffelbach. « Nonlinearity criteria for cryptographic functions ». In *Advances in Cryptology - EUROCRYPT'89*, volume 434 de *Lecture Notes in Computer Science*, pages 549–562. Springer-Verlag, 1990.

[MS02] S. Maitra et P. Sarkar. « Maximum Nonlinearity of Symmetric Boolean Functions on Odd Number Of Variables ». *IEEE Transactions on Information Theory*, 48(9), 2002.

[MvOV97] A.J. Menezes, P.C. van Oorschot et S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[NES01] NESSIE. « Security evaluation of NESSIE first phase ». `https://www.cryptonessie.org/`, 2001.

[NK93] K. Nyberg et L.R. Knudsen. « Provable security against differential cryptanalysis ». In *Advances in Cryptology - CRYPTO'92*, volume 740 de *Lecture Notes in Computer Science*, pages 566–574. Springer-Verlag, 1993.

[NK95] K. Nyberg et L.R. Knudsen. « Provable security against a differential attack ». *Journal of Cryptology*, 8(1):27–37, 1995.

[Nyb91] K. Nyberg. « Perfect nonlinear S-boxes ». In *Advances in Cryptology - EUROCRYPT'91*, volume 547 de *Lecture Notes in Computer Science*, pages 378–385. Springer-Verlag, 1991.

[Nyb93] K. Nyberg. « Differentially uniform mappings for cryptography ». In *Advances in Cryptology - EUROCRYPT'93*, volume 765 de *Lecture Notes in Computer Science*, pages 55–64. Springer-Verlag, 1993.

[Nyb94] K. Nyberg. « Linear approximation of block ciphers ». In *Advances in Cryptology - EUROCRYPT'94*, volume 950 de *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[PLL+91] B. Preneel, W.V. Leekwijck, L.V. Linden, R. Govaerts et J. Vandewalle. « Propagation characteristics of Boolean functions ». In *Advances in Cryptology - EUROCRYPT'90*, volume 437 de *Lecture Notes in Computer Science*, pages 155–165. Springer-Verlag, 1991.

[PW83] N.J. Patterson et D.H. Wiedemann. « The covering radius of the $[2^{15},16]$ Reed-Muller code is at least 16276 ». *IEEE Transactions on Information Theory*, IT-36(2):443, 1983.

[Qui04] M. Quisquater. « *Security of a threshold scheme and applications of character theory in symmetric cryptography* ». Thèse de doctorat, Katholieke Universiteit Leuven, 2004.

[Riv01] R. Rivest. « RSA security response to weaknesses in key scheduling algorithm of RC4 ». Rapport Technique, RSA Data Security, Inc., October 2001.

[Rot76] O.S. Rothaus. « On bent functions ». *Journal of Combinatorial Theory. Series A*, 20:300–305, 1976.

[Rue86] R.A. Rueppel. *Analysis and Design of stream ciphers*. Springer-Verlag, 1986.

[Sav94] P. SAVICKY. « On the Bent Boolean Functions That are Symmetric ». *European Journal of Combinatorics*, 15:407–410, 1994.

[Sha49] C.E. SHANNON. « The communication theory of secrecy systems ». *Bell System Technical Journal*, 28:656–715, 1949.

[Sid71] V.M. SIDELNIKOV. « On mutual correlation of sequences ». *Soviet Mathematics Doklady*, 12:197–201, 1971.

[Sie84] T. SIEGENTHALER. « Correlation-immunity of nonlinear combining functions for cryptographic applications ». *IEEE Transactions on Information Theory*, 30(5):776–780, 1984.

[Sie85] T. SIEGENTHALER. « Decrypting a class of stream ciphers using ciphertext only ». *IEEE Transactions on Computers*, 34(1):81–84, 1985.

[SM03] P. SARKAR et S. MAITRA. « Balancedness and Correlation Immunity of Symmetric Boolean Functions ». In *Proc. R.C. Bose Centenary Symposium*, volume 15 de *Electronic Notes in Discrete Mathematics*, pages 178–183, 2003.

[SP80] D.V. SARWATE et M.B. PURSLEY. « Crosscorrelation properties of pseudorandom and related sequences ». *Proceedings of the IEEE*, 68(5):593–619, 1980.

[TCG91] A. TARDY-CORFDIR et H. GILBERT. « A known plaintext attack of FEAL-4 and FEAL-6 ». In *Advances in Cryptology - CRYPTO'91*, volume 576 de *Lecture Notes in Computer Science*, pages 172–181. Springer-Verlag, 1991.

[THK99] H. TANAKA, K. HISAMATSU et T. KANEKO. « Strength of MISTY1 without FL function for Higher Order Differential Attack ». In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 1719 de *Lecture Notes in Computer Science*, pages 221–230. Springer-Verlag, 1999.

[Vau98] S. VAUDENAY. « Provable security for block cipers by decorrelation ». In *Proceedings of STACS'98*, volume 1373 de *Lecture Notes in Computer Science*, pages 249–275. Springer-Verlag, 1998.

[Vau99] S. VAUDENAY. « Vers une théorie du chiffrement symétrique ». Thèse d'habilitation, Université Paris 7, 1999.

[Vid04] M. VIDEAU. « On some properties of symmetric Boolean functions ». In *Proc. 2004 IEEE International Symposium on Information Theory*, page 500. IEEE, 2004.

[Vid05] M. VIDEAU. « Symmetric Boolean functions with high nonlinearity ». In *Conference records of We Worc*, pages 87–88, 2005.

[vzGR97] J. von zur GATHEN et J.R. ROCHE. « Polynomials with two values ». *Combinatorica*, 17(3):345–362, 1997.

[Weg87] I. WEGENER. *The complexity of Boolean functions*. Wiley, 1987.

[WT86] A.F. WEBSTER et S.E. TAVARES. « On the design of S-boxes ». In *Advances in Cryptology - CRYPTO'85*, volume 218 de *Lecture Notes in Coputer Science*, pages 523–534. Springer-Verlag, 1986.

[XM88] G. XIAO et J.L. MASSEY. « A spectral characterization of correlation-immune combining functions ». *IEEE Transactions on Information Theory*, IT-34(3):569–571, 1988.

[YG95] Y.X. YANG et B. GUO. « Further enumerating Boolean functions of cryptographic significance ». *Journal of Cryptology*, 8(3):115–122, 1995.

Index

Table des figures

Liste des tableaux

Table des matières

Résumé

Les travaux de cette thèse portent sur les critères de sécurité des algorithmes de chiffrement à clé secrète et ont été menés suivant deux axes. Le premier concerne la sécurité des chiffrements symétriques itératifs par blocs contre les attaques par distingueur sur le dernier tour. Les résultats portent en particulier sur la généralisation d'une attaque différentielle d'ordre supérieur menée sur l'algorithme MISTY1. L'origine de cette attaque ainsi que de sa généralisation a pu être expliquée grâce aux propriétés du spectre de Walsh des fonctions de non-linéarité maximale utilisées. Ainsi il a été possible d'élaborer une attaque générique sur tous les chiffrements de Feistel à cinq tours utilisant des fonctions dont le spectre de Walsh est divisible par une grande puissance de 2 car cette propriété permet d'obtenir une borne supérieure sur le degré de la composition de telles fonctions, nettement plus faible que la borne triviale. Cette attaque suggère ainsi un nouveau critère de sécurité qui porte sur la divisibilité du spectre de Walsh des fonctions de tour utilisées dans les chiffrements itératifs par blocs. La deuxième partie de la thèse porte sur l'étude des fonctions booléennes symétriques, et en particulier sur l'existence éventuelle de propriétés cryptographiques. À partir d'une propriété structurelle de périodicité d'une représentation d'une fonction booléenne symétrique, les propriétés de degré algébrique, d'équilibre, de résilience, de critère de propagation et de non-linéarité ont été étudiées, ce qui a permis d'améliorer les résultats existants. Par ailleurs, le calcul explicite du spectre de Walsh des fonctions booléennes symétriques de degré 2 et 3 a été réalisé, ainsi que la détermination de toutes les fonctions symétriques équilibrées de degré inférieur ou égal à 7, indépendamment du nombre de variables.

Mots-clé : chiffrement symétrique, chiffrement itératif par bloc, attaque par distingueur, cryptanalyse différentielle d'ordre supérieur, chiffrement à flot, fonctions booléennes symétriques

Abstract

The work done during my thesis concerns two different aspects of the security of secret key ciphers. The first part is devoted to the study of the security of iterated block ciphers against last round attacks based on distinguishers. The results especially concern a generalisation of a higher order differential attack that was lead against MISTY1 algorithm. The origin of this attack and of its generalisation has been explained thanks to the properties of the Walsh spectra of the highly nonlinear functions used in the cipher. Hence, it has been possible to mount a generic attack against all Feistel ciphers using confusion functions whose Walsh spectra are divisible by a high power of 2. Indeed, this property leads to an upper bound for the degree of the composition of such functions which can be noticeably smaller than the trivial bound. Thus the attack we have mounted leads to a new security criterion for iterated block ciphers which lies on the divisibility of the Walsh spectra of the round functions. The second part of my work is a study of cryptographic properties of symmetric Boolean functions. Starting from a structural property of one representation of symmetric Boolean functions, we improve existing results concerning algebraic degree, balance, resiliency, propagation criterion and nonlinearity of such functions. Besides, we compute explicitly the Walsh spectra of all symmetric Boolean functions of degree 2 and 3. We also determine all the balance symmetric Boolean functions of degree less than or equal to 7, for all number of variables.

Key words: symmetric cipher, iterated block cipher, distinguisher based attack, higher order differential cryptanalysis, stream cipher, symmetric Boolean functions