Analyse et conception de chiffrements à clef secrète

Anne Canteaut

MÉMOIRE D'HABILITATION À DIRIGER DES RECHERCHES

Université Pierre et Marie Curie, Paris 6

Spécialité
INFORMATIQUE

présenté par

Anne CANTEAUT

Institut National de Recherche en Informatique et Automatique, Rocquencourt

ANALYSE ET CONCEPTION de CHIFFREMENTS à CLEF SECRÈTE

soutenance le 15 septembre 2006 devant un jury composé de

Rapporteurs

Marc GIRAULT	France Telecom R&D
Thomas JOHANSSON	Université de Lund (Suède)
Serge VAUDENAY	École Polytechnique Fédérale de Lausanne (Suisse)

Examinateurs

Thierry BERGER	Université de Limoges
Claude CARLET	Université Paris 8
Pascale CHARPIN	INRIA-Rocquencourt
Tor HELLESETH	Université de Bergen (Norvège)
Claude KIRCHNER	INRIA-LORIA
Daniel LAZARD	Université Paris 6

version du 18/09/2006

À la mémoire de Hans Dobbertin

Ce document n'existerait pas sans les encouragements permanents de Damien qui, non content de m'inciter à le rédiger, m'a permis de mener à bien cette tâche dans des conditions idéales, au prix de quinze jours de vacances et de nombreux week-ends sacrifiés. Ces quelques mots ne suffisent certainement pas à excuser mon style tudesque qui, à grand renfort de propositions relatives, a tant heurté sa sensibilité littéraire.

Les travaux que je présente ici doivent énormément à Pascale Charpin. Pascale m'a transmis le goût des mathématiques discrètes et au-delà le virus de la recherche. À son contact, au cours de nos nombreux travaux communs et de nos discussions quotidiennes, j'ai véritablement appris à aborder les sujets, à prendre les problèmes à bras le corps. Travailler avec Pascale permet de mesurer combien l'ouverture d'esprit, la générosité et le travail d'équipe sont des qualités infiniment précieuses. À mes yeux, Pascale restera toujours un modèle, malheureusement bien difficile à reproduire.

J'ai été très touchée que Marc Girault accepte d'être rapporteur de ce mémoire, qui a pourtant dû lui sembler parfois bien indigeste. Je le remercie aussi pour sa gentillesse et son humour qui m'accompagnent régulièrement depuis ma thèse.

En écoutant un exposé de Serge Vaudenay à la fin de sa thèse, j'ai compris que, contrairement à toutes les idées reçues, la cryptographie symétrique ne se réduisait pas à un infâme bidouillage. Je le remercie donc pour ses travaux sans lesquels je serais peut-être passée à côté d'un domaine passionnant et amusant (c'est vrai que c'est rigolo). Tous mes remerciements enfin à Marc et Serge pour la rapidité dont ils ont fait preuve pour écrire leur rapport, et qui m'a été d'une aide incommensurable.

Je suis infiniment redevable à Thomas Johansson qui suit mes travaux de longue date et que j'ai si souvent mis à contribution. Je lui témoigne toute ma gratitude pour avoir lu ce document écrit dans une langue qui n'est pas la sienne et rédigé un rapport. Je souhaite le remercier pour son aide constante et aussi pour avoir remis les attaques par corrélation rapides au goût du jour. Tack så mycket!

Tor Helleseth ne cesse de m'impressionner à la fois par ses connaissances immenses qui semblent couvrir tous les domaines et par sa vision limpide des problèmes. J'ai beaucoup apprécié mes visites à Bergen au sein d'un groupe formidable et je suis très heureuse qu'il ait pu venir à Paris à l'occasion de ma soutenance.

Claude Carlet a suivi, avec attention et gentillesse, ma recherche dès ses débuts. J'ai eu le plaisir de travailler régulièrement avec lui — ce qui a parfois donné lieu à de grands débats sur la notation de la transformée de Walsh. Je souhaite le remercier pour tout cela et pour avoir, après le jury de thèse, accepté de récidiver dans ce jury d'HDR.

Je dois aussi beaucoup à Thierry Berger, notamment pour m'avoir permis d'enseigner dans le DEA Cryptographie, Codage, Calcul. Je garde un souvenir ému (et même nostalgique) de ces visites hebdomadaires à Limoges, des cours de programmation entrecoupés de discussions sur le chiffrement à flot ou la recherche de mots de poids faible.

Je tiens également à remercier Claude Kirchner et Daniel Lazard d'avoir accepté de faire partie de ce jury.

Je souhaite aussi témoigner toute ma reconnaissance à Philippe Robert et Sophie Cluet, pour leurs encouragements et leurs précieux conseils.

Un grand merci à Paul Zimmermann et à Marion Videau pour leur relecture attentive — ils offrent une récompense à quiconque trouve une coquille.

Merci à tous ceux que j'ai eu le plaisir de côtoyer au cours de ces travaux, notamment à Jim Massey, Matt Robshaw, Henk van Tilborg, Grisha Kabatyanskiy, Bimal Roy, Subho

Maitra, Gohar Kyureghyan, Gregor Leander (et la bande de Bochum), Pascal Véron (et la bande de Toulon), Jean-Marc Couveignes et Thierry Hénocq (je confirme, malgré ce que croient les Parisiens, que Toulouse ne manque pas d'animation), Marc Rybowicz, Philippe Gaborit, Mathieu Ciet, François Morain, Isabelle de Lamberterie, Stéphanie Lacour, Françoise Banat-Berger, sans oublier mes co-designers de SOSEMANUK *et DECIM.*

Grâce à Claude Vouillarmet, j'ai le privilège de soutenir dans un lieu où résonnèrent les mots de Camille Desmoulins, ce qui est tout de même beaucoup plus agréable que les poussières d'amiante de Jussieu.

Lucile et Thomas m'ont offert un séjour digne d'un cinq étoiles au Portugal pour rédiger ce mémoire. Ils m'ont excellement nourrie, logée, blanchie, et offert en prime une expertise bien utile en probabilités. Je les en remercie d'autant plus qu'il me sera bien difficile de leur rendre cette dernière au centuple.

La bonne humeur du projet CODES m'est chaque jour précieuse. Un immense merci à tous ses membres (par ordre d'apparition, et non d'âge) : Daniel Augot, Nicolas Sendrier, Christelle Guiziou, Francoise Lévy-dit-Véhel, Caroline Fontaine, Jean-Pierre Tillich, Pierre Loidreau, Eric Filiol, Ayoub Otmani, Hervé Alavoine, Christine Pourcelot, Fabien Galand, Cédric Tavernier, Carmen Nedeloaia, Marine Minier, Matthieu Finiasz, Raghav Bhaskar, Ludovic Perret, Avishek Adhikari, Yann Laigle-Chapuy, Frédéric Didier, Michael Quisquater, Andrea Röck...

Merci enfin, du fond du cœur, à ceux qui ont donné à ces années de travail une valeur à mes yeux inestimable : Vincent Bétis, Lionel Brunnengreber (dits Boule et Bill), Michaël Trabbia, Fatou Diop, Philippe Quanty, Domitille Heitzler, Maria Naya Plasencia, Cédric Lauradoux, Mathieu Cluzeau et Marion Videau.

Overview

Basic cryptographic algorithms split into two families: symmetric algorithms, otherwise known as secret-key algorithms, which normally require a key to be shared and simultaneously kept secret within a restricted group, and public-key algorithms where the private key is almost never shared. From outside, this may give the impression that symmetric techniques become obsolete after the invention of public-key cryptography in the mid 1970's. However, symmetric techniques are still widely used because they are the only ones that can achieve high-speed or low-cost encryption. Today, we find symmetric algorithms in GSM mobile phones, in credit cards, in WLAN connections... Therefore, symmetric cryptology is a very active research area which is stimulated by a pressing industrial demand for low-cost implementations (in terms of power consumption, gate count...). These extremely restricting implementation requirements are crucial when designing secure symmetric primitives. Actually, these constraints seem quite incompatible with the rather complex mathematical tools needed for constructing a provably secure system.

Research in symmetric cryptography is obviously characterized by a sequence of defenses and attacks — go to sleep as a cryptographer and you will wake up as a cryptanalyst! as explained by Marc Girault. Our work aims at considering this continual ping-pong game in a formal way such that it does not reduce to an infinite repetition of trails and errors. Each new dedicated attack against a given cryptosystem must actually be formalized, its scope must be analyzed and the structural properties which make it feasible must be highlighted. This approach is the only one which can lead to new design criteria and to the constructions of building blocks which guarantee to a provable resistance to the known attacks. However, such an analysis yields a practical system only if it includes the implementation requirements arising from the applications. Go to sleep as a cryptographer and you will wake up as a programmer or as a hardware specialist! Therefore, our work considers all aspects of the field, from the practical ones (new attacks, concrete specifications of new systems) to the most theoretical ones (study of the algebraic structure of underlying mathematical objects, definition of optimal objects). But, our purpose is to study these aspects not separately but as several sides of the same domain.

Our approach mainly relies on the idea that, in order to guarantee a provable resistance to the known attacks and to achieve extremely good performance, a symmetric cipher must use very particular building blocks, whose algebraic structures may introduce unintended weaknesses. This paradox is stressed by Lars Knudsen: *What is provably secure is probably not.* The AES Sbox provides a perfect illustration of this: it offers an optimal resistance against differential and linear attacks but the existence of many quadratic relations between its inputs and outputs may appear as a potential weakness. Moreover, Lars Knudsen's joke can be replaced by the more pessimistic idea that what is sufficiently fast is probably not secure.

Indeed, the previous paradox seems more general because all building blocks which have a low-cost realization are highly structured — since the average implementation complexity of a Boolean function is exponential. But, the structures that facilitate the implementation are usually the very same structures which can be exploited by the cryptanalyst.

In this context, it is very important to precisely identify all requirements for the different components of a symmetric cipher — regarding both the implementation and the security. It is actually essential to avoid purely intuitive or very general criteria which may introduce unnecessary constraints and may affect the performance. This issue is here investigated for some stream ciphers. Then, a theoretical study is required in order to exhibit some optimal building blocks with respect to the relevant criteria, and to analyze their properties. If these objects possess very special structures, a second step consists in determining whether such structures can be exploited by the cryptanalyst. If the conclusion is that the structures of these optimal objects are at the origin of a new attack, a more careful investigation is needed in order to isolate suboptimal objects which do not possess such structures. This is a rather difficult problem since the structures exploited by the cryptanalyst are usually the very same structures that are used by the mathematician to exhibit such components and by the implementor to obtain an efficient realization.

Our research work captures this conflict for both families of symmetric ciphers, stream ciphers and block ciphers. It includes new attacks and the search for new building blocks which ensure both a high resistance to the known attacks and a low implementation cost.

Summary of the main contributions

The manuscript is naturally split into two parts devoted to both types of symmetric ciphers. However, some generic tools involved in both contexts are presented in an introductory chapter. A list of notation can be found in Page 161, before the table of contents. Here, we briefly summarize our main contributions. However, we do not give any technical details on the results which have already been published — the reader may refer to the corresponding papers.

Chapter 1. This chapter briefly presents the statistical framework of most attacks which are considered here. Most notably, Lemma 1.1 recalls some results detailed by Baignères, Junod and Vaudenay [BJV04] on the number of samples required by the best distinguisher between two distributions which are close together. In particular, the number of samples needed for distinguishing a distribution $\mathcal{D}_0$ from the uniform distribution is proportional to

$$\frac{1}{\Delta^2(\mathcal{D}_0)} \text{ with } \Delta^2(\mathcal{D}_0) = |\mathcal{X}| \sum_{x \in \mathcal{X}} \varepsilon_x^2,$$

where $\mathcal{X}$ is the support of $\mathcal{D}_0$ and, for all $x \in \mathcal{X}$,

$$\Pr_{\mathcal{D}_0}[X = x] = \frac{1}{|\mathcal{X}|} + \varepsilon_x, \ \varepsilon_x \ll \frac{1}{|\mathcal{X}|}.$$

Corollary 1.3 then applies this result to the case where the considered distribution is the distribution $\mathcal{D}_f$ of $f(X)$ where f is an n-variable Boolean function (i.e., a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2$) and X is a random variable with uniform distribution over $\mathbf{F}_2^n$. In this case, the bias of $\mathcal{D}_f$ is given by

$$\Delta^2(\mathcal{D}_f) = \left(\frac{\mathcal{F}(f)}{2^n} \right)^2 ,$$

where the quantity $\mathcal{F}(f)$ is derived from the Hamming weight of f:

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

Corollary 1.5 provides a similar result where a vectorial function F from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$ is considered. The bias of $\mathcal{D}_F$ depends on the biases of all its *components* F_λ, which are the Boolean functions corresponding to the linear combinations of the coordinates of F, i.e., $F_\lambda(x) = \lambda \cdot F(x)$.

Section 1.3 recalls some classical definitions and properties of Boolean functions. The previously defined quantity $\mathcal{F}(f)$ is also used in the expression of the Walsh transform of f since the Walsh coefficient of f at point $a \in \mathbf{F}_2^n$ corresponds to

$$\mathcal{F}(f + \varphi_a) = \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+a \cdot x},$$

where φ_a denotes the n-variable linear function $x \mapsto a \cdot x$.

Stream ciphers

Chapter 2. This chapter is a short summary chapter on stream ciphers. It notably presents the general model of synchronous stream cipher and the related notation (see Figure 2.1 and Table 2.1). It recalls the basic security requirements derived from generic attacks and it also presents a general classification of synchronous stream ciphers into different families, depending on whether the next-state function is linear or not.

Chapter 3. In this chapter, we concentrate on some distinguishing attacks against keystreams produced by filtering a shift register (with linear or nonlinear feedback):

$$s_t = f(x_{t+\gamma_1}, x_{t+\gamma_2}, \ldots, x_{t+\gamma_n}) \ ,$$

where $(x_t, \ldots, x_{t+L-1})$ is the content of the register at time t.

We mainly focus on some existing attacks but we provide a careful analysis of their complexities in order to exhibit the corresponding security requirements on the filtering function. Section 3.1 first investigates distinguishers for the augmented function: Proposition 3.1 gives the exact expression of the number of samples required by the best distinguisher between the uniform distribution and the distribution of the pairs $(s_t, s_{t+\tau})$ of keystream digits produced at time t and $t + \tau$ for a given value of τ. The data complexity of the distinguisher depends on some Walsh coefficients of the filtering function:

$$N = \frac{1}{\Delta^2} \text{ with } \Delta^2 = \frac{1}{2^{4n}} \left(\sum_{u \in \mathbf{F}_2^\ell} \mathcal{F}(f + \varphi_{(u,0)_I}) \mathcal{F}(f + \varphi_{(u,0)_J}) \right)^2 ,$$

where both sets I and J are defined by

$$I = \{i, 1 \le i \le n, \ \exists j, \gamma_i - \gamma_j = \tau\} \text{ and } J = \{j, 1 \le j \le n, \ \exists i, \gamma_i - \gamma_j = \tau\},$$

$\ell = |I| = |J|$ and $x = (a,b)_I$ denotes the vector defined by $(x_i, i \in I) = a$ and $(x_i, i \notin I) = b$.

Most notably, we deduce that, for any $1 \le \tau \le \gamma_n - \gamma_1$, $(s_t, s_{t+\tau})$ is uniformly distributed if one of the following conditions holds:

- $\mathcal{F}(f + \varphi_u) = 0$ for all $u \in \langle e_1, \ldots, e_{n-1} \rangle$ such that $wt(u) \le \ell$;

- $\mathcal{F}(f+\varphi_u)=0$ for all $u \in \langle e_2,\ldots,e_n\rangle$ such that $wt(u) \leq \ell$,

where $(e_1,\ldots,e_n)$ is the canonical basis of $\mathbf{F}_2^n$.

Conversely, if $\mathcal{F}(f+\varphi_{e_1}) \neq 0$ (resp. if $\mathcal{F}(f+\varphi_{e_n}) \neq 0$) and there exists $i \in \{2,\ldots,n-1\}$ such that $\mathcal{F}(f+\varphi_{e_i}) \neq 0$, then the distribution of $(s_t,s_{t+\tau})$ is not uniform for $\tau=\gamma_i-\gamma_1$ (resp. for $\tau=\gamma_n-\gamma_i$). It is worth noticing that, contrary to what is sometimes believed, f does not need to be ℓ-resilient for resisting this attack. We also obtain a similar formula (Page 23) for the correlations between $s_t+s_{t+\tau}$ and any linear combination of the internal state bits.

More generally, if we consider the distribution of a $(\gamma_n-\gamma_1+1)$-tuple of keystream digits, it is known [Gol96] that this distribution is uniform when the filtering function f has one the following forms:

$$f(x_1,\ldots,x_n)=x_1+g(x_2,\ldots,x_n)$$

or

$$f(x_1,\ldots,x_n)=g(x_1,\ldots,x_{n-1})+x_n \ .$$

Here, we prove that this condition is actually a sufficient and necessary condition (Prop. 3.2) — the converse part was mentioned as an open problem in [Gol96].

Section 3.2 then focuses on filtered LFSRs, especially on distinguishing attacks based on sparse parity-check equations. We especially discuss the complexity of the distinguisher described in [EJ05, MH04]. If one parity-check equation of weight w is used, this complexity is derived from the (normalized) w-th power moment of the Walsh spectrum of the filtering function (Prop. 3.5)

$$\Delta_w = 2^{-wn} \sum_{\lambda \in \mathbf{F}_2^n} \mathcal{F}^w(f+\varphi_\lambda) \ .$$

We first focus on the case $w=4$ and we provide some bounds on the 4-th power moment of the Walsh spectrum of a Boolean function — this quantity is proportional to the so-called sum-of-square indicator. More generally, Proposition 3.13 exhibits some new upper bounds on the w-th power moment of the Walsh spectrum when w is any even integer:

$$\Delta_{2p+2} \leq \Delta_{2p} \left[\frac{\mathcal{L}(f)}{2^n}\right]^2$$

with equality if and only if f is a plateaued function. In particular, we have

$$\Delta_{2p} \leq \left[\frac{\mathcal{L}(f)}{2^n}\right]^{2(p-1)} \quad \text{where } \mathcal{L}(f)=\max_{a \in \mathbf{F}_2^n} |\mathcal{F}(f+\varphi_a)| \ .$$

Theorem 3.14 provides some general lower bounds on the same quantity, especially

$$\Delta_{2p} \geq 2^{n(1-p)}$$

with equality if and only if f is bent. By combining both results, we derive Corollary 3.15 which gives a lower and an upper bound on the complexity of the distinguisher based on a parity-check equation of weight $2p$. When the complexity of the search for the parity-check relation is not included, the optimal weight for the relation is roughly $\sqrt{\frac{L}{n-1}}$ where L is the LFSR length and n is the number of variables of the filtering function. This leads to an attack with complexity $2^{\sqrt{L(n-1)}}$.

Section 3.2.3 gives similar results for the case of a combination generator. Section 3.2.4 then presents another attack based on low-weight parity-check equations, which applies in

the very different context of "reverse-engineering". In this case, the attacker wants to recover the unknown specifications of a combination generator (i.e., the feedback polynomials of all constituent LFSRs and the filtering function) from the knowledge of some keystream bits only. This attack relies on the same basic principle. It is detailed in a paper presented at FSE 2000 [CF01].

Finally, Section 3.2.5 investigates the complexity of the distinguishing attack presented in [EJ05, MH04] against a filter generator where the attacker uses m parity-check relations

$$x_t + x_{t+\tau_{1,i}} + \ldots + x_{t+\tau_{w-1,i}}, \forall t \geq 0, 1 \leq i \leq m \ .$$

We show that considering the distribution of the m-tuple

$$\left(\sum_{j=0}^{w-1} s_{t+\tau_{j,1}}, \ldots, \sum_{j=0}^{w-1} s_{t+\tau_{j,m}} \right)$$

does not provide the best distinguishing attack. The reason is that all parity-check relations are not independent since they have one term in common. Instead, the distribution of the $(m+1)$-tuple

$$\left(s_t, \sum_{j=1}^{w-1} s_{t+\tau_{j,1}}, \ldots, \sum_{j=1}^{w-1} s_{t+\tau_{j,m}} \right)$$

must be considered. However, the general expression for this distribution is rather complicated. It is detailed in Proposition 3.23 for the case of $m = 2$ equations only. We also provide a toy example where the complexity of the best distinguisher with 2 equations differs from the complexity of the simpler but not optimal distinguisher.

Chapter 4. This chapter is devoted to correlation attacks. These divide-and-conquer attacks originally apply when the internal state of a keystream generator can be decomposed into several parts which are updated separately (see Figure 4.1). In this case, the attacker tries to recover the first ℓ-bit part of the initial state independently from the other bits. This can be done if there exists an ℓ-variable Boolean function g which is correlated to the filtering function f. In this case, an exhaustive search on the targeted part of the initial state can be made based on a distinguisher between the uniform distribution and the distribution of $s_t + \sigma_t$ where $(\sigma_t)_{t\geq 0}$ is the sequence produced by filtering the targeted part of the internal state by g. Since g is chosen by the attacker, it raises the problem of finding the best approximation of a function by a function of fewer variables x_i,$i \in I$. This issue is investigated in Section 4.1.2 which details some results from [Can02]. Proposition 4.1 (independently found by Zhang [Zha00]) shows that the best ℓ-variable approximation g of a function f is given by the biases of the restrictions of f to all subspaces $x + \langle e_i, i \notin I \rangle$. Moreover, the maximum value of the correlation between f and g is upper bounded by a quantity depending on the non-linearity of f as shown in Corollary 4.2:

$$\max_g \mathcal{F}(f+g) \leq \left(\sum_{x \in V} \mathcal{F}^2(f+\varphi_x) \right)^{\frac{1}{2}} \ ,$$

where $V = \langle e_i, i \in I \rangle$. Then, when such an approximation of the filtering function exists, a correlation attack can be performed. As pointed out by Meier and Staffelbach, this attack can

be seen as a decoding problem. Therefore, the rest of this chapter presents some decoding algorithms that can be used in this context, when the underlying code is linear. The first attack described in Section 4.3.2 and published in [CT00] is based on an iterative decoding algorithm as Meier-Staffelbach's original attack. When parity-check equations of weight w are used, the data and time complexities are given by

$$D = \left(\frac{1}{2\varepsilon}\right)^{\frac{2(w-2)}{w-1}} 2^{\frac{\ell}{w-1}} \text{ and } T = \left(\frac{1}{2\varepsilon}\right)^{\frac{2w(w-2)}{w-1}} 2^{\frac{\ell}{w-1}},$$

where the complexity of the preprocessing step (i.e., of the search for the parity-check equations) is here excluded. Section 4.4 finally focuses on the particular case of LFSR-based generators. We first determine the complexity of the previous attack for the combination generator, using the fact that the best approximation of a t-resilient filtering function by a $(t+1)$-variable function is affine — this can be directly deduced from the previous results on the approximation of a function by a function of fewer variables. Then, we focus on the the filter generator. In this case, we detail an attack presented in [CF02] which exploits not only one but all linear approximations of the filtering function together — this is a generalization of an attack due to Jönsson and Johansson [JJ02]. If the filtering function has B nonzero Walsh coefficients, then the attack consists in decoding an $[NB,L]$-code where N is the keystream length and L the size of the LFSR. The received word corresponds to B blocks derived from the keystream, each of them is related to a particular linear approximation. The decoding step is then performed by an algorithm proposed by Chepyshov, Johansson and Smeets [CJS00]. Here, the key-point in the complexity analysis is that the involved channel is not a stationary channel anymore since the nonzero Walsh coefficients may take different values. However, we provide in Theorem 4.4 a lower and an upper bound on its capacity where linear combinations of w columns of the generator matrix are used in Chepyzhov-Johansson-Smeets algorithm — the lower bound provides a good estimate of the capacity when f has a high nonlinearity. Therefore, the data and time complexity of our attack are given by

$$D = \mathcal{O}(2^{\frac{L-k}{w}}) \text{ and } T = \mathcal{O}(k2^k B^w) \ ,$$

where k and w are some parameters of the algorithm. An interesting point here is that the data complexity does not depend on the filtering function. But, the time complexity of the decoding step increases with the number of nonzero Walsh coefficients of the filtering function.

Chapter 5. Chapter 5 essentially corresponds to a paper published in the proceedings of WCC 2005 [Can06] on algebraic attacks. It aims at evaluating the complexity of these recent attacks on LFSR-based stream ciphers in order to complete the list of relevant security requirements for the filtering function in such keystream generators. This survey paper especially discusses two important issues. The first one is the commonly used hypothesis on the independence of the relations $g \circ \Phi^t$ obtained at different times t for linearly independent annihilators g, where Φ denotes the next-state function of the generator. This hypothesis may be important since it determines the number of linearly independent equations available to the attacker when the algebraic system is solved by linearization. In the case where Φ corresponds to the next-state function of an LFSR of length n, we show that, for any given annihilator g, all functions $g \circ \Phi^t$ for $0 \leq t < 2^L$ are different if $2^n - 1$ is a prime. When $(2^n - 1)$ is a composite number, one can always find some filtering functions for which this property does not hold.

But, we can show that, even if algebraic attacks do not apply in this case, this property can be exploited for mounting an efficient distinguishing attack. However, the general problem of determining the rank of the obtained linear equations remains open. The second part of the paper focuses on some general properties related to the algebraic immunity of Boolean functions. Most notably, we prove in Theorem 5.6 that, for any odd n, an n-variable function f has maximal algebraic-immunity $\frac{n+1}{2}$ if and only if its annihilator ideal $AN(f)$ does not contain any nonzero function of degree strictly less than $\frac{n+1}{2}$. In other words, if $\deg AN(F) = \frac{n+1}{2}$, then this property also holds for $AN(1+f)$.

Chapter 6. Taking into account all previous security criteria, Chapter 6 is dedicated to the search for appropriate filtering functions. These functions must have a low implementation complexity (in terms of gate count); they must be balanced, have a high degree, a high nonlinearity and the power moments of their Walsh spectrum must be as low as possible. Section 6.1 briefly recalls the known results on the highest possible nonlinearity for a Boolean function and it also gives all possible Walsh spectra for a function f with $\mathcal{L}(f) = 2^{\frac{n+1}{2}}$, n odd and $n \leq 9$ — these results are derived from an extensive study presented in [Can01b]. When n is even, the highest possible nonlinearity for a balanced function is not known in general. Some recursive constructions can be found in the literature but they are clearly inappropriate in most implementations — except for some software environments where the function is implemented by a lookup table. Therefore, we investigate in Section 6.2 another technique for constructing highly nonlinear balanced functions. Our method consists in fixing a (small) number of the inputs of a function which has a low implementation cost but which may present some cryptographic weaknesses. The resulting function (which corresponds to the *restriction* of another function to a given subspace) has then a similar implementation complexity but better cryptographic properties. This technique is extensively studied in [CCCF00, CCCF01, CC03]. For instance, we can start from a bent function, which cannot be used directly as a filtering function since it is not balanced. If one input variable is fixed, a highly nonlinear plateaued function is obtained (Proposition 6.5). Similarly, balanced functions depending on an odd number of variables can be constructing by fixing two inputs if the second-order derivative of the dual function with respect to the corresponding vectors is not the all-one function (Theorem 6.7). A related construction consists in adding to the bent function the indicator of a subspace of codimension 2 corresponding the case where the second-order derivative of the dual function differs from the null function (Theorem 6.8). Good candidates for low-cost bent functions which can be used in this context are the components of some power functions over a finite field (the so-called monomial bent functions).We performed an exhaustive search for all exponents s for which there exists $\lambda \in \mathbf{F}_{2^n}$ such that $S_\lambda : x \mapsto \mathrm{Tr}(\lambda x^s)$ is an n-variable bent function, up to $n = 20$. These simulation results have exhibited two new families of monomial bent functions described in Section 6.3 (see e.g. [CCK06]).

Section 6.3 also investigates the possibility to directly use the components of some power functions as a filtering function. This requires that the exponent s is coprime with $2^n - 1$, otherwise the function is not balanced. In this particular case, we give a lower bound on the third and fourth power moments of the Walsh spectrum of such functions S_λ (Prop. 6.11 and 6.12): when n is odd,

$$\sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^3(S_\lambda + \varphi_\mu) \geq 2^{2n+1} \text{ and } \sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^4(S_\lambda + \varphi_\mu) \geq 2^{3n+1}$$

with equality when $\mathcal{L}(S_\lambda) = 2^{\frac{n+1}{2}}$. When n is even,

$$\sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^3(S_\lambda + \varphi_\mu) \geq 2^{2n+2} \text{ and } \sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^4(S_\lambda + \varphi_\mu) \geq 2^{3n+1} + 2^{2n+3} \ .$$

Note that the power function used in the stream cipher proposal SFINKS achieves both of these lower bounds. The other family of low-cost Boolean function we examine here is the class of symmetric functions. This class seems quite appropriate for a hardware implementation since it can be realized by a circuit whose number of gates is linear in the number of input variables. A detailed study of the cryptographic properties of these functions can be found in [CV05]. Most notably, we point out that the balancedness requirement is a very restrictive property for a symmetric function: Theorem 6.15 gives the algebraic normal forms of all[a] balanced symmetric functions of degree at most 7. Since all maximally nonlinear symmetric functions are quadratic (which is an important weakness), we also focus on symmetric functions with suboptimal nonlinearity (Proposition 6.17) and we provide the algebraic normal forms of all n-variable symmetric functions f with

$$\mathcal{L}(f) = 2^{\lfloor \frac{n+1}{2} \rfloor} + 4 \ .$$

However, these functions cannot be used directly since they have a low algebraic immunity [BP05][b].

Block ciphers

The second part of the manuscript is devoted to S(ubstitution)-boxes for iterated block ciphers. It has its origin in a long-standing joint work with Pascale Charpin, Hans Dobbertin and with Marion Videau on almost bent functions.

Chapter 7. The first chapter of this second part is an introductory chapter on statistical attacks against iterated block ciphers. The basic principle of these attacks is described in Table 7.1 and in Figure 7.2. We mainly focus on the two most famous attacks in this class: differential cryptanalysis and linear cryptanalysis. The resistance to each of these attacks highly depends on some properties of the nonlinear part of the cipher, i.e., on the so-called Sboxes. Most notably, the resistance to differential cryptanalysis can be quantified by the following parameter

$$\delta(F) = \max_{a \neq 0, b} |\{x \in \mathbf{F}_2^n, \ F(x+a) + F(x) = b\}| \ .$$

It is known that, for any function F from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$, $\delta(F) \geq 2^{n-m}$. Moreover, if $m = n$, we have $\delta(F) \geq 2$ with equality for *Almost Perfect Nonlinear (APN) functions*.

Similarly, the resistance to linear cryptanalysis is estimated by the nonlinearity of the substitution function, i.e., by the minimal nonlinearity of its components:

$$\mathcal{NL}(F) = 2^{n-1} - \frac{1}{2}\mathcal{L}(F) \text{ where } \mathcal{L}(F) = \max_{a \in \mathbf{F}_2^n, b \in (\mathbf{F}_2^m)^*} |\mathcal{F}(F_b + \varphi_a)| \ .$$

a. Note that this result holds for any number of input variables.

b. It should be mentioned that most of the results on the algebraic immunity of symmetric functions published in [BP05] are wrong. For instance, the function $f = 1 + x_1 \dots x_n$ provides a counter-example of Theorem 1 since it has a unique annihilator, $(1+f)$, which does not belong to the claimed family. Theorem 2 which gives a bound on the number of elements in this subset is also wrong (e.g. for $n = 10$). Moreover, most constructions of infinite families of symmetric functions with optimal algebraic immunity are not valid.

Here, we have that $\mathcal{L}(F) \geq 2^{\frac{n}{2}}$ where the bound is tight for the bent functions. Moreover, if the number of outputs of the function equals the number of its inputs, we have

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

with equality for *Almost Bent (AB) functions* — note that AB functions exist for odd n only.

Chapter 8. This chapter is devoted to the construction of optimal (and suboptimal) functions with respect to both previous criteria. In order to clarify the terminology and the context, Section 8.1 first recalls the link between the APN and AB properties and other optimal objects which appear in different applications, like in coding theory and in the theory of sequences. Then, Section 8.2 investigates the link between the security criteria corresponding to differential and to linear attacks. The relationship between both properties can be exhibited through the values of the third and fourth power moments of the Walsh spectrum of a vectorial function $F : \mathbf{F}_2^n \to \mathbf{F}_2^n$ (Theorem 8.3). Actually, for $F(0) = 0$, we have

$$\begin{aligned} \sum_{\lambda \in \mathbf{F}_2^n \setminus \{0\}} \sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^3(F_\lambda + \varphi_\mu) &= 2^{2n+1}(2^n - 1) + 2^{2n} D_0(F) \\ \sum_{\lambda \in \mathbf{F}_2^n \setminus \{0\}} \sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^4(F_\lambda + \varphi_\mu) &= 2^{3n+1}(2^n - 1) + 2^{2n} D(F) \end{aligned}$$

where

$$\begin{aligned} D(F) &= |(a,b,x),\ a,b \in \mathbf{F}_2^n \setminus \{0\},\, x \in \mathbf{F}_2^n,\, a \neq b,\ D_a D_b F(x) = 0\}| \\ D_0(F) &= |(a,b),\ a,b \in \mathbf{F}_2^n \setminus \{0\},\, a \neq b,\ D_a D_b F(0) = 0\}| \ . \end{aligned}$$

Note that this theorem provides a new characterization of APN functions [BCCLC06] since those functions correspond to the case $D(F) = 0$. Moreover, these expressions can be simplified in the case of power mappings as shown in Corollary 8.4. Then, Theorem 8.6 shows that the previous quantity $D(F)$ can be bounded by an expression involving the nonlinearity of F and the minimal nonzero magnitude of its Walsh coefficient. As a corollary of the first item of Theorem 8.6, we recover a result proved by Chabaud and Vaudenay [CV95] which states that F is AB implies that F is APN and plateaued. The second item provides a kind of reciprocal statement since it enables us to exhibit a necessary and sufficient condition for an APN function to be almost bent (Corollary 8.7, see also [CCD00b] and [CCD99]): $F : \mathbf{F}_2^n \to \mathbf{F}_2^n$, n odd, is AB if and only if it is APN and all its Walsh coefficients are divisible by $2^{\frac{n+1}{2}}$. This condition has a nice formulation when F is a power function since the divisibility of its Walsh spectrum can be reduced to a purely combinatorial problem due to McEliece theorem (see Prop. 8.9). Actually, the divisibility of the Walsh spectrum of $x \mapsto x^s$ exactly corresponds to the divisibility of the weights of a cyclic code. Using this new condition, we were able to prove [CCD00a] a 30-year old conjecture due to Welch, which states that the power function $x \mapsto x^s$ over $\mathbf{F}_{2^n}$ with

$$s = 2^{\frac{n-1}{2}} + 3$$

is almost bent. Section 8.3 gives some more general results on all exponents of the form $2^{\frac{n-1}{2}} + 2^i - 1$, including Welch exponent; it also provides some bounds on the divisibility of the Walsh spectrum of a power function over $\mathbf{F}_2^n$ when n is not a prime, which can be efficiently

used for proving that some exponents are not AB — much more detailed results can be found in [CCD00a].

The case of functions depending on an even number of variables is still open as explained in Sections 8.4 and 8.5. Almost bent functions do not exist in this case and the lowest possible value for $\mathcal{L}(F)$ satisfies

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

However, the lowest known value corresponds to the upper bound and it is conjectured that this value is the optimal one for power functions. Here, we prove this conjecture for power functions which are not permutations, i.e., $x \mapsto x^s$ with $\gcd(s,2^n-1) > 1$. Another open problem is the existence of APN permutations for even n. Since none such function is known in the even case, practical applications have to use suboptimal functions, i.e., functions for which $\delta(F)$ exceeds 2 but is still low. We here characterize the third and fourth power moments of the Walsh spectrum of power permutations S with $\delta(S) = 4$ (Prop. 8.17). We also give the list of all exponents corresponding to a power permutation with $\delta(S) = 4$ and $\delta(S) = 6$. Other criteria such as the degree and the nonlinearity of these functions are also provided up to 17 variables. As an example, we prove that, for any n, the power function $S : x \mapsto x^7$ over $\mathbf{F}_{2^n}$ satisfies $\delta(S) = 6$.

Chapter 9. This chapter presents some results published at EUROCRYPT 2002 [CV02] on higher-order differential attacks against some block ciphers. We mainly show that a high divisibility of the Walsh spectrum of the S-box (i.e., the fact that all Walsh coefficients are divisible by a high power of 2) introduces some vulnerabilities regarding higher-order differential cryptanalysis, whereas this property characterizes the Sboxes with a very high nonlinearity. The extremal situation corresponds to the almost bent functions which provide the best resistance to linear and differential attacks, but possess the highest possible divisibility as proved in the previous chapter.

The key-idea in this work consists in improving the trivial bound on the degree of the composition of two functions

$$\deg(F' \circ F) \leq \deg(F') \deg(F) .$$

Since any coordinate of $F' \circ F$ corresponds to a polynomial in the coordinates of F, we establish a relationship between the Walsh spectrum of the product of some Boolean functions and the Walsh spectrum of their sum (Proposition 9.3). This implies that the divisibility of any product of the coordinates of F can be deduced from the divisibility of F (Theorem 9.4), leading to the following result: if all Walsh coefficients of F over $\mathbf{F}_2^n$ are divisible by 2^ℓ, then

$$\deg(F' \circ F) \leq n - \ell + \deg(F').$$

This result exhibits the origin of the higher-order differential attack presented by Tanaka, Hisamatsu and Kaneko [THK99] against an alleged version of MISTY1[c]. This attack actually comes from the use of an Sbox whose Walsh coefficients are divisible by $2^{\frac{n+1}{2}}$. This explains and generalizes the fact that the attack can be always be mounted even if the Sbox S_7 is replaced by any other almost bent power function of degree 3 [BF00]. Since a high divisibility of the Walsh coefficients may have some unintended consequences, Section 9.4 finally provides some properties of all Sboxes having a very low divisibility.

c. This alleged version is the one which has been proved secure against linear and differential attacks.

Chapter 10. The last chapter briefly focuses on other design criteria for Sboxes, especially on their algebraic-immunity and on the number of quadratic relations between their inputs and outputs. Most notably, we show that any power permutation over $\mathbf{F}_{2^8}$ possesses some quadratic relations while this is not the case in general for permutations over $\mathbf{F}_{2^8}$ — the only power function over $\mathbf{F}_{2^8}$ with algebraic-immunity 3 is $x \mapsto x^{27}$ which is not a permutation. But, there exists power permutations over $\mathbf{F}_{2^n}$ with algebraic-immunity at least 3 for $n \geq 9$. All relevant properties of power permutations over $\mathbf{F}_{2^n}$ for $n \in \{8,9,10\}$ are summarized in Tables 10.1, 10.2 and 10.3. For each $S : x \mapsto x^s$, we give the degree, the value of $\delta(S)$ which measures the resistance to differential attacks, the nonlinearity, the complete Walsh spectrum, and the number of quadratic relations between the inputs and outputs of the function. We finally discuss some issues related to the use of a power function in an Sbox and to the construction of the substitution layer of an iterated block cipher by concatenation of several small Sboxes. The underlying open question is to determine whether such constructions which are quite appropriate in terms of implementation complexity do not introduce weaknesses in the cipher.

Présentation générale

Les algorithmes cryptographiques se répartissent en deux grandes familles : les algorithmes *symétriques*, ou *à clef secrète*, qui nécessitent le partage d'un secret par les deux protagonistes, et les algorithmes *à clef publique* pour lesquels le secret reste connu d'un seul des deux acteurs. De l'extérieur, cette classification pourrait laisser penser que les techniques symétriques seraient devenues obsolètes dès le milieu des années 70, avec l'apparition de la cryptographie à clef publique. Elles sont pourtant très largement répandues car elles sont les seules qui atteignent les débits de chiffrement requis par la plupart des applications et qui permettent une mise en œuvre par des circuits de taille raisonnable. Ainsi, ce sont des algorithmes à clef secrète qui assurent la confidentialité des échanges dans les téléphones portables, les réseaux sans fil... La cryptographie symétrique est donc un domaine de recherche particulièrement actif, constamment sollicité par une demande industrielle pressante d'implémentations à faible coût (en termes d'encombrement, de consommation...). Ces exigences d'implémentation extrêmement contraignantes sont au cœur même de sa fragilité et paraissent difficilement compatibles avec la complexité des outils mathématiques nécessaires à la construction d'un système dont la sécurité ferait l'objet d'une preuve formelle.

La recherche en cryptographie symétrique se caractérise donc naturellement par l'enchaînement de phases de défense et d'attaque — on s'endort cryptographe et se réveille cryptanalyste, selon le bon mot de Marc Girault. L'ensemble de mes travaux vise à aborder cette incessante partie de ping-pong de manière formelle, afin qu'elle ne se résume pas à une simple succession d'essais-erreurs. Il est en effet indispensable de formaliser toute nouvelle attaque dédiée à un système donné, d'en analyser la portée et de mettre en évidence les propriétés structurelles qui la rendent opérationnelle. Seule une telle approche peut conduire à de nouveaux critères de conception et à la construction d'objets qui permettent de lui résister de manière certaine. Ma recherche s'inscrit donc dans un enchaînement, souvent décrit par la métaphore de l'obus et de la cuirasse, qui débute par l'élaboration d'une nouvelle attaque sur un algorithme donné, suivi par sa formalisation, puis par la recherche d'objets permettant de lui résister, et qui aboutit enfin à la conception d'un nouveau système de chiffrement dont on peut démontrer qu'il résiste à ce type d'attaques. Toutefois, cette analyse n'aboutit à la définition concrète d'un nouveau système que si elle tient compte des contraintes d'implémentation inhérentes aux applications. Il faut donc aussi s'endormir mathématicien et se réveiller programmeur ou électronicien. C'est dans cette perspective que je m'efforce d'aborder l'ensemble des aspects du domaine, des plus pratiques (élaboration de nouvelles attaques, conception concrète de nouveaux systèmes) aux plus théoriques (étude de la structure algébrique des différents objets utilisés, définition d'objets optimaux), en les imbriquant étroitement.

Mon approche repose essentiellement sur l'idée que, pour résister de manière certaine aux cryptanalyses connues et pour atteindre de bonnes performances, un chiffrement symé-

trique doit utiliser des objets aux propriétés exceptionnelles, dont la structure algébrique forte ouvre paradoxalement une brèche exploitable dans une nouvelle attaque. Paradoxe résumé sous forme de boutade par Lars Knudsen : *What is provably secure is probably not*, et magistralement illustré par la fonction de substitution de l'AES qui offre la meilleure résistance possible aux attaques classiques mais pour laquelle l'existence de relations quadratiques liant ses entrées et ses sorties peut représenter une faiblesse potentielle. Par ailleurs, l'assertion de Lars Knudsen pourrait être remplacée par l'idée, peut-être plus pessimiste encore, que tout système suffisamment rapide n'est probablement pas sûr. Ce constat a d'ailleurs été explicité par Jim Massey dans le contexte particulier des chiffrements par blocs itératifs : à la remarque de Luke O'Connor selon laquelle *Most ciphers are secure after sufficiently many rounds*, il a objecté *Most ciphers are too slow after sufficiently many rounds.* Ce paradoxe me semble en fait plus général car les objets qui permettent une réalisation à faible coût sont nécessairement très structurés — sinon, on se heurte au comportement exponentiel de la complexité d'implémentation en moyenne d'une fonction booléenne.

Dans ce contexte, il est donc essentiel d'énoncer avec précision l'ensemble des propriétés requises — au regard des contraintes de sécurité et de mise en œuvre — pour les différentes composantes d'un chiffrement symétrique, et d'écarter certains critères trop réducteurs ou purement intuitifs qui font peser des contraintes inutiles et nuisent aux performances. C'est à cette tâche que je me suis attelée dans le cas de certains chiffrements à flot. Ensuite, il convient de mener une étude théorique afin de mettre en évidence des objets optimaux vis-à-vis des critères pertinents et de caractériser leurs propriétés. Dans le cas où ces objets présentent des particularités structurelles extrêmement fortes, une seconde étape consiste à déterminer s'il est possible de les exploiter dans une attaque. Enfin, si l'on conclut que ces objets optimaux sont à l'origine de nouvelles failles de sécurité, il est nécessaire de mener une recherche plus approfondie en quête d'objets sous-optimaux mais moins structurés. Ce travail est souvent ardu, puisque la structure optimale exploitée par le cryptanalyste est généralement aussi celle qui est l'origine de sa découverte et de la possibilité d'une réalisation efficace.

Mes travaux déclinent donc cette idée pour les deux familles d'algorithmes symétriques, les chiffrements par blocs et les chiffrements à flot. Ils portent à la fois sur de nouvelles attaques et sur la recherche de briques de base qui offrent une excellente résistance aux attaques et un faible coût de mise en œuvre.

Contributions et organisation du document

Ce document est découpé naturellement en deux parties, consacrées aux deux grands types d'algorithmes de chiffrement symétriques. Toutefois, un certain nombre d'outils interviennent dans les deux contextes applicatifs. J'ai donc choisi de les regrouper au sein d'un chapitre liminaire qui rappelle rapidement le cadre statistique de la plupart des attaques que je vais étudier ; il présente également certains paramètres qui interviendront par la suite pour quantifier les qualités cryptographiques des fonctions utilisées dans les différents algorithmes.

Chiffrements à flot. Les algorithmes de chiffrement à flot répondent à des contraintes pratiques très fortes puisqu'ils visent des débits extrêmement élevés en logiciel ou des réalisations matérielles à faible coût. Un des problèmes sous-jacents est le choix de la fonction de filtrage, c'est-à-dire de la fonction qui extrait à chaque instant un ou plusieurs bits d'information de l'état interne du générateur pseudo-aléatoire pour produire la suite chiffrante,

choix particulièrement crucial quand l'état interne évolue de façon linéaire. Afin de concevoir cette fonction de manière pertinente, il convient de mettre en lumière les propriétés qui déterminent la complexité des différentes attaques que l'on peut mener sur un tel chiffrement. À cette fin, le chapitre 3 présente une analyse détaillée de la complexité de certaines attaques par distingueur sur les générateurs pseudo-aléatoires classiques. Nous verrons alors que la résistance à ces attaques dépend fortement de la non-linéarité de la fonction de filtrage, mais qu'elle fait aussi intervenir plus précisément les différents moments de son spectre de Walsh (ou de Fourier) — et pas uniquement sa valeur maximale. Par ailleurs, l'une de ces attaques, fondées sur l'existence d'équations de parité de poids faible, permet également dans certains cas de cryptanalyser le système même lorsque certaines de ses spécifications ne sont pas connues, comme nous l'avons montré lors d'un travail commun avec E. Filiol présenté à FSE 2000 [CF01] — les techniques développées ici peuvent aussi être exploitées dans un tout autre contexte, celui de la reconstitution des spécifications d'un schéma de codage à partir de messages interceptés. Le chapitre 4 présente ensuite deux nouvelles attaques par corrélation rapides sur les générateurs pseudo-aléatoires à base de registres à décalage à rétroaction linéaire. La première, qui repose sur un algorithme de décodage itératif, a été présentée au colloque EUROCRYPT 2000 [CT00] ; la seconde [CF02], restreinte aux registres filtrés, met notamment en lumière l'influence de la fonction de filtrage dans ce contexte. L'inventaire des critères de sécurité serait clairement incomplet s'il n'incluait pas la résistance aux attaques algébriques, objet du chapitre 5 qui reprend essentiellement l'exposé invité que j'ai donné au colloque WCC 2005 [Can06]. Enfin, le chapitre 6 présente plusieurs voies de recherche que j'ai explorées dans le but d'identifier des fonctions de filtrage qui résistent aux attaques précédentes et, élément essentiel, qui puissent être réalisées par un circuit de taille raisonnable. Après une étude exhaustive des fonctions ayant un petit nombre de variables, je détaille plusieurs constructions peu onéreuses appropriées dans le cas d'un nombre de variables élevé — présentant chacune des avantages et inconvénients — en particulier l'utilisation de composantes de permutations puissances, de restrictions de fonctions courbes résultant de fonctions puissances et l'emploi de fonctions symétriques. Une partie de ces résultats est extraite des articles de revue [Can01b, CCCF01, CC03, CDLD06, CV05].

Fonctions de substitution pour les chiffrements par blocs. La deuxième partie du document a pour origine un travail que j'ai mené sur plusieurs années, avec P. Charpin, H. Dobbertin puis avec M. Videau, consacré aux fonctions presque courbes, qui sont les fonctions de substitution offrant une résistance optimale aux cryptanalyses linéaires et différentielles. Après un bref rappel du contexte cryptanalytique, j'exposerai au chapitre 8 une caractérisation de ces fonctions au moyen d'un nouveau paramètre : la divisibilité de leurs coefficients de Walsh. Cette quantité nous a permis d'étudier sous un angle nouveau ces fonctions et en particulier de démontrer une conjecture formulée par Welch en 1968. Ces résultats sont issus de divers articles parus dans les actes du colloque FSE'99 [CCD99] et dans les revues *IEEE Transactions on Information Theory* [CCD00a] et *SIAM Journal on Discrete Mathematics* [CCD00b]. Cependant, comme les fonctions presque courbes n'existent que pour un nombre impair de variables, j'étudierai également dans ce chapitre différentes classes de fonctions « sous-optimales » pour ces critères. Le chapitre 9 montre alors que, paradoxalement, le fait que le spectre de Walsh d'une fonction de substitution soit divisible par une grande puissance de 2, caractéristique des fonctions optimales pour les deux principales attaques, introduit une faiblesse dans les systèmes qui les utilisent. En effet, cette propriété de divisibilité des coefficients de Walsh permet

généralement de monter une attaque d'ordre supérieur. Cette situation paradoxale a été mise en évidence dans un travail commun avec M. Videau présenté à EUROCRYPT 2002 [CV02]. Enfin, le chapitre 10 détaille quelques résultats qui peuvent également guider le choix de la fonction de substitution, en relation avec d'autres types d'attaques. Nous exposons également différents résultats numériques, et donnons en particulier les propriétés cryptographiques de l'ensemble des permutations puissances à 8, 9 et 10 variables.

Une liste des principales notations est située à la fin de ce document, page 161, juste avant la table des matières.

Les démonstrations des résultats exposés dans les différents chapitres se trouvent toutes dans les articles cités en référence après chaque en-tête de théorème, proposition... ; celles des résultats nouveaux sont par contre détaillées.

Chapitre 1

Cadre statistique et propriétés spectrales des fonctions cryptographiques

En guise de préliminaires, nous rappelons brièvement quelques fondements communs aux attaques statistiques dont nous allons étudier quelques exemples au chapitre 3 pour les chiffrements à flot, et au chapitre 7 pour les algorithmes par blocs. Nous définissons également ici certains paramètres essentiels qui quantifient les qualités cryptographiques des fonctions employées dans les algorithmes de chiffrement — celles des fonctions de filtrage dans les générateurs pseudo-aléatoires, ou des fonctions de substitution dans les algorithmes par blocs.

1.1 Cadre statistique

L'analyse des performances des attaques que nous étudions se situe dans le cadre plus général des attaques statistiques, décrit par exemple dans [CHJ02, Jun05, BJV04].

Le problème sous-jacent à toutes ces attaques peut s'exprimer de la manière suivante : étant donnée une source qui génère une suite $(X_1,\ldots,X_N)$ de N variables aléatoires à valeurs dans $\mathcal{X}$, indépendantes et identiquement distribuées suivant une distribution $\mathcal{D}$, l'objectif est de déterminer si la distribution $\mathcal{D}$ correspond à $\mathcal{D}_0$ ou à $\mathcal{D}_1$, où $\mathcal{D}_0$ et $\mathcal{D}_1$ sont deux distributions connues. Dans ce contexte, un *distingueur* à N échantillons est un algorithme qui prend en entrée une suite de N réalisations $(x_1,\ldots,x_n)$ de ces variables aléatoires, et produit la valeur 0 ou 1 en fonction de la distribution choisie. Un distingueur est caractérisé par sa probabilité d'erreur donnée par $(P_{e,1} + P_{e,2})/2$ où $P_{e,1}$ est la probabilité que le distingueur retourne 0 alors que $\mathcal{D} = \mathcal{D}_1$ et $P_{e,2}$ est la probabilité que le distingueur retourne 1 alors que $\mathcal{D} = \mathcal{D}_0$.

D'après le lemme de Neyman-Pearson [CT91], le *distingueur optimal* entre deux distributions (i.e., celui dont la probabilité d'erreur est minimale) consiste à calculer le logarithme du rapport de vraisemblance

$$L(x_1,\ldots,x_N) = \sum_{t=1}^{N} \log \frac{\Pr_{\mathcal{D}_0}[X_t = x_t]}{\Pr_{\mathcal{D}_1}[X_t = x_t]}.$$

La distinction entre les deux distributions est alors opérée par le signe de cette quantité : on choisit l'hypothèse $\mathcal{D} = \mathcal{D}_0$ si et seulement si $L(x_1,\ldots,x_N) > 0$. Pour éviter les opérations sur

les nombres flottants, plus coûteuses que sur les entiers, le calcul du logarithme du rapport de vraisemblance est naturellement implémenté en pratique par des compteurs qui évaluent le nombre d'occurrences de chaque valeur $x \in \mathcal{X}$ parmi $(x_1, \ldots, x_N)$.

Le succès de ce distingueur dépend naturellement du nombre N d'échantillons utilisés. Cette dépendance s'exprime de manière relativement simple lorsque les deux distributions $\mathcal{D}_0$ et $\mathcal{D}_1$ sont proches l'une de l'autre.

Lemme 1.1 *[BJV04, Th. 6] Soit $\mathcal{D}_0$ et $\mathcal{D}_1$ deux distributions sur un alphabet fini $\mathcal{X}$. Supposons que $\mathcal{D}_0$ et $\mathcal{D}_1$ ont le même support et que, pour chaque élément x de leur support, on a*

$$\Pr_{\mathcal{D}_0}[X = x] - \Pr_{\mathcal{D}_1}[X = x] = \varepsilon_x \text{ avec } |\varepsilon_x| \ll \Pr_{\mathcal{D}_1}[X = x].$$

Alors, pour un nombre d'échantillons égal à

$$N = \frac{d}{\sum_{x \in \mathcal{X}} \frac{\varepsilon_x^2}{p_x}},$$

où $p_x = \Pr_{\mathcal{D}_1}[X = x]$ et d est un réel quelconque, la probabilité d'erreur du distingueur est de l'ordre de $\Phi(-\sqrt{d}/2)$ où Φ est la fonction de distribution de la loi normale :

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{t^2}{2}\right) dt.$$

En particulier, si $\mathcal{D}_1$ est la distribution uniforme, le nombre d'échantillons nécessaires au distingueur pour une probabilité d'erreur inférieure à $\Phi(-\sqrt{d}/2)$ est égal à

$$N = \frac{d}{\Delta^2(\mathcal{D}_0)} \text{ avec } \Delta^2(\mathcal{D}_0) = |\mathcal{X}| \sum_{x \in \mathcal{X}} \varepsilon_x^2. \tag{1.1}$$

La quantité $\Delta^2(\mathcal{D}_0)$ mesure donc le biais d'une distribution proche de la distribution uniforme ; elle est appelée *squared Euclidean imbalance* dans [BJV04]. Pour des variables aléatoires à valeurs binaires, il sera souvent plus commode de manipuler la racine carrée de cette valeur, qui sera notée $\Delta(\mathcal{D}_0)$.

Dans toute la suite, on considérera que le nombre d'échantillons nécessaires pour distinguer les deux distributions est obtenu en appliquant le résultat précédent avec $d = 1$, ce qui correspond à une probabilité d'erreur de l'ordre de 0.3. Il suffit en effet d'augmenter légèrement la valeur de d pour obtenir une probabilité d'erreur très faible (par exemple, 0.16 pour $d = 4$).

1.2 Biais d'une fonction booléenne vectorielle

De manière générale, les fonctions qui interviennent tant dans la conception que dans l'analyse de la sécurité des algorithmes de chiffrement sont des fonctions définies sur l'ensemble des mots de n bits, $\mathbf{F}_2^n$, et à valeurs dans $\mathbf{F}_2^m$. Nous qualifierons ces fonctions de *booléennes vectorielles*, la dénomination *fonction booléenne* étant réservée au cas où $m = 1$. On utilisera également l'expression de *fonction de substitution* dans le contexte du chiffrement par blocs. Les attaques statistiques consistent généralement à distinguer la distribution $\mathcal{D}_F$ de la sortie d'une fonction vectorielle F de la distribution uniforme quand son entrée est une variable aléatoire uniformément distribuée — les fonctions F pour lesquelles $\mathcal{D}_F$ est la distribution

uniforme sont dites *équilibrées*. Dans ce but, nous introduisons la notation suivante, qui sera utilisée tout au long de ce document.

Notation 1.2 *Soit f une fonction booléenne à n variables. On note $\mathcal{F}(f)$ la quantité*

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

où $wt(f)$ désigne le poids de Hamming de f, c'est-à-dire le nombre de points de $\mathbf{F}_2^n$ en lesquels la fonction vaut 1.

Corollaire 1.3 *Soit f une fonction booléenne à n variables et $\mathcal{D}_f$ la distribution de $f(X)$ où X est une variable aléatoire uniformément distribuée dans $\mathbf{F}_2^n$. Supposons que $|\mathcal{F}(f)| \ll 2^n$. Alors, le nombre d'échantillons N requis pour distinguer $\mathcal{D}_f$ de la distribution uniforme avec le distingueur optimal est donné par*

$$N = \frac{1}{\Delta^2(\mathcal{D}_f)} \text{ avec } \Delta^2(\mathcal{D}_f) = \left(\frac{\mathcal{F}(f)}{2^n}\right)^2.$$

Preuve. Le résultat découle directement de la dernière partie du lemme 1.1 en considérant la distribution $\mathcal{D}_f$. Avec les notations précédentes, on a en effet, d'après l'équation (1.1) :

$$\begin{aligned}
\Delta^2(\mathcal{D}_f) &= 2\left(\Pr[f(X)=1] - \frac{1}{2}\right)^2 + 2\left(\Pr[f(X)=0] - \frac{1}{2}\right)^2 \\
&= (2\Pr[f(X)=1] - 1)^2 \\
&= \left(\frac{\mathcal{F}(f)}{2^n}\right)^2.
\end{aligned}$$

◇

Dans le cas où la fonction considérée est une fonction vectorielle à valeurs dans $\mathbf{F}_2^m$ avec $m > 1$, il faut considérer ses composantes au sens de la définition suivante, terminologie employée par exemple dans [BCCLC06, Nyb95].

Définition 1.4 (Composantes d'une fonction vectorielle) *Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$. On appelle composantes de F les fonctions booléennes correspondant aux combinaisons linéaires des fonctions coordonnées de F, c'est-à-dire les fonctions*

$$\begin{array}{rccc} F_\lambda : & \mathbf{F}_2^n & \to & \mathbf{F}_2 \\ & x & \mapsto & \lambda \cdot F(x) \end{array}$$

où $x \cdot y$ désigne le produit scalaire usuel sur $\mathbf{F}_2^m$.

Le nombre d'échantillons nécessaires pour distinguer la distribution de F de la distribution uniforme dépend alors du biais des composantes de F. Ce résultat [BJV04] se déduit de la même relation que celle utilisée pour montrer qu'une fonction vectorielle est équilibrée si et seulement si toutes ses composantes le sont [LN83, Th. 7.37] (voir aussi [CP05]).

Corollaire 1.5 *Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$ et soit F_λ, $\lambda \in \mathbf{F}_2^m$ ses composantes. Soit $\mathcal{D}_F$ la distribution de $F(X)$ où X est une variable aléatoire uniformément distribuée dans $\mathbf{F}_2^n$. Supposons que, pour tout $\lambda \in \mathbf{F}_2^m$, $\lambda \neq 0$, $|\mathcal{F}(F_\lambda)| \ll 2^n$. Alors, le nombre optimal d'échantillons N requis pour distinguer $\mathcal{D}_F$ de la distribution uniforme est donné par*

$$N = \frac{1}{\Delta^2(\mathcal{D}_F)} \text{ avec } \Delta^2(\mathcal{D}_F) = \frac{1}{2^{2n}} \sum_{\lambda \in (\mathbf{F}_2^m)^*} \mathcal{F}^2(F_\lambda).$$

Toutefois, quand le nombre m de sorties est grand, il devient généralement difficile de déterminer l'ensemble de la distribution $\mathcal{D}_F$ dans la mesure où elle fait intervenir les 2^m valeurs que peut prendre la fonction. Ce problème se pose typiquement quand F est la fonction de filtrage d'un chiffrement à flot opérant sur des mots-machine de 32 bits ou, dans les attaques différentielles, quand F est la dérivée du chiffrement réduit d'un algorithme par blocs (m correspond alors à la taille de bloc). En pratique, on applique donc le distingueur à la fonction $g \circ F$ où g est une fonction choisie par l'attaquant de $\mathbf{F}_2^m$ dans $\mathbf{F}_2^b$, b étant alors petit, souvent égal à 1. Le choix de la fonction g optimale dépend alors naturellement de la distribution $\mathcal{D}_F$. On peut ainsi ne considérer que la composante de F la plus biaisée, autrement dit celle qui apporte la contribution principale à la somme définissant $\Delta^2(\mathcal{D}_F)$. La fonction booléenne g est alors une fonction linéaire. C'est le principe de la cryptanalyse linéaire (et de ses généralisations utilisant des approximations linéaires multiples [BDQ04, BJV04]). S'il existe une valeur y de $\mathbf{F}_2^m$ telle que le biais $|\Pr[F(X) = y] - 2^{-m}|$ est particulièrement élevé, on peut choisir pour g l'indicatrice de cette valeur y. C'est ce que l'on fait dans les attaques différentielles classiques et par différentielles impossibles. Dans le cas de la cryptanalyse différentielle tronquée, on utilise pour g l'indicatrice d'un sous-espace vectoriel $V \subset \mathbf{F}_2^m$. Des résultats partiels sur le choix de la fonction g dans le cas où elle est équilibrée sont présentés dans [ZC00].

1.3 Spectre de Walsh et propriétés associées

1.3.1 Spectre de Walsh

On emploiera également la notation $\mathcal{F}$ pour définir la transformée de Walsh (ou de Fourier) d'une fonction booléenne puisque $\mathcal{F}(f)$ désigne aussi le coefficient de Walsh de f en 0. Cette transformée intervient dans de nombreuses attaques, en particulier parce qu'il est relativement naturel d'exploiter, au-delà du biais de la fonction f, sa corrélation avec une fonction linéaire, c'est-à-dire le biais d'une fonction de la forme $f + \varphi$ où φ est une fonction linéaire.

Notation 1.6 (Fonctions linéaires) *Pour tout $a \in \mathbf{F}_2^n$, on note φ_a la fonction linéaire à n variables définie*[a] *par $x \mapsto a \cdot x$, φ_0 correspondant à la fonction nulle.*

Définition 1.7 *Soit f une fonction booléenne à n variables. Le* coefficient de Walsh de f au point $a \in \mathbf{F}_2^n$ *est alors la quantité*

$$\mathcal{F}(f + \varphi_a) = \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+a\cdot x}.$$

On appelle spectre de Walsh de f *le multi-ensemble*[b] *formé par ses coefficients de Walsh :*

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

Le spectre de Walsh étendu (ou symétrisé) est, lui, le multi-ensemble formé par

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

a. Dans certains cas, il est utile d'identifier l'espace vectoriel $\mathbf{F}_2^n$ au corps fini à 2^n éléments. Avec cette représentation, les fonctions linéaires s'écrivent sous la forme $x \mapsto \mathrm{Tr}(\lambda x)$, $\lambda \in \mathbf{F}_{2^n}$.

b. c'est-à-dire les valeurs apparaissant dans cet ensemble et leur multiplicité

La notion de spectre de Walsh étendu peut sembler assez peu naturelle de prime abord, mais c'est souvent celle qui intervient dans les attaques puisque c'est la valeur absolue du biais $\mathcal{F}(f)$ d'une fonction qui importe généralement dans les attaques statistiques et non son signe.

Une quantité essentielle, qui intervient naturellement dans la plupart des attaques que nous verrons par la suite, est la valeur maximale apparaissant dans le spectre de Walsh étendu d'une fonction car elle détermine sa non-linéarité.

Définition 1.8 (Non-linéarité) *La* non-linéarité *d'une fonction booléenne f à n variables, notée $\mathcal{NL}(f)$, est sa distance à l'ensemble des fonctions affines. Elle est donnée par*

$$\mathcal{NL}(f) = 2^{n-1} - \frac{1}{2}\mathcal{L}(f) \text{ où } \mathcal{L}(f) = \max_{a \in \mathbf{F}_2^n} |\mathcal{F}(f + \varphi_a)|.$$

Les deux notions suivantes sont alors étroitement liées au spectre de Walsh d'une fonction.

1.3.2 Dérivées

La notion de dérivée intervient de plusieurs manières dans les attaques. Les attaques différentielles exploitent en effet un biais dans la distribution des différences des sorties de la fonction correspondant à deux entrées dont la différence est fixée. Mais, la distribution des dérivées peut également déterminer la complexité de certaines attaques simplement de par l'existence d'une relation générale avec le spectre de Walsh de la fonction que nous avons mise en évidence dans [CCCF01, Lemme V.2] et que nous rappelons ici.

Définition 1.9 (Dérivées, structures linéaires) *Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$. Pour tout élément $a \in \mathbf{F}_2^n$, la* dérivée de F en a *est la fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$, notée D_aF, définie par*

$$D_aF(x) = F(x + a) + F(x).$$

On appelle alors structure linéaire *(de type ε) tout élément a non nul tel que la fonction D_aF est constante (égale à ε).*

Proposition 1.10 *[CCCF01, Lemme V.2] Soit f une fonction booléenne à n variables et V un sous-espace vectoriel de $\mathbf{F}_2^n$ de dimension k. Alors, pour tout $b \in \mathbf{F}_2^n$, on a*

$$\sum_{a \in V} \mathcal{F}^2(f + \varphi_{a+b}) = 2^k \sum_{\alpha \in V^\perp} (-1)^{b \cdot \alpha} \mathcal{F}(D_\alpha f),$$

où $V^\perp$ désigne l'orthogonal de V.

1.3.3 Restrictions

La dernière notion que je souhaite introduire dans ce chapitre liminaire apparaît naturellement dans les attaques statistiques : il s'agit de la notion de restriction d'une fonction à un sous-espace.

Définition 1.11 (Restriction d'une fonction à un sous-espace) *Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$, V un sous-espace vectoriel de $\mathbf{F}_2^n$ et $a \in \mathbf{F}_2^n$. On appelle* restriction de F à $a+V$ *la fonction*

$$\begin{array}{rccc} F_{a+V} : & V & \to & \mathbf{F}_2^m \\ & x & \mapsto & F(a + x). \end{array}$$

Cette fonction est généralement identifiée[c] *à une fonction de* $\mathbf{F}_2^k$ *dans* $\mathbf{F}_2^m$.

Pour W *supplémentaire de* V *dans* $\mathbf{F}_2^n$, *l'ensemble des restrictions* F_{a+V}, $a \in W$ *forme la* décomposition de F relativement à V.

Le biais des différentes restrictions d'une fonction booléenne est un paramètre qui intervient dans de nombreuses attaques sur les chiffrements à flot. Il permet notamment de définir la notion d'immunité aux corrélations et de résilience [Sie84, CGH$^+$85].

Définition 1.12 (Immunité aux corrélations, résilience) *Soit* F *une fonction de* $\mathbf{F}_2^n$ *dans* $\mathbf{F}_2^m$ *et* $\mathcal{D}_F$ *la distribution de* $F(X)$ *quand* X *est uniformément distribué dans* $\mathbf{F}_2^n$. *Alors,* F *est dite* sans corrélation *d'ordre* t *si sa distribution est inchangée quand on fixe* t *variables d'entrées. De plus, une fonction équilibrée et sans corrélation d'ordre* t *est dite* t*-résiliente.*

Il existe une relation générale entre le biais de la restriction d'une fonction booléenne à un sous-espace affine et son spectre de Walsh.

Proposition 1.13 *[CC03, Prop. 1] Soit* f *une fonction booléenne à* n *variables,* V *un sous-espace vectoriel de* $\mathbf{F}_2^n$ *de dimension* k *et* W *un sous-espace en somme directe avec* V. *Alors*

$$\sum_{v \in V^\perp} (-1)^{a \cdot v} \mathcal{F}(f + \varphi_v) = 2^{n-k} \mathcal{F}(f_{a+V}),$$

et

$$\mathcal{L}(f_{a+V}) \leq \mathcal{L}(f).$$

Nous allons maintenant utiliser abondamment l'ensemble des outils précédents pour analyser la sécurité de divers algorithmes à clef secrète, à flot et par blocs.

c. Cette dernière est définie à une translation $x \mapsto x + v$, $v \in V$ près, mais ceci n'aura pas d'influence sur les propriétés que nous étudierons ici.

Première partie

Chiffrement à flot

Chapitre 2

Introduction au chiffrement à flot

Parmi les algorithmes de chiffrement à clef secrète, la distinction entre chiffrement à flot et chiffrement par blocs est parfois délicate. On appelle usuellement *chiffrement à flot* un algorithme de chiffrement qui opère sur des blocs de clair de taille relativement petite (typiquement un bit, un octet ou un mot) au moyen d'une transformation qui varie au cours du temps. Au contraire, un algorithme par blocs applique la même fonction aux différents blocs de clair, qui sont ici de taille plus importante, typiquement 64, 128 ou 256 bits [MvOV97].

Cette définition permet de différencier aisément les chiffrements à flot synchrones additifs (qui opèrent au niveau du bit) des chiffrements par blocs en mode ECB (Electronic CodeBook). Elle permet également de classer clairement certains modes opératoires sur les algorithmes par blocs, par exemple les modes OFB et CTR, dans la catégorie des chiffrements à flot. Toutefois, elle devient beaucoup plus floue lorsque l'on veut caractériser le mode CBC, puisqu'il peut également être considéré comme un chiffrement à flot auto-synchronisant opérant sur des blocs de grande taille. Aussi, dans la suite de ce document, la classe des chiffrements à flot sera-t-elle restreinte aux seuls algorithmes synchrones. Cet abus de langage est également motivé par le fait que les chiffrements à flot auto-synchronisants sont plus ou moins tombés en désuétude : leur principal intérêt, qui est de permettre au destinataire de resynchroniser le générateur pseudo-aléatoire après la transmission d'un certain nombre de blocs de chiffré, est également offert par tous les algorithmes synchrones récents qui incluent une procédure de re-synchronisation (opérée par un changement de valeur initiale). Ce sont donc ces derniers qui sont désormais préférés dans l'immense majorité des applications, cette orientation étant par ailleurs renforcée par la difficulté de concevoir des chiffrements auto-synchronisants qui résistent aux attaques à chiffré choisi.

2.1 Modèle et propriétés des chiffrements à flot synchrones

2.1.1 Modèle général

Un algorithme de chiffrement à flot synchrone consiste à combiner le texte clair avec une suite binaire de même longueur, la suite chiffrante. Cette suite, notée $\boldsymbol{s} = (s_t)_{t\geq 0}$, est engendrée indépendamment du texte clair et du chiffré par un automate à états finis, appelé *générateur pseudo-aléatoire*. Ce dernier produit à chaque instant t un bloc de m bits, s_t, déterminé par la valeur de son état interne $\boldsymbol{x}_t$. Le fonctionnement du générateur pseudo-aléatoire peut donc

être décrit au moyen de trois fonctions (*cf.* Figure 2.1) :

- *une procédure d'initialisation* qui détermine l'état initial du générateur, $\boldsymbol{x}_0$, à partir de la clef secrète et d'une valeur initiale publique, notée IV, qui correspond souvent à un numéro de trame. Cette initialisation est parfois divisée en deux phases : l'une, dite de chargement de clef, calcule une certaine quantité qui ne dépend que de la clef (et non de la valeur initiale), l'autre, dite d'injection d'IV ou de re-synchronisation, détermine l'état initial du générateur à partir de la valeur calculée précédemment et de la valeur initiale. Le fait de découper de la sorte la phase d'initialisation permet de réduire le coût de la procédure qui consiste à changer la valeur initiale sans modifier la clef. Il s'agit en effet d'une opération beaucoup plus fréquente en pratique que le changement de clef, notamment pour les protocoles de communication pour lesquels la longueur des paquets échangés est relativement petite. Par exemple, dans les communications GSM, on change l'IV tous les 228 bits alors que la clef reste inchangée tout au long de la conversation.
- *une fonction de transition*, notée Φ, qui fait évoluer l'état interne du générateur entre les instants t et $(t+1)$. Cette fonction peut dépendre de la clef, de la valeur initiale et du temps, mais elle est fixe dans l'immense majorité des générateurs destinés à une mise en œuvre matérielle, pour des raisons évidentes de simplicité et d'encombrement.
- *une fonction de filtrage*, notée f, qui à chaque instant, produit le bloc de suite chiffrante à partir de l'état interne courant. Tout comme la fonction de transition, la fonction de filtrage peut varier avec la clef, la valeur initiale et le temps, mais elle est généralement fixe pour les raisons évoquées précédemment.

Le chiffrement consiste alors à combiner la suite chiffrante au texte clair au moyen d'une fonction de combinaison h inversible (éventuellement paramétrée par la clef secrète et la valeur initiale), qui à un couple de blocs de ℓ bits de suite chiffrante et de clair, associe un bloc de ℓ bits de texte chiffré [a]. Dans l'immense majorité des cas, la fonction h correspond à l'addition modulo 2 (c'est-à-dire au XOR bit à bit). On parle alors de chiffrement synchrone additif.

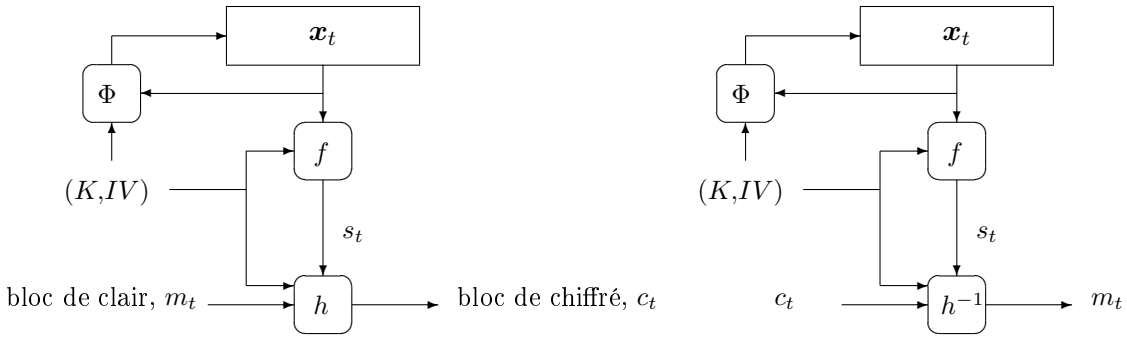

FIG. 2.1 – *Chiffrement à flot synchrone et déchiffrement associé*

Plusieurs paramètres influencent la sécurité d'un tel algorithme de chiffrement, notamment la taille de la clef secrète, celle de l'état interne .... Les notations utilisées dans la suite sont récapitulées à la table 2.1. En particulier, le nombre de bits de l'état interne $\boldsymbol{x}_t$ du générateur

a. Les blocs sur lesquels opère la fonction h ne sont pas nécessairement de même taille que ceux qui sont produits par la fonction de filtrage : h peut par exemple prendre en entrée un bloc formé de plusieurs sorties consécutives du générateur.

est désigné par L. Il s'agit de la taille de l'état du plus petit automate à états finis permettant d'engendrer cet ensemble de suites. Cette taille peut différer de l'entropie de l'état initial, qui correspond au nombre d'états valides produits par la fonction d'initialisation. Il est également important de remarquer que, pour des raisons de facilité de mise en œuvre, la fonction de filtrage n'opère généralement pas sur tous les bits de l'état interne. Toutefois, sauf mention contraire, on assimilera f à une fonction dont le nombre de variables est égal à la taille de l'état interne mais qui ne dépend pas de certaines d'entre elles, autrement dit qui possède des structures linéaires. Dans ce cas, on utilisera la lettre n pour désigner à la fois le nombre de variables de f et la taille de l'état interne du générateur.

$\boldsymbol{s}$	suite chiffrante
s_t	bloc de suite chiffrante de taille m, produit à l'instant t
m	taille de bloc de suite chiffrante
k	taille de la clef secrète
n	nombre de variables de la fonction de filtrage
L	taille de l'état interne du générateur, aussi notée n si elle est identique au nombre de variables de la fonction de filtrage
$\boldsymbol{x}_t$	état interne du générateur à l'instant t
Φ	fonction de transition, de $\mathbf{F}_2^L$ dans $\mathbf{F}_2^L$
f	fonction de filtrage de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$
$\Gamma = \{\gamma_1, \ldots, \gamma_n\}$	positions des bits de l'état interne (dans l'ordre croissant) donnant les entrées de la fonction de filtrage, i.e., $s_t = f(\boldsymbol{x}_{t,\gamma_1}, \boldsymbol{x}_{t,\gamma_2}, \ldots, \boldsymbol{x}_{t,\gamma_n})$

TAB. 2.1 – *Notations utilisées pour décrire un chiffrement à flot synchrone (additif)*

2.1.2 Contextes d'utilisation

La petite taille de bloc utilisée dans les algorithmes par flot (un seul bit dans le cas des chiffrements additifs) présente de multiples avantages. Elle permet naturellement de réduire les délais et la taille de la mémoire-tampon nécessaires pour stocker le message jusqu'à l'obtention d'un bloc complet. De plus, les chiffrements à flot additifs ne requièrent évidemment pas de padding pour atteindre une longueur multiple de la taille de bloc. Ceci peut s'avérer particulièrement souhaitable dans les applications où la bande passante est très limitée ou quand le protocole employé impose la transmission de paquets relativement courts (auquel cas, le padding représente une proportion non négligeable des données échangées). Un autre avantage présenté par l'emploi de blocs de petite taille est de limiter la propagation des erreurs de transmission au cours du déchiffrement.

Outre le fait qu'ils sont bien adaptés pour les transmissions bruitées ou à faible bande passante, les procédés de chiffrement à flot sont généralement privilégiés dans des contextes où il est primordial de pouvoir chiffrer et déchiffrer très rapidement ou au moyen de ressources très limitées. Leur utilisation est par exemple systématique dans les applications qui imposent de fortes contraintes sur la taille et la consommation électrique du circuit électronique dédié au chiffrement. C'est le cas de la plupart des systèmes embarqués, tels les téléphones mobiles.

2.1.3 Les grandes familles de générateurs pseudo-aléatoires

Les chiffrements à flot qui font l'objet de ce document visent uniquement l'un des deux contextes d'utilisation mentionnés précédemment et identifiés par le projet eSTREAM [ECR05] : un très haut débit dans un environnement logiciel ou une mise en œuvre très peu coûteuse dans un contexte matériel. Pour cette raison, nous écartons d'emblée du champ de cette étude les deux classes suivantes de générateurs pseudo-aléatoires qui présentent un intérêt évident du point de vue de la sécurité, mais dont la mise en œuvre ne répond pas aux contraintes applicatives :

- les générateurs sûrs d'un point de vue calculatoire, dont la sécurité repose sur la difficulté de certains problèmes mathématiques. Il s'agit de générateurs pour lesquels l'existence d'un distingueur de complexité polynomiale est équivalente à un problème mathématique connu réputé difficile. A l'instar de la plupart des cryptosystèmes à clef publique, ces générateurs pseudo-aléatoires sont fondés sur des problèmes issus de la théorie des nombres. On peut mentionner par exemple le générateur RSA, qui consiste à appliquer récursivement l'algorithme RSA à partir d'une graine $\boldsymbol{x}_0$ et dont la sortie à l'instant t correspond au bit de poids faible de $\boldsymbol{x}_t$ [ACGS88], ou le générateur Blum-Blum-Shub [BBS86] qui consiste à itérer l'élévation au carré modulo pq et dont la sécurité repose sur la difficulté du problème des résidus quadratiques. Par la nature des opérations effectuées, tous ces générateurs sont extrêmement lents ; leur débit est très insuffisant et leur mise en œuvre beaucoup trop compliquée pour qu'ils puissent être utilisés en pratique dans un chiffrement à flot ;
- les générateurs inconditionnellement sûrs selon certains modèles, dont l'objectif est d'apporter une sécurité démontrable identique à celle du masque jetable mais sous l'hypothèse que l'adversaire, s'il peut disposer d'une puissance de calcul infinie, a certaines contraintes (par exemple, sa capacité de stockage est limitée) [Mau92, AR99, Rab05]. Ces générateurs nécessitent la mise en place d'une infrastructure extrêmement lourde et ne peuvent être déployés qu'à grande échelle : ils utilisent par exemple une source d'aléa commune diffusée par un satellite, ou à travers le réseau.

Les constructions qui nous intéressent ici sont donc uniquement les constructions dédiées au sens où elles sont spécialement conçues pour cet usage. Leur intérêt pratique se mesure par comparaison avec un algorithme par blocs en mode OFB ou CTR, pour l'un des deux critères précédents. Ainsi, la vitesse de chiffrement en logiciel d'un système par bloc classique tel l'AES est de l'ordre d'une vingtaine de cycles du processeur pour chiffrer un octet, alors que certains algorithmes par flot permettent d'atteindre des vitesses de l'ordre de 3 à 5 cycles par octet — même si ce débit dépend naturellement de la longueur de trame visée, ce qui accorde une importance plus ou moins grande à la vitesse de re-synchronisation. De même, la réalisation matérielle d'un algorithme par blocs nécessite généralement au minimum de l'ordre de 20 000 portes logiques. Un des plus performants de ce point de vue est l'algorithme KASUMI [rGPP01b], utilisé dans le chiffrement à flot de l'UMTS [rGPP01a], qui peut être mis en œuvre par un circuit comportant moins de 10 000 portes.

Ces contraintes liées à la mise en œuvre, dans un environnement logiciel ou matériel, interviennent directement dans la conception des chiffrement à flot dédiés et rendent la tâche relativement difficile. Ainsi, à l'heure actuelle, les seuls générateurs dédiés considérés comme sûrs sont de conception très récente. Parmi les plus anciens, on peut mentionner SNOW 2.0 [EJ02] et MUGI [Hit01], qui ont été pris en compte dans la dernière version de travail de la norme internationale de chiffrement ISO/IEC 18033-4 [ISO05]. Plus récemment, une trentaine de

nouveaux générateurs dédiés ont été proposés en avril 2005 suite à l'appel lancé par le projet eSTREAM [ECR05] du réseau européen ECRYPT ; leur sécurité et leurs performances font actuellement l'objet d'une évaluation approfondie.

2.2 Sécurité

Pour évaluer la sécurité d'un algorithme de chiffrement à flot synchrone, on se place usuellement dans le contexte d'une attaque à clair connu, ce qui signifie que l'attaquant dispose d'une certaine quantité D de bits de suite chiffrante — on supposera en effet que la fonction de combinaison h est publique et que $s \mapsto c = h(m,s)$ est inversible, c'est-à-dire que la connaissance d'un couple clair-chiffré permet de déterminer la suite chiffrante.

2.2.1 Classification des attaques

Sous ces hypothèses, les attaques potentielles se répartissent en quatre grandes catégories en fonction de leur portée :

- les attaques *par recouvrement de clef*, qui visent à retrouver la clef secrète à partir de la connaissance d'un certain nombre de bits de suite chiffrante ;
- les attaques *par recouvrement de l'état initial*, dont l'objectif est de retrouver l'initialisation du générateur (ou de façon équivalente un état interne complet). La connaissance de la clef secrète suffit naturellement à retrouver l'initialisation, mais la réciproque n'est pas nécessairement vraie[b]. Si ces attaques permettent à l'adversaire de calculer autant de bits qu'il le souhaite à partir de cette initialisation, elles ne lui permettent pas nécessairement d'engendrer les suites produites à partir de la même clef secrète, mais de valeurs initiales différentes. Leur portée peut donc être considérablement plus faible que celle des attaques par recouvrement de clef, notamment si la taille des paquets échangés est petite, puisque les paquets sont tous chiffrés à l'aide de valeurs initiales différentes ;
- les attaques *par prédiction du bit suivant*, qui consistent, à partir de la connaissance des n premiers bits de la suite engendrée par le générateur pour une certaine clef, à prédire la valeur du bit suivant ;
- les attaques *par distingueur*, qui déterminent si une suite de n bits correspond à la sortie du générateur pseudo-aléatoire considéré ou s'il s'agit véritablement d'une suite aléatoire.

Les attaques de cette dernière catégorie sont évidemment beaucoup moins puissantes que les autres ; elles ne fournissent à l'adversaire que des informations partielles sur le message clair. Elles permettent par exemple de vérifier si un chiffré intercepté correspond à un texte clair donné, alors que retrouver la clef secrète permet de déchiffrer tous les messages interceptés lors de la communication. Par ailleurs, l'existence d'attaques par distingueur de complexité polynomiale est strictement équivalente à celle d'attaques polynomiales par prédiction du bit suivant [Yao82, BM84].

b. Toutefois, si la fonction qui, à valeur initiale fixée, détermine l'état en fonction de la clef n'est pas inversible, l'entropie de l'état initial est strictement inférieure à celle de la clef, ce qui peut être une source de faiblesse.

2.2.2 Complexité en données

On considère usuellement qu'un algorithme de chiffrement est sûr s'il n'est vulnérable à aucune attaque dont la complexité (en temps, en mémoire et en données) est inférieure à la taille de l'espace des clefs, c'est-à-dire à 2^k. Toutefois, dans le cas des algorithmes à flot, il n'est pas aisé de déterminer la complexité en données admissible, car les D bits de suite chiffrante nécessaires à l'attaque peuvent provenir d'une seule ou d'un grand nombre de valeurs initiales. Dans la plupart des applications, les trames sont relativement courtes, ce qui signifie qu'en pratique, un attaquant dispose d'un nombre très limité de données engendrées à partir d'un même état initial. Si une attaque nécessitant D bits de suite chiffrante, avec D inférieur mais relativement proche de 2^k n'est évidemment pas réaliste dans la pratique, le fait qu'elle permette de conclure qu'un chiffrement est théoriquement cassé est sujet à discussion — certains concepteurs limitent la sécurité de leur système aux attaques utilisant moins de $2^{\frac{k}{2}}$ bits pour un couple clef / IV donné.

Ainsi, on trouve aux deux extrémités de la catégorie des attaques par distingueur nécessitant D bits de suite chiffrante :

- les attaques sur l'algorithme de génération de la suite, qui permettent de distinguer une suite de D bits engendrée à partir d'un seul état initial d'une suite aléatoire ;
- les attaques sur l'algorithme de re-synchronisation, qui permettent de distinguer d'une fonction aléatoire la fonction qui, pour une clef fixée, associe à chaque IV de taille v les v premiers bits de la suite chiffrante engendrée pour cette IV.

2.2.3 Contraintes imposées par les attaques génériques

Dans toute la suite, on supposera que les composants du chiffrement respectent les critères élémentaires suivants, dictés par la nécessité de résister à des attaques classiques qui s'appliquent à tous les chiffrements à flot.

- La taille de l'état interne L est supérieure ou égale au double de la taille de la clef secrète afin de parer les attaques dites par compromis temps-mémoire-données [Bab95, Gol97].
- La fonction de filtrage est équilibrée afin que la sortie du générateur soit uniformément distribuée. On voit en effet grâce au corollaire 1.3 qu'un générateur utilisant une fonction booléenne de filtrage non équilibrée est toujours vulnérable à l'attaque par distingueur triviale, qui exploite le déséquilibre de la suite chiffrante, dès que le nombre de variables de la fonction de filtrage est inférieur à la moitié de la taille de la clef. Ainsi, pour $|\mathcal{F}(f)| = 2$ qui est le plus petit biais possible, on obtient un distingueur de complexité $2^{2(n-1)}$.
- La fonction de transition Φ garantit que la suite chiffrante possède une période élevée quel que soit l'état initial. Sinon, il devient facile de distinguer la suite produite d'une suite aléatoire.
- L'une des deux fonctions au moins, Φ ou f, est non linéaire. Sinon, la suite produite dépend linéairement des bits de l'état initial et la connaissance de L bits de suite chiffrante fournit un système linéaire de L équations à L inconnues (les bits de l'état initial), que l'on peut résoudre simplement au moyen d'un pivot de Gauss. Cette attaque permettrait de retrouver l'état initial du générateur en L^3 opérations, complexité très inférieure à celle de la recherche exhaustive de la clef puisque L est généralement de l'ordre de deux fois la taille de la clef.

2.3 Les grandes familles de constructions dédiées

La classification des différents types de générateurs pseudo-aléatoires dédiés est une tâche délicate dans la mesure où leur conception est largement liée à l'environnement applicatif auquel le générateur est destiné. On peut toutefois distinguer trois grandes familles dépendant du type de fonction de transition employée. Mais ces classes peuvent parfois être elles-mêmes subdivisées suivant que le générateur vise une mise en œuvre logicielle ou matérielle.

2.3.1 Les chiffrements à transition linéaire

L'utilisation d'une fonction de transition linéaire est en effet un choix naturel en termes de simplicité d'implémentation — à condition que la fonction de filtrage garantisse que la suite produite ne dépend pas linéairement de l'état initial du générateur. Parmi les fonctions de transition linéaires, celles qui sont mises en œuvre au moyen de registres à décalage à rétroaction linéaire (LFSR) sont privilégiées à la fois pour le faible coût de leur implémentation matérielle et parce que l'on dispose de nombreux résultats théoriques sur les propriétés statistiques des suites produites. Les générateurs utilisant des registres à décalage à rétroaction linéaire sont sans aucun doute ceux qui ont fait l'objet des études les plus nombreuses. Ces systèmes peuvent être destinés soit à un environnement matériel, soit à un environnement logiciel. Mais, on utilise généralement dans ce dernier cas des registres à décalage à rétroaction linéaire non plus binaires, mais opérant sur un alphabet plus grand (typiquement sur des octets ou des mots de 32 bits). On inclut souvent dans cette catégorie les générateurs à transition linéaire mais dont l'horloge est irrégulière. Parmi les générateurs à base de LFSR d'utilisation courante, on peut mentionner E0, déployé dans la norme Bluetooth, A5/1 utilisé pour chiffrer les communications des téléphones mobiles dans la norme GSM, SNOW 2.0 qui est inclus dans la dernière version de la norme ISO/IEC 18033 ou SOSEMANUK qui est l'une des deux soumissions à eSTREAM auxquelles j'ai participé [BBC+05b]. Les progrès récents dans le domaine de la cryptanalyse, notamment les attaques dites algébriques proposées dernièrement, mettent toutefois en évidence des faiblesses inhérentes à de nombreux générateurs de ce type. De multiples précautions liées à l'émergence de ces attaques doivent donc être prises lors de la conception de générateurs utilisant une fonction de transition linéaire.

2.3.2 Les chiffrements à transition non-linéaire

Afin d'éviter les faiblesses pouvant résulter du caractère linéaire de la fonction de transition, certaines conceptions récentes privilégient une évolution non-linéaire. Toutefois, la fonction de transition choisie doit garantir que les états internes du générateur ne forment pas une suite de faible période, et ce quelle que soit la valeur de l'état initial. Contrairement aux fonctions linéaires, il est relativement difficile d'obtenir de tels résultats théoriques pour des fonctions non-linéaires. Cette difficulté peut être contournée si la taille de l'état interne du générateur n'est pas limitée drastiquement par des contraintes d'implémentation (c'est le cas des applications logicielles destinées aux ordinateurs usuels). Dans ce cas, même en l'absence de résultats théoriques, on peut estimer qu'il est très peu probable qu'une suite de faible période soit produite si l'état interne est suffisamment grand. L'exemple type est celui de RC4, dont l'état interne correspond à un tableau de 512 octets dont les valeurs sont modifiées de façon non-linéaire à chaque itération de l'algorithme, ou de la proposition plus récente Py [BS05]. Toutefois, ce type de systèmes possède deux inconvénients majeurs. Du point

de vue de la sécurité, ils sont extrêmement vulnérables aux attaques par canaux cachés qui exploitent l'analyse du comportement du cache [Ber05, CLS06]. Par ailleurs, leur procédure d'initialisation est nécessairement coûteuse, ce qui les rend peu adaptés aux contextes où les trames sont de petite taille.

Dans les applications matérielles, les contraintes sur la taille du circuit correspondant imposent que l'état interne du générateur ne soit pas trop grand, autrement dit que sa taille n'excède pas sensiblement le double de la longueur de la clef (qui est la taille minimale pour résister aux attaques par compromis temps-mémoire-données). Dans ce cas, il est indispensable de disposer de résultats théoriques sur la période de la fonction de transition. A l'heure actuelle, très peu de fonctions offrent ces garanties et les générateurs pseudo-aléatoires les utilisant sont encore au stade de développement. On peut mentionner certains registres à décalage à rétroaction non-linéaire, les registres à décalage à rétroaction avec retenues [KG97, AB05] et les T-fonctions [KS02, KS04], cette dernière proposition s'avérant finalement peu souhaitable tant à cause de certaines faiblesses liées à son emploi que de sa lenteur même en logiciel [MH05].

2.3.3 Les conceptions hybrides

Dans certains générateurs pseudo-aléatoires, l'état interne est divisé en deux parties, l'une étant mise à jour par une fonction linéaire, l'autre par une fonction non-linéaire. Lorsque la partie qui évolue de manière non-linéaire est beaucoup plus petite que l'autre, elle est généralement assimilée à une mémoire interne. Autrement dit, le générateur est souvent classé comme un générateur à transition linéaire avec mémoire. C'est le cas par exemple des générateurs SNOW 2.0 et E0, qui sont usuellement qualifiés de systèmes à transition linéaire. Toutefois, il existe des générateurs dans lesquels les parties à évolution linéaire et non-linéaire de l'état interne sont de tailles similaires. Dans cette catégorie, on peut citer MUGI [Hit01] qui figure dans la dernière version de travail de la norme ISO/IEC 18033-4, ou l'algorithme Grain [HJM05], soumis à eSTREAM.

Chapitre 3

Quelques attaques par distingueur

Lors de la conception d'un chiffrement à flot, le choix de la fonction de filtrage est en partie conditionné par le fait que l'algorithme de génération de la suite chiffrante doit résister aux attaques par distingueur, qui sont particulièrement simples lorsque la fonction de transition du générateur est celle d'un registre à décalage, notamment d'un registre à décalage à rétroaction linéaire. L'objet n'est pas ici de présenter de nouvelles attaques par distingueur, mais de formaliser quelques attaques bien connues [Gol96, CHJ02, MH04, EJ05] et de mesurer précisément l'impact de la fonction de filtrage sur leur efficacité. En particulier, les outils que nous avons développés dans [CCCF01, CCCF00, BCCLC06] nous permettent de mettre en évidence l'influence dans la complexité de ces cryptanalyses de la non-linéarité de la fonction et des différents moments de son spectre de Walsh, c'est-à-dire de la somme des puissances w des coefficients de Walsh.

3.1 Distingueur sur la fonction augmentée

En plus de la distribution des blocs de suite chiffrante pris individuellement, il est important que la distribution des ℓ-uplets de blocs de suite chiffrante $(s_t,s_{t+1},\ldots,s_{t+\ell-1})$ soit uniforme pour toutes les valeurs de ℓ raisonnables. Dans la nomenclature des tests statistiques, on parle parfois de test de fréquence par blocs. Dans cette partie, nous nous focaliserons sur le cas où la fonction de transition correspond à celle d'un registre à décalage (dont la rétroaction n'est pas nécessairement linéaire). Autrement dit, nous supposons que, pour tout $0 \leq i \leq L-1$, le bit i de l'état interne à l'instant $(t+1)$ est égal au bit $(i+1)$ de l'état interne à l'instant t. Dans ce cas, on peut facilement modéliser les dépendances entre les entrées successives de la fonction de filtrage. L'hypothèse essentielle ici est que ce registre produit une suite de variables aléatoires $(x_t)_{t\geq 0}$ indépendantes et uniformément distribuées. Ainsi, la sortie du générateur à l'instant t est donnée par

$$s_t = f(x_{t+\gamma_1},x_{t+\gamma_2},\ldots,x_{t+\gamma_n}) \ ,$$

où la fonction f est équilibrée et les positions de connection $\gamma_1,\ldots,\gamma_n$ sont classées dans l'ordre croissant. On s'intéresse donc à la distribution de la fonction qui, à partir de ℓ blocs $(x_{t+\gamma_1},x_{t+\gamma_2},\ldots,x_{t+\gamma_n})$, $0 \leq t < \ell$ produit ℓ blocs consécutifs de suite chiffrante. Cette fonction a été introduite par Anderson [And95] sous le nom de *fonction augmentée* associée au registre filtré — notons que, dans [And95, Gol96], la fonction augmentée à ℓ sorties est une fonction possédant un nombre de variables plus élevé, puisqu'elle est en fait définie comme une fonction dépendant de $(x_{t+i},\ 0 \leq t < \ell,\ \gamma_1 \leq i \leq \gamma_n)$.

3.1.1 Distingueur par comparaison de deux blocs de suite chiffrante

Une attaque usuelle par distingueur consiste à exploiter un éventuel biais dans la distribution des couples de blocs de suite chiffrante engendrés aux instants t et $t+\tau$ où l'écart τ est fixé. La proposition suivante traite le cas binaire ($m = 1$), c'est-à-dire où le générateur ne produit qu'un bit par unité de temps. Rappelons que, dans toute la suite, on supposera implicitement que la fonction de filtrage f est équilibrée.

Proposition 3.1 *Soit τ un entier strictement positif, $\tau \leq \gamma_n - \gamma_1$ et I et J les deux ensembles définis par*

$$I = \{i, 1 \leq i \leq n,\ \exists j, \gamma_i - \gamma_j = \tau\} \text{ et } J = \{j, 1 \leq j \leq n,\ \exists i, \gamma_i - \gamma_j = \tau\}\ .$$

Soit ℓ le cardinal de chacun de ces deux ensembles. Alors, le nombre optimal d'échantillons N requis pour distinguer la distribution du couple $(s_t,s_{t+\tau})$ de la distribution uniforme sur $\mathbf{F}_2^2$ est donné par

$$N = \frac{1}{\Delta^2} \text{ avec } \Delta^2 = \frac{1}{2^{4n}} \left(\sum_{u \in \mathbf{F}_2^\ell} \mathcal{F}(f + \varphi_{(u,0)_I}) \mathcal{F}(f + \varphi_{(u,0)_J}) \right)^2 ,$$

où la notation $x = (a,b)_I$ désigne le vecteur défini par $(x_i, i \in I) = a$ et $(x_i, i \notin I) = b$.

En particulier, la distribution de $(s_t,s_{t+\tau})$ est uniforme pour tout $1 \leq \tau \leq \gamma_n - \gamma_1$ si l'une des deux conditions suivantes est satisfaite :

- *$\mathcal{F}(f + \varphi_u) = 0$ pour tout $u \in \langle e_1, \ldots, e_{n-1} \rangle$ tel que $wt(u) \leq \ell$;*
- *$\mathcal{F}(f + \varphi_u) = 0$ pour tout $u \in \langle e_2, \ldots, e_n \rangle$ tel que $wt(u) \leq \ell$,*

où $(e_1, \ldots, e_n)$ désigne la base canonique de $\mathbf{F}_2^n$.

A l'inverse, si $\mathcal{F}(f + \varphi_{e_1}) \neq 0$ (resp. si $\mathcal{F}(f + \varphi_{e_n}) \neq 0$) et qu'il existe $i \in \{2, \ldots, n-1\}$ tel que $\mathcal{F}(f + \varphi_{e_i}) \neq 0$, alors la distribution de $(s_t,s_{t+\tau})$ n'est pas uniforme pour $\tau = \gamma_i - \gamma_1$ (resp. pour $\tau = \gamma_n - \gamma_i$).

Preuve. Les bits s_t et $s_{t+\tau}$ sont les images par f de deux vecteurs ayant ℓ bits en commun. La distribution du couple $(s_t,s_{t+\tau})$ correspond donc à celle de la fonction à $2n - \ell$ variables :

$$\begin{array}{rccc} F_2 : & \mathbf{F}_2^\ell \times \mathbf{F}_2^{n-\ell} \times \mathbf{F}_2^{n-\ell} & \to & \mathbf{F}_2^2 \\ & (x,y,z) & \mapsto & (f(x,y)_I, f(x,z)_J) \end{array}$$

Comme f est équilibrée, on déduit du corollaire 1.5 que le nombre d'échantillons N requis pour distinguer la distribution de F_2 de la distribution uniforme est donné par

$$\frac{1}{N} = \frac{1}{2^{4n-2\ell}} \mathcal{F}^2(g)$$

où g est la fonction booléenne définie par

$$g(x,y,z) = f(x,y)_I + f(x,z)_J\ .$$

Notons $f_{(x,\mathbf{F}_2^{n-\ell})_I}$ (resp. $f_{(x,\mathbf{F}_2^{n-\ell})_J}$) la restriction de f (au sens de la définition 1.11) à l'ensemble $\{(x,y)_I, y \in \mathbf{F}_2^{n-\ell}\}$ (resp. $\{(x,y)_J, y \in \mathbf{F}_2^{n-\ell}\}$). En utilisant la proposition 1.13, on a

$$\begin{aligned}
\mathcal{F}(g) &= \sum_{x\in\mathbf{F}_2^\ell} \sum_{y\in\mathbf{F}_2^{n-\ell}} (-1)^{f(x,y)_I} \sum_{z\in\mathbf{F}_2^{n-\ell}} (-1)^{f(x,z)_J} \\
&= \sum_{x\in\mathbf{F}_2^\ell} \mathcal{F}\left(f_{(x,\mathbf{F}_2^{n-\ell})_I}\right) \mathcal{F}\left(f_{(x,\mathbf{F}_2^{n-\ell})_J}\right) \\
&= \frac{1}{2^{2\ell}} \sum_{x\in\mathbf{F}_2^\ell} \sum_{u\in\mathbf{F}_2^\ell} (-1)^{x\cdot u} \mathcal{F}(f+\varphi_{(u,0)_I}) \sum_{v\in\mathbf{F}_2^\ell} (-1)^{x\cdot v} \mathcal{F}(f+\varphi_{(v,0)_J}) \\
&= \frac{1}{2^\ell} \sum_{u\in\mathbf{F}_2^\ell} \mathcal{F}(f+\varphi_{(u,0)_I}) \mathcal{F}(f+\varphi_{(u,0)_J}) \ .
\end{aligned}$$

Enfin comme, par définition, I ne contient jamais la position 1 et J ne contient jamais la position n, on voit que la quantité $\mathcal{F}(g)$ définie ci-dessus est toujours nulle si $\mathcal{F}(f+\varphi_u) = 0$ pour tous les u de poids de Hamming inférieur ou égal à ℓ dans $\langle e_1,\ldots,e_{n-1}\rangle$ ou dans $\langle e_2,\ldots,e_n\rangle$ [a]. ◇

On obtient un résultat similaire dans le cas où la fonction de filtrage est une fonction vectorielle à valeurs dans $\mathbf{F}_2^m$ avec $m > 1$, c'est-à-dire quand le générateur produit plusieurs bits à chaque instant. Dans ce cas, le nombre d'échantillons N requis pour distinguer le couple $(s_t,s_{t+\tau})$ d'une suite aléatoire d'éléments de $\mathbf{F}_2^{2m}$ est donné par

$$N = \frac{1}{\Delta^2} \text{ avec } \Delta^2 = \frac{1}{2^{4n}} \sum_{\lambda,\mu\in(\mathbf{F}_2^m)^*} \left(\sum_{u\in\mathbf{F}_2^\ell} \mathcal{F}(f_\lambda+\varphi_{(u,0)_I}) \mathcal{F}(f_\mu+\varphi_{(u,0)_J}) \right)^2 .$$

De la même manière, si le registre à décalage considéré est un registre opérant sur des éléments de $\mathbf{F}_2^m$ (ou de $\mathbf{F}_{2^m}$), le résultat est obtenu en remplaçant ℓ par ℓm dans la formule précédente.

Enfin, un calcul identique permet de déterminer les corrélations de la fonction g qui produit $s_t + s_{t+\tau}$:

$$\mathcal{F}(g+\varphi_{(a,b,c)}) = \frac{1}{2^\ell} \sum_{u\in\mathbf{F}_2^\ell} \mathcal{F}(f+\varphi_{(u,b)_I}) \mathcal{F}(f+\varphi_{(u+a,c)_J}),\ \forall a \in \mathbf{F}_2^\ell,\ b,c \in \mathbf{F}_2^{n-\ell} \ .$$

Ainsi, pour limiter la contrainte pesant sur la fonction de filtrage, il est important de réduire autant que possible le nombre d'entrées communes de la fonction de filtrage à deux instants, c'est-à-dire le nombre de couples (i,j), $i > j$ pour lesquels les écarts $\gamma_i - \gamma_j$ sont identiques [Gol96]. Il est donc extrêmement souhaitable que tous les écarts $\gamma_i - \gamma_j$ soient distincts. L'ensemble $\{\gamma_1\ldots,\gamma_n\}$ est alors parfois qualifié de *full positive difference set* [LC04, Page 678] ; cette propriété est également utilisée dans la construction de codes convolutifs auto-duaux [RB67]. Un tel ensemble de n éléments ne peut exister que si les deux connections extrémales vérifient

$$\gamma_n - \gamma_1 \geq \frac{n(n-1)}{2} \ ,$$

a. Notons que, contrairement à ce qui est souvent affirmé, le fait que f soit ℓ-résiliente est suffisant mais n'est pas nécessaire pour résister à cette attaque. L'ordre de résilience de f n'intervient que dans les attaques par corrélation conditionnées par la sortie de la fonction augmentée introduites dans [And95].

c'est-à-dire si le nombre de variables de la fonction de filtrage est inférieur à $\sqrt{2L}$ où L est la taille du registre.

3.1.2 Critère d'équilibre de la fonction augmentée

Plus généralement, pour qu'il ne soit pas possible de monter une attaque par distingueur sur la distribution des ℓ-uplets de blocs de suite chiffrante, il faut que la fonction augmentée qui produit $(\gamma_n - \gamma_1 + 1)$ blocs consécutifs de suite chiffrante soit équilibrée. Cette condition est équivalente[b] au fait que le premier ou le dernier vecteur de la base canonique soit une structure linéaire de type 1 pour la fonction de filtrage, au sens de la définition 1.9 de la page 9.

Proposition 3.2 *La distribution de $(s_{t+\gamma_1},\ldots,s_{t+\gamma_n})$ est uniforme si et seulement si la fonction de filtrage f est de la forme*

$$f(x_1,\ldots,x_n) = x_1 + g(x_2,\ldots,x_n)$$

ou

$$f(x_1,\ldots,x_n) = g(x_1,\ldots,x_{n-1}) + x_n \ .$$

Preuve. Le caractère suffisant de la condition est démontré dans [Gol96]. Pour prouver son caractère nécessaire, on utilise le fait que, si $(s_{t+\gamma_1},\ldots,s_{t+\gamma_n})$ est uniformément distribué, le dernier point de la proposition précédente implique que

$$\Pr[X_{t+\gamma_1+\gamma_n} = x | s_{t+\gamma_1},\ldots,s_{t+\gamma_n-1}] = 2^{\gamma_n-\gamma_1-1}\Pr[s_{t+\gamma_1},\ldots,s_{t+\gamma_n-1}|X_{t+\gamma_1+\gamma_n} = x] = \frac{1}{2}$$

ou que

$$\Pr[X_{t+\gamma_1+\gamma_n} = x | s_{t+\gamma_1+1},\ldots,s_{t+\gamma_n}] = \frac{1}{2} \ .$$

Donc, la distribution de $s_{t+\gamma_n}$ (resp. de $s_{t+\gamma_1}$) est uniforme pour toute valeur fixée de $(s_{t+\gamma_1},\ldots,s_{t+\gamma_n-1})$ (resp. de $(s_{t+\gamma_1+1},\ldots,s_{t+\gamma_n})$) si et seulement si le biais moyen de la restriction de la fonction de filtrage à $x+\langle e_n\rangle$ (resp. à $x+\langle e_1\rangle$) quand x est uniformément distribué dans $\langle e_1,\ldots,e_{n-1}\rangle$ (resp. dans $\langle e_2,\ldots,e_n\rangle$) est nul. Dans le premier cas, cette condition s'écrit donc [LMV05]

$$\sum_{x\in\langle e_1,\ldots,e_{n-1}\rangle} \mathcal{F}^2(f_{x+\langle e_n\rangle}) = 0 \ .$$

Or, on a

$$\begin{aligned}\sum_{x\in\langle e_1,\ldots,e_{n-1}\rangle} \mathcal{F}^2(f_{x+\langle e_n\rangle}) &= \frac{1}{2^{n-1}} \sum_{v\in\langle e_1,\ldots,e_{n-1}\rangle} \mathcal{F}^2(f+\varphi_v) \\ &= \sum_{a\in\langle e_n\rangle} \mathcal{F}(D_a f) \\ &= 2^n + \mathcal{F}(D_{e_n} f) \ .\end{aligned}$$

On en déduit donc que la fonction dérivée $D_{e_n}f : x \mapsto f(x+e_n)+f(x)$ est constante et égale à 1. De même, le second cas implique que $D_{e_1}f = 1$. ◇

b. Son caractère suffisant a été démontré par Golic [Gol96], mais le problème réciproque était jusqu'ici ouvert.

3.2 Distingueurs exploitant des équations de parité

Dans le cas où la fonction de rétroaction du registre à décalage est linéaire, il est possible d'étudier la distribution de paquets formés par plusieurs sorties du générateur prises à certains intervalles de temps supérieurs à la longueur du registre. En effet, de l'équation de récurrence qui régit la suite $(x_t)_{t\geq 0}$ produite par le LFSR, on peut déduire des relations linéaires qui lient certains des éléments de la suite. Ces relations linéaires sont appelées *équations de parité*.

L'utilisation d'équations de parité, notamment de petit poids, pour cryptanalyser les systèmes utilisant des LFSRs remonte aux attaques par corrélation rapide sur les générateurs par combinaison de LFSRs [MS88]. Ces équations sont également le fondement d'une attaque par recouvrement de l'état initial sur les LFSRs filtrés, l'attaque par l'algorithme dit SOJA [Lev04, LZGB03]. Les propriétés sur lesquelles repose le SOJA peuvent être utilisées plus directement dans le cadre d'attaques par distingueur [CHJ02, EJ05, MH04].

3.2.1 Les équations de parité

Équations de parité et codes cycliques. Considérons un LFSR de longueur L et de polynôme caractéristique P, ce qui signifie que P désigne ici le polynôme réciproque du polynôme de rétroaction du LFSR. La plus petite période de toutes les suites produites par ce LFSR, à l'exception de la suite nulle, est l'ordre de P, c'est-à-dire le plus petit entier positif T tel que $P(X)$ divise $X^T - 1$. L'ensemble des suites de longueur T engendrées par ce LFSR forme donc un code cyclique de longueur T et de dimension L, noté $\mathcal{C}_P$, défini par la matrice génératrice suivante :

$$\begin{bmatrix} 1 & 0 & \dots & 0 & c_L & \dots & g_{0,t} & \dots \\ 0 & 1 & \dots & 0 & c_{L-1} & \dots & g_{1,t} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & c_1 & \dots & g_{L-1,t} & \dots \end{bmatrix}$$

où les c_i sont les coefficients de rétroaction du registre. Autrement dit, la colonne d'indice t de cette matrice, représentée sous forme d'un polynôme est donnée par

$$\sum_{i=0}^{L-1} g_{i,t} X^i = X^t \bmod P(X) \ .$$

Une équation de parité pour un code linéaire (c'est-à-dire une relation linéaire liant certaines positions des mots du code) correspond à un ensemble de colonnes de la matrice génératrice dont la somme s'annule — ou de manière équivalente, à un mot du code dual. Comme, dans le cas d'un LFSR, le code dual est cyclique, les équations de parité sont des relations de la forme

$$x_t + x_{t+\tau_1} + \ldots + x_{t+\tau_{w-1}} = 0, \text{ pour tout } t \geq 0$$

satisfaites par toutes les suites $(x_t)_{t\geq 0}$ engendrées par le LFSR. Elles sont trivialement en bijection avec les polynômes

$$1 + X^{\tau_1} + \ldots + X^{\tau_{w-1}}$$

multiples[c] de P. On appelle usuellement poids d'une telle équation son nombre de termes dans la mesure où il correspond au poids de Hamming du mot du dual associé. Par abus de

c. On appelle parfois équations de parité les équations dérivées des multiples du polynôme de rétroaction qui, elles, sont de la forme $x_t + x_{t-\tau_1} + \ldots + x_{t-\tau_{w-1}} = 0$ pour tout $t \geq \tau_{w-1}$.

langage, on parle parfois de son degré pour désigner le degré du polynôme multiple associé, c'est-à-dire la valeur de τ_{w-1}.

Distribution du degré des équations de parité. Un problème fréquent dans le contexte cryptographique est de déterminer, pour un poids donné, le nombre d'équations de parité de degré inférieur à d, quand d varie. Pour en donner une approximation, on estime généralement que, pour w fixé et d suffisamment grand, les valeurs prises par l'ensemble des polynômes de poids w de la forme $(1+X^{\tau_1}+X^{\tau_2}+\ldots+X^{\tau_{w-1}}) \bmod P(X)$ pour $0 < \tau_1 < \tau_2 < \ldots < \tau_{w-1} < d$ sont uniformément distribuées dans l'ensemble des polynômes de degré strictement inférieur à $\deg P$. Sous cette hypothèse, le nombre d'équations de parité de poids w et de degré inférieur ou égal à d est de l'ordre de :

$$m_w(d) = \frac{\binom{d}{w-1}}{2^{\deg P}} \simeq \frac{d^{w-1}}{(w-1)!\, 2^{\deg P}} \ . \tag{3.1}$$

Toutefois, il apparaît clairement que l'hypothèse précédente n'est pas toujours vérifiée. Elle suppose en particulier que le nombre de mots de poids w dans le code $\mathcal{C}_P^\perp$ de longueur T est proche de

$$\frac{T^{w-1}}{w!} \ . \tag{3.2}$$

Or, ce nombre dépend fortement des propriétés algébriques du polynôme caractéristique considéré. On peut notamment distinguer les cas suivants :

- **Lorsque P est primitif.** Le code $\mathcal{C}_P$ est alors un code de longueur $T = 2^{\deg P} - 1$ équivalent au code Simplex. Le code dual $\mathcal{C}_P^\perp$ est donc équivalent au code de Hamming de longueur $2^{\deg P} - 1$, pour lequel on peut vérifier que le nombre de mots de poids w, pour w petit, est proche de celui qui est donné par la formule (3.2) [MS77, Page 129]. Les simulations [CT00] permettent de vérifier que, quand d n'est pas trop petit, la formule (3.1) fournit une bonne approximation du nombre d'équations de parité, même quand P lui-même est de poids w. De la même manière, quand P est un polynôme primitif aléatoire, on estime que le degré minimal d'un polynôme de poids w multiple de P est proche de
$$(w-1)!^{\frac{1}{w-1}} 2^{\frac{\deg P}{w-1}} \ .$$

- **Lorsque P est un produit de polynômes primitifs.** La longueur du code cyclique $\mathcal{C}_P$ est alors
$$T = \mathrm{ppcm}(2^{\deg P_i} - 1) \ .$$
Il apparaît donc clairement que si les degrés des P_i ne sont pas tous deux-à-deux premiers entre eux, l'ensemble
$$\{X^t \bmod P(X), 0 \leq t < T\}$$
ne correspond qu'à une fraction de l'ensemble des polynômes de degré inférieur à $\deg P$. Par exemple, si $P = P_1 P_2$, cette fraction est de l'ordre de $1/(2^{\mathrm{pgcd}(\deg P_1, \deg P_2)} - 1)$ de toutes les valeurs possibles. Aussi, si l'approximation (3.1) reste raisonnable quand les degrés des polynômes P_i sont premiers entre eux, elle paraît plus hasardeuse dans le cas contraire en l'absence de résultats précis sur la distribution des poids du code cyclique $\mathcal{C}_P^\perp$. Un cas extrême de cette situation est celui où tous les P_i sont de même degré m. Alors, on peut associer à un tel produit $P_1 \ldots P_n$ une suite d'entiers $(s_2, \ldots, s_n)$

telle que, pour $2 \leq i \leq n$, chaque P_i est le polynôme minimal de α^{s_i} où $\alpha \in \mathbf{F}_{2^m}$ désigne une racine de P_1. Le code $\mathcal{C}_P^\perp$ est alors le code cyclique de longueur $2^m - 1$ ayant pour ensemble de définition[d] $\{1, s_2, \ldots, s_n\}$. La détermination de l'énumérateur des poids de ce code dans le cas général est un problème ouvert, mais l'on dispose de résultats partiels dans le cas $n = 2$ [Cha98]. On sait notamment que la distance minimale de $\mathcal{C}_P^\perp$ est inférieure ou égale à 5. La détermination du nombre de mots de poids 3 et 4 de $\mathcal{C}_P^\perp$ est liée à la résistance de la fonction puissance $x \mapsto x^s$ sur $\mathbf{F}_{2^m}$ à la cryptanalyse différentielle [CCD00b]. Ceci permet notamment d'identifier des LFSRs ne possédant aucune équation de parité de poids 3 :

Proposition 3.3 *Soit P_1 et P_2 deux polynômes primitifs de degré m à coefficients dans $\mathbf{F}_2$. Soit α une racine de P_1 et s un entier tel que α^s est racine de P_2. Alors, le polynôme P_1P_2 ne possède aucun multiple de poids 3 et tous ses multiples de poids 4 sont de degré strictement supérieur à 2^m-1 si et seulement si la fonction puissance $x \mapsto x^s$ sur $\mathbf{F}_{2^m}$ est APN[e], c'est-à-dire si l'équation $(x+1)^s + x^s = c$ admet au plus deux solutions $x \in \mathbf{F}_{2^m}$ pour toute valeur de $c \in \mathbf{F}_{2^m}$.*

La situation décrite par la proposition précédente est notamment obtenue pour toutes les valeurs de s données aux tables 8.3 et 8.4, page 105.

Algorithmes de recherche des équations de parité. Il existe plusieurs algorithmes permettant de trouver des équations de parité de poids w associées à un polynôme P de degré L.

L'algorithme classique, que nous décrivons dans [CT00], fournit la totalité des polynômes multiples de P de poids w et de degré inférieur ou égal à d, pour un degré d donné. Il dépend d'un paramètre ℓ qui régit le compromis entre les complexités en temps et en mémoire. Cet algorithme consiste à précalculer les valeurs de toutes les combinaisons linéaires de ℓ monômes modulo P, et à les stocker en les ordonnant suivant leurs valeurs. Cette dernière étape peut certes être effectuée par un tri, mais elle est mise en œuvre beaucoup plus efficacement si les valeurs sont stockées au moyen d'une structure de données adaptée, typiquement une table de hachage indexée par les $\log_2 \binom{n}{\ell}$ bits de poids fort des valeurs. Ensuite, pour chaque ensemble Λ_1 de $(w - 1 - \ell)$ monômes, on calcule la valeur de leur somme s, et on détermine grâce à la table précalculée tous les ensembles Λ_2 de ℓ monômes dont la somme vaut $s + 1$. On obtient de cette manière tous les polynômes de la forme $1 + \sum_{i \in \Lambda_1} X^i + \sum_{j \in \Lambda_2} X^j \equiv 0 \bmod P(x)$. La complexité en temps T et en mémoire M de cet algorithme est donc

$$T = \mathcal{O}(d^{w-\ell-1}) \text{ et } M = \mathcal{O}(d^\ell) \ .$$

Un compromis jugé intéressant d'un point de vue théorique — il l'est souvent moins en pratique car la mémoire est actuellement plus « chère » que le temps de calcul — est donc obtenu avec $\ell = \lfloor \frac{w-1}{2} \rfloor$, conduisant à un algorithme de complexité

$$T = \mathcal{O}(d^{\lceil \frac{w-1}{2} \rceil}) \text{ et } M = \mathcal{O}(d^{\lfloor \frac{w-1}{2} \rfloor}) \ .$$

Pour la recherche de multiples de poids supérieur ou égal à 5, une amélioration de cet algorithme décrite dans [CJM02] permet, avec une même complexité en temps, de réduire la complexité en mémoire à $M = \mathcal{O}(d^{\lfloor \frac{w}{4} \rfloor})$.

d. L'ensemble de définition d'un code cyclique de longueur $2^m - 1$ étant une réunion de classes cyclotomiques modulo $2^m - 1$, on ne mentionne ici qu'un seul représentant par classe.

e. Voir la proposition 7.1 page 90.

Ainsi, si l'on considère valide l'approximation (3.1), la recherche d'une équation de parité de poids w pour un polynôme de degré L permet de trouver un polynôme dont le degré est de l'ordre de

$$d \simeq 2^{\frac{L}{w-1}} .$$

En remplaçant d par cette valeur approchée dans les formules précédentes, on déduit que cette recherche a pour complexité

$$M \simeq 2^{\frac{L}{4}} \text{ et } T \simeq \begin{cases} 2^{\frac{L}{2}} & \text{pour } w \text{ impair} \\ 2^{\frac{L}{2-\frac{2}{w}}} & \text{pour } w \text{ pair} \end{cases} .$$

On constate ainsi que le temps de calcul nécessaire à la recherche d'un multiple de poids strictement inférieur au poids du polynôme P et de degré minimal est supérieur ou égal à $2^{\frac{L}{2}}$, ce qui rend parfois cette recherche hors de portée. On est donc amené dans ce dernier cas à utiliser un algorithme probabiliste moins coûteux fondé sur le paradoxe des anniversaires généralisé [Wag02], mais qui ne fournit pas le polynôme de degré minimal [Mol04, LV04b].

3.2.2 Distingueur d'un LFSR filtré utilisant une équation de parité

L'utilisation d'une équation de parité de poids faible dans une attaque par distingueur sur les LFSRs filtrés a été proposée en 2004, indépendamment par Molland et Helleseth [MH04] et Englund et Johansson [EJ05]. Nous détaillons ici la complexité de cette attaque en fonction du poids w de l'équation utilisée, ce qui nous permet de mettre en évidence l'influence du moment d'ordre w du spectre de Walsh de la fonction de filtrage dans ce contexte.

Lemme 3.4 *Soit f une fonction booléenne équilibrée à n variables, w un entier, $w \geq 3$, et x un élément de $\mathbf{F}_2^n$. Soit F_x la fonction*

$$\begin{array}{rccl} F_x : & \mathbf{F}_2^{(w-2)n} & \to & \mathbf{F}_2^{w-1} \\ & (x_1,\ldots,x_{w-2}) & \mapsto & (f(x_1),\ldots,f(x_{w-2}),f(x+x_1+\ldots+x_{w-2})). \end{array}$$

Alors, la distribution de F_x est donnée par :

$$|F_x^{-1}(y_1,\ldots,y_{w-1})| = \frac{1}{2^{w-1}} \left[2^{(w-2)n} + (-1)^{y_1+\ldots+y_{w-1}} 2^{-n} \sum_{\lambda \in \mathbf{F}_2^n} (-1)^{\lambda \cdot x} \mathcal{F}^{w-1}(f+\varphi_\lambda) \right]$$

pour tout $(w-1)$-uplet $(y_1,\ldots,y_{w-1})$.

Preuve. On utilise le fait que

$$\sum_{\lambda \in \mathbf{F}_2^n} (-1)^{\lambda \cdot (x+x_1+\ldots+x_{w-1})} = \begin{cases} 2^n & \text{si } x_{w-1} = x+x_1+\ldots+x_{w-2} \\ 0 & \text{sinon.} \end{cases}$$

On a alors, pour f équilibrée,

$$\begin{aligned} A &= |F_x^{-1}(y_1,\ldots,y_{w-1})| \\ &= \frac{1}{2^{w-1}} \left[\sum_{x_1,\ldots,x_{w-1} \in \mathbf{F}_2^n} [1+(-1)^{f(x_1)+y_1}]\ldots[1+(-1)^{f(x_{w-1})+y_{w-1}}] \left(2^{-n} \sum_{\lambda \in \mathbf{F}_2^n} (-1)^{\lambda \cdot (x+x_1+\ldots+x_{w-1})} \right) \right] \\ &= \frac{1}{2^{w-1}} \left[2^{(w-2)n} + (-1)^{y_1+\ldots+y_{w-1}} 2^{-n} \sum_{\lambda \in \mathbf{F}_2^n} (-1)^{\lambda \cdot x} \left(\sum_{x_i \in \mathbf{F}_2^n} (-1)^{f(x_i)+\lambda \cdot x_i} \right)^{w-1} \right] . \end{aligned}$$

$\diamond$

On en déduit alors la proposition suivante montrant que la distribution des w-uplets formés par les bits de suite chiffrante liés par une équation de parité est déterminée par le moment d'ordre w du spectre de Walsh de la fonction de filtrage.

Proposition 3.5 *Soit $(s_t)_{t\geq 0}$ la suite produite par un LFSR de polynôme caractéristique P filtré par une fonction équilibrée f à n variables. Soit $1 + X^{\tau_1} + \ldots + X^{\tau_{w-1}}$ un polynôme multiple de $P(X)$. Alors, la distribution du w-uplet $(s_t,s_{t+\tau_1},\ldots,s_{t+\tau_{w-1}})$ est donnée par*

$$\begin{aligned}\Pr[(s_t,s_{t+\tau_1},\ldots,s_{t+\tau_{w-1}}) = (\sigma_0,\ldots,\sigma_{w-1})] &= \frac{1}{2^{w-1}}\Pr[s_t + \ldots + s_{t+\tau_{w-1}} = \sigma_0 + \ldots + \sigma_{w-1}] \\ &= \frac{1}{2^w}\left[1 + (-1)^{\sigma_0+\ldots+\sigma_{w-1}}\Delta_w\right]\end{aligned}$$

avec

$$\Delta_w = 2^{-wn}\sum_{\lambda\in\mathbf{F}_2^n}\mathcal{F}^w(f+\varphi_\lambda)\ .$$

On en déduit le résultat suivant, exprimé de façon légèrement différente dans [MH04] et dans [EJ05].

Corollaire 3.6 *Avec les notations précédentes, la complexité en temps (T_w) et en données (D_w) de l'attaque qui consiste à distinguer la distribution du w-uplet $(s_t,s_{t+\tau_1},\ldots,s_{t+\tau_{w-1}})$ d'une suite aléatoire est*

$$T_w = w\frac{1}{\Delta_w^2}\ \textit{et}\ D_w = \frac{1}{\Delta_w^2} + \tau_{w-1}$$

où

$$\Delta_w = 2^{-wn}\sum_{\lambda\in\mathbf{F}_2^n}\mathcal{F}^w(f+\varphi_\lambda)$$

sous la condition que $|\Delta_w| \ll 1$. En particulier, si P est un polynôme primitif aléatoire de degré L, on a

$$D_w \simeq \frac{1}{\Delta_w^2} + 2^{\frac{L}{w-1}}\ .$$

Ce corollaire omet naturellement la complexité de la phase de précalcul qui détermine l'équation de parité. Il est important de noter que les deux termes intervenant dans la formule de la complexité en données de l'attaque jouent des rôles relativement différents. En effet, dans le contexte où l'on dispose de plusieurs trames, chacune de longueur D_T, l'attaque ne peut être menée à bien que si la longueur de trame est supérieure ou égale à τ_{w-1}. Dans ce cas, le nombre de trames nécessaires est égal à

$$\frac{1}{\Delta_w^2(D_T - \tau_{w-1})}\ .$$

On peut maintenant s'intéresser aux attaques correspondant aux différentes valeurs de w.

Utilisation d'une équation de poids 2. Le cas $w = 2$ est un cas extrémal puisque l'équation de parité est donnée par une période de la suite engendrée par le LFSR. On a alors trivialement

$$Pr[s_t + s_{t+\tau_1} = 0] = 1 \ ,$$

c'est-à-dire $\Delta_2 = 1$. Ce cas n'entre évidemment pas dans le cas du corollaire précédent et le distingueur utilisé pour l'attaque n'est pas le distingueur classique qui permet de détecter une distribution relativement proche de la distribution uniforme. On utilise au contraire un distingueur similaire à celui employé notamment dans la cryptanalyse par différentielle impossible. La complexité en données est naturellement déterminée par la période de la suite, et est de l'ordre de 2^L.

Utilisation d'une équation de poids 4. Dans ce cas, le biais Δ_4 est donné par la valeur d'un paramètre introduit par Zhang et Zheng [ZZ95] pour évaluer certaines propriétés cryptographiques des fonctions booléennes sous le nom de *sum-of-square indicator* : il s'agit en effet de la somme des carrés des biais des dérivées de la fonction. Comme nous l'avons montré dans [CCCF00], cette quantité peut s'exprimer à la fois en termes des dérivées de la fonction et de ses coefficients de Walsh.

Proposition 3.7 *[CCCF00] Soit f une fonction booléenne à n variables. Son indicateur par somme des carrés (*sum-of-square indicator*), noté $\nu(f)$, est défini par*

$$\nu(f) = \sum_{a \in \mathbf{F}_2^n} \mathcal{F}^2(D_a f) = 2^{-n} \sum_{\lambda \in \mathbf{F}_2^n} \mathcal{F}^4(f + \varphi_\lambda) \ .$$

On voit donc, avec les notations du corollaire 3.6 que la complexité de l'attaque par distingueur fondé sur une équation de poids 4 est déterminée par

$$\Delta_4 = \frac{1}{2^{3n}} \nu(f) \ .$$

Dans les articles [CCCF00, CCCF01, BCCLC06], nous avons mené une étude détaillée sur la valeur de cet indicateur dans divers contextes, et sur ses relations avec d'autres propriétés cryptographiques. Nous ne reprenons ici que certains points, importants dans le cadre de l'attaque par distingueur considérée. L'étude de cet indicateur fait également jouer un rôle particulier à la famille des fonctions booléennes dont le spectre de Walsh étendu (c'est-à-dire symétrisé) contient au plus 3 valeurs distinctes. Ces fonctions ont été introduites indépendamment sous le nom de fonctions *plateaux* par Zhang et Zheng [ZZ99b], et sous le nom de fonctions *3-valuées* dans [CCCF00].

Définition 3.8 *Une fonction booléenne f est dite* plateau *si tous ses coefficients de Walsh sont à valeurs dans $\{0, \pm\mathcal{L}(f)\}$.*

Cette définition inclut notamment les fonctions courbes, qui sont celles dont tous les coefficients de Walsh sont à valeurs dans $\{\pm 2^{n/2}\}$, les fonctions partiellement courbes définies dans [Car93] et les fonctions quadratiques.

Proposition 3.9 *[CCCF00, Th. 2] Soit f une fonction booléenne plateau à n variables. Alors, $\mathcal{L}(f) = 2^s$ pour un entier $s \geq n/2$ et le spectre de Walsh de f est donné par*

$\mathcal{F}(f + \varphi_u)$	*nombre de* $u \in \mathbf{F}_2^n$
0	$2^n - 2^{2n-2s}$
2^s	$2^{2n-2s-1} + (-1)^{f(0)} 2^{n-s-1}$
-2^s	$2^{2n-2s-1} - (-1)^{f(0)} 2^{n-s-1}$

La valeur de l'indicateur par somme des carrés d'une fonction à n variables vérifie trivialement

$$2^{2n} \leq \nu(f) \leq 2^{3n}$$

où la borne inférieure est atteinte si et seulement si f est courbe, et la borne supérieure si et seulement si f est de degré inférieur ou égal à 1. Toutefois, nous donnons dans [CCCF00] une borne supérieure plus précise qui dépend de la non-linéarité de f.

Théorème 3.10 *[CCCF00, Th. 1] Soit f une fonction booléenne à n variables. Alors,*

$$\nu(f) \leq 2^n \mathcal{L}(f)^2$$

avec égalité si et seulement si f est une fonction plateau.

On en déduit donc que la complexité de l'attaque par distingueur utilisant une équation de parité de poids 4 est d'autant plus grande que la non-linéarité de la fonction de filtrage est faible. Par ailleurs, nous avons vu à la proposition 3.2 que, pour que la fonction augmentée associée au registre filtré soit équilibrée, la fonction f devait posséder une structure linéaire de type 1. Dans ce cas, nous pouvons appliquer le résultat suivant, déduit du théorème 3 de [CCCF00].

Proposition 3.11 *[CCCF00] Soit f une fonction de filtrage à n variables possédant une structure linéaire,* i.e. *équivalente par transformation linéaire à une fonction du type*

$$f(x_1, \ldots, x_n) = \varepsilon x_n + g(x_1, \ldots, x_{n-1}) \ , \ \varepsilon \in \mathbf{F}_2.$$

Alors,

$$\mathcal{L}(f) \geq 2^{\frac{n+1}{2}} \ \textit{et} \ \nu(f) = 8\nu(g) \geq 2^{2n+1}$$

avec égalité dans les deux formules si et seulement si g est courbe, ce qui n'est possible que si n est impair.

On obtient alors une borne supérieure sur la complexité de l'attaque donnée par le corollaire 3.6 dans le cas où, conformément à la proposition 3.2, f possède une structure linéaire.

Corollaire 3.12 *Si la fonction de filtrage f possède une structure linéaire, notamment si elle est équivalente par transformation affine à une fonction du type*

$$f(x_1, \ldots, x_n) = x_n + g(x_1, \ldots, x_{n-1}) \ ,$$

alors la complexité en temps et en données de l'attaque par distingueur utilisant une équation de parité de poids 4 vérifie

$$T_4 \leq 2^{2n-2} \ \textit{et} \ D_4 \leq 2^{2n-2} + 2^{\frac{L}{3}}$$

avec égalité si et seulement si g est courbe. En particulier, si l'on omet le coût de la recherche de l'équation de parité, la résistance à cette attaque impose les tailles minimales suivantes pour le LFSR et le nombre de variables de la fonction de filtrage :

$$L \geq 3k \ \textit{ou} \ n > \frac{k}{2} \ ,$$

où k est la taille de la clef.

Une autre famille de fonctions booléennes pour lesquelles nous avons établi certains résultats sur la valeur de l'indicateur $\nu(f)$ est celle des composantes des fonctions puissances, c'est-à-dire

$$f : x \mapsto \mathrm{Tr}(\lambda x^s)$$

sur $\mathbf{F}_{2^n}$. Comme nous le verrons par la suite, il s'agit de bonnes candidates pour les fonctions de filtrage (pour la fonction f ou pour la fonction g) en raison de leur facilité d'implémentation. La valeur de $\nu(f)$ est alors liée aux propriétés de résistance de la fonction puissance $x \mapsto x^s$ à la cryptanalyse différentielle (voir les chapitres 6 et 8).

Comparaison des attaques quand le poids des équations varie. D'après le corollaire 3.12, l'attaque décrite précédemment avec une équation de poids 4 est clairement inopérante quand la taille du registre dépasse le triple de la taille de la clef. Dans ce cas, il faut utiliser un multiple du polynôme de rétroaction de degré inférieur à 2^k où k est la taille de la clef secrète. Ce multiple doit donc être recherché parmi les polynômes de poids plus élevé. Le problème est alors de déterminer si le biais dans la distribution du w-uplet étudié est encore suffisamment élevé pour que la complexité en temps de l'attaque ne dépasse pas celle de la recherche exhaustive. Pour cela, nous donnons des bornes sur les différents moments du spectre de Walsh d'une fonction booléenne.

Par une technique similaire à celle utilisée dans le théorème 1 de [CCCF00], on obtient la relation suivante entre les biais induits par les équations de parité de poids $2p$ et de poids $2p+2$.

Proposition 3.13 *Pour tout $p \geq 1$, on a*

$$\Delta_{2p+2} \leq \Delta_{2p} \left[\frac{\mathcal{L}(f)}{2^n}\right]^2$$

avec égalité si et seulement si f est une fonction plateau. En particulier,

$$\Delta_{2p} \leq \left[\frac{\mathcal{L}(f)}{2^n}\right]^{2(p-1)} .$$

Preuve.

$$\begin{aligned} 2^{2p}\mathcal{L}^2(f)\Delta_{2p} - 2^{2p+2}\Delta_{2p+2} &= \mathcal{L}^2(f) \sum_{\lambda \in \mathbf{F}_2^n} \mathcal{F}^{2p}(f+\varphi_\lambda) - \sum_{\lambda \in \mathbf{F}_2^n} \mathcal{F}^{2p+2}(f+\varphi_\lambda) \\ &= \sum_{\lambda \in \mathbf{F}_2^n} \mathcal{F}^{2p}(f+\varphi_\lambda) \left[\mathcal{L}^2(f) - \mathcal{F}^2(f+\varphi_\lambda)\right] . \end{aligned}$$

Cette dernière somme est donc positive puisque chacun de ses termes est positif. Elle est nulle si et seulement si chacun des termes est nul, c'est-à-dire si

$$\mathcal{F}^2(f+\varphi_\lambda) \in \{0, \mathcal{L}^2(f)\} .$$

◇

Le théorème suivant, partiellement démontré dans [ZZ99a] fournit, quant à lui, une borne inférieure de Δ_w quand w est pair.

Théorème 3.14 *Pour tout $k \geq 1$, on a*

$$\Delta_{4k} \geq \frac{\Delta_{2k}^2}{|\{\lambda \in \mathbf{F}_2^n, \mathcal{F}(f+\varphi_\lambda) \neq 0\}|}$$

et

$$\Delta_{4k+2} \geq \Delta_{2k+2}^2$$

avec égalité dans les deux formules si et seulement si f est une fonction plateau. On en déduit en particulier que, pour tout $p \geq 1$,

$$\Delta_{2p} \geq 2^{n(1-p)}$$

avec égalité si et seulement si f est courbe.

Preuve. Quand p est pair, $p = 2k$, l'inégalité de Cauchy-Schwarz donne

$$|\{\lambda \in \mathbf{F}_2^n, \mathcal{F}(f+\varphi_\lambda) \neq 0\}| \sum_{\lambda \in \mathbf{F}_2^n} \mathcal{F}^{4k}(f+\varphi_\lambda) \geq \left(\sum_{\lambda \in \mathbf{F}_2^n} \mathcal{F}^{2k}(f+\varphi_\lambda) \right)^2$$

avec égalité si et seulement si toutes les valeurs non nulles de $\mathcal{F}^{2k}(f+\varphi_\lambda)$ sont égales, c'est-à-dire si f est une fonction plateau. Quand p est impair, $p = 2k+1$, l'inégalité de Hölder donne

$$\sum_{\lambda \in \mathbf{F}_2^n} \mathcal{F}^{2}(f+\varphi_\lambda) \sum_{\lambda \in \mathbf{F}_2^n} \mathcal{F}^{4k+2}(f+\varphi_\lambda) \geq \left(\sum_{\lambda \in \mathbf{F}_2^n} \mathcal{F}^{2k+2}(f+\varphi_\lambda) \right)^2 .$$

La deuxième borne inférieure correspond à la valeur de Δ_{2p} quand f est une fonction courbe, et est obtenue par récurrence grâce au résultat précédent. ◇

Il est par contre plus délicat de donner un encadrement de la valeur de Δ_w quand w est impair. Molland et Helleseth [MH04] expliquent que l'utilisation d'une équation de poids impair $2p-1$ n'a que très peu d'intérêt car elle conduit à un biais qui est en moyenne égal à celui induit par l'équation de poids $2p$. On peut également vérifier que toute fonction plateau satisfait

$$\Delta_{2p-1} = \Delta_{2p}, \ \forall p \geq 2,$$

mais il nous semble difficile d'obtenir plus de résultats dans le cas impair — la difficulté venant alors essentiellement du fait que les différents termes intervenant dans la somme sont de signes différents. Toutefois, nous pouvons utiliser les bornes du cas pair pour obtenir une bonne estimation de la complexité de l'attaque générale.

Corollaire 3.15 *La complexité en temps (T_{2p}) et en données (D_{2p}) de l'attaque par distingueur utilisant une équation de parité de poids $2p$ vérifie*

$$\left(\frac{2^n}{\mathcal{L}(f)} \right)^{4(p-1)} \leq T_{2p} \leq 2^{2n(p-1)}$$

et

$$\left(\frac{2^n}{\mathcal{L}(f)} \right)^{4(p-1)} + 2^{\frac{L}{2p-1}} \leq D_{2p} \leq 2^{2n(p-1)} + 2^{\frac{L}{2p-1}} ,$$

avec égalité avec les bornes inférieures si et seulement si f est une fonction plateau.

De plus, si la fonction de filtrage f possède une structure linéaire, notamment si elle est équivalente par transformation affine à une fonction du type

$$f(x_1,\ldots,x_n) = x_n + g(x_1,\ldots,x_{n-1}) \ ,$$

alors

$$T_{2p} \leq 2^{2(n-1)(p-1)} \ \textit{et}\ D_{2p} \leq 2^{2(n-1)(p-1)} + 2^{\frac{L}{2p-1}}$$

avec égalité si et seulement si g est courbe.

On voit donc que le choix d'une fonction g courbe est optimal au regard de cette attaque par distingueur, quel que soit le poids de l'équation de parité utilisée. Dans ce cas, on voit aisément que la valeur optimale du poids de l'équation est celle pour laquelle les deux termes intervenant dans la complexité en données sont comparables.

Corollaire 3.16 *Si f est équivalente par transformation affine à une fonction du type*

$$f(x_1,\ldots,x_n) = x_n + g(x_1,\ldots,x_{n-1}) \ ,$$

où g est courbe, alors l'attaque par distingueur la plus efficace est obtenue avec une équation de poids w où

$$\sqrt{\frac{L}{n-1}} + 1 \leq w \leq \sqrt{\frac{L}{n-1}} + 2 \ .$$

Les complexités en temps et en données de l'attaque sont alors du même ordre et proches de

$$2^{\sqrt{L(n-1)}} \ .$$

On en déduit donc (si l'on omet le coût du précalcul) que, pour éviter cette attaque par distingueur, la longueur du LFSR et le nombre de variables de la suite chiffrante doivent être tels que la quantité $\sqrt{L(n-1)}$ excède la taille de la clef.

Enfin, le corollaire 3.15 est également utile pour déterminer le poids minimal admissible pour le polynôme de rétroaction du LFSR. En effet, quand w correspond au poids de P, le second terme dans l'expression de la complexité en données disparaît, puisqu'il correspond au degré du multiple de P utilisé.

3.2.3 Distingueur d'un générateur par combinaison utilisant une équation de parité

On peut également utiliser des équations de parité pour monter une attaque par distingueur sur un système par combinaison de LFSRs. Les équations de parité employées ici sont des relations vérifiées par la sortie d'un ou d'un petit nombre des LFSRs constituants. Un distingueur de ce type a été proposé pour la première fois dans [CF01], mais dans un contexte différent que nous détaillerons dans la section suivante.

Dans toute la suite, on considère un générateur composé de n LFSRs combinés par une fonction booléenne f à n variables. On s'intéresse alors à une équation de parité de poids w vérifiée simultanément par les ℓ LFSRs d'indice $I = \{i_1,\ldots,i_\ell\}$ du système. Autrement dit, cette équation correspond à un polynôme $1 + X^{\tau_1} + \ldots + X^{\tau_{w-1}}$ multiple du ppcm des ℓ polynômes caractéristiques des registres considérés. Le biais induit par cette équation de parité

dans la distribution du w-uplet $(s_t,s_{t+\tau_1},\ldots,s_{t+\tau_{w-1}})$ peut être calculé à l'aide du lemme suivant, appliqué au cas où V est l'espace vectoriel de dimension ℓ engendré par les vecteurs e_i, $i \in I$. Sa preuve est similaire à celle du lemme 3.4.

Lemme 3.17 *Soit f une fonction booléenne équilibrée à n variables, w un entier, $w \geq 3$, et x un élément de $\mathbf{F}_2^n$. Soit V et W deux espaces en somme directe avec $\dim V = \ell$. Soit F_x la fonction*

$$\begin{array}{lcl} (V \times W)^{w-2} \times W & \to & \mathbf{F}_2^{w-1} \\ ((x_1,y_1),\ldots,(x_{w-2},y_{w-2}),y_{w-1}) & \mapsto & (f(x_1+y_1),\ldots,f(x_{w-2}+y_{w-2}),f(x+x_1+\ldots+x_{w-2}+y_{w-1})) \end{array} .$$

Alors, la distribution de F_x est donnée par :

$$|F_x^{-1}(z_1,\ldots,z_{w-1})| = \frac{1}{2^{w-1+\ell}} \left[2^{(w-1)n} + (-1)^{z_1+\ldots+z_{w-1}} \sum_{\lambda \in V} (-1)^{\lambda \cdot x} \mathcal{F}^{w-1}(f+\varphi_\lambda) \right]$$

pour tout $(w-1)$-uplet $(z_1,\ldots,z_{w-1})$.

Proposition 3.18 *[BL06] Soit $(s_t)_{t\geq 0}$ la suite produite par un générateur par combinaison de n LFSRs et $1 + X^{\tau_1} + \ldots + X^{\tau_{w-1}}$ un polynôme multiple du ppcm des P_i, $i \in I$. Alors, la distribution du w-uplet $(s_t,s_{t+\tau_1},\ldots,s_{t+\tau_{w-1}})$ est donnée par*

$$\begin{array}{rcl} \Pr[(s_t,s_{t+\tau_1},\ldots,s_{t+\tau_{w-1}}) = (\sigma_0,\ldots,\sigma_{w-1})] & = & \frac{1}{2^{w-1}} \Pr[s_t + \ldots + s_{t+\tau_{w-1}} = \sigma_0 + \ldots + \sigma_{w-1}] \\ & = & \frac{1}{2^w} \left[1 + (-1)^{\sigma_0+\ldots+\sigma_{w-1}} \Delta_w(V) \right] \end{array}$$

avec

$$\Delta_w(V) = 2^{-wn} \sum_{\lambda \in V} \mathcal{F}^w(f+\varphi_\lambda) \text{ et } V = \langle e_i, i \in I \rangle .$$

L'attaque est donc naturellement impossible si le nombre ℓ de registres mis en jeu est inférieur ou égal à l'ordre de résilience de la fonction de combinaison.

Corollaire 3.19 *Avec les notations précédentes, la complexité en temps (T_w) et en données (D_w) de l'attaque qui consiste à distinguer la distribution du w-uplet $(s_t,s_{t+\tau_1},\ldots,s_{t+\tau_{w-1}})$ d'une suite aléatoire est*

$$T_w = \frac{1}{\Delta_w^2(V)} \text{ et } D_w = \frac{1}{\Delta_w^2(V)} + \tau_{w-1}$$

avec

$$\Delta_w(V) = 2^{-wn} \sum_{\lambda \in V} \mathcal{F}^w(f+\varphi_\lambda)$$

quand $|\Delta_w(V)| \ll 1$.

Utilisation d'une équation de poids 2. L'attaque avec $w = 2$ s'applique à tous les types de générateurs par combinaison, même quand les suites combinées ne sont pas engendrées par des LFSRs. En effet, l'équation de parité de poids 2 est alors donnée par

$$1 + X^\tau \text{ où } \tau = \text{ppcm}(T_i, i \in I) \ .$$

Contrairement au cas d'un système filtré, cette attaque peut avoir une complexité inférieure à celle de la recherche exhaustive. Il suffit qu'il existe $t+1$ suites constituantes, où t est l'ordre de résilience de f dont le produit des périodes est inférieur ou égal à la taille de la clef.

Corollaire 3.20 *Considérons un générateur dont la sortie est produite par la combinaison de n suites de périodes respectives T_i par une fonction booléenne f t-résiliente. L'attaque par distingueur optimale utilisant une équation de parité de poids 2 sur ce générateur a pour complexité la valeur minimale atteignable par le ppcm des périodes de $(t+1)$ suites constituantes, en supposant que cette valeur est supérieure ou égale à $2^{4(n-t-2)}$.*

Preuve. La complexité de l'attaque sur un sous-ensemble de suites de cardinal $(t+1)$, indexées par I, est

$$D_2 = \left(\frac{2^{2n}}{\sum_{\lambda \in V} \mathcal{F}^2(f + \varphi_\lambda)} \right)^2 + \text{ppcm}_{i \in I} T_i \ .$$

Le terme dominant dans cette formule est celui de droite puisque le terme de gauche vérifie

$$\frac{2^{2n}}{\sum_{\lambda \in V} \mathcal{F}^2(f + \varphi_\lambda)} \leq 2^{2(n-t-2)} \ .$$

En effet, seul un coefficient de Walsh est non nul dans cette somme, et tous les coefficients de Walsh d'une fonction t-résiliente sont divisibles par 2^{t+2} [SM00b]. Pour cette raison, considérer $(t+2)$ suites simultanément conduit nécessairement à une attaque plus coûteuse car le terme de droite dans la formule de la complexité augmenterait. ◇

Notons que, si les suites constituantes sont elles-mêmes produites par des générateurs de périodes maximales premières entre elles, la condition précédente équivaut à dire que la taille moyenne de leur état interne L est minorée par

$$L \geq \frac{4(n-1)}{t+1} - 4 \ ,$$

ce qui est pratiquement toujours le cas puisque l'état interne est généralement très grand devant le nombre de variables utilisées, en particulier pour résister aux attaques par corrélation.

Valeur optimale du poids de l'équation de parité. Dans le cas où les suites combinées sont des suites récurrentes linéaires, l'attaquant peut choisir la valeur de w qui conduit au distingueur de complexité minimale. Par contre, le nombre ℓ de LFSRs attaqués simultanément sera toujours choisi minimal car la complexité de l'étape de précalcul (c'est-à-dire de la recherche des équations de parité) est de l'ordre de la racine carrée du ppcm des périodes des suites considérées. La valeur de w optimale est alors celle pour laquelle les deux termes de la formule donnant la complexité en données de l'attaque sont du même ordre de grandeur.

Corollaire 3.21 *Considérons un générateur dont la sortie est produite par la combinaison des sorties de n LFSRs de longueurs respectives L_i par une fonction booléenne f t-résiliente.*

Parmi les sous-ensembles J de $\{1,\ldots,n\}$ de taille $(t+1)$ vérifiant[f] $\mathcal{F}(f+\varphi_{1_J}) \neq 0$, *on note I celui qui minimise la valeur de*

$$\sum_{i\in I} L_i(n - \log_2 |\mathcal{F}(f+\varphi_{1_I})|) \ .$$

Alors l'attaque par distingueur optimale utilise une équation de parité vérifiée simultanément par les $(t+1)$ LFSRs de I et de poids

$$w \simeq 1 + \sqrt{\frac{\sum_{i\in I} L_i}{2(n - \log_2 |\mathcal{F}(f+\varphi_{1_I})|)}} \ .$$

Sa complexité en temps et en données est de l'ordre de

$$D = T = 2^{\sqrt{2\sum_{i\in I} L_i(n - \log_2 |\mathcal{F}(f+\varphi_{1_I})|)}}$$

et vérifie notamment

$$\sqrt{2(n - \log_2 \mathcal{L}(f)) \left(\sum_{i\in I} L_i\right)} \leq \log_2 T \leq \sqrt{2(n-t-2)\left(\sum_{i\in I} L_i\right)} \ .$$

L'attaque nécessite un précalcul de complexité de l'ordre de

$$2^{\frac{\sum_{i\in I} L_i}{2}} \ .$$

On remarque donc que le facteur limitant ici est évidemment l'étape de précalcul. Pour résister à cette attaque, notamment pour que la phase de précalcul soit plus coûteuse que la recherche exhaustive de la clef, l'ordre de résilience t de la fonction de combinaison et la longueur moyenne L des LFSRs utilisés doivent vérifier

$$(t+1)L \geq 2k$$

où k est la taille de la clef secrète. On voit ici que l'implémentation d'un tel système est peu économique puisque l'ordre de résilience de la fonction de combinaison est limité par le fait que son degré doit être suffisamment élevé pour résister aux attaques algébriques[g]. Par exemple, dans un générateur par combinaison composé de 5 LFSRs utilisant une fonction de degré supérieur ou égal à 3, la taille moyenne des registres est de l'ordre de la taille de la clef, ce qui correspond à un état interne de taille $5k$. Si l'on impose que le degré de f soit supérieur ou égal à 4, un générateur composé de 8 LFSRs doit avoir un état interne de taille totale supérieure à 4 fois la taille de la clef secrète.

3.2.4 Reconstruction des spécifications d'un générateur à base de LFSRs

Nous avons utilisé le distingueur précédent pour la première fois dans [CF01], mais dans un contexte différent puisque notre objectif était alors de retrouver les spécifications d'un tel système, c'est-à-dire les polynômes de rétroaction et la fonction de combinaison, à partir

f. 1_J désigne ici le vecteur de $\mathbf{F}_2^n$ dont le support est l'ensemble J, c'est-à-dire $1_J = \sum_{j\in J} e_j$ où e_j est le j-ième vecteur de la base canonique.

g. En effet, l'ordre de résilience t d'une fonction f à n variables vérifie $t \leq n - 1 - \deg f$.

de la seule connaissance de la suite chiffrante (ou du chiffré correspondant à un texte clair possédant une redondance suffisante). Ces outils sont également employés dans les problèmes plus généraux de « reverse-engineering », dont l'objet est de retrouver les spécifications des différents éléments d'une chaîne de communication à partir de données interceptées (éventuellement chiffrées ou bruitées). Ainsi, les algorithmes de reconnaissance de brasseurs linéaires au sein d'une chaîne de communication développés par M. Cluzeau [Clu03, Clu04] reposent sur le même principe.

Supposons qu'un attaquant dispose de N_T trames de chiffré de longueur D_T chacune, correspondant à un texte clair inconnu, mais biaisé, au sens où il est produit par une source binaire sans mémoire émettant 0 avec une probabilité $p_0 > 1/2$ — une attaque à clair connu correspond alors à $p_0 = 1$. L'algorithme utilisé pour retrouver les polynômes caractéristiques des différents LFSRs composant un générateur par combinaison consiste à estimer la distribution de toutes les combinaisons linéaires de w bits du chiffré

$$c_t + c_{t+\tau_1} + \ldots + c_{t+\tau_{w-1}} \ .$$

Si l'on détecte un $(w-1)$-uplet $(\tau_1, \ldots, \tau_{w-1})$ tel que cette distribution n'est pas uniforme, c'est qu'il correspond à un multiple d'un ou de plusieurs des polynômes caractéristiques des LFSRs utilisés. Cette attaque est décrite précisément à la table 3.1. Le seuil T utilisé dans

Pour chaque $(w-1)$-uplet $(\tau_1, \ldots, \tau_{w-1})$ avec $0 < \tau_1 < \ldots < \tau_{w-1} < D_T$
    $Z \leftarrow 0$
    Pour chaque trame de chiffré $\boldsymbol{c} = (c_0, \ldots, c_{D_T-1})$
        Pour t de 0 à $D_T + \tau_{w-1} - 1$
            $z \leftarrow c_t + c_{t+\tau_1} + \ldots + c_{t+\tau_{w-1}}$
            $Z \leftarrow Z + (-1)^z$
    Si $|Z| > T$, retourner la valeur Z et $1 + X^{\tau_1} + \ldots + X^{\tau_{w-1}}$.

TAB. 3.1 – *Recherche des polynômes caractéristiques utilisés dans un générateur par combinaison de LFSRs*

l'algorithme de la table 3.1 est déterminé de manière à obtenir des probabilités de fausse alarme et de non-détection adéquates pour détecter un biais supérieur ou égal à ε où

$$\varepsilon = \frac{|2p_0 - 1|}{2^{n-1}}$$

et n est le nombre de variables estimé de la fonction de combinaison. L'algorithme fournit ainsi tous les polynômes de poids w multiples des produits $\prod_{i \in I} P_i$ pour tous les ensembles I de taille supérieure ou égale à $t+1$ et tel que $\mathcal{F}(f + \varphi_{1_I}) \neq 0$. La valeur du paramètre w doit être choisie de sorte que le degré du polynôme multiple recherché soit inférieur à la longueur de trame disponible, c'est-à-dire

$$\frac{L(t+1)}{w-1} \leq \log_2 D_T$$

où L est la longueur moyenne des registres utilisés et t l'ordre de résilience de la fonction. En pratique, si les valeurs de L et de t sont inconnues de l'attaquant, on peut exécuter l'algorithme avec des valeurs croissantes de w jusqu'à ce qu'un polynôme soit détecté. La complexité en temps de cet algorithme correspond à celle de l'énumération de tous les polynômes de poids $(w-1)$ et de degré inférieur ou égal à $2^{L(t+1)/(w-1)}$, c'est-à-dire de l'ordre de $2^{L(t+1)}$.

La valeur de Z retournée par l'algorithme fournit, elle, une approximation de $\frac{\mathcal{F}(f+\varphi_{1_I})}{2^n}$. Une fois les polynômes caractéristiques retrouvés, les valeurs des autres coefficients de Walsh peuvent alors être estimées en exécutant l'algorithme précédent pour un polynôme de poids raisonnable multiple de $\prod_{i\in I} P_i$ où I décrit les sous-ensembles de $\{1,\ldots,n\}$ par ordre de taille croissant. Une transformée de Fourier inverse permet alors de retrouver la fonction de combinaison.

Cette technique peut aussi être appliquée au cas du LFSR filtré, mais la complexité en temps de l'algorithme permettant de retrouver le polynôme de rétroaction du LFSR est alors de l'ordre de 2^L où L est la longueur du registre.

3.2.5 Distingueur d'un LFSR filtré à plusieurs équations de parité

Comme l'ont remarqué Englund et Johansson [EJ05], il est possible d'améliorer l'attaque par distingueur sur les LFSRs filtrés quand on dispose de plusieurs équations de parité de poids donné.

Pour m équations de poids w associées aux polynômes

$$1 + X^{\tau_{1,i}} + \ldots + X^{\tau_{w-1,i}}, 1 \leq i \leq m \ ,$$

l'attaque consiste alors à comparer la distribution du $(m+1)$-uplet

$$\left(s_t, \sum_{j=1}^{w-1} s_{t+\tau_{j,1}}, \ldots, \sum_{j=1}^{w-1} s_{t+\tau_{j,m}} \right)$$

à la distribution uniforme. Ainsi, on déduit du corollaire 1.5 que le biais Δ de la distribution de m-uplet

$$\left(\sum_{j=0}^{w-1} s_{t+\tau_{j,1}}, \ldots, \sum_{j=0}^{w-1} s_{t+\tau_{j,m}} \right)$$

vérifie

$$\Delta^2 \geq m\Delta_w^2$$

où Δ_w^2 est le biais correspondant à une équation de parité. On obtient donc une borne supérieure sur la complexité de l'attaque utilisant m équations de parité.

Corollaire 3.22 *Avec les notations précédentes, la complexité en temps ($T_{w,m}$) et en données ($D_{w,m}$) de l'attaque par distingueur utilisant m équations de parité de poids w est*

$$T_{w,m} \leq \frac{w}{\Delta_w^2} \text{ et } D_{w,m} \leq \frac{1}{m\Delta_w^2} + m^{\frac{1}{w-1}} 2^{\frac{L}{w-1}}$$

quand $\sqrt{m}|\Delta_w| \ll 1$.

Le calcul de la valeur optimale de w pour un nombre d'équations donné montre que l'utilisation de plusieurs équations (si ce nombre reste raisonnable) ne permet pas d'améliorer significativement l'attaque. Toutefois, en pratique, il est rarement possible d'augmenter la valeur de w avec le nombre d'équations. La situation usuelle est que l'attaquant possède un polynôme multiple de poids w donné et de degré raisonnable, et qu'il peut trouver aisément des multiples de même poids, par exemple par élévations au carré successives. On voit alors que, si le facteur dominant dans la complexité en données est le premier terme de la formule,

alors l'augmentation du nombre d'équations de parité d'un facteur m permet de réduire d'autant la complexité en données. Il est important de noter que la complexité en temps reste ici inchangée.

Un autre point intéressant est que l'attaque précédente prend en compte uniquement la distribution du m-uplet

$$\left(\sum_{j=0}^{w-1} s_{t+\tau_{j,1}}, \ldots, \sum_{j=0}^{w-1} s_{t+\tau_{j,m}} \right) .$$

Or, on peut observer que ce distingueur n'est pas optimal car les m combinaisons linéaires étudiées ne sont pas indépendantes puisqu'elles ont toutes un terme en commun. C'est donc la distribution du $(m+1)$-uplet

$$\left(s_t, \sum_{j=1}^{w-1} s_{t+\tau_{j,1}}, \ldots, \sum_{j=1}^{w-1} s_{t+\tau_{j,m}} \right) .$$

que doit considérer le distingueur optimal. Toutefois, celle-ci est généralement difficile à évaluer sauf pour de petites valeurs de m, et les estimations générales conduisent à utiliser le distingueur précédent. Il est cependant possible d'obtenir des résultats plus précis, notamment pour $m = 2$ comme le montre la proposition suivante.

Proposition 3.23 *Soit $(s_t)_{t\geq 0}$ la suite produite par un LFSR de polynôme caractéristique P filtré par une fonction équilibrée f à n variables. Soit $1 + X^{\tau_1} + \ldots + X^{\tau_{w-1}}$ et $1 + X^{\mu_1} + \ldots + X^{\mu_{w-1}}$ deux polynômes multiples de $P(X)$. Alors, la distribution du $(2w-1)$-uplet $(s_t, s_{t+\tau_1}, \ldots, s_{t+\mu_{w-1}})$ est donnée par*

$$\begin{aligned}&\Pr[(s_t, s_{t+\tau_1}, \ldots, s_{t+\mu_{w-1}}) = (\sigma_0, \ldots, \sigma_{2w-2})] = \\ &2^{-2w+4} \Pr[(s_t, s_{t+\tau_1} + \ldots + s_{t+\tau_{w-1}}, s_{t+\mu_1} + \ldots + s_{t+\mu_{w-1}}) = (\sigma_0, \sigma_1 + \ldots + \sigma_{w-1}, \sigma_w + \ldots + \sigma_{2w-2})]\end{aligned}$$

où, en notant $a = \sigma_1 + \ldots + \sigma_{w-1}$ et $b = \sigma_w + \ldots + \sigma_{2w-2}$, cette dernière distribution correspond à

$$\begin{aligned} p(\sigma_0, a, b) \;=\; & \frac{1}{8} \Bigg[1 + 2^{-nw} (-1)^{\sigma_0} ((-1)^a + (-1)^b) \sum_{\lambda \in \mathbf{F}_2^n} \mathcal{F}^w(f + \varphi_\lambda) \\ & + 2^{-n(2w-1)+1} (-1)^{a+b} \sum_{x \in f^{-1}(\sigma_0)} \left(\sum_{\lambda \in \mathbf{F}_2^n} (-1)^{\lambda \cdot x} \mathcal{F}^{w-1}(f + \varphi_\lambda) \right)^2 \Bigg] . \end{aligned}$$

Ceci conduit à un biais $\Delta_{w,2}$ dont le carré est donné par

$$\Delta_{w,2}^2 = 2\Delta_w^2 + 2^{-2n(2w-1)+1} (\mathcal{I}_0^2 + \mathcal{I}_1^2) ,$$

où

$$\mathcal{I}(i) = \sum_{x \in f^{-1}(i)} \left(\sum_{\lambda \in \mathbf{F}_2^n} (-1)^{\lambda \cdot x} \mathcal{F}^{w-1}(f + \varphi_\lambda) \right)^2 .$$

En particulier, pour $w = 3$, on a

$$p(\sigma_0,a,b) = \frac{1}{8}\left[1 + 2^{-3n}(-1)^{\sigma_0}((-1)^a + (-1)^b)\sum_{\lambda\in\mathbf{F}_2^n}\mathcal{F}^3(f+\varphi_\lambda) + 2^{-3n+1}(-1)^{a+b}\sum_{x\in f^{-1}(\sigma_0)}\mathcal{F}^2(D_x f)\right] ,$$

et

$$\Delta_{3,2}^2 = 2\Delta_3^2 + 2^{-6n+1}(\nu_0^2(f) + \nu_1^2(f)) \text{ avec } \nu_i(f) = \sum_{x\in f^{-1}(i)}\mathcal{F}^2(D_x f) .$$

Preuve. La formule générale se déduit du lemme 3.4. En effet, avec les notations de ce lemme, on a

$$p(\sigma_0,a,b) = \frac{1}{2^{n(2w-3)}}\sum_{x\in f^{-1}(\sigma_0)}\left(\sum_{y_1+\ldots+y_{w-1}=a}|F_x^{-1}(y_1,\ldots,y_{w-1})|\right)\left(\sum_{z_1+\ldots+z_{w-1}=b}|F_x^{-1}(z_1,\ldots,z_{w-1})|\right) .$$

On en déduit

$$\begin{aligned} p(\sigma_0,a,b) &= \frac{1}{4}\sum_{x\in f^{-1}(\sigma_0)}\left(2^{-n} + ((-1)^a + (-1)^b)2^{-nw}\sum_{\lambda\in\mathbf{F}_2^n}(-1)^{\lambda\cdot x}\mathcal{F}^{w-1}(f+\varphi_\lambda)\right) \\ &\quad + (-1)^{a+b}2^{-n(2w-1)-2}\sum_{x\in f^{-1}(0)}\left(\sum_{\lambda\in\mathbf{F}_2^n}(-1)^{\lambda\cdot x}\mathcal{F}^{w-1}(f+\varphi_\lambda)\right)^2 \\ &= \frac{1}{8}\left(1 + ((-1)^{a+\sigma_0} + (-1)^{b+\sigma_0})2^{-nw}\sum_{\lambda\in\mathbf{F}_2^n}\mathcal{F}^w(f+\varphi_\lambda)\right) \\ &\quad + \frac{1}{4}(-1)^{a+b}2^{-n(2w-1)}\sum_{x\in f^{-1}(0)}\left(\sum_{\lambda\in\mathbf{F}_2^n}(-1)^{\lambda\cdot x}\mathcal{F}^{w-1}(f+\varphi_\lambda)\right)^2 . \end{aligned}$$

Pour $w = 3$, on peut donner une expression plus simple du dernier terme en utilisant la proposition 1.10, page 9 :

$$\sum_{\lambda\in\mathbf{F}_2^n}(-1)^{\lambda\cdot x}\mathcal{F}^2(f+\varphi_\lambda) = 2^n\mathcal{F}(D_x f) .$$

Le carré du biais est obtenu directement à partir de cette distribution par

$$\Delta_{w,2}^2 = 8\sum_{(\sigma_0,a,b)\in\mathbf{F}_2^3}\left(p(\sigma_0,a,b) - \frac{1}{8}\right)^2 .$$

◇

On remarque que la prise en compte de la valeur de s_t en plus du nombre d'équations de parité satisfaites conduit à un distingueur strictement plus performant, même si le gain dépend fortement des fonctions étudiées et est parfois négligeable.

Exemple. Considérons la fonction à 7 variables équilibrée de non-linéarité optimale dont les propriétés spectrales sont décrites dans [Can01b] :

$$f(x_1,\ldots,x_7) = x_2x_3x_4x_5 + x_1x_2x_3 + x_2x_4 + x_3x_5 + x_6x_7 + x_4 + x_5 .$$

En particulier, cette fonction possède un spectre de Walsh à 5 valeurs : $\{0, \pm 2^3, \pm 2^4\}$. La distribution du triplet étudié précédemment pour $w = 3$ est donné par

(s_0,a,b)	(0,0,0)	(0,1,0)	(0,0,1)	(0,1,1)	(1,0,0)	(1,1,0)	(1,0,1)	(1,1,1)
$p(s_0,a,b) - \frac{1}{8}$	$\frac{13824}{2^{21}}$	$-\frac{5632}{2^{21}}$	$-\frac{5632}{2^{21}}$	$-\frac{2560}{2^{21}}$	$-\frac{7168}{2^{21}}$	$-\frac{1024}{2^{21}}$	$-\frac{1024}{2^{21}}$	$\frac{9216}{2^{21}}$

Son biais vaut

$$\Delta^2_{3,2} = 2\Delta^2_3 + 2^{-6n+1}(\nu_0^2(f) + \nu_1^2(f)) \ .$$

On voit que dans ce cas, les deux termes intervenant dans le calcul du biais sont pratiquement du même ordre de grandeur. En effet, on a ici

$$\Delta^2_3 \simeq 2^{-12} \text{ et } 2^{-6n}(\nu_0^2(f) + \nu_1^2(f)) \simeq 2^{-13.03} \ ,$$

ce qui conduit à un biais total de

$$\Delta^2_{3,2} \simeq 2^{-10.43} \ .$$

La recherche de fonctions pour lesquelles la valeur de $\Delta^2_{3,2}$ est maximale reste un problème ouvert en raison de la difficulté d'estimer les quantités $\nu_0(f)$ et $\nu_1(f)$ intervenant ici. De manière plus générale, il semble relativement difficile d'estimer le gain apporté par le distingueur optimal que nous venons de décrire par rapport au distingueur fondé sur le nombre d'équations de parité satisfaites quand le nombre total d'équations utilisées est plus élevé.

3.3 Impact sur le choix de la fonction de filtrage

La formalisation des attaques par distingueur précédentes met en évidence un certain nombre de critères que doit vérifier la fonction de filtrage d'un générateur pseudo-aléatoire. Elle doit bien entendu être équilibrée et doit posséder une structure linéaire de type 1 si l'évolution de l'état interne est opérée par un registre à décalage (à rétroaction linéaire ou non). Dans le cas particulier d'un LFSR filtré, la fonction de filtrage doit être de non-linéarité élevée dans la mesure où cette dernière fournit une borne sur la valeur des moments d'ordre w du spectre de Walsh qui interviennent dans les attaques exploitant des équations de parité de poids w.

Les générateurs par combinaison de plusieurs suites sont également vulnérables à ce type d'attaques, qui imposent en particulier que la fonction de combinaison ait un ordre de résilience élevé mais aussi que les suites composantes soient de période élevée. Si ce type de générateur paraît de prime abord très séduisant pour les réalisations matérielles dans la mesure où il offre la possibilité d'engendrer les suites composantes en parallèle, l'analyse précise de la complexité de cette attaque montre qu'il est en fait très peu économique car la taille de l'état interne du générateur doit être beaucoup plus élevée que celle imposée par l'attaque par compromis temps-mémoire-données usuelle.

Chapitre 4

Les attaques par corrélation

Contrairement aux attaques décrites au chapitre précédent, les attaques par corrélation visent à retrouver une partie (ou la totalité) de l'état interne du générateur pseudo-aléatoire.

Cette famille d'attaques a été introduite en 1985 par Siegenthaler [Sie85] pour cryptanalyser les générateurs par combinaison de LFSRs. Elle a ensuite été améliorée, sous le nom d'attaques par corrélation rapides, en 1988 par Meier et Staffelbach [MS88, MS89] qui ont montré que ces attaques se ramenaient à un problème de correction d'erreurs. L'algorithme de décodage alors proposé par Meier et Staffelbach pour attaquer les générateurs par combinaison de LFSRs était un algorithme itératif utilisant des équations de parité de poids 3. Il permettait alors de mener une cryptanalyse sur les générateurs utilisant des LFSRs définis par un polynôme de rétroaction creux (notamment par un trinôme). En 1999, Johansson et Jönsson ont proposé de transformer le problème sous-jacent en un problème de décodage soit d'un code convolutif [JJ99b], soit d'un turbo-code [JJ99a]. Ces deux algorithmes fournissaient pour la première fois une attaque par corrélation rapide sans condition particulière sur la forme du polynôme de rétroaction du LFSR considéré. Nous avons alors montré [CT00] qu'un algorithme itératif du même type que celui utilisé par Meier et Staffelbach pouvait également conduire à une attaque dans le cas général, dans la mesure où il est possible de construire des équations de parité de poids faible quelle que soit la forme du polynôme visé grâce aux techniques que nous avons exposées au chapitre précédent. Par ailleurs, nous avons également introduit l'idée qu'il était parfois plus performant d'utiliser des équations de parité de poids légèrement supérieur — typiquement de poids 4, 5 ou 6 — ce qui offre souvent un meilleur compromis entre le taux d'erreur à corriger et le nombre d'équations de parité disponibles. D'autres algorithmes de décodage appropriés au contexte des attaques par corrélation rapides ont été proposés plus récemment, notamment une technique fondée sur le décodage à maximum de vraisemblance d'un code de dimension inférieure à celle du code original [CJS00, JJ00] dont nous analyserons ici la complexité.

Enfin, dans le cas particulier d'une attaque par corrélation sur les LFSRs filtrés, nous avons également montré [CF02] que les algorithmes de décodage précédents pouvaient exploiter simultanément toutes les approximations linéaires de la fonction de filtrage. Cette technique conduit alors à une attaque dont la complexité en données ne dépend que de la longueur du LFSR, mais dont la complexité en temps augmente avec le nombre de coefficients de Walsh non nuls de la fonction de filtrage.

4.1 Principe général

Les attaques par corrélation entrent dans la catégorie plus générale des attaques de type *diviser pour mieux régner* qui s'appliquent à chaque fois que l'on peut décomposer un système en composantes plus petites, cryptographiquement faibles. Dans le cas du chiffrement à flot, ces attaques ont été introduites contre les générateurs par combinaison de LFSRs [Sie85]. Mais, cette cryptanalyse est en fait valide sur tous les générateurs pseudo-aléatoires dont l'état interne est décomposable en plusieurs parties mises à jour indépendamment les unes des autres. On peut alors chercher séparément la valeur initiale de chaque partie de l'état interne.

L'attaque repose sur l'existence d'éventuelles corrélations entre la sortie de la fonction de filtrage et un sous-ensemble de ses entrées qui correspond à la partie incriminée de l'état interne.

4.1.1 Description

Supposons comme à la figure 4.1 que l'on peut séparer l'état interne du générateur à l'instant t en deux parties $\boldsymbol{x}_t$ et $\boldsymbol{y}_t$ de tailles respectives ℓ et $(n-\ell)$, mises à jour indépendamment par Φ_0 et Φ_1.

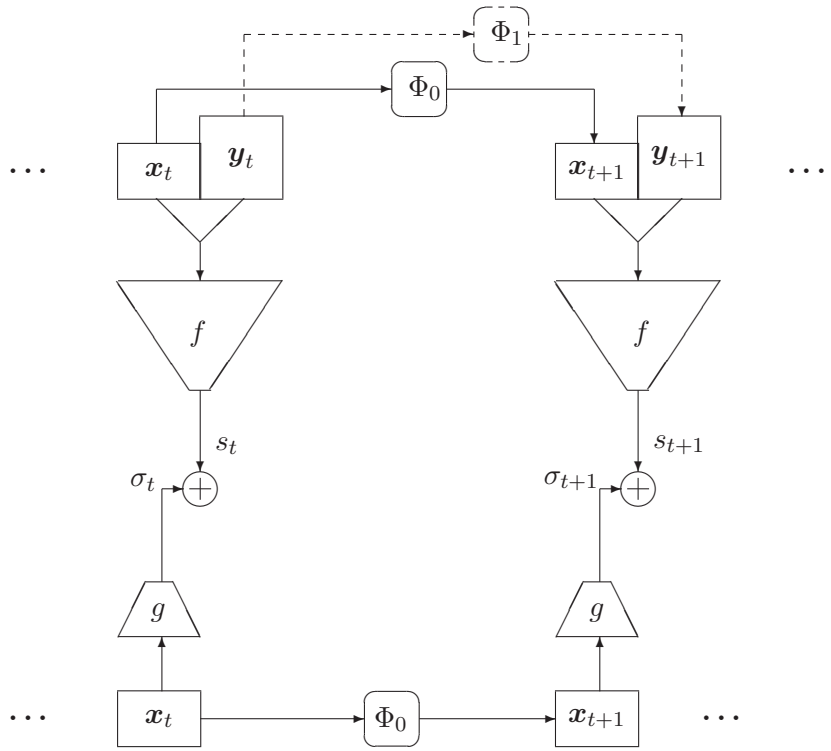

FIG. 4.1 – *Modèle de l'attaque par corrélation*

Plaçons-nous dans le cas où l'attaquant cherche à retrouver la valeur de la première partie de l'état initial, $\boldsymbol{x}_0$. Le vecteur d'entrée de la fonction de filtrage f se décompose de la même manière en deux parties, x et y. On peut alors appliquer l'attaque s'il existe une fonction g à ℓ variables (c'est-à-dire ne dépendant que de x) qui coïncide avec la sortie de f dans plus de

la moitié des cas, autrement dit si la probabilité

$$p_g = \Pr_{X,Y}[f(X,Y) = g(X)] > \frac{1}{2} \ .$$

La suite $\boldsymbol{\sigma}(\boldsymbol{x}_0)$ produite par le générateur réduit, d'état initial $\boldsymbol{x}_0$ et de fonction de filtrage g, est alors corrélée à la suite chiffrante $\mathbf{s}$ car, pour tout $t \geq 0$,

$$\Pr[s_t = \sigma_t] = p_g > \frac{1}{2} \ .$$

Dans ce cas, l'attaque par corrélation consiste à effectuer une recherche exhaustive sur la partie incriminée de l'état initial, $\boldsymbol{x}_0$, au moyen de l'algorithme décrit à la table 4.1.

Entrée. les D premiers bits de suite chiffrante, $(s_t)_{t<D}$.
Sortie. $\boldsymbol{x}_0 \in \mathbf{F}_2^\ell$, ℓ bits de l'état initial.
Algorithme

Pour chaque vecteur $\boldsymbol{x}_0$ de $\mathbf{F}_2^\ell$

- Calculer les D bits de la suite $\boldsymbol{\sigma}(\boldsymbol{x}_0)$ définie par

$$\sigma_t = g \circ \Phi_0^t(\boldsymbol{x}_0) \ .$$

- Calculer la corrélation sur D bits entre les suites $\boldsymbol{s}$ et $\boldsymbol{\sigma}(\boldsymbol{x}_0)$:

$$c(\boldsymbol{s},\boldsymbol{\sigma}(\boldsymbol{x}_0)) = \sum_{t=0}^{D-1} (-1)^{\sigma_t + s_t} \ .$$

Retourner la valeur de $\boldsymbol{x}_0$ qui maximise $c(\boldsymbol{s},\boldsymbol{\sigma}(\boldsymbol{x}_0))$.

TAB. 4.1 – *Attaque par corrélation*

Cet algorithme consiste en fait à distinguer la distribution de $(\sigma_t + s_t)$ de la distribution uniforme. Le biais associé à cette distribution étant défini par

$$\Delta = 2|p_g - \frac{1}{2}| \ ,$$

le nombre de bits de suite chiffrante nécessaires pour retrouver la valeur correcte de $\boldsymbol{x}_0$ est de l'ordre de

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

La complexité en temps est, elle, donnée par

$$T = \frac{2^\ell}{(2p_g - 1)^2} \ ,$$

mais peut être significativement réduite quand les fonctions Φ_0 et g sont des fonctions linéaires, en employant les attaques dites *par corrélation rapides*.

4.1.2 Approximation d'une fonction booléenne par une fonction à moins de variables

La fonction g étant choisie par l'attaquant, celui-ci doit naturellement utiliser celle qui conduit à la plus grande corrélation, c'est-à-dire celle pour laquelle la valeur de p_g est la plus éloignée possible de $1/2$. La forme générale de g est donnée par le résultat suivant, dont une formulation alternative est due à Zhang [Zha00, Th. 1].

Proposition 4.1 *[Can02] Soit f une fonction booléenne à n variables et I un sous-ensemble de $\{1,\ldots,n\}$ de cardinal ℓ. On note V (resp. $\overline{V}$) l'espace vectoriel engendré par les vecteurs de la base canonique e_i avec i dans I (resp. avec $i \notin I$). Alors, la valeur maximale*

$$\mathcal{F}(f+g) = \sum_{x\in V}\sum_{y\in\overline{V}} (-1)^{f(x+y)+g(x)}$$

atteignable par une fonction g qui ne dépend que des ℓ variables indexées par I est obtenue si et seulement si, pour tout $x \in V$,

$$\begin{cases} g(x) = 0 & si\ \mathcal{F}(f_{x+\overline{V}}) > 0 \\ g(x) = 1 & si\ \mathcal{F}(f_{x+\overline{V}}) < 0 \end{cases}$$

où $f_{x+\overline{V}}$ désigne la restriction de f à l'espace $x+\overline{V}$. De plus,

$$\max_g \mathcal{F}(f+g) = \sum_{x\in V} |\mathcal{F}(f_{x+\overline{V}})| \ .$$

Preuve. Pour toute fonction g définie sur V, on a

$$\begin{aligned} \mathcal{F}(f+g) &= \sum_{x\in V}\sum_{y\in\overline{V}} (-1)^{f(x+y)+g(x)} \\ &= \sum_{x\in V} (-1)^{g(x)} \left(\sum_{y\in\overline{V}} (-1)^{f(x+y)} \right) \\ &= \sum_{x\in V} (-1)^{g(x)} \mathcal{F}(f_{x+\overline{V}}) \ . \end{aligned}$$

On en déduit immédiatement que cette valeur est maximale si et seulement si tous les termes de la somme sont positifs, ce qui conduit à la définition de g et au maximum annoncés. ◇

De façon équivalente, la quantité précédente permet de mesurer la distance de f à l'ensemble des fonctions à n variables pour lesquelles tous les éléments de V sont des structures linéaires de type 0 :

$$d_H(f,LS^0(n,V)) = 2^{n-1} - \frac{1}{2}\sum_{x\in W} |\mathcal{F}(f_{x+V})|,$$

où V et W sont en somme directe. De même la distance de f à l'ensemble des fonctions à n variables pour lesquelles tous les éléments de V sont des structures linéaires est donnée par

$$d_H(f,LS(n,V)) = 2^{n-1} - \frac{1}{2}\max_{y\in W^\perp}\sum_{x\in W} |\mathcal{F}((f+\varphi_y)_{x+V})| \ ,$$

où W et V sont en somme directe [Can02] [a]. Par ailleurs, on déduit de la proposition précédente que la valeur maximale de $\mathcal{F}(f+g)$ est limitée par la non-linéarité de f.

Corollaire 4.2 *[Can02] Soit f une fonction booléenne à n variables et I un sous-ensemble de $\{1,\ldots,n\}$ de cardinal ℓ. On note V l'espace vectoriel engendré par les vecteurs de la base canonique e_i avec i dans I. Alors*

$$\max_g \mathcal{F}(f+g) \leq \left(\sum_{x \in V} \mathcal{F}^2(f+\varphi_x) \right)^{\frac{1}{2}} ,$$

où g est une fonction booléenne ne dépendant que des variables indexées par I.

En particulier,

$$\max_g \mathcal{F}(f+g) = 0$$

si et seulement si f est équilibrée et sans corrélation par rapport à I, et

$$\max_g \mathcal{F}(f+g) \leq 2^{\frac{\ell}{2}} \mathcal{L}(f) \ .$$

Dans le contexte de l'attaque par corrélation, on voit donc qu'il est indispensable que la fonction de filtrage soit ℓ-résiliente afin de se prémunir de la possibilité d'isoler ℓ entrées de la fonction de filtrage. Dans le cas où cette contrainte est trop forte — par exemple si plusieurs valeurs de ℓ sont possibles alors que l'ordre de résilience est limité par le nombre de variables et le degré de la fonction de filtrage — le choix d'une fonction ayant une non-linéarité élevée permet de réduire la portée de ce type d'attaques.

4.2 Modélisation par un problème de décodage

Considérons un générateur pseudo-aléatoire engendrant une suite chiffrante $\boldsymbol{s}$ à partir d'un état interne $\boldsymbol{x}_0$ de ℓ bits au moyen d'une fonction de transition Φ et d'une fonction de filtrage f. Comme précédemment, nous supposons alors qu'il existe un autre générateur de fonction de transition Ψ et de fonction de filtrage g qui, à partir du même état initial $\boldsymbol{x}_0$, produit une suite $\boldsymbol{\sigma}(\boldsymbol{x}_0)$ corrélée à la suite chiffrante, c'est-à-dire telle que, pour tout t,

$$p = \Pr[s_t \neq \sigma_t] = \frac{1}{2} - \varepsilon, \varepsilon > 0 \ .$$

Cette hypothèse revient à dire que la suite chiffrante $\boldsymbol{s}$ correspond au résultat de la transmission de la suite $\sigma(\boldsymbol{x}_0)$ à travers un canal de communication bruité qui suit le modèle d'un canal binaire symétrique de probabilité d'erreur p (voir Figure 4.2). Grâce à cette observation, Meier et Staffelbach [MS88, MS89] ont montré que toute attaque consistant à retrouver l'état initial du générateur, $\boldsymbol{x}_0$, (ou de manière équivalente la suite $\boldsymbol{\sigma}$) à partir de la connaissance de la suite chiffrante $\boldsymbol{s}$ peut être vue comme un problème classique de correction d'erreurs. En effet, les D premiers bits de $\boldsymbol{\sigma}(\boldsymbol{x}_0)$ correspondent à un mot du code correcteur d'erreurs $\mathcal{C}$ de longueur D formé par les 2^ℓ vecteurs $(\sigma_t)_{t<D}$ obtenus pour chacun des 2^ℓ états initiaux possibles $\boldsymbol{x}_0$. Ce code est donc défini par la donnée des fonctions Ψ et g.

Les attaques par corrélation rapides consistent alors à appliquer des algorithmes de décodage au code $\mathcal{C}$ afin de retrouver $\boldsymbol{x}_0$.

a. De ce résultat, on peut notamment déduire l'expression de la distance d'une fonction à l'ensemble des fonctions possédant exactement 1 ou 3 structures linéaires non nulles.

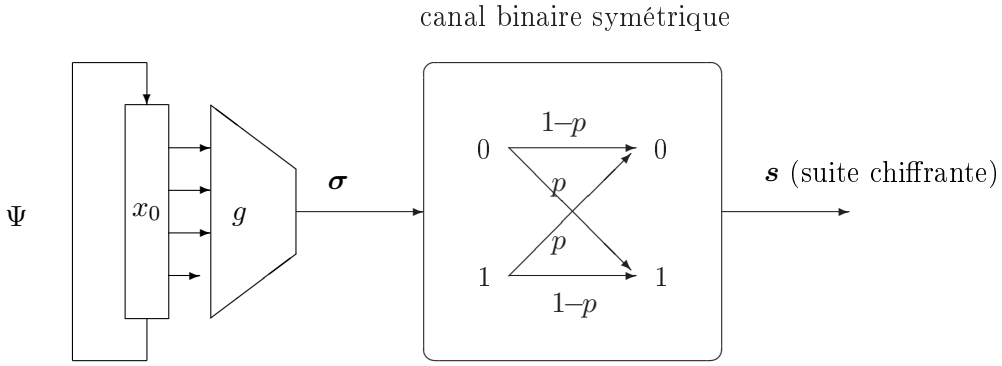

FIG. 4.2 – *Modèle de l'attaque par corrélation rapide*

4.3 Algorithmes de décodage pour les attaques par corrélation rapides

4.3.1 Décodage à maximum de vraisemblance

Lorsque le code $\mathcal{C}$ ne possède pas de propriété particulière qui facilite son décodage — ce qui est notamment le cas s'il s'agit d'un code non linéaire — le seul algorithme qui permette de corriger les erreurs de transmission est l'algorithme de décodage à maximum de vraisemblance. Il consiste à énumérer les 2^ℓ mots de code possibles, et à choisir celui qui est le plus proche du mot reçu. On voit que cet algorithme est strictement équivalent à la recherche exhaustive de l'état initial $\boldsymbol{x}_0$, autrement dit à l'algorithme d'attaque par corrélation décrit par Siegenthaler. Sa complexité en temps (T) et en données (D) est de l'ordre de

$$T = \frac{2^\ell}{\varepsilon^2} \text{ et } D = \frac{1}{\varepsilon^2}$$

où $(\frac{1}{2} - \varepsilon)$ désigne la probabilité d'erreur.

Sa complexité en temps est donc inaccessible, sauf dans les cas particuliers où la suite $\boldsymbol{\sigma}$ ne dépend que d'un petit nombre des bits de l'état initial. Cependant, elle peut être sensiblement réduite dans le cas où le code $\mathcal{C}$ est un code linéaire. En effet, le décodage à maximum de vraisemblance consiste à calculer la distance entre le mot de longueur D reçu (correspondant ici à la suite chiffrante) et les 2^ℓ mots du code $\mathcal{C}$. Mais pour un code linéaire, ce calcul est équivalent à celui de la transformée de Fourier de la fonction ternaire F de $\mathbf{F}_2^\ell$ dans $\{-1,0,1\}$ définie par

$$\begin{cases} F(g_t) = (-1)^{s_t} & \text{pour tout } 0 \le t < N \\ F(x) = 0 & \text{pour tout } x \notin \{g_t, 0 \le t < N\} \end{cases}$$

où g_t est la t-ième colonne de la matrice génératrice de $\mathcal{C}$. En effet, la corrélation entre la suite chiffrante et la suite $\boldsymbol{\sigma}(\boldsymbol{x}_0)$ engendrée à partir d'un état initial $\boldsymbol{x}_0$ donné correspond à la transformée de Fourier de F au point $\boldsymbol{x}_0$:

$$\begin{aligned} C(\mathbf{s},\boldsymbol{\sigma}(\boldsymbol{x}_0)) &= \sum_{t=0}^{D-1} (-1)^{s_t + \boldsymbol{x}_0 \cdot g_t} \\ &= \sum_{x \in \mathbf{F}_2^\ell} F(x)(-1)^{\boldsymbol{x}_0 \cdot x} = \widehat{F}(\boldsymbol{x}_0) \ . \end{aligned}$$

Ainsi, quand la longueur du code D (c'est-à-dire la complexité en données de l'attaque) est grande, la complexité en temps de l'algorithme peut être réduite à $2^\ell \ell$ au moyen d'une transformée de Fourier rapide [CJM02, Lu06]. Toutefois, le décodage à maximum de vraisemblance reste hors de portée pour les tailles de registres généralement utilisées en pratique, typiquement $\ell \geq 80$, à l'exception des attaques de type diviser pour mieux régner telles celles développées par Lu et Vaudenay contre E0 [LV04b, LV04a, LMV05].

Dans cette situation, l'attaque ne peut être menée que grâce à des algorithmes de décodage plus rapides que le décodage à maximum de vraisemblance. Toutefois, il est important de noter que, d'après le théorème fondamental de codage de canal de Shannon [Sha48], le décodage à maximum de vraisemblance est optimal dans le sens où c'est celui qui exige la longueur minimale D nécessaire pour pouvoir décoder un code contenant 2^ℓ mots. Aussi la longueur de suite chiffrante nécessaire pour retrouver les n bits de l'état initial $\boldsymbol{x}_0$ vérifie-t-elle toujours

$$D \geq \frac{\ell \ln 2}{2\varepsilon^2}$$

qui correspond au nombre de bits requis pour mener une attaque au moyen d'un décodage par maximum de vraisemblance. Les algorithmes de décodage beaucoup plus rapides que la recherche exhaustive utilisés dans les attaques par corrélation rapides vont donc avoir une complexité en données beaucoup plus élevée que la valeur minimale définie précédemment.

Les principaux algorithmes de décodage rapides employés dans ce contexte peuvent être répartis en deux grandes familles : les algorithmes itératifs, et des algorithmes qui consistent à appliquer successivement un décodage à maximum de vraisemblance à des codes de dimensions inférieures à celle du code initial.

4.3.2 Algorithmes itératifs utilisant des équations de parité

L'idée d'utiliser une procédure itérative fondée sur des équations de parité de poids faible revient à Meier et Staffelbach [MS88, MS89], même si l'algorithme proposé était alors moins performant que les techniques de décodage de ce type introduites par Gallager [Gal62]. Le principe général d'un tel décodage consiste à rechercher un grand nombre d'équations de parité de poids faible w pour $\mathcal{C}$ (c'est-à-dire un grand nombre de mots du code dual de poids w), et à assimiler par ce moyen le code $\mathcal{C}$ à un code à matrice de parité creuse (code LDPC, selon la terminologie anglo-saxonne). Les procédures de décodage peuvent alors se décomposer de la manière suivante :

- **Initialisation :**
$$Obs(\sigma_t) = \Pr[\sigma_t = 1 | \boldsymbol{s}] = \begin{cases} p & \text{si } s_t = 0 \\ 1-p & \text{si } s_t = 1 \end{cases}$$
- Pour t de 0 à $D-1$
  - Calculer l'information extrinsèque sur σ_t dans chacune de ses équations : si $\sigma_t = \sum_{j \in J_i} \sigma_j$ est la $i$ème équation faisant intervenir σ_t, alors .
$$Ext_i(\sigma_t) = \Pr[\sum_{j \in J_i} \sigma_j = 1 | \boldsymbol{s}]$$
  - Calculer les probabilités *a posteriori* partielles sur σ_t :
$$APP(\sigma_t,i) = Obs(\sigma_t) \prod_{j,j \neq i} Ext_j(\sigma_t)$$

– Itérer la mise à jour des informations extrinsèques $Ext_i(\sigma_t)$ à partir des APP partielles $APP(\sigma_t,i)$, puis la mise à jour des APPs.

Un point important des algorithmes itératifs classiques est que les probabilités *a posteriori* utilisées pour mettre à jour l'information extrinsèque dans l'équation i doivent exclure l'information apportée par cette équation. Il est donc nécessaire d'utiliser pour chaque position t du mot de code des APPs partielles correspondant à chacune des équations de parité dans lesquelles cette position intervient.

La complexité en temps et en mémoire de cet algorithme quand on utilise des équations de poids w est alors de l'ordre

$$T = \mathcal{O}(Dm_w^2) \text{ et } M = \mathcal{O}\left(Dm_w\right)$$

où D est la longueur du code (*i.e.* de la suite chiffrante) et m_w le nombre d'équations de parité de poids w disponibles. Ces complexités sont dans l'immense majorité des cas beaucoup trop élevées pour que cet algorithme puisse être employé pour décoder les codes intervenant dans les attaques par corrélation. Deux approximations sont donc classiquement utilisées pour réduire la complexité de cette algorithme :

- la valeur de la probabilité *a posteriori* globale

$$APP(\sigma_t) = Obs(\sigma_t) \prod_j Ext_j(\sigma_t)$$

est utilisée à la place de chacune des APPs partielles. La complexité de l'algorithme est alors réduite à

$$T = \mathcal{O}\left(Dm_w\right) \text{ et } M = \mathcal{O}\left(D\right) \ .$$

- les APPs et les informations extrinsèques sont calculées au moyen d'approximations qui évitent en particulier les problèmes liés à la manipulation de nombres flottants très petits [HOP96]. Cette modification influence uniquement la constante dans la complexité en temps, mais elle permet une amélioration significative en pratique.

Ces deux types de modifications, s'ils améliorent notablement la complexité de l'algorithme, dégradent légèrement ses performances. La complexité et les performances (au sens de la capacité de correction) de toutes ces variantes de l'algorithme classique ont été comparées avec précision par S. Leveiller [Lev04].

La variante dégradée de l'algorithme de Gallager que nous avons choisie dans le cas des attaques par corrélation rapides [CT00] manipule les logarithmes des rapports de vraisemblance pour chaque position. Elle est décrite à la table 4.2. La complexité de cet algorithme dépend naturellement du nombre d'équations de parité m_w nécessaires à sa convergence. S'il semble difficile de donner une estimation théorique de son nombre vu les modifications effectuées par rapport à l'algorithme de décodage classique, les nombreuses simulations effectuées dans [CT00] nous ont conduits à l'estimation suivante du nombre minimal d'équations de poids w requis pour que l'algorithme converge :

$$m_w \geq \frac{2\ln 2}{(2\varepsilon)^{2w-4}} \ .$$

En combinant cette condition avec l'approximation (3.1) usuelle du nombre de multiples de poids w de degré inférieur ou égal à D d'un polynôme de degré ℓ, rappelée page 26, on en

- *Initialisation :*
  pour t de 0 à $D-1$, $L[t] = \log(\frac{1-p}{p})$.
- *Jusqu'à ce que l'algorithme converge, itérer :*
  Pour t de 0 à $D-1$
  $L'[t] = (-1)^{s_t} L[t]$
  pour chaque équation de parité faisant intervenir le bit t, $\sigma_t = \sum_{j \in J} \sigma_j$,

$$L'[t] \leftarrow L'[t] + (-1)^{\sum_{j \in J} s_j} \min_{j \in J}(L[j])$$

  Pour t de 0 à $D-1$
  $s_t \leftarrow 0$ si $L'[t] < 0$ et $s_t \leftarrow 1$ sinon.
  $L[t] \leftarrow |L'[t]|$

TAB. 4.2 – *Algorithme de décodage itératif pour les attaques par corrélation rapides*

déduit que la complexité en données (D) et en temps (T) de l'attaque est donnée par

$$D = \left(\frac{1}{2\varepsilon}\right)^{\frac{2(w-2)}{w-1}} 2^{\frac{\ell}{w-1}} \text{ et } T = \left(\frac{1}{2\varepsilon}\right)^{\frac{2w(w-2)}{w-1}} 2^{\frac{\ell}{w-1}} . \tag{4.1}$$

On voit alors qu'il est préférable d'augmenter le poids des équations de parité, par rapport au choix $w = 3$ fait initialement par Meier et Staffelbach. Toutefois, le poids des équations reste limité par la complexité en mémoire de l'algorithme qui nécessite le stockage des m_w équations.

4.3.3 Décodage à maximum de vraisemblance d'un code de dimension inférieure

Le principe de ces algorithmes est de transformer, au moyen de manipulations sur sa matrice génératrice, le code $\mathcal{C}$ à décoder en un autre code linéaire $\mathcal{C}'$ de dimension k plus petite, auquel on peut appliquer un décodage à maximum de vraisemblance. L'idée fondatrice, due à Chepyshov, Johansson et Smeets [CJS00], est de s'abstraire des $\ell - k$ dernières positions de l'état initial en considérant toutes les combinaisons linéaires de w colonnes de la matrice génératrice de $\mathcal{C}$ qui s'annulent sur ces $\ell - k$ dernières positions. Ce nouveau code de dimension k est de longueur

$$\frac{D^w}{w! 2^{\ell-k}} ,$$

mais chaque bit du mot reçu correspond maintenant à la somme de w bits de la suite chiffrante, ce qui implique que la probabilité d'erreur atteint

$$\frac{1}{2} + 2^{w-1}\varepsilon^w .$$

Le nombre de bits de suite chiffrante nécessaires pour décoder ce nouveau code est donc de l'ordre de

$$\left(\frac{1}{2\varepsilon}\right)^2 2^{\frac{\ell-k}{w}} .$$

La complexité de l'algorithme, hors phase de précalcul, est alors donnée par

$$D \simeq \left(\frac{1}{2\varepsilon}\right)^2 2^{\frac{\ell-k}{w}} \text{ et } T = k2^k. \tag{4.2}$$

On voit ici que l'immense avantage de cet algorithme est sa complexité en temps, qui ne dépend que de la valeur de k, ce qui permet une grande flexibilité puisqu'il suffit de choisir pour k la valeur permettant d'atteindre le coût de décodage souhaité. Pour retrouver les $n-k$ bits restants de l'état initial, il suffit alors d'appliquer la même attaque aux k bits suivants et ainsi de suite. La phase de précalcul permettant de générer la nouvelle matrice avec les combinaisons linéaires de w colonnes a une complexité équivalente à la recherche d'équations de parité de poids $w+1$.

Certaines améliorations de cet algorithme sont présentées dans [JJ00]. Une première idée est de considérer toutes les combinaisons linéaires des colonnes dont la valeur sur les $(\ell-k)$ dernières positions appartient à un ensemble donné, qui n'est pas réduit au singleton $\{0\}$ comme c'était le cas précédemment. Une deuxième amélioration repose sur l'algorithme de Goldreich, Rubinfeld et Sudan [GRS95] permettant la reconstruction d'un polynôme dans le cas multivarié. Au lieu de répéter l'algorithme précédent pour chaque bloc de k bits de l'état initial, on peut utiliser une procédure séquentielle qui permet de retrouver les valeurs des $(\ell-k)$ bits restants les unes après les autres.

Une autre procédure de décodage, proposée dans [MFI00], peut sembler sur le papier plus efficace que les précédentes, mais un certain nombre de facteurs ont été négligés dans le calcul de sa complexité (en particulier la complexité en mémoire liée au stockage des équations de parité), et les simulations réalisées par P. Quanty [Qua02] montrent qu'elle est en pratique, pour les paramètres cryptographiques usuels, moins performante que les deux algorithmes que nous venons de décrire.

4.4 Attaques sur les générateurs à base de LFSRs

Dans le cas particulier des différents générateurs à base de LFSRs, la forme du code linéaire sous-jacent permet généralement d'améliorer les performances des algorithmes de décodage génériques.

4.4.1 Combinaison de LFSRs

Dans le cas d'un générateur par combinaison de n LFSRs assemblés par une fonction f à n variables, la meilleure suite cible $\boldsymbol{\sigma}$, c'est-à-dire celle qui correspond à la probabilité d'erreur la plus faible, est celle obtenue en additionnant les sorties de $(t+1)$ LFSRs si t est l'ordre d'immunité aux corrélations de la fonction f. Ce résultat, démontré dans [CT00], peut également être vu comme corollaire immédiat de la proposition 4.1 et du corollaire 4.2 précédents.

Proposition 4.3 *[CT00] Soit f une fonction booléenne à n variables t-résiliente. Soit I un sous-ensemble de $\{1,\ldots,n\}$ de cardinal $(t+1)$ et V (resp. $\overline{V}$) l'espace vectoriel engendré par les vecteurs de la base canonique e_i avec i dans I (resp. avec $i \notin I$). Alors, la valeur maximale de*

$$\mathcal{F}(f+g) = \sum_{x \in V} \sum_{y \in \overline{V}} (-1)^{f(x+y)+g(x)}$$

atteignable par une fonction g qui ne dépend que des $(t+1)$ variables indexées par I est obtenue pour la fonction affine

$$g(x_{i_1},\ldots,x_{i_{t+1}}) = x_{i_1} + \ldots + x_{i_{t+1}} + c$$

où la constante $c \in \mathbf{F}_2$ est donnée par

$$(-1)^c = \text{signe}(\mathcal{F}(f+\varphi_{1_I}))\ .$$

De plus

$$\max_g \mathcal{F}(f+g) = |\mathcal{F}(f+\varphi_{1_I})|\ .$$

Par conséquent, la suite cible $\boldsymbol{\sigma}$ optimale pour l'attaque par corrélation est obtenue en additionnant les sorties de $t+1$ LFSRs du système. Elle correspond donc à la sortie d'un unique LFSR dont le polynôme de rétroaction P est le ppcm des polynômes de rétroaction des $(t+1)$ LFSRs impliqués. Quand tous ces polynômes sont primitifs, la longueur du LFSR cible est donc la somme des longueurs des $(t+1)$ LFSRs. La suite chiffrante $\boldsymbol{s}$ correspond alors au résultat de la transmission de $\boldsymbol{\sigma}$ à travers un canal binaire symétrique de probabilité d'erreur p avec

$$p = Pr[s_t \neq \sigma_t] = \frac{1}{2} - \frac{1}{2^{n+1}} \max_{I,|I|=t+1} |\mathcal{F}(f+\varphi_{1_I})|.$$

Le fait de choisir pour f une fonction de non-linéarité élevée garantit donc un taux d'erreur important, et par conséquent que les attaques par corrélation rapides ont une complexité élevée. La complexité de l'attaque par corrélation rapide est alors donnée, pour les deux algorithmes de décodage décrits précédemment, par la formule (4.1) ou (4.2) avec

$$\ell = \sum_{i=1}^{t+1} L_i \text{ et } \varepsilon = \frac{|\mathcal{F}(f+\varphi_{1_I})|}{2^{n+1}}$$

où I est l'ensemble des indices de $(t+1)$ variables d'entrées qui conduit à la complexité minimale et les L_i désignent les longueurs des différents LFSRs du système.

4.4.2 LFSRs filtrés

Dans le cas du registre filtré de longueur L, on choisit pour $\boldsymbol{\sigma}$ une suite produite par un LFSR qui a le même polynôme de rétroaction que le LFSR constituant le générateur pseudo-aléatoire mais un état initial différent. Si la suite chiffrante est définie par la relation :

$$\forall t,\ \ s_t = f(u_{t+\gamma_1}, u_{t+\gamma_2}, \cdots, u_{t+\gamma_n})$$

alors, la suite $\boldsymbol{\sigma}$ optimale est donnée par :

$$\sigma_t = \sum_{i=1}^{n} \alpha_i u_{t+\gamma_i}$$

où $\alpha = (\alpha_1,\ldots,\alpha_n)$ est le vecteur qui maximise $|\mathcal{F}(f+\varphi_\alpha)|$. La probabilité d'erreur du canal binaire symétrique est alors de

$$p = Pr[s_t \neq \sigma_t] = \frac{\mathcal{NL}(f)}{2^n}$$

où $\mathcal{NL}(f)$ désigne la non-linéarité de f. La complexité de l'attaque par corrélation rapide est alors donnée, pour les deux algorithmes de décodage décrits précédemment, par la formule (4.1) ou (4.2) avec

$$\ell = L \text{ et } \varepsilon = \frac{\mathcal{L}(f)}{2^{n+1}} \ .$$

Toutefois, le modèle du canal binaire symétrique décrit précédemment et appliqué à l'origine aux générateurs par combinaison de LFSRs n'est plus adéquat. En effet, le canal de transmission considéré n'est pas un canal sans mémoire dans la mesure où les entrées de la fonction de filtrage aux différents instants ne sont pas indépendantes, comme en attestent les biais mis en évidence au chapitre précédent. En pratique, dans le cas d'un LFSR filtré, on constate que les performances de l'attaque par corrélation rapide sont légèrement moins bonnes que celles que l'on attend dans le cas d'un canal sans mémoire [Lev04]. Une technique pour mieux prendre en compte la forme particulière du canal de transmission et utiliser les dépendances entre les entrées de la fonction aux différents instants consiste à exploiter les corrélations de la fonction augmentée, et non de la fonction de filtrage. Cette approche a notamment conduit à des attaques dites par corrélation conditionnées, comme celles décrites dans [LCPP96, Löh03, LZGB03]. Nous avons exploré dans [CF02] une autre voie permettant d'améliorer les performances de l'attaque : il s'agit ici de considérer, non plus une seule approximation linéaire de la fonction de filtrage, mais toutes celles qui correspondent à un coefficient de Walsh non nul. Cette idée a également été développée partiellement par Jönsson et Johansson [JJ02] dans une attaque sur LILI-128 qui prend en compte simultanément tous les coefficients de Walsh dont la valeur absolue est maximale. Le code qui intervient dans l'attaque que nous avons présentée dans [CF02] est donc celui qui associe à chaque état initial $(u_0,\ldots,u_{L-1})$ possible du LFSR visé, la suite formée par la succession de B blocs de N bits, définie par

$$\boldsymbol{\sigma} = (\boldsymbol{\sigma}_{\alpha_1},\ldots,\boldsymbol{\sigma}_{\alpha_B})$$

où $\{\alpha_1,\ldots,\alpha_B\}$ est l'ensemble des vecteurs α tel que $\mathcal{F}(f+\varphi_\alpha) \neq 0$. Le bit t de $\boldsymbol{\sigma}_\alpha$ est donné par

$$\sigma_{\alpha,t} = \sum_{i=1}^{n} \alpha_i u_{t+\gamma_i} \ .$$

La matrice génératrice G de ce code de longueur BN et de dimension L correspond alors à la concaténation de B matrices génératrices similaires à celle employée dans l'attaque classique :

$$G = (G_{\alpha_1},\ldots,G_{\alpha_B})$$

où la colonne d'indice t de la matrice G_α, représentée sous forme polynomiale correspond à

$$\sum_{i=1}^{n} g_{i,t}^{(\alpha)} X^i = \left(\sum_{i=1}^{n} \alpha_i X^{t+\gamma_i}\right) \bmod P(X) \ ,$$

P désignant le polynôme caractéristique du LFSR.

Le mot à décoder $\boldsymbol{y}$ est de la même manière formé par la répétition B fois de N bits de suite chiffrante. Toutefois, contrairement au cas classique, la probabilité d'erreur affectant chacun de ces blocs n'est pas constante puisque, pour le bloc α,

$$\Pr[s_t \neq \sigma_{\alpha,t}] = \frac{1}{2} - \frac{1}{2^{n+1}} |\mathcal{F}(f+\varphi_\alpha)| \ .$$

Autrement dit, si l'on exprime le problème en termes de correction d'erreurs, le canal de transmission utilisé est maintenant un canal binaire non-stationnaire.

On peut toutefois le décoder avec la technique décrite au paragraphe 4.3.3 qui consiste à appliquer un décodage à maximum de vraisemblance au code $\mathcal{C}'$ de dimension k dérivé de $\mathcal{C}$ dont la matrice génératrice est formée des combinaisons linéaires de w colonnes de la matrice originale qui s'annulent sur les $(L-k)$ dernières positions. La probabilité d'erreur liée à ce nouveau problème de décodage est donc celle qui affecte la somme de w bits du mot reçu. Toutefois, si dans le cas classique décrit dans [CJS00], on avait une expression immédiate de cette probabilité d'erreur :

$$\frac{1}{2} - 2^{w-1}\varepsilon^w$$

où $\frac{1}{2} - \varepsilon$ est la probabilité d'erreur initiale, le calcul devient plus complexe dans le cas d'un canal non stationnaire puisque la probabilité dépend des indices des blocs utilisés dans la combinaison linéaire considérée.

Pour $w = 2$, on doit ainsi découper les positions du nouveau code en fonction du spectre de Walsh étendu de f. Dans la suite, nous noterons A_c le nombre de vecteurs α tels que $|\mathcal{F}(f+\varphi_\alpha)| = c$. Ainsi, pour chaque couple $(c_1,c_2) \in \{|\mathcal{F}(f+\varphi_\alpha)|\} \setminus \{0\}$, $c_1 < c_2$, on obtient

$$M_{c_1c_2} = \frac{N^2 A_{c_1} A_{c_2}}{2^{L-k}}$$

combinaisons de 2 colonnes de la matrice G provenant de blocs affectés par des probabilités d'erreur respectives correspondant à c_1 et c_2. De même, pour $c_1 = c_2 = c$, on a

$$M_{c^2} = \frac{N^2 A_c^2}{2^{L-k+1}}$$

combinaisons linéaires qui ont la propriété attendue. La probabilité d'erreur associée à ces positions du nouveau code $\mathcal{C}'$ est alors

$$\frac{1}{2} - \frac{c_1 c_2}{2^{2n+1}} \ .$$

Ainsi, la proportion des positions du nouveau code $\mathcal{C}'$ affectées par la probabilité d'erreur précédente est de

$$\mu_{c_1c_2} = \frac{2A_{c_1}A_{c_2}}{B} \text{ pour } c_1 < c_2, \ \mu_{c^2} = \frac{A_c^2}{B} \ .$$

Ce découpage permet donc de calculer la capacité du canal non-stationnaire associé à ce nouveau problème de décodage

$$C = \sum_{c_1 \le c_2} C\left(\frac{1}{2} - \frac{c_1 c_2}{2^{2n+1}}\right)$$

où $C(p)$ est la capacité du canal binaire symétrique (stationnaire) de probabilité d'erreur p. En utilisant l'approximation classique de la capacité quand p est proche de $1/2$ — ce qui revient ici à supposer que tous les coefficients de Walsh de f sont petits devant 2^n, c'est-à-dire que f est de non-linéarité élevée — on obtient, pour une fonction f équilibrée :

$$C \simeq \frac{1}{2B^2 \ln 2} \ .$$

Un calcul détaillé montre que cette quantité est en fait une borne inférieure de la capacité, atteinte quand f est de non-linéarité élevée, et qu'à l'inverse une borne supérieure est donnée par

$$C < \frac{1}{B^2}$$

sous l'hypothèse que le degré de f est supérieur ou égal à 2.

On peut mener de la même manière le calcul de la capacité du canal dans le cas général où le code $\mathcal{C}'$ est obtenu à partir des combinaisons linéaires de w colonnes de la matrice G. On obtient alors le résultat similaire suivant :

Théorème 4.4 *[CF02] Pour toute fonction f équilibrée de degré strictement supérieur à 1, la capacité du canal binaire symétrique non-stationnaire impliqué dans l'algorithme qui utilise des combinaisons linéaires de w colonnes de la matrice génératrice satisfait*

$$\frac{1}{2B^w \ln 2} \leq C < \frac{1}{B^w} \ .$$

De plus, la borne inférieure fournit une bonne approximation de C quand f est de non-linéarité élevée.

Corollaire 4.5 *La complexité de l'attaque par corrélation rapide utilisant l'algorithme de décodage de Chepyshov, Johansson et Smeets de paramètres w et k est donnée par*

$$D = \mathcal{O}(2^{\frac{L-k}{w}}) \ et \ T = \mathcal{O}(k2^k B^w)$$

où B est le nombre de coefficients de Walsh non nuls de f.

On voit donc que le nombre de bits nécessaires pour mener l'attaque ne dépend que de la longueur du LFSR et non de la fonction de filtrage. Par contre, la complexité en temps de l'algorithme augmente, elle, avec le nombre de coefficients de Walsh non nuls de f. Ces résultats sont confirmés par les simulations présentées dans [CF02], menées sur plusieurs fonctions de filtrage, dans le cas d'un LFSR de longueur $L = 40$.

4.5 Perspectives

Parmi les voies de recherche envisagées sur les attaques par corrélation, deux pistes semblent prometteuses. La première idée est la recherche d'une amélioration générale des algorithmes de décodage que nous venons de décrire, qui consisterait à enchaîner deux algorithmes de décodage souple de natures différentes. Le principe est d'utiliser l'information souple fournie par l'algorithme de décodage itératif, même si celui-ci ne converge pas, en entrée d'un autre algorithme de décodage, par exemple l'algorithme proposé par A. Valembois [Val00]. L'intérêt réside dans le fait que le premier algorithme permet notamment de sélectionner les positions les plus fiables du mot reçu, et donc de n'appliquer le second algorithme qu'à un code raccourci.

Une deuxième voie de recherche serait d'essayer de mieux adapter les algorithmes de décodage décrits précédemment, dont l'objectif est très général, au générateur pseudo-aléatoire attaqué. On peut penser par exemple que la connaissance des spécifications du générateur, et de la valeur de l'IV utilisée pourrait apporter de l'information susceptible d'être exploitée par l'algorithme de décodage.

Chapitre 5

Les attaques algébriques

Aux critères de conception mis en évidence par les attaques précédentes et essentiellement liés aux propriétés spectrales de la fonction de filtrage s'ajoute une autre propriété qui permet de garantir la résistance du système contre les attaques algébriques. Cette famille d'attaques, proposée en 2003 par Courtois et Meier [CM03a], est particulièrement dévastatrice pour tous les générateurs à base de LFSRs. Nous rappelons ici son principe général et mettons en évidence les propriétés de la fonction de filtrage qui influencent sa complexité [a].

5.1 Principe général

L'idée sous-jacente aux attaques algébriques remonte à l'article fondateur de Shannon [Sha49, Page 711] : il s'agit d'exprimer l'opération de chiffrement sous forme d'un système d'équations algébriques multivariées — liant les bits du chiffré, les bits du clair et ceux de la clef — d'instancier ce système à l'aide d'un ou plusieurs couples clairs / chiffrés, puis de le résoudre afin de retrouver les bits de clef. Dans le cas d'un générateur pseudo-aléatoire dont la fonction de transition Φ est linéaire, les équations de chiffrement sont celles qui donnent l'expression du bit de la suite chiffrante à l'instant t en fonction de l'état initial $\boldsymbol{x}_0$

$$s_t = f \circ \Phi^t(\boldsymbol{x}_0) \ .$$

Leur degré correspond donc au degré de la fonction de filtrage f ; pour cette raison, un critère bien connu, utilisé dans la conception de tous les générateurs de ce type, est que le degré de la fonction de filtrage doit être élevé.

Toutefois, Courtois et Meier ont montré en 2003 [CM03a] qu'il était parfois possible de mener une attaque de ce type même quand f est de degré élevé. Leur idée originale est que l'on peut appliquer ce principe général dès lors qu'il existe des relations de degré faible entre les entrées et la sortie de la fonction de filtrage. De telles relations correspondent à des multiples de petit degré de f, c'est-à-dire à des relations du type

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

où la fonction h est de petit degré.

Mais, dans le cas binaire qui nous intéresse ici, toute relation de ce type est équivalente à l'existence d'un annulateur de petit degré pour la fonction f ou pour la fonction $(1+f)$, au

a. Ce chapitre correspond essentiellement au contenu de l'article [Can06].

sens de la définition suivante [MPC04, FA03]. En effet, en multipliant la relation précédente par $f(x)$, on obtient

$$g(x)[f(x)]^2 = h(x)f(x) = g(x)f(x) = h(x)$$

qui conduit à

$$h(x)[1 + f(x)] = 0 \ .$$

Définition 5.1 (Idéal annulateur d'une fonction booléenne) *Soit f une fonction booléenne à n variables.* L'idéal annulateur de f, *noté $AN(f)$, est l'ensemble des fonctions booléennes g à n variables vérifiant*

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

Pour un degré d fixé, on note $AN_d(f)$ l'ensemble des annulateurs de f de degré inférieur ou égal à d :

$$AN_d(f) = \{g \in AN(f), \ \deg(g) \leq d\} \ .$$

Les annulateurs de f et $(1 + f)$ fournissent alors des relations algébriques liant les bits de l'état initial du générateur :

- si $s_t = 1$, alors pour toute fonction $g \in AN(f)$, on a

$$g \circ \Phi^t(\boldsymbol{x}_0) = 0 \ ;$$

- si $s_t = 0$, alors pour toute fonction $h \in AN(1 + f)$, on a

$$h \circ \Phi^t(\boldsymbol{x}_0) = 0 \ .$$

Ainsi, l'ensemble des relations obtenues à partir de N bits de suite chiffrante et de toutes les fonctions de $AN_d(f) \cup AN_d(1 + f)$ forme un système d'équations de degré inférieur ou égal à d à n variables $x_1, \ldots, x_n$ correspondant aux n bits de l'état initial :

$$\begin{cases} g \circ \Phi^t(x_1, \ldots, x_n) \ \forall g \in AN_d(f), & \forall 0 \leq t < N \text{ tel que } s_t = 1 \\ h \circ \Phi^t(x_1, \ldots, x_n) \ \forall h \in AN_d(1 + f), & \forall 0 \leq t < N \text{ tel que } s_t = 0. \end{cases} \tag{5.1}$$

Les n bits de l'état initial peuvent donc être retrouvés par simple résolution de ce système algébrique.

5.2 Complexité des attaques algébriques

La résolution d'un système de polynômes multivariés comme (5.1) est un problème classique de calcul formel. Une méthode extrêmement simple pour cela, appelée la linéarisation, consiste à identifier ce système à un système linéaire en $\sum_{i=1}^{d} \binom{n}{i}$ variables, chacune d'elles correspondant en fait au produit de i variables du système initial. L'inversion de ce système — par une élimination de Gauss ou une technique plus élaborée — permet donc de retrouver l'état initial en un nombre d'opérations de l'ordre de

$$\left(\sum_{i=1}^{d} \binom{n}{i} \right)^{\omega} \simeq n^{\omega d}$$

où ω est l'exposant donnant la complexité de l'inversion matricielle (on prendra typiquement $\omega \simeq 2.37$ [CW90]). Nous utiliserons par la suite cette formule du fait de sa simplicité, mais

nous garderons à l'esprit le fait qu'elle ne fournit qu'une borne supérieure grossière de la complexité de la résolution du système (5.1).

Toutefois, cette estimation utilisée dans la plupart des travaux sur les attaques algébriques, est fondée sur deux hypothèses. Tout d'abord, elle suppose que tous les monômes de degré d apparaissent dans le système, ce qui correspond au pire cas du point de vue de l'attaquant. La seconde hypothèse, sans doute beaucoup plus forte, est que le système (5.1) peut toujours être résolu. En effet, on suppose a priori que les équations résultant de

$$N \simeq \frac{2n^d}{d!(\dim AN_d(f) + \dim AN_d(1+f))}$$

bits de suite chiffrante sont linéairement indépendantes. Il arrive clairement que ce ne soit pas le cas, ce qui augmente la complexité en données de l'attaque. Mais on peut également se trouver parfois dans des situations extrêmes dans lesquelles le système (5.1) est sous-déterminé quel que soit le nombre de bits de suite chiffrante considérés. Ceci se produit par exemple lorsque les équations dérivées d'un annulateur g aux différents instants t ne sont pas toutes différentes, ou de façon équivalente, quand il existe un entier T strictement inférieur à la période de la suite $(\Phi^t(\boldsymbol{x}_0))_{t\geq 0}$ telle $g \circ \Phi^T(x) = g(x)$ pour tout $x \in \mathbf{F}_2^n$. Il est clair qu'un tel entier T divise la période de la suite $(\Phi^t(\boldsymbol{x}_0))_{t\geq 0}$, ce qui nous conduit au résultat suivant quand Φ correspond à la fonction de transition d'un LFSR.

Proposition 5.2 *Soit Φ la fonction de transition d'un LFSR de longueur n et de polynôme de rétroaction primitif. Soit g une fonction booléenne à n variables. Si $2^n - 1$ est un nombre premier, alors toutes les fonctions $g \circ \Phi^t$, pour $0 \leq t < 2^n$, sont distinctes.*

A l'inverse, lorsque $2^n - 1$ est un nombre composite, il est toujours possible de construire des fonctions de filtrage f dont certains annulateurs g sont tels que la suite $(g \circ \Phi^t)_{t\geq 0}$ a une période strictement inférieure à $(2^n - 1)$, comme le montre l'exemple suivant.

Exemple. Considérons le LFSR de longueur 4 défini par le polynôme de rétroaction primitif $P(x) = x^4 + x + 1$ et la fonction de filtrage à 4 variables f suivante

$$f(x_1,\ldots,x_4) = x_3 + x_4 + x_1x_2 + x_2x_3 + x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4 \ .$$

Alors, la fonction $g(x_1,\ldots,x_4) = 1 + x_2 + x_3 + x_4 + x_2x_4 + x_3x_4$ est un annulateur de f et vérifie

$$g \circ \Phi^t(x_1,\ldots,x_4) = g \circ \Phi^{t \bmod 5}(x_1,\ldots,x_4), \ \forall t \geq 0 \ .$$

En effet, si l'on identifie $\mathbf{F}_2^4$ avec le corps fini $\mathbf{F}_{2^4}$ défini par P, on constate que $g(x) = g(x\alpha^5)$ où α est une racine de P.

Toutefois, cette situation ne se produit que quand l'annulateur g possède une propriété de périodicité de ce type, ce qui induit également une régularité pour la fonction f. Il est donc clair que l'utilisation d'une telle fonction de filtrage conduirait immédiatement à une attaque par distingueur, comme l'atteste la proposition suivante.

Proposition 5.3 *[Can06] Soit f une fonction booléenne à n variables et g un élément non nul de $AN(f) \cup AN(1+f)$. Si $g \circ \Phi^T = g$ pour un entier T, alors il existe au moins un état initial du générateur qui engendre une suite chiffrante dont tous les bits s_{t_0+iT}, $i \geq 0$ sont égaux pour un certain $t_0 < T$. De plus, si Φ est la fonction de transition d'un LFSR dont le polynôme de rétroaction est primitif et si $\deg(g) \neq n$, alors il existe $t_0 < T$ tel que la suite chiffrante produite à partir de tout état initial non nul vérifie la propriété précédente.*

Cependant, nous avons uniquement envisagé ici la possibilité que certaines équations obtenues à partir d'un même annulateur soient identiques, et la question de l'existence d'éventuelles dépendances linéaires entre elles reste ouverte. Il est malgré tout fortement probable que la situation soit dans ce cas similaire à celle que nous venons de décrire : si le rang du système (5.1) diffère fortement du rang attendu pour un système aléatoire, alors le générateur étudié est sans doute vulnérable à une attaque par distingueur simple.

Aussi apparaît-il clairement, si nous supposons que le système (5.1) se comporte comme un système aléatoire, que le paramètre cryptographique pertinent dans le contexte d'une attaque algébrique est le degré de ce système, qui est égal à l'immunité algébrique de la fonction de filtrage [MPC04] [b].

Définition 5.4 (Immunité algébrique d'une fonction booléenne) L'immunité algébrique *d'une fonction booléenne f, notée $AI(f)$, est le degré minimal atteint par une fonction non nulle de $AN(f) \cup AN(1+f)$.*

Ainsi, la complexité en temps de l'attaque algébrique utilisant l'algorithme de linéarisation est donnée par

$$T = \mathcal{O}\left(n^{\omega AI(f)}\right) \text{ avec } \omega \simeq 2.37$$

et la complexité en données est de l'ordre de $n^{AI(f)}$. Il est probable que cette dernière diminue quand le nombre de fonctions de degré minimal de $AN(f) \cup AN(1+f)$ augmente. Cette estimation de la complexité fournit donc la valeur de l'immunité algébrique minimale que doit posséder la fonction de filtrage afin que le système résiste aux attaques algébriques :

$$AI(f) \geq 0.42 \frac{k}{\log_2 n}$$

où k est la taille de la clef et n celle de l'état interne. En particulier, si la taille de l'état interne correspond au double de celle de la clef, il faut que

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

Ainsi, pour une clef de 128 bits et un état interne de 256 bits, l'immunité algébrique de la fonction de filtrage doit être au moins égale à 7.

Cependant, la borne précédente sous-estime clairement la valeur minimale admissible pour l'immunité algébrique de la fonction de filtrage. En effet, elle repose sur la complexité de l'algorithme de linéarisation, alors qu'il existe des algorithmes beaucoup plus efficaces pour résoudre un système algébrique multivarié, notamment les techniques de calcul de bases de Gröbner. Les méthodes les plus récentes et les plus performantes sont les algorithmes F4 et F5 de Faugère [Fau99, Fau02]. L'algorithme F4 est par exemple plus efficace [AFI+04, Die04] qu'une version raffinée de la linéarisation, appelée algorithme XL, proposée par Courtois, Klimov, Patarin et Shamir [CKPS00], et l'algorithme F5 est strictement plus rapide que tous les autres. Une autre technique de résolution, appelée XSL, a également été proposée par Courtois et Pieprzyk [CP02] mais sa complexité est sujette à discussion [CL05]. Des résultats récents sur la complexité des algorithmes F4 et F5 ont été démontrés [BFSY05, BFSY04],

b. Notons que ce paramètre n'est pas pertinent dans l'évaluation de la complexité des attaques algébriques sur d'autres types de systèmes, comme les chiffrements par blocs, ou les générateurs par combinaison avec mémoire. Dans ce dernier cas, les idéaux annulateurs de f et $(1+f)$ jouent en effet des rôles différents [Arm05].

mais il est important de garder à l'esprit que ces résultats ne s'appliquent que dans le cas dit semi-régulier. Il est donc essentiel de déterminer si le système (5.1) étudié dans une attaque algébrique se comporte comme un système aléatoire vis-à-vis de ces algorithmes de résolution. En l'absence de tout résultat permettant de valider cette hypothèse, il ne semble pas pertinent d'appliquer les résultats de complexité du cas semi-régulier. La situation pourrait en effet être similaire à celle rencontrée pour la cryptanalyse du système HFE [Pat96] menée par Faugère à l'aide de F5 alors qu'elle était hors de portée d'après sa complexité dans le cas générique [FJ03].

5.3 Immunité algébrique des fonctions booléennes

Même si l'analyse de la complexité des attaques algébriques reste un problème ouvert, il paraît incontestable que celle-ci dépend essentiellement de l'immunité algébrique de la fonction de filtrage.

5.3.1 Propriétés générales

L'ensemble $AN(f)$ des annulateurs de f est clairement un idéal de l'anneau des fonctions booléennes, et il est engendré par la fonction $(1+f)$. Cet idéal est composé des $2^{2^n-wt(f)}$ fonctions à n variables qui s'annulent sur le support de f. Le nombre de fonctions de $AN(f)$ de degré au plus d est donc égal à 2^κ où κ est la dimension du noyau de la matrice obtenue par restriction au support de f de la matrice génératrice du code de Reed-Muller de longueur 2^n et d'ordre d. Autrement dit, chacune des lignes de cette matrice correspond à l'évaluation d'un monôme de degré au plus d en tous les points x tels que $f(x)=1$. Comme cette matrice possède $\sum_{i=0}^{d}\binom{n}{i}$ lignes et $wt(f)$ colonnes, son noyau est évidemment non réduit à $\{0\}$ dès que

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

De la même manière, $AN(1+f)$ contient une fonction de degré inférieur ou égal à d dès que

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

L'immunité algébrique de f est donc liée à son poids, comme l'ont constaté Dalai *et al.* [DGM04]. On retrouve notamment ainsi que l'immunité algébrique d'une fonction à n variables est inférieure ou égale à $\lceil n/2 \rceil$, résultat déjà démontré indépendemment dans [FA03] et dans [CM03b]. Par ailleurs, quand le nombre de variables n est impair, seules les fonctions équilibrées peuvent atteindre l'immunité algébrique maximale.

Une autre propriété simple [Can06] est que l'immunité algébrique est limitée par la dimension de l'espace $\mathcal{S}_0(f)$ des structures linéaires de type 0 de la fonction :

$$AI(f) \leq \left\lceil \frac{n - \dim \mathcal{S}_0(f)}{2} \right\rceil\ .$$

On en déduit en particulier que, si le nombre de variables n de la fonction de filtrage est inférieur à la taille de l'état interne L du générateur — c'est-à-dire si on assimile la fonction

de filtrage à une fonction à L variables pour laquelle $\dim \mathcal{S}_0(f) \geq (L-n)$ —, alors n doit satisfaire

$$n \geq 0.84 \frac{k}{\log_2 L}$$

où k est la taille de la clef. Ainsi, un LFSR filtré de longueur 256 pour une clef secrète de 128 bits doit utiliser une fonction de filtrage ayant au moins 16 variables[c]. Ici encore, la valeur minimale admissible pour le nombre de variables de la fonction de filtrage est sous-estimée puisqu'elle repose sur l'hypothèse que l'attaque utilise l'algorithme de linéarisation pour résoudre le système.

5.3.2 Immunité algébrique des fonctions à 5 variables

Il est possible de calculer l'immunité algébrique de toutes les fonctions équilibrées à 5 variables en utilisant la classification de Berlekamp-Welch [BW72]. Les résultats sont résumés à la table 5.1. Une autre quantité intéressante, donnée à la table 5.2, est le nombre d'annulateurs

$AI(f)$	1	2	3
nombre de f équilibrées	62	403 315 208	197 765 120
proportion de f équilibrées	10^{-7}	0.671	0.329

TAB. 5.1 – *Immunité algébrique des fonctions équilibrées à 5 variables*

linéairement indépendants de degré inférieur ou égal à 2 pour chacune des fonctions équilibrées à 5 variables. En menant ces calculs, nous avons notamment constaté que les ensembles

$\dim(AN_2(f))$	0	1	2	3	4	5
nb. de f équilibrées	197 765 120	345 283 456	56 801 920	1 213 960	15 872	62
proportion	0.329	0.574	0.094	0.002	$2 \cdot 10^{-5}$	10^{-7}

TAB. 5.2 – *Dimension de $AN_2(f)$ pour les fonctions équilibrées à 5 variables*

$AN_2(f)$ et $AN_2(1+f)$ avaient la même dimension pour toutes les fonctions équilibrées à 5 variables, sauf pour une fonction et son complément (à équivalence linéaire près). Cette remarque nous a conduit à nous demander s'il n'était pas possible d'établir une relation générale entre les ensembles $AN(f)$ et $AN(1+f)$ pour les fonctions f équilibrées.

5.3.3 Fonctions d'immunité algébrique maximale

Nous avons pu établir une relation entre les annulateurs de f et de $1+f$ dans le cas particulier des fonctions équilibrées d'immunité algébrique maximale dépendant d'un nombre impair de variables. En effet, tous les annulateurs d'une fonction f sont de degré supérieur ou égal à $\lfloor \frac{n+1}{2} \rfloor$ si et seulement si le support de f correspond à un sous-ensemble de 2^{n-1} colonnes de rang maximal de la matrice génératrice du code de Reed-Muller de longueur 2^n et d'ordre $\lfloor \frac{n-1}{2} \rfloor$. Si n est impair, un tel sous-ensemble est donc un ensemble d'information du code de Reed-Muller d'ordre $\frac{n-1}{2}$, qui est de dimension 2^{n-1}. Aussi peut-on mettre en évidence une

c. Cette condition n'était par exemple pas respectée dans le registre filtré utilisé pour produire les données dans le système LILI-128 puisqu'il s'agissait d'un LFSR de longueur 89 filtré par une fonction à 10 variables.

relation entre les degrés de $AN(f)$ et de $AN(1+f)$ en utilisant le fait que le code sous-jacent est auto-dual.

Proposition 5.5 *[Can06] Soit $\mathcal{C}$ un code auto-dual. Si I est un ensemble d'information de $\mathcal{C}$, alors c'est également un ensemble d'information de $\mathcal{C}^{\perp}$.*

Ainsi, dans le cas où n est impair, pour que tous les éléments non-nuls de l'ensemble $AN(f) \cup AN(1+1)$ soient de degré maximal, il suffit que cette propriété soit satisfaite pour les éléments de $AN(f)$.

Théorème 5.6 *[Can06] Soit n un entier impair et f une fonction booléenne équilibrée à n variables. Alors, f est d'immunité algébrique maximale $\frac{n+1}{2}$ si et seulement si $AN(f)$ ne contient aucune fonction non nulle de degré strictement inférieur à $\frac{n+1}{2}$.*

Quelques rares familles de fonctions d'immunité algébrique maximale ont été mises en évidence récemment. Dalai *et al.* ont notamment présenté une construction récursive qui conduit à une famille infinie de fonctions équilibrées ayant une immunité algébrique optimale [DGM05]. Un autre exemple de fonction optimale de ce point de vue est la fonction majorité à n variables pour n impair, c'est-à-dire la fonction qui vaut 1 si et seulement si le poids de Hamming de son vecteur d'entrée est supérieur ou égal à $\frac{n+1}{2}$. Ce résultat est exposé en termes d'ensemble d'information du code de Reed-Muller auto-dual dans [KMM05] ; il a été mentionné plus tard dans le contexte des attaques algébriques indépendamment dans [DSM05, BP05] [d]. Aussi, à l'heure actuelle, les rares familles générales de fonctions connues dont l'immunité algébrique est maximale présentent-elles d'autres inconvénients qui rendent leur utilisation directe peu souhaitable dans un chiffrement à flot.

5.3.4 Immunité algébrique des composantes des fonctions puissances

Dans un article récent [NGG06], Nawaz, Gong et Gupta donnent une borne supérieure sur l'immunité algébrique des composantes des fonctions puissances, qui montre que celle-ci reste relativement limitée quand n est élevé pour la plupart des familles d'exposants utilisés habituellement.

Théorème 5.7 *[NGG06] Soit $S : x \mapsto x^s$ sur $\mathbf{F}_{2^n}$. Alors, l'immunité algébrique de S vérifie*

$$AI(S) \leq r(s)\lfloor\sqrt{n}\rfloor + \left\lceil \frac{n}{\lfloor\sqrt{n}\rfloor} \right\rceil - 1 \leq r(s)\lfloor\sqrt{n}\rfloor + \lceil\sqrt{n}\rceil$$

où $r(s)$ est le nombre de plages de 1 consécutifs dans la décomposition binaire de l'exposant s.

Nawaz, Gong et Gupta améliorent cette borne dans le cas où n est impair et s est un des exposants pour lesquels toutes les composantes de $x \mapsto x^s$ ont la meilleure non-linéarité possible [e] ; ils montrent en particulier que toutes ces fonctions booléennes ont une immunité algébrique inférieure ou égale à

$$2\lfloor\sqrt{n}\rfloor + 1 \ .$$

5.3.5 Immunité algébrique et autres critères cryptographiques

L'immunité algébrique d'une fonction booléenne est également liée à d'autres propriétés usuelles des fonctions booléennes, notamment à la non-linéarité [DGM04]. Il est en effet clair

d. Dans [BP05], Braeken et Preneel prétendent donner d'autres constructions de familles infinies de fonctions symétriques mais il s'agit en fait de constructions qui ne sont valides que pour un petit nombre de variables et pour lesquelles on peut aisément trouver des contre-exemples dans le cas général.

e. Il s'agit des exposants des fonctions presque courbes (AB), listés au tableau 8.1 page 101.

que, pour toute fonction linéaire φ, l'immunité algébrique de $f+\varphi$ est au plus égale à $AI(f)+1$. La relation entre le poids d'une fonction et son immunité algébrique nous donne alors

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

Il s'ensuit que toute fonction d'immunité algébrique optimale est de non-linéarité relativement élevée, plus précisément

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

Par conséquent, une haute non-linéarité et une immunité algébrique élevée sont des critères compatibles. Les générateurs à base de LFSRs qui résistent aux attaques algébriques ne sont donc généralement pas vulnérables aux attaques par corrélation rapides[f].

On peut également établir un lien entre l'immunité algébrique et la *normalité*, propriété cette fois-ci antagoniste à une immunité algébrique élevée.

Définition 5.8 *[Dob94] Une fonction est dite* k-normale *(resp.* k-faiblement normale*) si elle est constante (resp. affine) sur un sous-espace affine de dimension* k*.*

En utilisant le fait que les mots de poids minimum de $RM(r,n)$ sont ceux dont le support est un espace affine de dimension $(n-r)$, on voit clairement que toute fonction k-normale à n variables est d'immunité algébrique au plus $(n-k)$. De même, une fonction k-faiblement normale est d'immunité algébrique inférieure ou égale à $(n-k+1)$. Aussi les fonctions non-normales comme celles trouvées pour la première fois dans [CDLD06] peuvent-elles être de bonnes candidates à la construction de fonctions d'immunité algébrique élevée.

L'existence de liens entre l'immunité algébrique et d'autres propriétés cryptographiques reste par contre ouverte. Notons toutefois que Claude Carlet a récemment mis en évidence une relation entre l'immunité algébrique d'une fonction et sa distance aux fonctions de bas degré [Car06] (*i.e.* sa distance à $RM(d,n)$).

5.4 Attaques algébriques rapides

Une variante plus générale et souvent plus efficace des attaques algébriques a été introduite par Courtois en 2003 [Cou03] sous le nom d'attaque algébrique rapide. Une première amélioration de l'attaque précédente, permettant de diminuer le degré du système d'équations à résoudre, consiste à rechercher des équations de petit degré entre les bits de l'état initial, qui font intervenir plusieurs bits consécutifs de suite chiffrante. Autrement dit, on cherche des relations de bas degré entre les entrées et les sorties de la fonction augmentée

$$\begin{array}{rccl} F_m\colon & \mathbf{F}_2^n & \to & \mathbf{F}_2^m \\ & \boldsymbol{x} & \mapsto & (f(\boldsymbol{x}), f(\Phi(\boldsymbol{x})), \ldots, f(\Phi^{m-1}(\boldsymbol{x}))). \end{array}$$

Toutefois, la recherche de ces relations est souvent un problème ardu, dont la complexité augmente considérablement quand le nombre m de bits de suite chiffrante intervenant simultanément dans les équations augmente. L'algorithme usuel employé pour trouver de telles relations entre les entrées et les sorties d'une fonction vectorielle F de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$ consiste à

f. L'inverse n'est évidemment pas vrai, comme en atteste l'existence de fonctions courbes quadratiques.

rechercher les annulateurs de petit degré de la fonction caractéristique Φ_F de F, qui est la fonction booléenne à $(n+m)$ variables définie par

$$\Phi_F(x_1,\ldots,x_n,y_1,\ldots,y_m) = 1 \text{ si et seulement si } y_i = F_i(x_1,\ldots,x_n), \forall i \ .$$

On voit donc clairement qu'il ne peut être utilisé que pour des petites valeurs de m. Par exemple, si l'on considère une fonction de filtrage booléenne à 20 variables, son immunité algébrique peut atteindre 10. Mais il existe toujours des relations de degré au plus 7 liant les bits de l'état initial et 4 bits consécutifs de suite chiffrante. La recherche d'éventuelles relations de degré 6 entre ces bits correspond cependant au calcul du noyau d'une matrice de 120 Gigaoctets. Même la recherche de relations de degré 3 dans ce contexte met en jeu une matrice de 2.7 Gigaoctets. Cependant, on peut légitimement se demander si la recherche des relations de degré minimal entre les entrées et les sorties de la fonction augmentée F_m ne peut pas être simplifiée du fait qu'il s'agit d'une fonction vectorielle extrêmement particulière. L'existence d'une relation entre le degré minimal des relations pour F_m et l'immunité algébrique de f est notamment un problème ouvert. Plus généralement, une question importante est de déterminer si le fait que toutes les composantes de la fonction vectorielle considérée soient affinement équivalentes, ou qu'elles aient le même spectre de Walsh étendu, influe sur ce degré — nous verrons au chapitre 10 que cette question est également au cœur de la recherche de fonctions de substitution pour les chiffrements par blocs qui offrent une bonne résistance aux attaques algébriques dans la mesure où les permutations puissances, souvent utilisées dans ce contexte, vérifient aussi cette propriété.

Pour contourner le problème général de la recherche de relations de bas degré pour la fonction augmentée, généralement trop coûteux, Courtois a proposé de n'employer que les relations obtenues par combinaison linéaire de relations connues de la forme

$$g(x_0,\ldots,x_{n-1},s_t,\ldots,s_{t+m})$$

où les termes de plus haut degré ne font intervenir aucun bit de suite chiffrante. En effet, ces termes peuvent alors être annulés par combinaison linéaire au cours d'une phase de précalcul, ce qui permet de faire baisser le degré [Cou03, HR04, Arm04]. Un exemple d'attaque algébrique rapide de ce type est celle qui permet de cryptanalyser le générateur par LFSR filtré SFINKS, candidat à l'appel eSTREAM [Cou06].

Des variantes plus sophistiquées des attaques algébriques s'appliquent également lorsque la fonction de filtrage fait intervenir quelques bits de mémoire mis à jour de manière non linéaire, comme dans le système E0 [AK03, Arm02], ou lorsqu'il s'agit non pas d'une fonction booléenne, mais d'une fonction vectorielle qui produit plusieurs bits de suite à chaque instant [Cou05].

Chapitre 6

Conception de fonctions de filtrage

Les différentes attaques connues sur les générateurs pseudo-aléatoires utilisant une fonction de transition linéaire imposent donc des contraintes très fortes sur la fonction de filtrage utilisée : celle-ci doit être équilibrée et de degré relativement élevé, posséder une haute non-linéarité et les différents moments de son spectre de Walsh doivent être petits. De plus, la résistance aux attaques algébriques nécessite l'emploi d'une fonction ayant un nombre de variables élevé, ce qui paraît incompatible avec la possibilité d'une réalisation matérielle peu coûteuse.

Mon expérience dans la conception de ce type d'algorithmes m'a montré qu'il est illusoire d'atteindre des performances raisonnables si les contraintes de mise en œuvre ne sont pas au cœur du choix de la fonction de filtrage. Quand le nombre de variables devient élevé (au minimum une quinzaine), nous sommes à peu près sûrs qu'une fonction simplement choisie pour ses qualités cryptographiques sera trop onéreuse pour les applications visées. Aussi avons-nous délibérément choisi une autre approche : l'étude de certaines familles de fonctions appropriées à la réalisation matérielle, et la recherche au sein de ces classes particulières de fonctions qui soient de bonne qualité cryptographique. Les classes qui nous ont particulièrement intéressée sont d'une part celle constituée des composantes des fonctions puissances, c'est-à-dire les fonctions du type $x \mapsto \mathrm{Tr}(\lambda x^s)$, et d'autre part celle des fonctions symétriques qui sont les seules fonctions pour lesquelles on connaisse une réalisation matérielle en un nombre de portes logiques linéaire en le nombre de variables. Toutefois, les fonctions de ces deux familles ayant rarement toutes les qualités souhaitées, en particulier la propriété d'équilibre, nous détaillons une nouvelle technique peu coûteuse permettant d'obtenir une fonction équilibrée de non-linéarité élevée à partir d'une fonction courbe. Afin de pouvoir mesurer les qualités cryptographiques de ces fonctions et de les comparer à la situation optimale, nous donnons auparavant quelques résultats exhaustifs sur les fonctions de haute non-linéarité dépendant d'un petit nombre de variables, $5 \leq n \leq 9$.

6.1 Fonctions équilibrées de haute non-linéarité à 9 variables et moins

Contrairement au cas pair, la non-linéarité maximale atteignable par une fonction à n variables quand n est impair n'est pas connue dans le cas général et sa détermination est un problème difficile qui a donné lieu à de nombreux travaux (voir par exemple [Fon98]). On sait

cependant que

$$2^{\frac{n}{2}} < \min_{f \in \mathcal{B}ool_n} \mathcal{L}(f) \leq 2^{\frac{n+1}{2}}$$

où la borne supérieure est appelée *borne quadratique*[a] puisqu'elle est atteinte par des fonctions de degré 2. De plus, il a été démontré que, pour $n \leq 7$, la non-linéarité maximale est égale à la borne quadratique [BW72, Myk80, Hou96b]. A l'inverse, cette borne est surpassée à partir de $n \geq 9$ comme en atteste l'exemple récemment mis en évidence par Kavut, Maitra et Yücel [KMY06]. De plus, pour $n \geq 15$, Patterson et Wiedemann [PW83, PW90] ont montré que :

$$\min_{f \in \mathcal{B}ool_n} \mathcal{L}(f) \leq \frac{27}{32} 2^{\frac{n+1}{2}} \ .$$

Toutefois, même pour les petites valeurs de n pour lesquelles la non-linéarité optimale est connue, il n'est pas toujours aisé de trouver des fonctions qui l'atteignent et qui puissent être utilisées en cryptographie. En effet, les fonctions quadratiques, si elles constituent un exemple immédiat de fonctions de non-linéarité optimale, sont évidemment à proscrire car leur emploi conduit immédiatement à de nombreuses attaques, notamment les attaques algébriques, ou les attaques par l'algorithme de Berlekamp-Massey [Mas69] exploitant la faible complexité linéaire de la suite engendrée. De plus, comme nous l'avons vu aux chapitres 3 et 4, la complexité de certaines attaques fait intervenir des éléments plus précis du spectre de Walsh de la fonction de filtrage que la seule non-linéarité. Il est donc important lors de la conception d'un chiffrement à flot de calculer le spectre de Walsh de la fonction de filtrage dans sa totalité — le spectre étendu, au sens de la définition 1.7 suffit souvent — et de déterminer s'il possède les meilleurs caractéristiques cryptographiques parmi tous les spectres possibles pour une fonction ayant ces paramètres.

Pour $n = 5$, le spectre de Walsh de toutes les fonctions — à équivalence affine près — a été déterminé par Berlekamp et Welch [BW72]. On constate alors qu'il existe seulement deux spectres de Walsh étendus optimaux, un spectre plateau et un spectre à 5 valeurs. Ils sont décrits dans le tableau 6.1 sous une forme « symétrisée », ce qui signifie que l'on donne uniquement le nombre de u tels que $\mathcal{F}(f + \varphi_u) \in \{\pm x\}$.

$\mathcal{F}(f+\varphi_u)$	± 8	± 4	0	degré de f
nombre de u	16		16	2 ou 3
	12	16	4	4

TAB. 6.1 – *Spectre de Walsh étendu des fonctions à 5 variables de non-linéarité maximale*

En généralisant des techniques introduites par Brouwer [Bro93], Simonis [Sim94] et Hou [Hou96c, Hou96a], nous avons déterminé la forme de tous les spectres de Walsh optimaux pour les fonctions à 7 variables [Can01b]. Cette étude repose essentiellement sur les propriétés de divisibilité des coefficients de Walsh, et sur la forme des éléments $u \in \mathbf{F}_2^n$ pour lesquels $\mathcal{F}(f + \varphi_u)$ satisfait une congruence particulière modulo une puissance de 2. Ces résultats peuvent donc être vus comme des généralisations du théorème de McEliece sur la divisibilité des poids d'un code cyclique [McE72] et du théorème suivant dû à Katz, que nous exprimons ici en termes de transformée de Walsh et non de poids des translatés du code de Reed-Muller.

a. Quand il n'y aura pas d'ambiguïté, on utilisera cette expression pour désigner à la fois la valeur de $\mathcal{L}(f)$ et la valeur de $\mathcal{NL}(f)$ associée. Cette non-linéarité sera souvent qualifiée de « presque optimale ».

Proposition 6.1 *[Kat71] Soit f et g deux fonctions booléennes à n variables de degrés respectifs d_1 et d_2 avec $d_2 \leq d_1 \leq n$. Alors*

$$\mathcal{F}(f+g) \equiv \mathcal{F}(f) \bmod 2^{\lceil \frac{n-d_2}{d_1} \rceil + 1} .$$

On en déduit donc que les coefficients de Walsh d'une fonction à n variables de degré d vérifient

$$\mathcal{F}(f+\varphi_u) \equiv \mathcal{F}(f) \bmod 2^{\lceil \frac{n-1}{d} \rceil + 1}, \ \forall u \in \mathbf{F}_2^n .$$

Notre approche repose alors sur l'étude de la structure du sous-ensemble des u pour lesquels

$$\mathcal{F}(f+\varphi_u) \not\equiv \mathcal{F}(f) \bmod 2^{\lceil \frac{n-1}{d} \rceil + 2} .$$

Notre objectif est en effet de faciliter l'étude du spectre de Walsh en répartissant les éléments u en deux sous-ensembles distincts en fonction de la divisibilité des coefficients de Walsh associés.

Proposition 6.2 *[Can01b] Soit n et d deux entiers tels que $2 < d < n$ et d ne divise pas $(n-2)$. Soit f une fonction à n variables de degré d et $\ell = \lceil \frac{n-1}{d} \rceil + 1$. Supposons qu'il existe un élément $\alpha \in \mathbf{F}_2^n$ tel que $\mathcal{F}(f+\varphi_\alpha) \not\equiv \mathcal{F}(f) \bmod 2^{\ell+1}$. Alors*

$$E_f = \{\varphi \in \mathcal{B}ool_n \text{ avec } \deg(\varphi) \leq 1, \ \mathcal{F}(f+\varphi) \equiv \mathcal{F}(f) \bmod 2^{\ell+1}\}$$

est un hyperplan de l'espace des fonctions affines. De plus, E_f contient la fonction constante égale à 1 si et seulement si $\mathcal{F}(f) \equiv 0 \bmod 2^\ell$.

En particulier, pour une fonction f équilibrée, l'ensemble des éléments $u \in \mathbf{F}_2^n$ tels que $\mathcal{F}(f+\varphi_u) \not\equiv 0 \bmod 2^{\ell+1}$ est soit vide, soit un hyperplan de $\mathbf{F}_2^n$. Ceci permet notamment de montrer que l'existence d'une fonction f de non-linéarité supérieure ou égale à la borne quadratique avec $E_f \neq \emptyset$ implique l'existence d'une fonction ayant une variable de moins et un spectre de Walsh très particulier [Can01b]. Grâce à ce résultat, nous avons pu démontrer le théorème suivant qui détermine le spectre de Walsh étendu des fonctions de non-linéarité optimale à 7 variables de degré inférieur ou égal à 4, ainsi que des fonctions cubiques à 9 variables.

Théorème 6.3 *[Can01b] Soit f une fonction booléenne à n variables de degré d telle que $\mathcal{L}(f) \geq 2^{\frac{n+1}{2}}$. Si $n \in \{5,7\}$ et $d = 4$ ou si $n = 9$ et $d = 3$, alors le spectre de Walsh étendu de f est l'un des deux spectres suivants :*

$\mathcal{F}(f+\varphi_u)$	$\pm 2^{\frac{n+1}{2}}$	$\pm 2^{\frac{n-1}{2}}$	0
nombre de u	2^{n-1}		2^{n-1}
	$3 \cdot 2^{n-3}$	2^{n-1}	2^{n-3}

De plus, pour chacun de ces jeux de paramètres, il existe des fonctions possédant ces deux spectres de Walsh étendus.

Ce théorème contredisait alors l'idée jusque-là communément admise selon laquelle toutes les fonctions cubiques de non-linéarité optimale étaient des fonctions plateaux — ce qui est le cas jusqu'à $n = 7$. En effet, nous avons mis en évidence la fonction cubique à 9 variables suivante dont le spectre de Walsh a 5 valeurs :

$$\begin{aligned} f \ = \ & x_1x_2x_3 + x_4x_5x_6 + x_1x_7x_8 + x_4x_7x_8 + x_9x_1x_4 + x_1x_5 + x_1x_6 + x_1x_7 + x_1x_8 \\ & + x_2x_4 + x_2x_5 + x_2x_7 + x_3x_4 + x_5x_6 + x_5x_7 + x_9x_1 . \end{aligned}$$

On peut naturellement construire une fonction cubique à n variables ayant un spectre à 5 valeurs similaire pour tout n impair, $n \geq 9$, par somme directe de la fonction précédente avec une fonction courbe quadratique.

La proposition 6.2 ne s'appliquant pas aux fonctions à n variables de degré $(n-2)$, nous avons alors dû traiter ce cas séparément.

Proposition 6.4 *[Can01b] Soit f une fonction à n variables de degré $(n-2)$ avec $n \geq 5$ telle que $\mathcal{F}(f) \equiv 0 \bmod 16$. Alors, tous les éléments du spectre de Walsh de f sont divisibles par 8 et*

$$|\{\varphi \in \mathcal{B}ool_n \text{ avec } \deg(\varphi) \leq 1,\ \mathcal{F}(f+\varphi) \equiv 0 \bmod 16\}| = 2^n + 2^i \text{ ou } 2^n$$

avec $\lceil \frac{n}{2} \rceil \leq i \leq n-1$.

Ce dernier résultat nous permet donc d'établir la liste complète des spectres de Walsh étendus de toutes les fonctions de non-linéarité optimale à 7 variables. Elle est récapitulée au tableau 6.2. Ici encore, il est possible de trouver des fonctions qui possèdent chacun de ces

$\mathcal{F}(f+\varphi_u)$	± 16	± 8	0	degré de f
nombre de u	64		64	2, 3 ou 4
	48	64	16	4 ou 5
	50	56	22	5
	52	48	28	5
	56	32	40	5

TAB. 6.2 – *Spectre de Walsh étendu des fonctions à 7 variables de non-linéarité maximale*

spectres pour chacun des degrés mentionnés [Fon99, Can01b].

6.2 Fonctions équilibrées obtenues par restriction

Quand n est pair, on sait que la plus haute non-linéarité possible pour une fonction à n variables est atteinte par les fonctions courbes. Mais, pour le cryptographe, le problème n'est en fait pas tellement plus facile que dans le cas n impair puisque ces fonctions ne sont pas équilibrées. La détermination de la plus haute non-linéarité atteignable par une fonction équilibrée est une question ouverte depuis de nombreuses années pour $n \geq 8$ (voir par exemple [Fon98]). Dobbertin a conjecturé que la valeur minimale L_n de $\mathcal{L}(f)$ pour une fonction équilibrée à n variables, n pair, était régie par la récurrence

$$L_n = 2^{\frac{n}{2}} + L_{\frac{n}{2}} \ .$$

Cette borne est atteinte par modification d'une fonction courbe normale [Dob94]. En effet, une fonction courbe normale f est par définition constante sur un espace V de dimension $n/2$ — et, par conséquent, équilibrée sur tous les translatés de V. Ainsi, en remplaçant la restriction de f à V par une fonction équilibrée g à $n/2$ variables, on obtient une fonction équilibrée f' qui vérifie

$$\mathcal{L}(f') = 2^{\frac{n}{2}} + \mathcal{L}(g) \ .$$

Une autre possibilité pour obtenir des fonctions équilibrées de haute non-linéarité consiste à modifier une fonction de haute non-linéarité en un petit nombre de points, choisis de manière heuristique. Une méthode de ce type a ainsi permis, à partir de la fonction de Patterson et Wiedemann, de trouver pour tout n impair, $n \geq 15$, une fonction équilibrée à n variables de non-linéarité strictement supérieure à la borne quadratique [SM00a]. Toutefois, si de telles constructions présentent un intérêt indéniable puisqu'elles fournissent des indications sur la

meilleure non-linéarité possible pour une fonction équilibrée, elles ne sont pas utilisables en pratique — sauf dans des applications logicielles et quand le nombre de variables est suffisamment petit pour que la fonction puisse être mise en table.

Les constructions précédentes ayant une mise en œuvre matérielle relativement onéreuse, il nous a donc semblé intéressant de rechercher d'autres techniques permettant de construire des fonctions équilibrées à partir d'une fonction courbe pour un surcoût minime [b] du point de vue de l'implémentation matérielle — même si ces constructions produisent des fonctions de non-linéarité sous-optimale. La somme directe avec une fonction linéaire en est une, mais elle conduit, par définition, à des fonctions qui possèdent des structures linéaires, ce qui introduit souvent des faiblesses [c]. Nous avons donc exploré dans [CCCF00, CCCF01, CC03] une autre piste : la possibilité d'obtenir une fonction adéquate par restriction, c'est-à-dire en fixant un petit nombre des entrées d'une fonction ayant un nombre de variables plus élevé — cette méthode n'est naturellement viable que si le nombre de variables de la fonction de départ n'est pas significativement plus grand que celui de la fonction finalement construite.

En effet, le poids et la non-linéarité de la restriction d'une fonction f à un espace affine sont liés au spectre de Walsh de f par la relation donnée dans la proposition 1.13 page 10. Mais nous avons obtenu des résultats plus précis dans le cas de la restriction de f à un espace V de codimension 1 ou 2 pour lequel $D_a f$ est équilibrée pour tous les a non nuls de V. Dans la suite, on notera H_α l'hyperplan $\langle\alpha\rangle^\perp$ pour tout élément non nul $\alpha \in \mathbf{F}_2^n$.

Proposition 6.5 *[CCCF01, Th. V.3] Soit $n \geq 4$ un entier pair et f une fonction à n variables. Alors, les propriétés suivantes sont équivalentes :*

(i) *Il existe un élément non nul $\alpha \in \mathbf{F}_2^n$ tel que $D_u f$ est équilibrée pour tout u non nul de H_α ;*

(ii) *f est une fonction courbe ;*

(iii) *Pour tout élément non nul $\alpha \in \mathbf{F}_2^n$, les restrictions f_{H_α} et $f_{\overline{H_\alpha}}$ sont des fonctions plateaux vérifiant*

$$\mathcal{L}(f_{H_\alpha}) = \mathcal{L}(f_{\overline{H_\alpha}}) = 2^{\frac{n}{2}}$$

et

$$\mathcal{F}(f_{H_\alpha} + \varphi_\lambda) \neq \mathcal{F}(f_{\overline{H_\alpha}} + \varphi_\lambda),\ \forall \lambda \in \mathbf{F}_2^{n-1}\ .$$

Ainsi, une méthode simple pour obtenir une fonction plateau de haute non-linéarité dépendant d'un nombre impair de variables consiste à fixer une des variables d'entrée d'une fonction courbe. A l'inverse, dans le théorème V.2 de [CCCF01], nous montrons que, pour n impair, les restrictions d'une fonction à n variables à un hyperplan H_α (et à son complémentaire) pour lequel $D_u f$ est équilibrée pour tout $u \in H_\alpha^*$ sont des fonctions courbes.

De la même manière, on peut obtenir des fonctions équilibrées en utilisant les restrictions à un espace de codimension 2, comme le montre le résultat suivant dérivé d'un théorème plus général de [CCCF01] qui inclut également le cas n impair.

Proposition 6.6 *[CCCF01, Th. V.4] Soit $n \geq 4$ un entier pair, f une fonction à n variables et V un espace de dimension $(n-2)$ pour lequel $D_u f$ est équilibrée pour tout $u \in V^*$. Soit W tel que $V \times W = \mathbf{F}_2^n$. Alors, $\mathcal{L}(f) \in \{2^{\frac{n}{2}}, 2^{\frac{n}{2}+1}\}$ et toutes les restrictions f_{b+V}, $b \in W$, ont le même spectre de Walsh étendu et leurs coefficients de Walsh sont à valeurs dans $\{0, \pm 2^{\frac{n}{2}}, \pm 2^{\frac{n}{2}+1}\}$.*

b. On verra en effet dans la partie suivante des exemples de fonctions courbes possédant une réalisation matérielle de coût raisonnable.

c. Nous avons certes vu au chapitre 3 que la fonction de filtrage d'un registre à décalage devait posséder une structure linéaire, mais l'existence d'un espace linéaire de dimension supérieure est une faille au regard de la plupart des attaques.

Si la fonction f d'origine est courbe, on peut alors utiliser la fonction duale $\widetilde{f}$ [Car94a, Wol99] définie par

$$\mathcal{F}(f+\varphi_u) = 2^{n/2}(-1)^{\widetilde{f}(u)} \ , \ \ u \in \mathbf{F}_2^n \ ,$$

pour préciser la forme du spectre de Walsh des restrictions obtenues en fixant deux variables.

Théorème 6.7 *[CC03] Soit f une fonction courbe à n variables, $n \geq 4$ et $\widetilde{f}$ sa duale. Soit a et b deux éléments non nuls distincts de $\mathbf{F}_2^n$, $V = \langle a,b\rangle^\perp$ et W tel que $V \times W = \mathbf{F}_2^n$. Alors, toutes les restrictions de f à $b+V$, $b \in W$ ont le spectre de Walsh étendu suivant :*

$\mathcal{F}(f_{b+V}+\varphi_u)$	$\pm 2^{n/2}$	$\pm 2^{(n-2)/2}$	0
nombre de $u \in \mathbf{F}_2^{n-2}$	$2^{n-4} - 2^{-4}\lambda$	$\lambda/4$	$3(2^{n-4} - 2^{-4}\lambda/4)$

où $\lambda = wt(D_aD_b\widetilde{f})$. En particulier,

- *toutes les f_{b+V} sont courbes si et seulement si $D_aD_b\widetilde{f} = 1$;*
- *toutes les f_{b+V} sont plateaux avec $\mathcal{L}(f) = 2^{\frac{n}{2}}$ si et seulement si $D_aD_b\widetilde{f} = 0$.*

En utilisant les mêmes outils fondés sur la fonction duale nous avons également développé à partir des fonctions courbes une autre construction de fonctions équilibrées de non linéarité presque optimale, inspirée de [Car94b]. Il s'agit ici d'ajouter à f une fonction quadratique correspondant à l'indicatrice d'un espace vectoriel de codimension 2.

Théorème 6.8 *[CC03] Soit f une fonction courbe à n variables, $n \geq 4$ et $\widetilde{f}$ sa duale. Soit a et b deux éléments non nuls distincts de $\mathbf{F}_2^n$, $V = \langle a,b\rangle^\perp$ et Φ_V la fonction indicatrice de V. Alors, le spectre de Walsh de la fonction $f + \phi_V$ est donné par :*

$\mathcal{F}(f+\phi_V+\varphi_u)$	$\pm 2^{n/2+1}$	$\pm 2^{n/2}$	0
nombre de $u \in \mathbf{F}_2^n$	$\frac{1}{4}\lambda$	$2^n - \lambda$	$\frac{3}{4}\lambda$

où $\lambda = wt(D_aD_b\widetilde{f})$. En particulier,

- *$f + \phi_V$ est courbe si et seulement si $D_aD_b\widetilde{f} = 0$;*
- *$f + \phi_V$ est plateau avec $\mathcal{L}(f+\phi_V) = 2^{\frac{n}{2}+1}$ si et seulement si $D_aD_b\widetilde{f} = 1$.*

L'étude des dérivées secondes de la duale de certaines fonctions courbes, en particulier des fonctions de Maiorana-McFarland [McF73] et des fonctions composantes des fonctions puissances nous a permis d'expliciter la forme des sous-espaces qui donnent des fonctions ayant chacun des spectres précédents.

6.3 Composantes des fonctions puissances

Parmi les fonctions booléennes pour lesquelles on possède une réalisation matérielle relativement raisonnable figurent les fonctions composantes des fonctions puissances (ou monômes) $x \mapsto x^s$ sur un corps fini. L'utilisation d'une base normale permet en effet de réduire considérablement le nombre de portes logiques du circuit qui réalise $x \mapsto x^s$, en particulier quand la décomposition binaire de l'exposant s présente des motifs réguliers [ABMV93] ; on peut également diminuer la taille de ce circuit en utilisant d'éventuels sous-corps de $\mathbf{F}_{2^n}$, notamment dans le cas classique où n est une puissance de 2. Les fonctions booléennes composantes d'une fonction puissance sont donc les fonctions de la forme

$$S_\lambda : x \mapsto \mathrm{Tr}(\lambda x^s)$$

où l'espace vectoriel $\mathbf{F}_2^n$ est ici identifié au corps fini à 2^n éléments et Tr désigne la fonction Trace de $\mathbf{F}_{2^n}$ vers $\mathbf{F}_2$. Dans ce contexte, le produit scalaire entre deux éléments x et y est égal à $\mathrm{Tr}(xy)$ et les fonctions linéaires sont les fonctions $\varphi_\mu : x \mapsto \mathrm{Tr}(\mu x)$.

Deux catégories d'exposants s nous semblent particulièrement intéressantes dans le contexte du chiffrement à flot : ceux qui produisent des fonctions équilibrées, et dans le cas où n est pair, ceux qui sont susceptibles de posséder des composantes courbes, puisque ces fonctions de non-linéarité maximale pourront ensuite être « équilibrées » par l'une des méthodes décrites précédemment. Ces deux catégories sont bien entendu caractérisées par le poids de la fonction recherchée, ce qui conditionne la valeur de l'exposant s, comme l'indique la proposition suivante.

Proposition 6.9 *Soit n et s deux entiers avec $s < 2^n - 1$. Alors, pour tout $\lambda \in \mathbf{F}_{2^n}^*$, le poids de Hamming de la fonction $S_\lambda : x \mapsto \mathrm{Tr}(\lambda x^s)$ est divisible par $\mathrm{pgcd}(s,2^n-1)$. En particulier,*

- *S_λ est équilibrée si et seulement si $\mathrm{pgcd}(s,2^n-1) = 1$;*
- *si S_λ est courbe pour $n = 2t$, alors $\mathrm{pgcd}(s,2^{2t}-1) > 1$ et s est premier soit avec (2^t-1), soit avec (2^t+1).*

Preuve. Soit $d = \mathrm{pgcd}(s,2^n-1)$, $e = \frac{2^n-1}{d}$ et U_e le sous-ensemble de taille d défini par

$$U_e = \{x^e,\ x \in \mathbf{F}_{2^n}^*\} .$$

Alors, on peut décomposer $\mathbf{F}_{2^n}^*$ en cosets multiplicatifs de U_e :

$$\mathbf{F}_{2^n}^* = \bigcup_{i=0}^{e-1} \alpha^i U_e$$

où α est un élément primitif de $\mathbf{F}_{2^n}^*$. On voit alors que la fonction S_λ est constante sur chacun de ces cosets : pour tout $x \in \alpha^i U_e$,

$$S_\lambda(x) = \mathrm{Tr}(\lambda\alpha^{is}) .$$

Son poids de Hamming vaut donc

$$wt(S_\lambda) = d\ \left|\{i,\ 0 \le i < e,\ \mathrm{Tr}(\lambda\alpha^{is})\right| .$$

◇

Par ailleurs, le nombre de valeurs de λ étudiées pour un exposant donné peut être réduit en tirant parti du résultat suivant.

Proposition 6.10 *Soit n et s deux entiers avec $s < 2^n - 1$, S la fonction $x \mapsto x^s$ sur $\mathbf{F}_{2^n}$ et $d = \mathrm{pgcd}(s,2^n-1)$. Soit $U_d = \{x^d,\ x \in \mathbf{F}_{2^n}^*\}$. Alors, pour tout $u \in U_d$ et pour tout $\lambda \in \mathbf{F}_{2^n}$, $S_{\lambda u}$ et S_λ sont linéairement équivalentes.*

Preuve. Comme $U_d = U_s$, à tout $u \in U_d$ correspond un élément $v \in \mathbf{F}_{2^n}^*$ tel que $u = v^s$. On a donc

$$S_{\lambda u}(x) = \mathrm{Tr}(\lambda u x^s) = \mathrm{Tr}(\lambda (vx)^s) = S_\lambda(vx) .$$

◇

En particulier, il est bien connu que toutes les composantes (au sens de la définition 1.4) d'une permutation puissance sont linéairement équivalentes. Enfin, dans le cas d'une permutation puissance, le spectre de Walsh de toute composante de $S : x \mapsto x^s$ est identique à celui de toute composante de la fonction inverse $S^{-1} : x \mapsto x^{s^{-1}}$ avec $s^{-1}s \equiv 1 \bmod (2^n-1)$. En effet

$$\mathcal{F}(S_\lambda + \varphi_\mu) = \mathcal{F}(S_\mu^{-1} + \varphi_\lambda) .$$

6.3.1 Composantes équilibrées d'une fonction puissance

Les deux propositions précédentes nous montrent donc que l'étude des fonctions S_λ équilibrées se ramène à l'étude des permutations puissances $x \mapsto x^s$ dans la mesure où le spectre de Walsh étendu de $x \mapsto x^s$ correspond à celui d'une seule composante S_λ avec $\lambda \neq 0$, le nombre d'occurrences de chaque élément du spectre étant multiplié par $2^n - 1$. On en déduit donc notamment les bornes suivantes sur la non-linéarité des fonctions S_λ quand $\mathrm{pgcd}(s, 2^n - 1) = 1$:

- si n est impair,
$$\mathcal{L}(S_\lambda) \geq 2^{\frac{n+1}{2}}$$
avec égalité si et seulement si $S : x \mapsto x^s$ est une permutation presque courbe (AB) [d] de $\mathbf{F}_{2^n}$;
- si n est pair, on conjecture que
$$\mathcal{L}(S_\lambda) \geq 2^{\frac{n}{2}+1} .$$

Les composantes de permutations puissances qui atteignent ces deux valeurs sont étudiées en détail au chapitre 8. En particulier, les tableaux 8.1 et 8.2 donnent la liste des exposants connus s à l'origine de telles fonctions. Nous résumons toutefois brièvement les résultats du chapitre 8 concernant la valeur des moments d'ordre 3 et 4 de leur spectre de Walsh.

Proposition 6.11 *Soit n un entier impair et S_λ la fonction $x \mapsto \mathrm{Tr}(\lambda x^s)$ où s est premier avec $(2^n - 1)$, $\lambda \in \mathbf{F}_{2^n}^*$. Alors,*

$$\begin{aligned} \sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^3(S_\lambda + \varphi_\mu) &\geq 2^{2n+1} \\ \sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^4(S_\lambda + \varphi_\mu) &\geq 2^{3n+1} \end{aligned}$$

avec égalité quand $\mathcal{L}(S_\lambda) = 2^{\frac{n+1}{2}}$.

Dans le cas d'un nombre pair de variables, on a :

Proposition 6.12 *Soit n un entier pair et S_λ la fonction $x \mapsto \mathrm{Tr}(\lambda x^s)$ où s est premier avec $(2^n - 1)$, $\lambda \in \mathbf{F}_{2^n}^*$. Alors,*

$$\begin{aligned} \sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^3(S_\lambda + \varphi_\mu) &\geq 2^{2n+2} \\ \sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^4(S_\lambda + \varphi_\mu) &\geq 2^{3n+1} + 2^{2n+3} . \end{aligned}$$

Entre en particulier dans le cas d'égalité de la proposition précédente la fonction inverse dont une composante est utilisée pour filtrer un LFSR dans le système SFINKS [BLM+05]. Il est important de noter à ce stade que les bornes inférieures sur les moments d'ordre 3 et 4 du spectre de Walsh d'une fonction booléenne équilibrée ne sont valides que dans le cas des composantes d'une permutation puissance, et non pour une fonction f quelconque — même si l'on impose $f(0) = 0$. Par exemple [BCCLC06], la fonction dite de Dobbertin [Dob00], $x \mapsto x^{339}$ sur $\mathbf{F}_2^{10}$, possède une fonction composante S_λ pour laquelle il existe une fonction linéaire φ telle que $f = S_\lambda + \varphi$ est équilibrée et

$$\sum_{\mu \in \mathbf{F}_2^{10}} \mathcal{F}^4(f + \varphi_\mu) = 2^{31} - 19 \cdot 2^{24} .$$

d. La notion de fonction presque courbe est définie à la proposition 7.3.

6.3.2 Composantes courbes d'une fonction puissance

Nous avons mené une recherche exhaustive[e] des exposants s pour lesquels $S : x \mapsto x^s$ sur $\mathbf{F}_{2^n}$, n pair, possède au moins une composante S_λ courbe. Trois familles d'exposants de ce type étaient connues, et leurs propriétés sont détaillées par exemple dans [Lea04] :

- *les exposants quadratiques (de Gold)*, notés $\mathcal{Q}(i)$: $s = 2^i + 1$ avec $1 \leq i \leq \frac{n}{2}$ et $\frac{n}{\text{pgcd}(n,i)}$ pair ;
- *les exposants $\mathcal{PS}_{ap}$ (de Dillon)*, notés $\mathcal{P}(a)$: $s = a(2^{\frac{n}{2}} - 1)$ avec $\text{pgcd}(a,2^{\frac{n}{2}} + 1) = 1$ [Dil74, LW90, CG06] ;
- *les exposants de Kasami (dits aussi de Dillon-Dobbertin) [DD04]*, notés $\mathcal{K}(i)$: $s = 2^{2i} - 2^i + 1$ avec $1 \leq i \leq \frac{n}{2}$ et $\text{pgcd}(i,n) = 1$.

Si les fonctions correspondant aux deux premiers exposants appartiennent à des classes de fonctions courbes bien identifiées, la classe de Maiorana-McFarland [McF73] et celle des Partial Spreads [Dil74, Car94a], la situation est différente pour les exposants de Kasami. En effet, alors que toutes les autres fonctions courbes connues sont faiblement normales, c'est-à-dire affines sur un espace affine de dimension $n/2$, nous avons montré que, pour $n \in \{10,14\}$, cela n'était pas le cas des fonctions courbes obtenues à partir de l'exposant $\mathcal{K}(3) = 57$ [CDLD03, CDLD06]. Ce résultat apportait la réponse à la question ouverte suivante posée par Dobbertin en 1995 [Dob94] : toutes les fonctions courbes sont-elles faiblement normales ?

Nous avons alors également effectué des simulations afin de déterminer la liste complète des exposants conduisant à une fonction courbe pour toutes les valeurs de n paires inférieures ou égales à 20. Nous avons ainsi identifié deux nouvelles familles d'exposants :
(I) $s = (2^{\frac{n}{4}} + 1)^2$ quand n est divisible par 4 ;
(II) $s = 2^{\frac{n}{3}} + 2^{\frac{n}{6}} + 1$ quand n est divisible par 6.
Suite aux conjectures que nous avions établies, le caractère courbe de ces fonctions a alors été démontré par G. Leander [Lea06] pour la première classe, et dans un travail commun avec P. Charpin et G. Kyureghyan [CCK06] pour la seconde. Il s'avère en fait que ces deux fonctions sont équivalentes à des fonctions de la classe Maiorana-McFarland.

exposant s	condition	λ tels que $f_{\lambda,s}$ est courbe	famille[a]
$2^i + 1$	$\frac{n}{\text{pgcd}(n,i)}$ pair	$\lambda \notin \{x^s, x \in \mathbf{F}_{2^n}\}$	$\mathcal{M}$
$a(2^{\frac{n}{2}} - 1)$	$\text{pgcd}(a,2^{\frac{n}{2}} + 1) = 1$	$K(\lambda) = \sum_{x \in \mathbf{F}_{2^t}} (-1)^{\text{Tr}_1^t(1/x + \lambda x)} = 0$[b]	$\mathcal{PS}_{ap}$
$2^{2i} - 2^i + 1$	$\text{pgcd}(i,n) = 1$	$\lambda \notin \{x^3, x \in \mathbf{F}_{2^n}\}$	
$(2^{\frac{n}{4}} + 1)^2$	$n \equiv 0 \bmod 4$	notamment $\lambda = \beta^5$ où $\beta \in \mathbf{F}_{16}$ primitif	$\mathcal{M}$
$2^{\frac{n}{3}} + 2^{\frac{n}{6}} + 1$	$n \equiv 0 \bmod 6$	$\text{Tr}_r^{3r}(\lambda) = 0$ où $r = n/6$	$\mathcal{M}$

TAB. 6.3 – *Composantes courbes des fonctions puissances sur $\mathbf{F}_{2^n}$, n pair*

[a] à équivalence affine près
[b] $\mathcal{K}$ est ici la somme de Kloosterman sur $\mathbf{F}_{2^t}$, $t = n/2$ [LW90].

Les origines de ces cinq types de fonctions courbes semblent notablement différentes, comme en témoigne en particulier la forme du spectre de Walsh des autres composantes de la

e. Les fonctions $x \mapsto \text{Tr}(\lambda x^s)$ et $x \mapsto \text{Tr}(\lambda x^{2s})$ étant équivalentes, on ne considère pour s que le plus élément de sa classe cyclotomique modulo $(2^n - 1)$

fonction puissance associée. Ainsi, les exposants quadratiques et de Kasami proviennent d'une fonction APN. Les autres composantes de ces fonctions sont des fonctions plateaux[f] vérifiant $\mathcal{L}(f_{\lambda,s}) = 2^{\frac{n}{2}+1}$. Par contre, le tableau 6.4 montre par exemple que les spectres obtenus pour les autres familles sont beaucoup plus complexes.

TAB. 6.4 – *Spectre de Walsh des fonctions $f_{\alpha^i,s}$ présentant au moins une composante courbe pour $n = 12$ (on ne donne ici que le premier exposant $\mathcal{PS}_{ap}$). Conformément à la proposition 6.10, on ne considère qu'un seul élément i par classe cyclotomique modulo* $\text{pgcd}(s,2^n-1)$.

	s	$\text{pgcd}(s,2^n-1)$	i	spectre de Walsh	courbe
$\mathcal{Q}(1)$	3	3	0	128 [528] 0 [3072] -128 [496]	
			1	64 [2080] -64 [2016]	oui
$\mathcal{Q}(2)$	5	5	0	256 [136] 0 [3840] -256 [120]	
			1	64 [2080] -64 [2016]	oui
$\mathcal{Q}(3)$	9	9	0	512 [36] 0 [4032] -512 [28]	
			1	64 [2080] -64 [2016]	oui
			3	64 [2080] -64 [2016]	oui
(II)	21	21	0	112 [546] 48 [1092] -16 [1365] -80 [1092] -272 [1]	
			1	112 [546] 48 [1092] -16 [1365] -80 [1092] -272 [1]	
			3	112 [546] 48 [1092] -16 [1365] -80 [1092] -272 [1]	
			5	400 [1] 144 [441] 80 [84] 16 [1764] -48 [1764] -240 [42]	
			7	64 [2080] -64 [2016]	oui
			9	64 [2080] -64 [2016]	oui
$\mathcal{Q}(5)$	33	3	0	128 [528] 0 [3072] -128 [496]	
			1	64 [2080] -64 [2016]	oui
$\mathcal{P}(1)$	63	63	0	568 [1] 56 [2331] -72 [1764]	
			1	316 [1] 60 [2205] -68 [1890]	
			3	68 [1953] -60 [2142] -188 [1]	
			5	820 [1] 52 [2457] -76 [1638]	
			7	64 [2080] -64 [2016]	oui
			9	80 [1575] -48 [2520] -944 [1]	
			11	72 [1827] -56 [2268] -440 [1]	
			13	316 [1] 60 [2205] -68 [1890]	
			15	568 [1] 56 [2331] -72 [1764]	
			21	76 [1701] -52 [2394] -692 [1]	
			23	76 [1701] -52 [2394] -692 [1]	
			27	72 [1827] -56 [2268] -440 [1]	
			31	64 [2080] -64 [2016]	oui
$\mathcal{Q}(6)$	65	65	0	4096 [1] 0 [4095]	
			1	64 [2080] -64 [2016]	oui
			3	64 [2080] -64 [2016]	oui
			5	64 [2080] -64 [2016]	oui
			7	64 [2080] -64 [2016]	oui
			11	64 [2080] -64 [2016]	oui
			13	64 [2080] -64 [2016]	oui
(I)	81	9	0	256 [27] 192 [144] 64 [864] 0 [1755] -64 [1296] -256 [9] -512 [1]	
			1	192 [72] 128 [144] 64 [856] 0 [2016] -64 [792] -128 [144] -192 [72]	
			3	64 [2080] -64 [2016]	oui

f. Dans le cas des exposants de Kasami, cette propriété est uniquement conjecturée.

	s	pgcd(s,2^n-1)	i	spectre de Walsh	courbe
$\mathcal{K}(5)$	159	3	0	128 [528] 0 [3072] -128 [496]	
			1	64 [2080] -64 [2016]	oui

On peut donc utiliser comme fonction de filtrage une de ces fonctions courbes en fixant l'une de ses variables d'entrées conformément à la proposition 6.5, ce qui produit une fonction équilibrée dépendant d'un nombre impair de variables dont la non-linéarité est égale à la borne quadratique. On peut également ajouter à l'une de ces fonctions courbes l'indicatrice d'un sous-espace de dimension 2 correspondant à une dérivée seconde de la fonction duale non nulle, comme au théorème 6.8, ce qui produit une fonction f équilibrée dépendant du même nombre n de variables que la fonction courbe de départ (n pair) qui vérifie $\mathcal{L}(f) = 2^{\frac{n}{2}+1}$. Remarquons toutefois que, pour un emploi direct dans un générateur pseudo-aléatoire, seules les fonctions dérivées des deuxième et troisième lignes du tableau 6.3 sont en général pertinentes puisque les autres fonctions sont de degré au plus 3. Par ailleurs, leur immunité algébrique peut être étudiée à l'aide du résultat de Nawaz, Gong et Gupta énoncé au théorème 5.7, page 63.

6.4 Fonctions symétriques

L'autre famille de fonctions booléennes à faible coût de mise en œuvre à laquelle nous nous sommes intéressées est celle des fonctions symétriques au sens de la définition suivante.

Définition 6.13 *Une fonction booléenne est dite* symétrique *si sa sortie est invariante par toute permutation de ses variables d'entrées.*

Autrement dit, la valeur d'une fonction symétrique ne dépend que du poids de Hamming de son vecteur d'entrée. Une fonction de ce type peut donc être représentée par une fonction

$$v_f : \{0,\ldots,n\} \to \mathbf{F}_2$$

définie par

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

La suite des valeurs prises par v_f, $v(f) = (v_f(0),\ldots,v_f(n))$, est appelée *vecteur simplifié des valeurs de* f. De la même manière, la propriété de symétrie d'une fonction est caractérisée par la symétrie de sa forme algébrique normale, ce qui signifie que f peut être représentée par un polynôme de la forme

$$f(x_1,\ldots,x_n) = \sum_{i=0}^{n} \lambda_f(i) \sum_{\substack{u\in\mathbf{F}_2^n \\ wt(u)=i}} \left(\prod_{j=1}^{n} x_j^{u_j} \right)\ .$$

Les $(n+1)$ coefficients $\lambda_f(i)$, facteurs des polynômes symétriques élémentaires dans la forme algébrique normale de f, forment alors le *vecteur simplifié de l'ANF de* f. Ils sont liés au vecteur simplifié des valeurs de la fonction par les formules [CV05, Prop. 2] :

$$\forall i \in \{0,\ldots,n\},\ v_f(i) = \sum_{k\preceq i} \lambda_f(k) \bmod 2 \text{ et } \lambda_f(i) = \sum_{k\preceq i} v_f(k) \bmod 2\ ,$$

où l'ordre partiel $a \preceq b$ entre deux mots de $\mathbf{F}_2^n$ (ou entre les représentations binaires de deux entiers) est défini par $a_i \leq b_i$ pour tout $1 \leq i \leq n$.

On voit donc que, par définition, une fonction booléenne symétrique à n variables peut être représentée par une table de $(n+1)$ bits, au lieu des 2^n bits nécessaires pour stocker une fonction quelconque. La mise en œuvre matérielle de f est également considérablement simplifiée puisqu'il suffit alors d'implémenter le calcul du poids de Hamming d'un vecteur de n bits et une fonction booléenne de $\lceil \log_2 n \rceil$ bits. Muller et Preparata [MP75] ont proposé une réalisation simple du calcul du poids à l'aide de $(n - \lceil \log_2 n \rceil)$ additionneurs complets et de $(\lceil \log_2 n \rceil - 1)$ demi-additionneurs[g]. On voit alors que toute fonction symétrique à n variables peut être réalisée par un circuit comportant un nombre de portes logiques de l'ordre de $\mathcal{O}(n)$. Il s'agit d'ailleurs de la seule classe connue de fonctions booléennes qui possède cette propriété [Weg87, Hil94], ce qui rend leur utilisation particulièrement attractive dans les algorithmes de chiffrement à flot.

Pour cette raison, nous avons entrepris, dans un travail commun avec M. Videau [CV05], une étude très poussée de leurs propriétés cryptographiques. Ce travail nous a également permis de montrer que les fonctions symétriques de petit degré possédaient une propriété particulièrement intéressante qui facilite à la fois leur étude et leur mise en œuvre matérielle : leur vecteur simplifié des valeurs est en effet périodique.

Théorème 6.14 *[CV05, Th. 1] Soit f une fonction booléenne symétrique à n variables de vecteur simplifié des valeurs $v(f) = (v_0,\ldots,v_n)$ et de vecteur simplifié de l'ANF $\lambda(f) = (\lambda_0,\ldots,\lambda_n)$. Alors $v(f)$ est périodique de période 2^t si et seulement si $\deg(f) \leq 2^t - 1$. De plus, $(v_0,\ldots,v_{2^t-1})$ est le vecteur simplifié des valeurs de la fonction symétrique à (2^t-1) variables ayant $(\lambda_0,\ldots,\lambda_{2^t-1})$ pour vecteur simplifié de l'ANF.*

Ce résultat peut, de plus, être amélioré dans le cas des fonctions dont le degré est une puissance de 2 [CV05, Prop. 4]. Par conséquent, la mise en œuvre d'une fonction symétrique de degré inférieur à 2^t ne nécessite pas le calcul complet du poids de Hamming de son vecteur d'entrée, mais seulement de ses t premières coordonnées, ce qui diminue considérablement la complexité du circuit (voir [Lau05] pour plus de détails). Nous avons notamment utilisé cette propriété dans la mise en œuvre du chiffrement DECIM [BBC+05a, BBC+06] soumis à eSTREAM, qui utilise une fonction de filtrage symétrique quadratique[h]. Ce théorème, associé à une étude précise des propriétés des dérivées et des restrictions des fonctions symétriques, nous a entre autres permis de calculer l'intégralité du spectre de Walsh de toutes les fonctions symétriques de degré 2 et 3. Nous avons également pu nous atteler pour les petits degrés au problème ardu de la recherche de fonctions symétriques équilibrées. En effet, la condition d'équilibre, qui s'exprime par une relation simple sur les coefficients binomiaux : trouver $v_f \in \mathbf{F}_2^{n+1}$ tel que

$$\sum_{i=0}^{n} (-1)^{v_f(i)} \binom{n}{i} = 0 \ ,$$

semble cependant difficile à satisfaire, à l'exception des fonctions que nous avons appelées *trivialement équilibrées*, qui sont les fonctions dépendant d'un nombre n impair de variables qui vérifient

$$v_f(i) = v_f(n-i) + 1, \ \forall\, 0 \leq i \leq n \ .$$

g. Un additionneur complet est une fonction à 3 entrées et 2 sorties (y_0,y_1) qui calcule la somme de ses entrées. Il peut être implémenté à l'aide de 5 portes logiques [MP75, BPP00], par exemple : $y_0 = (x_1 \oplus x_2) \oplus x_3$ et $y_1 = ((x_1 \oplus x_2) \wedge x_3) \vee (x_1 \wedge x_2)$. Un demi-additionneur ne comporte que 2 entrées et peut être réalisé à l'aide de 2 portes.

h. Le fait que la fonction soit quadratique ne pose pas de problème ici car la sortie du LFSR filtré est ensuite décimé de façon irrégulière par l'algorithme ABSG [GSB+05].

Les résultats de simulation [vzGR97] montrent qu'il n'existe pas d'autres fonctions équilibrées non affines que les fonctions trivialement équilibrées pour n impair, $n \leq 128$ sauf pour

$$n \in \{13,29,31,33,35,41,47,61,63,73,97,103\} .$$

De même, pour n pair, $n \leq 128$, les fonctions équilibrées n'existent que si $n \equiv 2 \bmod 6$ ou si

$$n \in \{24,34,48,54\} .$$

Les fonctions équilibrées triviales présentent malheureusement un certain nombre d'inconvénients, ce qui nous a amené à nous demander si l'on pouvait isoler des familles de fonctions équilibrées (et non trivialement équilibrées) de petit degré. Nous avons alors obtenu le résultat négatif suivant, sur l'impossibilité d'équilibre des fonctions symétriques de degré inférieur ou égal 7.

Théorème 6.15 *[CV05, Th. 4 et 5]*

- *Pour tout n pair, $n \geq 2$, il n'y a aucune fonction symétrique équilibrée à n variables de degré inférieur ou égal à 7 à l'exception des deux fonctions de degré 1 et des 4 fonctions à 8 variables suivantes :*

$$\lambda(f) = (\varepsilon,1,1,0,0,0,0,1,0) \text{ et } \lambda(f) = (\varepsilon,1,1,1,0,1,0,1,0),\ \varepsilon \in \mathbf{F}_2 .$$

- *Pour tout n impair, $n \geq 3$, il n'y a aucune fonction symétrique équilibrée à n variables de degré inférieur ou égal à 7 à l'exception des fonctions équilibrées triviales.*

Nous avons aussi établi une nouvelle borne supérieure sur l'ordre de résilience des fonctions symétriques de petit degré.

Proposition 6.16 *[CV05, Th. 2] Soit f une fonction symétrique à n variables telle que*

$$1 < \deg(f) \leq 2^\ell .$$

Alors, f est au plus $(2^\ell - 2)$-résiliente.

Ceci améliore de façon significative la borne générale de Siegenthaler, mais reste malgré tout très loin de la conjecture de von zur Gathen et Roche [vzGR97] selon laquelle il n'existe pas de fonction symétrique 3-résiliente.

Dans l'idée d'utiliser des fonctions symétriques au sein d'un chiffrement à flot, nous nous sommes tout naturellement intéressées à la non-linéarité de ces fonctions. La meilleure non-linéarité possible pour une fonction symétrique à n variables est donnée par

$$\mathcal{L}(f) = 2^{\lfloor \frac{n+1}{2} \rfloor}$$

mais elle est uniquement atteinte pour des fonctions quadratiques. Nous nous sommes donc focalisées sur les fonctions de non-linéarité légèrement moins élevée, espérant obtenir de la sorte des fonctions de plus grand degré.

Proposition 6.17 *[CV05, Th. 6] Soit f une fonction 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

$$v_f(i+2) = v_f(i) + 1, \text{ pour tout } t \leq i \leq n-2-t .$$

Ceci nous a donc permis de déterminer toutes les fonctions symétriques dont la non-linéarité diffère au plus de 2 de la valeur optimale.

Théorème 6.18 *[CV05, Prop 20 et 21]*

- *Les fonctions symétriques f à n variables vérifiant*

$$\mathcal{L}(f) = 2^{\lfloor \frac{n+1}{2} \rfloor} + 2$$

sont les 8 fonctions de degré n ayant pour vecteur simplifié de l'ANF:

$$\lambda_f = (a,b,1,0,\ldots,0,1) \text{ ou } \lambda_f = (a,b,0,1,\ldots,1,1),\ a,b \in \mathbf{F}_2\ .$$

- *Les fonctions symétriques f à n variables vérifiant*

$$\mathcal{L}(f) = 2^{\lfloor \frac{n+1}{2} \rfloor} + 4$$

sont les 4 fonctions de degré $(n-1)$ ayant pour vecteur simplifié de l'ANF:

$$\lambda_f = (a,b,0,1,\ldots,1,0),\ a,b \in \mathbf{F}_2\ .$$

Parmi les quatre fonctions de non-linéarité

$$\mathcal{NL}(f) = 2^{n-1} - 2^{\lfloor \frac{n-1}{2} \rfloor} - 2$$

figurent d'ailleurs des fonctions équilibrées. Leur faible immunité algébrique [BP05] ne permet sans doute pas une utilisation directe comme fonction de filtrage, mais elles pourraient cependant constituer un solide point de départ dans la construction d'une fonction équilibrée de haute non-linéarité ayant un très faible coût de mise en œuvre.

De manière générale, l'immunité algébrique des fonctions symétriques est un sujet qui, s'il a récemment donné lieu à plusieurs articles [BP05, LQ06], demande à être étudié en profondeur. Il est en effet probable que les annulateurs de degré minimal d'une fonction symétrique possèdent une forme particulière[i]. Toutefois, il est important de noter que les résultats publiés à ce sujet dans [BP05] sont erronés. Ainsi, le théorème 1 de [BP05] qui prétend que toute fonction symétrique f à n variables possède un annulateur de degré minimal qui appartient à un sous-ensemble donné de fonctions de degré au plus $\lceil n/2 \rceil$ n'est pas correct : par exemple pour $f = 1 + x_1 \ldots x_n$, le seul annulateur de f est la fonction $1 + f$, de degré n, qui ne peut donc pas appartenir à ce sous-ensemble. De même, la borne donnée au théorème 2 de cet article sur le nombre d'éléments dans ce sous-ensemble est également fausse, le cas $n = 10$ constituant un contre-exemple.

Par ailleurs l'existence au sein de la famille des fonctions symétriques d'une fonction d'immunité algébrique maximale, la fonction majorité, peut aussi laisser penser que ces fonctions n'ont pas a priori un comportement catastrophique vis-à-vis de ce critère.

6.5 Conclusion

Nous avons donc mis en lumière et étudié plusieurs familles de fonctions de filtrage souhaitables ayant un nombre n de variables élevé et une réalisation matérielle de coût raisonnable, en particulier :

- pour n impair, deux classes de fonctions équilibrées f vérifiant $\mathcal{L}(f) = 2^{\frac{n+1}{2}}$ et de spectre de type plateau : les composantes des fonctions puissances presque courbes (AB)

i. Notons que l'ensemble des annulateurs de degré minimal ne contient pas toujours une fonction symétrique.

que nous étudierons plus en détail au chapitre 8, et les restrictions à un hyperplan de fonctions courbes dérivées de fonctions puissances ;

- pour n pair, deux classes de fonctions équilibrées f vérifiant $\mathcal{L}(f) = 2^{\frac{n}{2}+1}$ et de spectre de Walsh à 5 valeurs, déduites des composantes courbes de certaines fonctions puissances soit par restriction à un espace de codimension 2, soit par ajout de l'indicatrice d'un espace de codimension 2 ;
- des fonctions symétriques équilibrées vérifiant $\mathcal{L}(f) = 2^{\lfloor \frac{n+1}{2} \rfloor} + 2^t$, pour $0 \leq t < \lfloor \frac{n+1}{2} \rfloor$.

Des résultats, pour l'instant partiels, tendent à montrer que l'immunité algébrique de ces fonctions n'est pas optimale. Toutefois, une étude plus détaillée est clairement nécessaire car une immunité algébrique maximale n'est pas indispensable dès lors que le nombre de variables est suffisamment élevé. Par ailleurs, il semble pertinent d'examiner d'autres types de constructions qui permettraient, à un moindre coût, de remédier aux points faibles de ces fonctions.

Deuxième partie

Fonctions de substitution pour le chiffrement par blocs

Chapitre 7

Résistance aux attaques statistiques classiques

Les chiffrements par blocs sont les primitives utilisées le plus fréquemment dans les applications pour protéger la confidentialité des données. La possibilité de choisir un mode opératoire approprié, ce qui offre une relative flexibilité, et une vitesse de chiffrement suffisante dans la grande majorité des contextes — de l'ordre d'une vingtaine de cycles du processeur pour chiffrer un octet — plaident clairement en faveur de leur utilisation dans la plupart des cas. Une autre motivation, tout aussi importante, est l'existence depuis une trentaine d'années d'un algorithme standardisé, le DES puis l'AES, ce qui garantit à l'utilisateur l'absence de failles de sécurité grossières et le soulage de bon nombre de problèmes de mise en œuvre.

Au sein de la communauté cryptographique, l'existence du DES, alliée aux nombreuses rumeurs sur une hypothétique « trappe » qu'y aurait placée la NSA, est indéniablement à l'origine du développement d'outils théoriques dédiés tant à la conception qu'à la cryptanalyse des chiffrements par blocs [Vau99, Wag04]. Initié par l'invention par Biham et Shamir de la cryptanalyse différentielle [BS91], ce travail de formalisation a notamment conduit à la définition de plusieurs critères de conception utilisés dans bon nombre des propositions au concours AES. Parmi ces critères, les plus anciens, mais sans doute les plus fondamentaux, sont ceux qui quantifient la résistance aux attaques différentielles et linéaires. Ils ont clairement guidé le choix de la fonction de substitution de l'AES et, alliés au souci de minimiser le coût de la réalisation matérielle, ont conduit à l'emploi d'une fonction dont la forte structure algébrique apparaît comme une faille potentielle qui pourrait être exploitée dans de futures attaques dont les attaques algébriques [CP02] apparaissent comme une première tentative.

C'est dans ce contexte qu'il nous a semblé important d'étudier précisément les propriétés des fonctions de substitution afin de déterminer par exemple si une structure algébrique forte était une nécessaire contrepartie à la possibilité de résister aux attaques connues et/ou d'assurer un débit de chiffrement élevé.

7.1 Le chiffrement itératif par blocs

Un chiffrement par blocs de taille de blocs n est un ensemble de permutations E_K de $\mathbf{F}_2^n$ paramétrées par une clef secrète $K \in \mathbf{F}_2^k$. Définir un chiffrement par blocs pour ce jeu de paramètres revient donc à choisir 2^k permutations de $\mathbf{F}_2^n$. Idéalement, il faut que l'ensemble ainsi sélectionné ne soit pas distinguable d'un ensemble de 2^k éléments choisis aléatoirement

dans l'ensemble des $2^n!$ permutations possibles de $\mathbf{F}_2^n$.

L'immense majorité des chiffrements par blocs utilisés à l'heure actuelle sont des chiffrements dits itératifs (ou par composition). Ils reposent en effet sur l'idée que la technique la plus simple pour atteindre à la fois un niveau de sécurité et un débit élevés consiste à enchaîner un certain nombre de permutations de $\mathbf{F}_2^n$ du même type qui, prises individuellement, sont cryptographiquement faibles mais dont la succession va garantir la solidité. Un chiffrement itératif par blocs est donc formé de la composition de r exemplaires d'une fonction interne paramétrée par une quantité secrète K_i, la clef de tour (ou sous-clef), qui change à chaque itération :

$$E_K = F_{K_r} \circ F_{K_{r-1}} \circ \ldots \circ F_{K_1} \ .$$

Les sous-clefs $(K_1, \ldots, K_r)$ sont, elles, dérivées de la clef secrète par un algorithme dit *de cadencement de clef*. La complexité de la mise en œuvre d'un tel algorithme résulte évidemment d'un compromis entre la complexité de la fonction interne (appelée également fonction de tour) et le nombre d'itérations. Il existe ainsi des algorithmes fondés sur une fonction interne très simple itérée un grand nombre de fois (par exemple Serpent [BAK98]) et d'autres qui emploient une fonction interne plus complexe mais effectuent un nombre d'itérations plus faible.

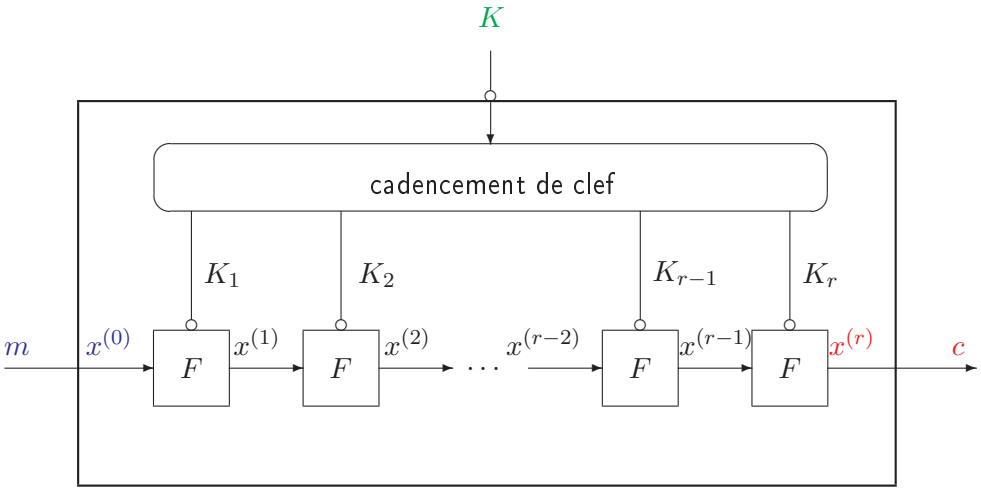

FIG. 7.1 – *Principe du chiffrement par blocs itératif*

La structure de la fonction interne varie donc suivant les algorithmes. Elle doit cependant répondre à deux principes fondamentaux, la diffusion et la confusion, respectivement décrits par Shannon [Sha49] de la manière suivante :

- *In the method of diffusion the statistical structure of M [the plaintext] which leads to its redundancy is "dissipated" into long range statistics.*
- *The method of confusion is to make the relation between the simple statistics of E [the ciphertext] and the simple description of K a very complex and involved one.*

Ces deux principes sont généralement mis en œuvre au sein de deux grandes familles de chiffrements par blocs caractérisées par la forme de leur fonction interne : les réseaux de Feistel et les réseaux de substitution-permutation.

Réseaux de Feistel. Cette structure, formalisée par Feistel au début des années 70 et utilisée notamment dans le DES, repose sur l'emploi d'une fonction interne opérant sur un nombre pair de bits qui traite différemment les deux moitiés de son vecteur d'entrée :

$$\begin{array}{rccc} F_K : & \mathbf{F}_2^{2n} & \to & \mathbf{F}_2^{2n} \\ & (L,R) & \mapsto & (R,L+f_K(R)) \end{array}$$

où f_K est une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$. Cette fonction est généralement elle-même composée d'une fonction affine et d'une fonction de substitution non-linéaire, usuellement appelée boîte-S. La structure de la fonction interne d'un réseau de Feistel garantit ainsi son inversibilité quelle que soit la fonction f_K employée. Elle présente de plus l'avantage que la fonction inverse, utilisée pour le déchiffrement, est identique à une permutation près des deux moitiés de l'entrée. Grâce à cette propriété, le chiffrement et le déchiffrement peuvent être opérés par le même algorithme en inversant simplement l'ordre des sous-clefs. La structure de réseau de Feistel peut également être généralisée au cas où le vecteur d'entrée est divisé en un plus grand nombre de parties [SK96].

Réseaux de substitution-permutation. Cette structure, mise en œuvre par exemple dans l'AES, utilise une fonction interne formée par la composition d'une permutation linéaire assurant la diffusion, d'une fonction de substitution (usuellement appelée boîte-S) assurant la confusion et d'une opération d'insertion de la sous-clef. Contrairement aux réseaux de Feistel, cette structure ne garantit pas l'inversibilité. Il faut donc que chacune des composantes de la fonction interne soit inversible. De plus, le déchiffrement est naturellement opéré en itérant les inverses de chacune de ces composantes.

7.2 Principes des attaques statistiques sur le dernier tour

Il est naturel de tenter de mener sur les chiffrements décrits précédemment une attaque de type « diviser pour mieux régner » afin de tirer parti de leur structure itérative. Ainsi, la famille des attaques dites *sur le dernier tour* vise à retrouver la sous-clef K_r utilisée à la dernière itération de l'algorithme à partir de la connaissance d'un certain nombre de couples clairs-chiffrés[a]. Ces attaques consistent à exploiter la possibilité de distinguer d'une permutation aléatoire la famille $\mathcal{G}$ des chiffrements réduits, c'est-à-dire des fonctions G_K formées des $(r-1)$ premières itérations du chiffrement :

$$G_K = F_{K_{r-1}} \circ \ldots \circ F_{K_1} \ .$$

Elles reposent donc sur l'existence d'un distingueur de $\mathcal{G}$, c'est-à-dire d'une fonction $\mathcal{T}$ qui, à d éléments de $\mathbf{F}_2^n$ $\boldsymbol{x} = (x_1,\ldots,x_d)$ et leurs images par une permutation $(\pi(x_1),\ldots,\pi(x_d))$, associe une valeur binaire telle que la probabilité qu'elle soit égale à 1 quand π est dans l'ensemble des chiffrements réduits est significativement plus élevée que quand π est une permutation aléatoire de $\mathbf{F}_2^n$ — la différence entre ces deux probabilités constitue l'avantage du distingueur. Cet avantage peut être augmenté en faisant appel au distingueur pour N d-uplets d'entrées indépendants, ce qui définit un distingueur non-adaptatif itératif d'ordre d [Vau99].

a. Le reste de la clef peut être retrouvé soit en répétant cette attaque sur les chiffrements obtenus successivement en retirant le dernier tour, soit par une recherche exhaustive des bits restants de la clef secrète (ou par une combinaison des deux techniques précédentes).

L'algorithme pour distinguer le chiffrement réduit d'une permutation aléatoire consiste alors à calculer le nombre de fois s où la fonction $\mathcal{T}$ retourne la valeur 1 et à décider que $\pi \in \mathcal{G}$ quand le logarithme du rapport de vraisemblance

$$L(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N) = \log \frac{\Pr[\sum_{i=1}^{N} \mathcal{T}(\boldsymbol{x}_i,\pi(\boldsymbol{x}_i)) = s | \pi \in \mathcal{G}]}{\Pr[\sum_{i=1}^{N} \mathcal{T}(\boldsymbol{x}_i,\pi(\boldsymbol{x}_i)) = s | \pi \in_R \mathrm{Perm}(\mathbf{F}_{2^n})]} \ .$$

est positif, autrement dit quand s dépasse un certain seuil dont la valeur dépend des probabilités de non-détection et de fausse alarme de $\mathcal{T}$. Toutefois, il faut ici supposer que toutes les fonctions de chiffrement réduit se comportent sensiblement de la même manière vis-à-vis du distingueur. Cette hypothèse est appelée hypothèse d'équivalence des clefs fixées [Har96, HKM95, Kuk99] pour la cryptanalyse linéaire et hypothèse d'équivalence stochastique pour la cryptanalyse différentielle [LMM91] ; le fait qu'elle soit satisfaite est parfois peu évident, comme dans le cas des attaques bilinéaires [Cou04]. Le nombre de requêtes N à effectuer pour distinguer $\mathcal{G}$ avec une probabilité d'erreur raisonnable dépend des probabilités de non-détection et de fausse alarme de $\mathcal{T}$ [Jun05].

Dans ce cadre d'une attaque statistique sur le dernier tour, on applique un distingueur de ce type à des couples d'entrées-sorties de la fonction

$$H_{\widehat{K}} = F_{\widehat{K}}^{-1} \circ E_K = F_{\widehat{K}}^{-1} \circ F_{K_r} \circ F_{K_{r-1}} \circ \ldots \circ F_{K_1}$$

où $\widehat{K}$ décrit l'ensemble des sous-clefs possibles pour le dernier tour. Les couples $(x,H_{\widehat{K}}(x))$ sont déduits des couples clairs-chiffrés en appliquant au chiffré la fonction $F_{\widehat{K}}^{-1}$. On voit ainsi que si $\widehat{K} = K_r$, alors la fonction $H_{\widehat{K}}$ appartient à la famille des chiffrements réduits. Dans le cas contraire, on supposera que $H_{\widehat{K}}$ se comporte comme une permutation tirée aléatoirement selon l'hypothèse dite de répartition aléatoire par fausse clef (*wrong-key randomization*) [Har96, Kuk99]. Un distingueur d'ordre d du chiffrement réduit permet donc d'attaquer le chiffrement itératif par blocs à l'aide de l'algorithme générique décrit à la table 7.1. Nous avons ici supposé

Entrée. N d-uplets de clairs $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N$ et les N d-uplets de chiffrés $\boldsymbol{c}_1,\ldots,\boldsymbol{c}_N$ correspondants.
Sortie. Un ensemble de candidats pour la sous-clef du dernier tour.

Pour toutes les valeurs $\widehat{K}$ possibles pour la sous-clef du dernier tour
  compteur $\leftarrow 0$
  Pour i de 1 à N
    $\boldsymbol{y}_i \leftarrow (F_{\widehat{K}}^{-1}(c_{i,1})\ldots,F_{\widehat{K}}^{-1}(c_{i,d}))$ où $\boldsymbol{c}_i = (c_{i,1},\ldots,c_{i,d})$
    compteur $\leftarrow$ compteur $+ \mathcal{T}(\boldsymbol{x}_i,\boldsymbol{y}_i)$
  Si compteur $\geq T$, alors retourner $\widehat{K}$

TAB. 7.1 – *Algorithme général d'une attaque sur le dernier tour utilisant un distingueur non-adaptatif d'ordre d*

que l'algorithme effectuait une recherche exhaustive sur la sous-clef du dernier tour, ce qui est souvent hors de portée, et même sans intérêt lorsque les sous-clefs ne sont pas plus petites que la clef-maître comme dans l'AES. Dans ce cas, il importe que le choix du distingueur permette de répartir les sous-clefs en classes d'équivalence au sein desquelles

$$\mathcal{T}(x_1,\ldots,x_d,F_{\widehat{K}}^{-1}(y_1),\ldots,F_{\widehat{K}}^{-1}(y_d))$$

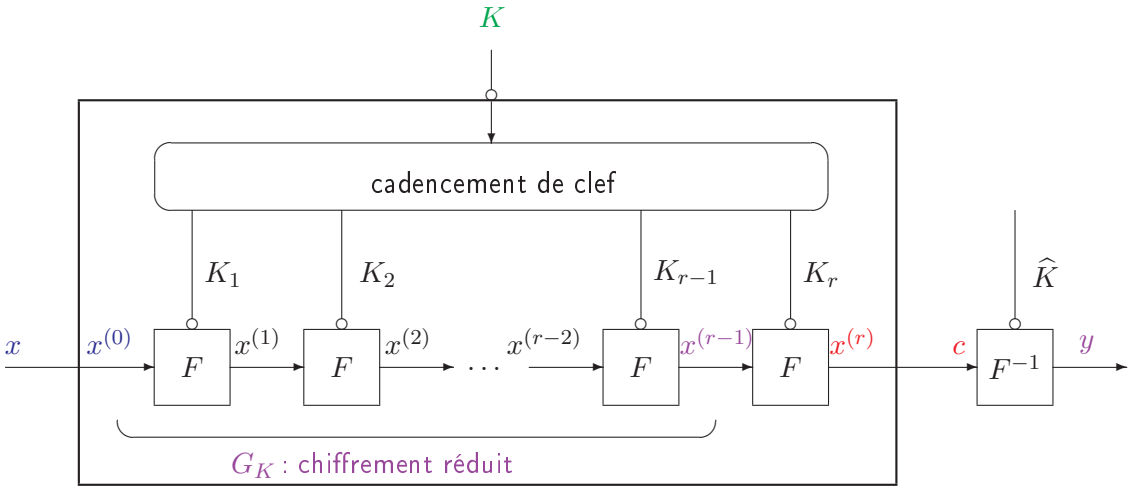

FIG. 7.2 – *Principe des attaques sur le dernier tour*

prend la même valeur. C'est typiquement le cas quand le distingueur ne fait intervenir que certains bits des $F_{\widehat{K}}^{-1}(y_i)$ qui ne dépendent eux-mêmes que de certains bits de $\widehat{K}$.

Les trois attaques classiques qui vont nous intéresser font appel à des distingueurs d'ordres différents : la cryptanalyse linéaire utilise un distingueur d'ordre 1, la cryptanalyse différentielle un distingueur d'ordre 2 et les attaques différentielles d'ordre supérieur un distingueur d'ordre $d+1$, où d est l'ordre de la dérivée utilisée.

7.3 Cryptanalyse différentielle

La cryptanalyse différentielle, introduite par Biham et Shamir [BS91] exploite l'existence d'une dérivée D_aG du chiffrement réduit dont la distribution est éloignée de la distribution uniforme[b]. Cependant, du fait de la grande taille de blocs, la détermination de la distribution complète de la dérivée est généralement hors de portée. On teste donc usuellement une propriété particulière de cette distribution.

Ainsi, l'attaque différentielle classique développée par Biham et Shamir utilise le fait que D_aG prend une certaine valeur b de $\mathbf{F}_2^n$ avec une probabilité plus élevée que celle qui est attendue pour la dérivée d'une permutation choisie aléatoirement. Autrement dit, elle tire parti d'un couple d'éléments (a,b) de $\mathbf{F}_2^n$ tel que toute fonction de chiffrement réduit $G \in \mathcal{G}$ vérifie

$$|\{x \in \mathbf{F}_2^n,\ D_aG(x) = b\}| \gg 1 \ .$$

Certaines variantes de l'attaque différentielle originale exploitent d'autres propriétés de cette distribution. Ainsi, l'attaque par différentielle impossible [BBS99] repose sur l'existence d'une

b. On ne s'intéresse ici qu'au cas où la différence correspond à l'addition sur $\mathbf{F}_2^n$; il s'agit du choix naturel quand la sous-clef est insérée par addition au sein de la fonction interne, car il permet souvent une meilleure maîtrise de l'influence de la clef dans la propagation des différences. Toutefois, il est évidemment possible de mener une attaque similaire pour une différence définie par une loi de groupe autre que celle utilisée pour introduire la sous-clef, mais l'hypothèse de l'équivalence stochastique est alors généralement remise en cause, ce qui rend l'analyse statistique plus délicate.

valeur $b \in \mathbf{F}_2^n$ qui n'est jamais prise par D_aG. L'attaque par différentielle tronquée introduite par Knudsen [Knu95] utilise notamment le fait que la probabilité que la valeur de la dérivée D_aG appartienne à un sous-espace $V \in \mathbf{F}_2^n$ est plus élevée que pour une permutation aléatoire.

Dans l'attaque différentielle classique, la recherche de « bonnes » différentielles se fonde sur l'existence de dérivées de la fonction interne qui présentent cette propriété, autrement dit sur l'existence de couples (a,b) pour lesquels

$$\delta_{F_K}(a,b) = |\{x \in \mathbf{F}_2^n,\ D_aF_K(x) = b\}|$$

est élevé pour toute valeur de la sous-clef K. Le nombre maximum de solutions d'une équation de ce type,

$$\delta(F_K) = \max_{a \neq 0, b} \delta_{F_K}(a,b)$$

étant invariant par composition à gauche et à droite par une permutation linéaire de $\mathbf{F}_2^n$, il apparaît clairement que $\delta(F_K)$ est entièrement déterminé par la fonction de substitution S. De plus, la valeur de $\delta(S)$ fournit parfois une borne supérieure sur la valeur maximale pour tous les couples (a,b), $a \neq 0$, de la moyenne de $\delta_{G_K}(a,b)$ quand G_K décrit l'ensemble des chiffrements réduits — généralement sous l'hypothèse que les différentes sous-clefs sont choisies de manière indépendante selon la distribution uniforme. Un résultat de ce type a été démontré par Nyberg et Knudsen [NK93, NK95] dans le cas des réseaux de Feistel, et dans [HLL$^+$01, PSLL03] notamment pour certains réseaux de substitution-permutation.

Pour ces raisons, nous considérerons que le paramètre $\delta(S)$ permet de quantifier la résistance d'un chiffrement par blocs à la cryptanalyse différentielle classique. Il est donc légitime de rechercher des fonctions de substitution S pour lesquelles cette quantité est minimale.

Proposition 7.1 *[NK93] Soit S une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$ et*

$$\delta(S) = \max_{a \neq 0, b} |\{x \in \mathbf{F}_2^n,\ D_aS(x) = b\}| \ .$$

Alors,

$$\delta(S) \geq 2^{n-m}$$

et les fonctions S atteignant cette borne sont dites parfaitement non-linéaires.

De plus, si $m = n$, alors

$$\delta(S) \geq 2$$

et les fonctions S atteignant cette borne sont dites presque parfaitement non-linéaires (APN).

La dernière assertion de la proposition précédente vient simplement de la constatation que le nombre $\delta_S(a,b)$ est toujours pair puisque D_aS prend la même valeur aux points x et $(x + a)$. Notons que les fonctions S vérifiant $\delta(S) = \delta$ sont qualifiées de *différentiellement δ-uniformes* [NK93].

7.4 Cryptanalyse linéaire

La cryptanalyse linéaire, dont le principe a été initialement développé par Gilbert, Chassé et Tardy-Corfdir [GC91, TCG91] sur le chiffrement FEAL-8 puis appliqué au DES par Matsui [Mat94, Mat95], exploite, elle, l'existence d'une relation linéaire biaisée entre les entrées

et les sorties du chiffrement réduit. Autrement dit, on utilise un couple d'éléments (a,b) non nuls tels que la distribution de la fonction

$$x \mapsto G_a(x) + \varphi_b(x)$$

n'est pas uniforme pour tout chiffrement réduit $G \in \mathcal{G}$, où G_a désigne la composante $x \mapsto a \cdot G(x)$ de G. Alors, si

$$\Pr[\varphi_a \circ G(X) + \varphi_b(X) = 1] = \frac{1}{2} + \varepsilon \text{ avec } \varepsilon \ll 1 \ ,$$

il est possible de distinguer cette distribution à l'aide d'un nombre de couples clairs-chiffrés de l'ordre de ε^{-2}. Notons cependant que le signe du biais ε est généralement inconnu de l'attaquant car il dépend de la clef. Ceci n'entrave toutefois pas la possibilité de développer un test statistique et ce signe est de plus déterminé au cours de l'attaque, fournissant ainsi un bit d'information supplémentaire sur la clef.

La fonction affine utilisée dans l'attaque est obtenue en chaînant des relations affines biaisées entre les entrées, les sorties et la sous-clef de la fonction interne. Le paramètre intervenant ici est donc clairement la non-linéarité de la fonction interne au sens suivant.

Définition 7.2 (Non-linéarité d'une fonction vectorielle) *Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$. Le spectre de Walsh de F est formé par la réunion des spectres de Walsh de ses composantes $F_b = \varphi_b \circ F$, c'est-à-dire*

$$\{\mathcal{F}(F_b + \varphi_a),\ a \in \mathbf{F}_2^n,\ b \in \mathbf{F}_2^m \setminus \{0\}\} \ .$$

La non-linéarité de F est donnée par

$$\mathcal{NL}(F) = 2^{n-1} - \frac{1}{2}\mathcal{L}(F) \text{ où } \mathcal{L}(F) = \max_{a \in \mathbf{F}_2^n, b \in (\mathbf{F}_2^m)^*} |\mathcal{F}(F_b + \varphi_a)| \ .$$

La quantité $\mathcal{L}(F)$ étant trivialement invariante si on la compose à gauche ou à droite avec une permutation affine, la valeur de la non-linéarité de la fonction interne ne dépend ici encore que de la fonction de substitution. De plus, comme pour la cryptanalyse différentielle, il est parfois possible de majorer le biais de la meilleure relation affine pour le chiffrement réduit à partir de la valeur de $\mathcal{L}(S)$ [KMT01, HLL$^+$01, PSLL03]. Nous nous intéresserons donc à cette dernière quantité, dont la valeur maximale est donnée par la proposition suivante.

Proposition 7.3 *Soit S une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$. Alors,*

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

et les fonctions S atteignant cette borne sont dites courbes.

De plus, si $m = n$, alors

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

et les fonctions S atteignant cette borne sont dites presque courbes (AB).

La première partie de ce résultat, due à Nyberg [Nyb91], généralise la notion de fonction booléenne courbe introduite par Rothaus [Rot76] au cas vectoriel. La seconde a été démontrée par Chabaud et Vaudenay [CV95] ; le résultat était toutefois déjà connu dans le cas particulier des fonctions puissances mais dans un cadre applicatif différent [Sid71].

Comme dans le cas booléen traité entre autres par Meier et Staffelbach [MS89], les notions de fonctions courbes et de fonctions parfaitement non-linéaires coïncident.

Proposition 7.4 *[Nyb91] Soit S une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$. Alors, S est courbe si et seulement si elle est parfaitement non-linéaire. De plus, de telles fonctions n'existent que quand n est pair et $n \geq 2m$.*

La détermination de la meilleure non-linéarité atteignable par une fonction dont le nombre de sorties m est compris strictement entre $\frac{n}{2}$ et n reste, elle, un problème ouvert.

A partir de ces résultats, nous allons désormais nous focaliser sur l'étude des fonctions de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ — en particulier parce que ce sont celles qui sont employées dans les réseaux de substitution-permutation — qui offrent la meilleure résistance possible aux cryptanalyses linéaires et différentielles. Nous garderons à l'esprit que ces deux attaques ne sont évidemment pas les seules menaces pesant sur les chiffrements par blocs et que d'autres critères doivent naturellement intervenir dans le choix de la fonction de substitution. Certains d'entre eux seront d'ailleurs étudiés plus en détail par la suite.

Chapitre 8

Fonctions optimales pour les attaques différentielles et linéaires

La caractérisation des fonctions optimales pour les deux critères précédents est donc importante dans l'objectif de concevoir un chiffrement par blocs sûr. Elle a également pour but de déterminer si cette optimalité n'est pas incompatible avec d'autres propriétés indispensables dans ce contexte, par exemple avec un degré univarié et multivarié élevé [JK97]. Mais, le fait que les fonctions presque courbes (AB), qui offrent la meilleure résistance possible à la fois aux attaques différentielles et linéaires, n'existent que pour un nombre impair de variables — ce qui est peu commode dans les applications logicielles — nous a amenée à étudier également des fonctions « sous-optimales » pour ces deux critères. Dans ce contexte, il est indispensable de déterminer précisément la nature du lien entre les critères de résistance aux deux types de cryptanalyse. Nous montrons ici que cette relation s'exprime, de manière générale, au moyen des moments d'ordre 3 et 4 du spectre de Walsh de la fonction de substitution. Ceci nous a alors conduite à introduire un nouveau paramètre dont la valeur permet dans certains cas de rapprocher les deux critères : la divisibilité de son spectre de Walsh, c'est-à-dire la plus grande puissance de 2 qui divise tous les coefficients de Walsh de la fonction. Nous verrons par la suite que cette quantité influence notablement les qualités cryptographiques d'une fonction de substitution.

8.1 Lien avec d'autres objets optimaux

Les fonctions APN et AB sont des objets combinatoires extrémaux. Il n'est donc pas étonnant que ces propriétés apparaissent, sous d'autres noms, dans d'autres cadres applicatifs utilisant des outils de même nature, tels que la correction d'erreurs ou la synchronisation des transmissions numériques. Nous rappelons ici brièvement le lien entre les propriétés mises en jeu dans ces différents domaines, afin notamment de faciliter le passage d'une terminologie à l'autre.

8.1.1 Lien avec la théorie des codes

Le lien entre les propriétés APN et AB d'une part, et les propriétés métriques de certains codes a été mis en lumière par Carlet, Charpin et Zinoviev [CCZ98]. On ne considérera ici que les fonctions F de $\mathbf{F}_2^n$ dans lui-même qui s'annulent en 0 puisque les propriétés étudiées sont

invariantes par translation. On supposera par ailleurs que F ne possède pas de composante [a] constante, ce qui serait clairement une faiblesse cryptographique importante (sauf à considérer F comme une fonction ayant moins de sorties).

Proposition 8.1 *[CCZ98, CCD00b] Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ avec $F(0) = 0$ et $\dim Im(F) = n$. Soit $\mathcal{C}_F$ le code linéaire binaire de longueur $(2^n - 1)$ défini par la matrice de parité*

$$H_F = \begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{2^n-2} \\ F(1) & F(\alpha) & F(\alpha^2) & \dots & F(\alpha^{2^n-2}) \end{pmatrix} , \tag{8.1}$$

où chaque élément de la matrice est vu comme un vecteur-colonne binaire, α est un élément primitif de $\mathbf{F}_{2^n}$ et F est identifiée à une fonction définie sur $\mathbf{F}_{2^n}$. Alors,

- *F vérifie*
$$\mathcal{L}(F) = \max_{c \in \mathcal{C}_F^\perp,\, c \neq 0} |2^n - 2wt(c)| .$$
En particulier, pour n impair, F est AB si et seulement si tout mot non nul c du code $\mathcal{C}_F^\perp$ vérifie
$$2^{n-1} - 2^{\frac{n-1}{2}} \leq wt(c) \leq 2^{n-1} + 2^{\frac{n-1}{2}} .$$
- *F est APN si et seulement si la distance minimale de $\mathcal{C}_F$ est égale à 5* [b].

Dans le cas où la fonction F est une fonction puissance, $F : x \mapsto x^s$ dans $\mathbf{F}_{2^n}$, le code $\mathcal{C}_F$ défini par (8.1) est le code cyclique de longueur $2^n - 1$ dont les zéros sont α et α^s.

8.1.2 Lien avec la théorie des séquences

L'anglicisme « séquence » désigne ici, suivant l'usage en traitement du signal, une suite binaire transmise au cours d'une communication à des fins de synchronisation. Pour qu'une suite $\boldsymbol{s}$ de longueur N soit appropriée pour synchroniser une communication, il faut qu'elle puisse être distinguée facilement de ses décalées, autrement dit que tous ses coefficients d'auto-corrélation

$$\sum_{t=0}^{N-1} (-1)^{s_t + s_{t+\tau}},\ 0 \leq \tau < N$$

soient proches de 0 quand $\tau \neq 0$. Les suites produites par un LFSR de polynôme de rétroaction primitif sont donc particulièrement adaptées à cet usage, et généralement utilisées dans ce contexte.

Quand un système de communication utilise un ensemble de signaux, correspondant généralement à plusieurs utilisateurs, on se trouve donc en présence de plusieurs suites de ce type. Il faut alors que chacune des suites puisse être distinguée facilement de ses décalées, mais aussi de toutes les autres suites et de leurs décalées. Cette propriété s'exprime alors au travers de la fonction de corrélation mutuelle entre deux suites de même longueur [HK98].

Définition 8.2 *Soit $\boldsymbol{u}$ et $\boldsymbol{v}$ deux suites binaires de longueur N. La fonction de corrélation mutuelle entre $\boldsymbol{u}$ et $\boldsymbol{v}$ est définie par*

$$\theta_{\boldsymbol{u},\boldsymbol{v}}(\tau) = \sum_{t=0}^{N-1} (-1)^{u_t + v_{t+\tau}},\ 0 \leq \tau < N .$$

a. au sens de la définition 1.4

b. Notons que la distance minimale du code $\mathcal{C}_F$ est toujours comprise entre 3 et 5 [DZ84, BT93].

Le spectre de corrélation mutuelle entre $\boldsymbol{u}$ et $\boldsymbol{v}$ est donc le multi-ensemble composé de tous les coefficients de corrélation mutuelle.

Si $\boldsymbol{u}$ et $\boldsymbol{v}$ sont toutes deux produites par des LFSRs de longueur n et de polynômes de rétroaction primitifs, il existe un entier s premier avec $N = (2^n - 1)$ et un couple (λ,μ) d'éléments de $\mathbf{F}_{2^n}^*$ tels que

$$u_t = \mathrm{Tr}(\lambda\alpha^t) \text{ et } v_t = \mathrm{Tr}(\mu\alpha^{st}),\ \forall t \geq 0$$

où α est un élément primitif de $\mathbf{F}_{2^n}$. Quand $\lambda = \mu$, on dit que $\boldsymbol{v}$ correspond à la *décimation de $\boldsymbol{u}$ par l'entier s*. En exprimant λ et μ sous forme de puissances de α, $\lambda = \alpha^i$ et $\mu = \alpha^j$, on obtient

$$\theta_{\boldsymbol{u},\boldsymbol{v}}(\tau) = \sum_{t=0}^{N-1} (-1)^{\mathrm{Tr}(\alpha^{i+t}+\alpha^{j+st+\tau})} = \sum_{x\in\mathbf{F}_{2^n}^*} (-1)^{\mathrm{Tr}(\alpha^{\tau'}[\alpha^{i-\tau'}x+x^s])}$$

où $\tau' = j + s\tau$. Le spectre de corrélation mutuelle ne dépendant donc pas de la valeur de j, il suffit d'étudier la corrélation mutuelle entre une suite et sa décimée par s. En utilisant la proposition 6.10, on voit alors que le spectre correspondant est composé des valeurs

$$\mathcal{F}(f + \varphi_\mu) - 1,\ \mu \in \mathbf{F}_{2^n}^*$$

où f est la fonction $x \mapsto \mathrm{Tr}(x^s)$. En particulier[c],

$$\max_\tau |\theta_{\boldsymbol{u},\boldsymbol{v}}(\tau) + 1| = \mathcal{L}(f) = \mathcal{L}(S)$$

où S est la fonction puissance $x \mapsto x^s$. On a donc

$$\max_\tau |\theta_{\boldsymbol{u},\boldsymbol{v}}(\tau) + 1| \geq 2^{\frac{n+1}{2}}$$

et les valeurs de décimation s pour lesquelles on a égalité sont celles pour lesquelles la fonction $x \mapsto x^s$ est presque courbe. Les fonctions presque courbes n'existant pas dans le cas où n est pair, on utilise alors des couples de suites $(\boldsymbol{u},\boldsymbol{v})$ dits *favoris* [SP80] pour lesquels la fonction de corrélation mutuelle prend trois valeurs $\{-1, -1 \pm L\}$ où L est minimal, c'est-à-dire égal à $2^{\frac{n}{2}+1}$.

8.2 Relation entre les deux critères de sécurité

Une question naturelle pour le concepteur qui doit choisir une fonction de substitution est de savoir si les critères de résistance aux attaques linéaires et différentielles sont plutôt incompatibles ou s'ils coïncident dans certains cas. Le fait que les propriétés de fonction courbe et de fonction parfaitement non-linéaire soient en fait équivalentes semble plaider pour la seconde option. Nous formulons ici ce lien d'une manière plus précise à travers l'expression des moments d'ordre 3 et 4 du spectre de Walsh d'une fonction vectorielle.

c. De manière équivalente, ce lien est décrit dans [CCD00b, Prop. 2.9] sous forme de la relation entre le spectre de corrélation mutuelle et l'énumérateur des poids du code cyclique dont les zéros sont α et α^s.

8.2.1 Moments d'ordre 3 et 4 du spectre de Walsh

Le théorème suivant donne la valeur des troisième et quatrième moments du spectre d'une fonction à partir de certaines quantités liées à ses dérivées secondes. L'expression du quatrième moment est donnée à la proposition 6 de [Can01a], et peut également se déduire de [Nyb95]. Si l'on formule le problème en termes de codes linéaires comme à la proposition 8.1, ces expressions correspondent à l'ensemble des identités de Pless [Ple63] jusqu'à l'ordre 4.

Théorème 8.3 *Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ telle que $F(0) = 0$ et F_λ ses composantes, $\lambda \in \mathbf{F}_2^n$. Alors*

$$\begin{aligned}\sum_{\lambda\in\mathbf{F}_2^n\setminus\{0\}}\sum_{\mu\in\mathbf{F}_2^n}\mathcal{F}^3(F_\lambda+\varphi_\mu) &= 2^{2n+1}(2^n-1)+2^{2n}D_0(F)\\ \sum_{\lambda\in\mathbf{F}_2^n\setminus\{0\}}\sum_{\mu\in\mathbf{F}_2^n}\mathcal{F}^4(F_\lambda+\varphi_\mu) &= 2^{3n+1}(2^n-1)+2^{2n}D(F)\end{aligned}$$

où

$$\begin{aligned}D(F) &= |\{(a,b,x),\ a,b\in\mathbf{F}_2^n\setminus\{0\}, x\in\mathbf{F}_2^n, a\neq b,\ D_aD_bF(x)=0\}|\\ D_0(F) &= |\{(a,b),\ a,b\in\mathbf{F}_2^n\setminus\{0\}, a\neq b,\ D_aD_bF(0)=0\}|\end{aligned}$$

Ce théorème nous donne notamment une nouvelle caractérisation des fonctions APN.

Corollaire 8.4 *[BCCLC06, Coro. 1] Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ avec $F(0) = 0$ et F_λ ses composantes, $\lambda \in \mathbf{F}_2^n$. Alors*

$$\sum_{\lambda\in\mathbf{F}_2^n\setminus\{0\}}\sum_{\mu\in\mathbf{F}_2^n}\mathcal{F}^4(F_\lambda+\varphi_\mu)\geq 2^{3n+1}(2^n-1)$$

avec égalité si et seulement si F est APN. De plus,

$$\sum_{\lambda\in\mathbf{F}_2^n\setminus\{0\}}\sum_{\mu\in\mathbf{F}_2^n}\mathcal{F}^3(F_\lambda+\varphi_\mu)\geq 2^{2n+1}(2^n-1)$$

avec égalité notamment dans le cas où F est APN.

Le cas des permutations puissances est particulièrement intéressant car, toutes leurs composantes ayant le même spectre, on peut en déduire des informations sur les moments d'ordre 3 et 4 des spectres de chacune d'entre elles, et non uniquement sur la somme de ces valeurs.

Corollaire 8.5 *Soit S la fonction $x \mapsto x^s$ sur $\mathbf{F}_{2^n}$ où s est premier avec $(2^n - 1)$ et S_λ ses composantes. Alors, pour tout $\lambda \in \mathbf{F}_2^n \setminus \{0\}$, on a*

$$\begin{aligned}\sum_{\mu\in\mathbf{F}_2^n}\mathcal{F}^3(S_\lambda+\varphi_\mu) &= 2^{2n+1}+2^{2n}(\delta_1-2)\\ \sum_{\mu\in\mathbf{F}_2^n}\mathcal{F}^4(S_\lambda+\varphi_\mu) &= 2^{3n+1}+2^{2n}\sum_{c\in\mathbf{F}_2^n}\delta_c(\delta_c-2)\end{aligned}$$

où

$$\delta_c = |x\in\mathbf{F}_{2^n},\ (x+1)^s+x^s=c\}|\ .$$

Golomb et Gong [GG99a] avaient conjecturé que, quand n est premier, toutes les valeurs de s telles que pgcd(s,2^n-1) $=1$ vérifiaient $\delta_1 = 2$, autrement dit

$$\sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^3(S_\lambda + \varphi_\mu) = 2^{2n+1},\ \forall \lambda \neq 0\ .$$

Nos simulations nous ont fourni un contre-exemple de cette conjecture, rapporté dans [CTZ01], pour $n = 17$ et $s = 281$.

8.2.2 Relation entre les propriétés APN et AB

Un certain nombre de caractéristiques des fonctions presque courbes (AB) a été démontré par Chabaud et Vaudenay [CV95] : ils ont notamment prouvé que toutes les composantes d'une fonction AB sont des fonctions plateaux, et que toute fonction AB est également APN, propriété essentielle d'un point de vue cryptographique. Autrement dit, la résistance optimale à la cryptanalyse linéaire garantit une résistance optimale à la cryptanalyse différentielle. La fonction inverse dans $\mathbf{F}_{2^n}$ montre par ailleurs que la réciproque est fausse. Dans ce contexte, nous avons souhaité déterminer sous quelle condition une fonction APN était AB, afin de pouvoir énoncer une sorte de réciproque du résultat de Chabaud et Vaudenay. De plus, cette condition pouvait alors constituer le chaînon manquant pour prouver que certaines fonctions APN étaient également AB.

La condition que nous avons établie dans un travail commun avec H. Dobbertin et P. Charpin est exprimée dans [CCD00b] par un résultat liant les poids de certains codes linéaires et ceux de leur dual. Ce résultat est d'ailleurs plus général que celui que nous allons énoncer ici en termes de fonctions APN et AB car il concerne tous les codes linéaires ayant les mêmes paramètres que ceux décrits dans la proposition 8.1, et pas seulement ceux qui ont la forme (8.1).

Théorème 8.6 *[CCD00b, Th. 3.1] et [CCD99] Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$. Soit*

$$D(F) = |\{(a,b,x),\ a,b \in \mathbf{F}_2^n \setminus \{0\},\, x \in \mathbf{F}_2^n,\, a \neq b,\ D_aD_bF(x) = 0\}|\ .$$

Alors :

(i)
$$D(F) \leq (2^n-1)(\mathcal{L}(F)^2 - 2^{n+1})$$
avec égalité si et seulement si F est une fonction plateau[d].

(ii) *pour tout entier positif L_0 tel que tous les coefficients de Walsh de F satisfont*
$$|\mathcal{F}(F_\lambda + \varphi_\mu)| \geq L_0\ ,\forall \lambda,\mu \in \mathbf{F}_2^n$$
on a
$$D(F) \geq (2^n-1)(L_0^2 - 2^{n+1})\ ,$$
avec égalité si et seulement si F est une fonction plateau vérifiant $\mathcal{L}(F) = L_0$.

Preuve. D'après le théorème 8.3, on a

$$\sum_{\lambda \in \mathbf{F}_2^n \setminus \{0\}} \sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^4(F_\lambda + \varphi_\mu) = 2^{3n+1}(2^n-1) + 2^{2n}D(F)\ .$$

d. Par analogie avec le cas booléen, on appelle *plateau* une fonction vectorielle dont les coefficients appartiennent à $\{0, \pm\,\mathcal{L}(F)\}$.

On en déduit, pour toute valeur de L,

$$\begin{aligned}\sum_{\lambda\in\mathbf{F}_2^n\setminus\{0\}}\sum_{\mu\in\mathbf{F}_2^n}\mathcal{F}^4(F_\lambda+\varphi_\mu)-L^2 2^{2n}(2^n-1) &= \sum_{\lambda\in\mathbf{F}_2^n\setminus\{0\}}\sum_{\mu\in\mathbf{F}_2^n}\left(\mathcal{F}^4(F_\lambda+\varphi_\mu)-L^2\mathcal{F}^2(F_\lambda+\varphi_\mu)\right)\\ &= \sum_{\lambda\in\mathbf{F}_2^n\setminus\{0\}}\sum_{\mu\in\mathbf{F}_2^n}\mathcal{F}^2(F_\lambda+\varphi_\mu)\left(\mathcal{F}^2(F_\lambda+\varphi_\mu)-L^2\right)\\ &= 2^{2n}(2^n-1)\left(2^{n+1}-L^2\right)+2^{2n}D(F)\ .\end{aligned}$$

On obtient alors le résultat en constatant que tous les termes

$$\mathcal{F}^2(F_\lambda+\varphi_\mu)\left(\mathcal{F}^2(F_\lambda+\varphi_\mu)-L^2\right)$$

sont négatifs ou nuls quand $L=\mathcal{L}(F)$, positifs ou nuls quand $L=L_0$ avec L_0 défini au point **(ii)** du théorème, et valent 0 si et seulement si $\mathcal{F}(F_\lambda+\varphi_\mu)\in\{0,\pm L\}$. ◇

Comme la quantité $D(F)$ du théorème précédent s'annule si et seulement si F est APN, on retrouve en corollaire immédiat du point **(i)** le résultat de Chabaud et Vaudenay, qui montre que $\mathcal{L}(F)\geq 2^{\frac{n+1}{2}}$ et que, si cette borne est atteinte, F est plateau et APN. Le second point nous permet maintenant de mettre en évidence la condition sous laquelle une fonction APN est AB.

Corollaire 8.7 *[CCD99] Soit n un entier impair et F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$. Alors, F est AB si et seulement si elle est APN et tous ses coefficients de Walsh sont divisibles par $2^{\frac{n+1}{2}}$.*

Ce résultat nous a donc amené à introduire la notion de divisibilité du spectre de Walsh d'une fonction, au même sens que celle des poids d'un code [McE72].

Définition 8.8 *On dit que le spectre de Walsh d'une fonction vectorielle F est 2^ℓ-divisible si tous les coefficients de Walsh de F sont divisibles par 2^ℓ.*

On dit qu'il est exactement *2^ℓ-divisible si, en plus, au moins l'un des coefficients de Walsh de F n'est pas divisible par $2^{\ell+1}$.*

Notons qu'une fonction de degré d est au plus 2^{n-d+1}-divisible [Car94b, Lemme 3]. Le corollaire 8.7 permet donc de retrouver le fait que le degré d'une fonction AB est inférieur ou égal à $\frac{n+1}{2}$, comme l'ont montré Carlet, Charpin et Zinoviev [CCZ98].

Dans ce contexte, les fonctions puissances semblent particulièrement intéressantes car on dispose du théorème de McEliece [McE72] qui permet de déterminer la divisibilité exacte des poids d'un code cyclique. Dans le cas qui nous intéresse ici, il se formule de la manière suivante :

Proposition 8.9 *[CCD99] Soit S la fonction puissance $x\mapsto x^s$ sur $\mathbf{F}_{2^n}$. Le spectre de Walsh de S est exactement divisible par 2^ℓ si et seulement si, pour tout entier u, $0\leq u\leq 2^n-1$,*

$$w_2(us \bmod (2^n-1))\leq w_2(u)+n-\ell \tag{8.2}$$

où $w_2(i)$ désigne le poids de la représentation binaire de l'entier i.

Grâce à cet outil, démontrer qu'une fonction puissance APN est AB est donc réduit à un problème combinatoire. Cette propriété fournit également un algorithme pour vérifier qu'une telle fonction est AB de manière beaucoup plus efficace que le calcul de la transformée de Walsh. En effet, il suffit de vérifier la condition (8.2) pour les entiers u tels que $w_2(u)\leq (n-1)/2$ — elle est d'ailleurs toujours satisfaite pour tous les u tels que $w_2(u)=(n-1)/2$ quand $\mathrm{pgcd}(s,2^n-1)=1$. Enfin, il suffit naturellement de la vérifier seulement pour un

représentant par classe cyclotomique modulo $(2^n - 1)$. La détermination de la divisibilité exacte du spectre de Walsh d'une fonction puissance revient donc à tester environ $2^{n-1}/n$ valeurs de u, le test de chacune d'entre elles demandant une multiplication modulaire et deux calculs de poids. Un autre facteur important est que cet algorithme ne nécessite aucun stockage, contrairement à tous les algorithmes efficaces de calcul de la transformée de Walsh.

8.3 Fonctions puissances AB

Nous avons donc utilisé intensivement le théorème de McEliece pour déterminer la divisibilité du spectre de Walsh de certaines fonctions puissances afin de conclure, soit qu'une fonction APN était AB, soit à l'inverse qu'une fonction ne pouvait pas être AB car la divisibilité de son spectre était trop faible. Il suffit en effet de considérer la condition (8.2) pour un u bien choisi pour obtenir une borne supérieure sur la divisibilité du spectre de Walsh d'une fonction. Nous nous focaliserons ici essentiellement sur les permutations puissances, c'est-à-dire les fonctions $x \mapsto x^s$ sur $\mathbf{F}_{2^n}$ où s est premier avec $2^n - 1$ dans la mesure où une fonction puissance AB est nécessairement une permutation [e] [Dob, BCCLC06].

8.3.1 Exposants $2^{\frac{n-1}{2}} + 2^i - 1$

Dans [CCD00b] et [CCD00a], nous nous sommes intéressés tout particulièrement à la divisibilité des fonctions puissances sur $\mathbf{F}_{2^n}$, $n = 2t + 1$ impair, dont les exposants sont de la forme

$$s = 2^t + 2^i - 1 \ .$$

En effet, cette classe d'exposants contient deux valeurs pour lesquelles il était conjecturé que la fonction puissance correspondante était AB : dans le cas où $i = 2$, c'est-à-dire $s = 2^t + 3$, il s'agit d'une conjecture formulée par Welch et rapportée par Golomb dans un article datant de 1968 [Gol68b] ; une conjecture similaire figure dans la thèse de Niho [Nih72] pour les cas $i = \frac{t}{2}$ quand $n \equiv 1 \bmod 4$ et $i = \frac{3t+1}{2}$ quand $n \equiv 3 \bmod 4$. Comme H. Dobbertin avait déjà démontré que ces deux fonctions étaient APN [Dob99b, Dob99a], la preuve (toutefois relativement technique) du problème combinatoire posé à la proposition 8.9 nous a permis de démontrer la conjecture de Welch.

Théorème 8.10 *[CCD00a] Soit n un entier impair. La fonction $x \mapsto x^s$ sur $\mathbf{F}_{2^n}$ avec*

$$s = 2^{\frac{n-1}{2}} + 3$$

est AB.

La même technique a ensuite été utilisée par Xiang et Hollmann [HX01] pour démontrer la conjecture de Niho.

Par ailleurs, motivés par la possibilité de trouver d'autres fonctions AB au sein de cette famille d'exposants, nous avons également donné des bornes supérieures sur la divisibilité des autres fonctions puissances de cette famille. Ces résultats sont résumés au théorème suivant.

Théorème 8.11 *[CCD00b] Soit $n = 2t + 1$ un entier impair et S la fonction $x \mapsto x^s$ sur $\mathbf{F}_{2^n}$ avec*

$$s = 2^t + 2^i - 1 \ .$$

e. Ceci n'est pas vrai dans le cas général, comme l'atteste la fonction AB mise en évidence au théorème 1 de [BCP05].

Notons 2^ℓ la divisibilité exacte du spectre de Walsh de S. Alors :

- *si $2 < i < t-1$ et $t \bmod i = 0$, $i \neq \frac{t}{2}$, alors $\ell \leq t$;*
- *si $2 < i < t-1$ et $t \bmod i = 1$, alors $\ell \leq t - i + 3$;*
- *si $2 < i < t-1$ et $t \bmod i \geq 2$, alors $\ell \leq t - i + (t \bmod i) + 1$;*
- *si $t+1 < i < \frac{3t+1}{2}$, alors $\ell \leq n - i + 1$;*
- *si $\frac{3t+1}{2} < i < n-2$, alors $\ell \leq 2(n-i)$;*
- *si $t = n-2$, alors $\ell \leq 4$.*

En conséquence, les valeurs de i pour lesquelles S est 2^{t+1}-divisible sont 1, 2, $\frac{t}{2}$, t, $t+1$, $\frac{3t+1}{2}$, $2t$, et éventuellement $t-1$.

Les valeurs de i de la liste précédente correspondent aux fonctions suivantes :

- $i = 1$ correspond à la fonction quadratique d'exposant $\mathcal{Q}(t)$;
- $i = 2$ correspond à la fonction de Welch ;
- $i = t$ correspond à l'inverse d'une fonction quadratique dont l'exposant appartient à la classe cyclotomique de $\mathcal{Q}(t)$ puisque $(2^{t+1}-1)(2^t+1) \equiv 2^t \bmod 2^n - 1$;
- $i = t+1$ correspond à une fonction équivalente à la fonction de Kasami d'exposant $\mathcal{K}(t)$ car $2^t(2^{t+1}+2^t-1) \equiv 2^{2t} - 2^t + 1 \bmod 2^n - 1$;
- $i = 2t$ appartient à la même classe cyclotomique que l'exposant du cas $i = t$;
- $i = \frac{t}{2}$ et $i = \frac{3t+1}{2}$ correspondent à la fonction de Niho.

Le seul cas non résolu du théorème précédent est donc le cas $i = t-1$, correspondant à l'exposant $s = 2^t + 2^{t-1} - 1$, pour lequel nous avons conjecturé [CCD00b] que la fonction S avait un spectre de Walsh exactement 2^{t+1}-divisible, propriété que nous avons vérifiée pour $n \leq 39$. Il semble toutefois que, de manière générale, cet exposant n'est pas celui d'une fonction AB — sauf dans les cas particuliers $n = 5$ et $n = 7$ où il correspond respectivement à un exposant quadratique et à l'exposant de Welch. En effet, la fonction S ne semble pas être APN, mais ceci a uniquement été démontré pour n multiple de 3 [CTZ97].

8.3.2 Liste des fonctions puissances AB

Ces travaux nous ont donc conduits à la liste des fonctions puissances AB connues donnée au tableau 8.1. Dans cette liste d'exposants, nous ne donnons qu'un représentant par classe cyclotomique, et un seul élément du couple (s, s^{-1}) où s^{-1} désigne l'exposant de la fonction inverse. Nous conjecturons à ce stade que cette liste est complète. Il est important de noter que cette liste ne couvre pas toutes les fonctions AB mais seulement les fonctions puissances, puisque Budaghyan, Carlet et Pott ont mis à jour récemment une fonction AB qui n'est pas affinement équivalente à une fonction puissance [BCP05].

8.3.3 Cas où n n'est pas premier

Une manière de progresser dans la démonstration de l'exhaustivité de la liste précédente consiste à éliminer la plupart des autres exposants en démontrant que le spectre de la fonction S ne possède pas la divisibilité attendue. Nous avons entamé ce travail pour certaines classes particulières d'exposants dans [CCD00b], mais nous avons également obtenu un résultat plus général quand n n'est pas premier.

Théorème 8.12 *[CCD00b, Coro. 7.2] Soit n un entier impair et g un diviseur de n. Si la fonction $x \mapsto x^s$ est AB sur $\mathbf{F}_{2^n}$, alors*

$$s_0 = s \bmod (2^g - 1)$$

	exposants s	
fonctions quadratiques $\mathcal{Q}(i)$	2^i+1 avec pgcd(i,n) $=1$, $1 \leq i \leq t$	[Gol68a, Nyb93]
fonctions de Kasami $\mathcal{K}(i)$	$2^{2i}-2^i+1$ avec pgcd(i,n) $=1$ $2 \leq i \leq t$	[Kas71]
fonction de Welch	2^t+3	[Dob99b, CCD00a]
fonction de Niho	$2^t+2^{\frac{t}{2}}-1$ si t est pair $2^t+2^{\frac{3t+1}{2}}-1$ si t est impair	[Dob99a, HX01]

TAB. 8.1 – *Fonctions puissances AB connues $S : x \mapsto x^s$ sur $\mathbf{F}_{2^n}$ avec $n = 2t+1$*

n'est pas une puissance de 2 et le spectre de $x \mapsto x^{s_0}$ sur $\mathbf{F}_{2^g}$ est $2^{\frac{g+1}{2}}$-divisible.

Ce théorème fournit donc un moyen automatique pour restreindre significativement la liste des exposants à examiner si l'on veut établir la liste des fonctions AB pour une valeur de n élevée mais composite. Notons qu'il est en fait dérivé d'un résultat plus général détaillé dans [CCD00b] qui donne un encadrement de la divisibilité exacte de $x \mapsto x^s$ sur $\mathbf{F}_{2^n}$ en fonction de celle de $x \mapsto x^{s_0}$ sur $\mathbf{F}_{2^g}$ et est également satisfait pour n pair. Appliqué à ce dernier cas, on obtient par exemple la proposition 8.15 que nous détaillerons par la suite.

Une autre famille d'exposants pour lesquels on peut établir une borne sur la divisibilité quand n n'est pas premier est la suivante.

Proposition 8.13 *Soit n un entier et g un diviseur de n. Le spectre de Walsh de la fonction $x \mapsto x^s$ sur $\mathbf{F}_2^n$ est au plus $2^{g(w_2(s_0)+1)}$-divisible où*

$$s \equiv -s_0 \bmod \frac{2^n-1}{2^g-1} \text{ avec } 0 < s_0 < \frac{2^n-1}{2^g-1} .$$

En particulier, si

$$w_2(s_0) \leq \frac{1}{2}\left(\frac{n}{g}-3\right)$$

alors $x \mapsto x^s$ sur $\mathbf{F}_2^n$ n'est pas AB.

Nous avons notamment déduit de cette proposition que l'exposant APN de la fonction de Dobbertin [Dob00] $s = 2^{4g}+2^{3g}+2^{2g}+2^g-1$ pour $n = 5g$ ne correspondait pas à une fonction AB.

8.4 Fonctions de substitution de haute non-linéarité pour n pair

Quand n est pair, la non-linéarité optimale (c'est-à-dire celle des fonctions AB) ne peut pas être atteinte. La plus petite valeur possible de $\mathcal{L}(F)$ vérifie donc

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

Toutefois, la plus petite valeur de $\mathcal{L}(F)$ connue correspond à la borne supérieure et l'existence de fonctions ayant une meilleure non-linéarité est un problème ouvert. Il est notamment conjecturé [SP80] qu'il n'existe pas de fonction puissance S vérifiant $\mathcal{L}(S) < 2^{\frac{n}{2}+1}$.

8.4.1 Fonctions puissances qui ne sont pas des permutations

Nous résolvons ici partiellement cette conjecture puisque nous démontrons qu'elle est satisfaite dans le cas des fonctions puissances qui ne sont pas des permutations — les fonctions puissances APN pour n pair entrent en particulier dans cette catégorie, comme l'avait déjà remarqué H. Dobbertin [Dob].

Théorème 8.14 *Soit $S : x \mapsto x^s$ une fonction puissance sur $\mathbf{F}_{2^n}$, n pair, telle que* $\gcd(s,2^n-1) > 1$. *Alors,*

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

De plus, si $\mathcal{L}(S) = 2^{\frac{n}{2}+1}$, alors $\gcd(s,2^n-1) = 3$ *et*

$$\mathcal{F}(S_\lambda) = \begin{cases} (-1)^{\frac{n}{2}+1} 2^{\frac{n}{2}+1} & si\ \lambda \in \{x^3, x \in \mathbf{F}_{2^n}^*\} \\ (-1)^{\frac{n}{2}} 2^{\frac{n}{2}} & si\ \lambda \notin \{x^3, x \in \mathbf{F}_{2^n}^*\} . \end{cases}$$

Preuve. Soit $d = \gcd(s,2^n-1)$. Alors, $s = ed$, avec $\gcd(e,2^n-1) = 1$. Il s'ensuit que

$$\mathcal{F}(S_\lambda) = \sum_{x \in \mathbf{F}_{2^n}} (-1)^{\mathrm{Tr}(\lambda x^{de})} = \sum_{y \in \mathbf{F}_{2^n}} (-1)^{\mathrm{Tr}(\lambda y^d)} = \mathcal{F}(F_\lambda) ,$$

où F est la fonction $x \mapsto x^d$. Or,

$$\begin{aligned} \sum_{\lambda \in \mathbf{F}_{2^n}} \mathcal{F}^2(F_\lambda) &= \sum_{\lambda \in \mathbf{F}_{2^n}} \sum_{x,y \in \mathbf{F}_{2^n}} (-1)^{\mathrm{Tr}(\lambda x^d) + \mathrm{Tr}(\lambda y^d)} \\ &= \sum_{x,y \in \mathbf{F}_{2^n}} \left(\sum_{\lambda \in \mathbf{F}_{2^n}} (-1)^{\mathrm{Tr}(\lambda (x^d + y^d))} \right) \\ &= 2^n \ |\{(x,y),\ x,y \in \mathbf{F}_2^n,\ x^d = y^d\}| \\ &= 2^n \left(\frac{(2^n-1)}{d}\ d^2 + 1 \right) \\ &= 2^n (2^n-1)d + 2^n \end{aligned}$$

puisque $\{x^d,\ x \in \mathbf{F}_{2^n}^*\}$ est de cardinal $(2^n-1)/d$ et que chacun de ses éléments a d antécédents. On en déduit donc que

$$\sum_{\lambda \in \mathbf{F}_{2^n}^*} \mathcal{F}^2(F_\lambda) = 2^n (2^n-1)(d-1) .$$

Donc

$$\max_{\lambda \in \mathbf{F}_{2^n}^*} \mathcal{F}^2(F_\lambda) \geq 2^n (d-1)$$

avec égalité si et seulement si tous les $\mathcal{F}^2(F_\lambda)$, $\lambda \neq 0$, sont égaux. En particulier,

$$\mathcal{L}(S) \geq \max_{\lambda \in \mathbf{F}_{2^n}^*} |\mathcal{F}(F_\lambda)| \geq 2^{\frac{n}{2}} \sqrt{d-1} . \tag{8.3}$$

On voit ainsi que $\mathcal{L}(S)$ ne peut être inférieur ou égal à $2^{\frac{n}{2}+1}$ que si $d \in \{3,5\}$. Or, les poids des composantes des deux fonctions $x \mapsto x^3$ et $x \mapsto x^5$ sont connus. En particulier, si $d = 5$, on a $\mathcal{L}(S) = 2^{\frac{n}{2}+1}$ d'après (8.3) uniquement si tous les $|\mathcal{F}(F_\lambda)|$ sont identiques, ce qui n'est pas

le cas puisque certaines composantes F_λ sont courbes [Lea06]. Par conséquent, $\mathcal{L}(S) > 2^{\frac{n}{2}+1}$ dans ce cas. Et, pour $F : x \mapsto x^3$, on a d'après [Car79, Th. 1] :

$$\mathcal{F}(F_\lambda) = \begin{cases} (-1)^{\frac{n}{2}+1} 2^{\frac{n}{2}+1} & \text{si } \lambda \in \{x^3, x \in \mathbf{F}_{2^n}^*\} \\ (-1)^{\frac{n}{2}} 2^{\frac{n}{2}} & \text{si } \lambda \notin \{x^3, x \in \mathbf{F}_{2^n}^*\} \end{cases}$$

◇

8.4.2 Liste des permutations puissances de meilleure non-linéarité connue

Intéressons-nous maintenant aux permutations puissances atteignant la meilleure non-linéarité connue, c'est-à-dire celles qui vérifient $\mathcal{L}(S) = 2^{\frac{n}{2}+1}$. La liste des exposants connus (à équivalence par inversion et par le Frobenius près) qui correspondent à de telles fonctions est donnée au tableau 8.2.

exposants s	condition sur n	divisibilité	
$2^{n-1}-1$		2^2	[LW90]
2^i+1 avec $\text{pgcd}(i,n)=2$	$n \equiv 2 \bmod 4$	$2^{\frac{n}{2}+1}$	[Gol68a, Nyb93]
$2^{2i}-2^i+1$ avec $\text{pgcd}(i,n)=2$	$n \equiv 2 \bmod 4$	$2^{\frac{n}{2}+1}$	[Kas71]
$\sum_{i=0}^{n/2} 2^{ik}$ avec $\text{pgcd}(k,n)=1$	$n \equiv 0 \bmod 4$	$2^{\frac{n}{2}}$	[Dob98]
$2^{\frac{n}{2}}+2^{\frac{n+2}{4}}+1$	$n \equiv 2 \bmod 4$	$2^{\frac{n}{2}+1}$	[CD96]
$2^{\frac{n}{2}}+2^{\frac{n}{2}-1}+1$	$n \equiv 2 \bmod 4$	$2^{\frac{n}{2}+1}$	[CD96]
$2^{\frac{n}{2}}+2^{\frac{n}{4}}+1$	$n \equiv 4 \bmod 8$	$2^{\frac{n}{2}}$	[Dob98]

TAB. 8.2 – *Fonctions puissances connues $S : x \mapsto x^s$ sur $\mathbf{F}_{2^n}$ telles que $\mathcal{L}(S) = 2^{\frac{n}{2}+1}$*

8.4.3 Fonctions puissances plateaux de meilleure non-linéarité connue

Les fonctions de meilleure non-linéarité connue les plus simples à étudier sont naturellement les fonctions plateaux. D'après le théorème 8.14, les fonctions puissances de ce type sont nécessairement des permutations. En utilisant la formule des moments d'ordre 3 et 4 pour les permutations puissances, on voit que, si $x \mapsto x^s$ est une fonction plateau ayant la meilleure non-linéarité connue, alors le nombre δ_c de solutions de l'équation $(x+1)^s + x^s = c$ vérifie

$$\delta_0 = 4 \text{ et } \sum_{c \in \mathbf{F}_{2^n}^*} \delta_c(\delta_c - 2) = 2^{n+1} - 8 \ .$$

Toutefois, cette condition ne permet pas de conclure sur la résistance à la cryptanalyse différentielle offerte par cette fonction.

Par ailleurs, notre résultat sur la divisibilité du spectre de Walsh d'une fonction puissance sur $\mathbf{F}_{2^n}$ quand n est un nombre composite, déjà évoqué dans le cas impair, fournit également une condition sur les exposants donnant des fonctions de meilleure non-linéarité connue.

Proposition 8.15 *[CCD00b, Prop. 7.3] Soit n un entier pair et g un diviseur de n. Si la fonction $S : x \mapsto x^s$ sur $\mathbf{F}_{2^n}$ est une fonction plateau avec $\mathcal{L}(S) = 2^{\frac{n}{2}+1}$, alors*

$$s_0 = s \bmod (2^g - 1)$$

n'est pas une puissance de 2 et la fonction $S_0 : x \mapsto x^{s_0}$ sur $\mathbf{F}_{2^g}$ est une fonction plateau avec $\mathcal{L}(S_0) = 2^{\frac{g}{2}+1}$. En particulier[f]*, il n'existe pas de fonction puissance plateau avec $\mathcal{L}(S) = 2^{\frac{n}{2}+1}$ quand n est multiple de 4.*

8.5 Fonctions APN

Comme le montre le corollaire 8.4, les fonctions APN sont caractérisées par la somme des indicateurs par somme des carrés de leurs composantes :

$$\sum_{\lambda \in (\mathbf{F}_2^n)^*} \nu(F_\lambda) = 2^{2n+1}(2^n - 1) \ .$$

Toutefois, cette condition nécessaire et suffisante correspond à une certaine diversité dans les spectres de Walsh.

8.5.1 Fonctions APN pour n impair

Si n est impair, trois spectres de Walsh étendus correspondant à des fonctions APN sont connus :

- celui de $x \mapsto x^3$ dont les éléments sont $\{0, \ \pm 2^{\frac{n+1}{2}}\}$. Il s'agit du spectre des fonctions AB ;
- celui de la fonction inverse, $x \mapsto x^{2^n-2}$ dont les éléments prennent toutes les valeurs $\pm k$ avec k multiple de 4 tel que $0 \leq k < 2^{\frac{n}{2}+1}$ [LW90] ;
- celui de la fonction de Dobbertin, $x \mapsto x^s$ avec $x = 2^{4g} + 2^{3g} + 2^{2g} + 2^g - 1$ pour $n = 5g$ [Dob00], qui diffère des précédents puisqu'il est divisible par 2^g et qu'il contient au moins une valeur qui n'est pas divisible par 2^{2g+1}. Par exemple, pour $n = 15$ (i.e., $g = 3$), il s'agit du spectre $0, \ \pm 2^{2g}, \ \pm 3 \cdot 2^{2g}, \ \pm 2^{2g+2}, \ \pm 5 \cdot 2^{2g}, \ \pm 9 \cdot 2^{2g}$.

Plusieurs familles de fonctions APN non équivalentes à des fonctions puissances ont été mises en évidence dans [BCP05, BCP06, EKP06, BCFL05] mais aucune d'entre elles ne présente un spectre différent des trois précédents. En particulier, le corollaire 8.4 montre qu'une fonction puissance est APN si et seulement si chacune de ses composantes vérifie

$$\nu(F_\lambda) = 2^{2n+1} \ ,$$

qui est la valeur minimale atteignable par l'indicateur par la somme des carrés de la composante d'une fonction puissance. Mais, curieusement, toutes les fonctions APN connues pour n impair vérifient cette condition, même celles qui ne sont pas équivalentes à des fonctions puissances.

La liste des fonctions APN puissances connues pour n impair est donnée au tableau 8.3.

8.5.2 Fonctions APN pour n pair

La question de l'existence de permutations APN quand n est pair est un problème ouvert depuis plusieurs années. On peut aisément prouver qu'il n'existe pas de permutations puissances APN. De plus, la non-existence de permutations APN de $\mathbf{F}_2^n$, n pair, a été démontrée dans les cas particuliers suivants :

- si $n = 4$ [Hou03] ;

f. Ce cas particulier avait été démontré antérieurement par McGuire et Calderbank [MC95].

	exposants s	
fonctions quadratiques $\mathcal{Q}(i)$	2^i+1 avec pgcd$(i,n)=1$, $1\leq i\leq t$	[Gol68a, Nyb93]
fonctions de Kasami $\mathcal{K}(i)$	$2^{2i}-2^i+1$ avec pgcd$(i,n)=1$ $2\leq i\leq t$	[Kas71]
fonction de Welch	2^t+3	[Dob99b, CCD00a]
fonction de Niho	$2^t+2^{\frac{t}{2}}-1$ si t est pair $2^t+2^{\frac{3t+1}{2}}-1$ si t est impair	[Dob99a, HX01]
fonction inverse	$2^{2t}-1$	[Nyb93, BD93]
fonction de Dobbertin	$2^{4g}+2^{3g}+2^{2g}+2^g-1$ avec $n=5g$	[Dob00]

TAB. 8.3 – *Fonctions puissances APN connues $S: x\mapsto x^s$ sur $\mathbf{F}_{2^n}$ avec $n=2t+1$*

- si $F\in\mathbf{F}_{2^t}[X]$ avec $n=2t$ [Hou03] ;
- si toutes les composantes de F, F_λ, $\lambda\in\mathbf{F}_2^n\setminus\{0\}$ sont plateaux [BCCLC06] [g] ;
- si $\sum_{i=0}^{\frac{2^n-1}{3}} a_{3i}=0$ où $\sum_{i=0}^{2^n-1}a_iX^i$ est l'expression de F dans $\mathbf{F}_{2^n}[X]$ [Can97].

L'ensemble des fonctions APN connues dans le cas pair correspond à deux spectres étendus :

- celui de $x\mapsto x^3$ dont les éléments sont $\{0,\ \pm 2^{\frac{n}{2}},\ \pm 2^{\frac{n}{2}+1}\}$;
- celui de la fonction de Dobbertin qui diffère du précédent pour les raisons de divisibilité déjà évoquées.

Mais, la situation est différente du cas impair où toutes les composantes des fonctions APN connues avaient le même $\nu(F_\lambda)$. Par exemple, la fonction APN trouvée dans [BCP05]:

$$G: x\mapsto x^{2^i+1}+(x^{2^i}+x+1)\mathrm{Tr}(x^{2^i+1}),\ 1\leq i<\frac{n}{2},\ \mathrm{pgcd}(i,n)=1$$

a par construction le même spectre de Walsh que $x\mapsto x^3$. Mais, pour $n=6$ et $i=1$, $\nu(G_\lambda)$, $\lambda\neq 0$, prend 30 fois la valeur 2^{12}, 24 fois la valeur $2^{13}+2^{11}$ et 9 fois la valeur 2^{14}.

La liste des exposants APN connus dans le cas pair est donnée au tableau 8.4. Du fait que

	exposants s	
fonctions quadratiques $\mathcal{Q}(i)$	2^i+1 avec pgcd$(i,n)=1$	[Gol68a, Nyb93]
fonctions de Kasami $\mathcal{K}(i)$	$2^{2i}-2^i+1$ avec pgcd$(i,n)=1$ $2\leq i\leq t$	[Kas71]
fonction de Dobbertin	$2^{4g}+2^{3g}+2^{2g}+2^g-1$ avec $n=5g$	[Dob00]

TAB. 8.4 – *Fonctions puissances APN connues $S: x\mapsto x^s$ sur $\mathbf{F}_{2^n}$ avec $n=2t$*

les fonctions puissances APN correspondent à des exposants s pour lesquels pgcd$(s,2^n-1)=$

g. Le cas particulier des fonctions quadratiques est dû à Nyberg [Nyb95]

3 [BCCLC06, Prop. 3][Dob], on déduit, de manière similaire au théorème 8.14, une borne sur leur non-linéarité.

Théorème 8.16 *[BCCLC06, Th. 4] Soit $S : x \mapsto x^s$ une fonction puissance APN sur $\mathbf{F}_{2^n}$, n pair. Alors,* $\mathrm{pgcd}(s,2^n-1) = 3$ *et*

$$\mathcal{F}(S_\lambda) = \begin{cases} (-1)^{\frac{n}{2}+1} 2^{\frac{n}{2}+1} & si\ \lambda \in \{x^3, x \in \mathbf{F}_{2^n}^*\} \\ (-1)^{\frac{n}{2}} 2^{\frac{n}{2}} & si\ \lambda \notin \{x^3, x \in \mathbf{F}_{2^n}^*\} \end{cases}$$

ce qui implique en particulier que

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

8.6 Permutations puissances différentiellement δ-uniformes

Comme aucune permutation APN n'est à ce jour connue pour un nombre pair de variables, les chiffrements à vocation logicielle — qui opèrent naturellement sur des mots dont la taille est une puissance de 2 — doivent utiliser de préférence des fonctions de substitution sous-optimales pour la cryptanalyse différentielle, c'est-à-dire différentiellement 4-uniformes. Malheureusement, le lien entre résistance à la cryptanalyse linéaire et résistance à la cryptanalyse différentielle semble se relâcher au fur et à mesure que l'on s'éloigne de la situation optimale. Le quatrième moment du spectre de Walsh de la fonction est encore minoré par une quantité qui dépend de la valeur de $\delta(F)$ [BCCLC06, Prop. 6], mais il peut prendre la même valeur pour des $\delta(F)$ très différents. Dans le cas particulier des permutations puissances de $\mathbf{F}_{2^n}$ avec n pair, on peut toutefois obtenir les résultats suivants sur les moments d'ordre 3 et 4 des fonctions différentiellement 4-uniformes.

Proposition 8.17 *[BCCLC06] Soit S une permutation puissance sur $\mathbf{F}_{2^n}$, n pair. Alors,*

$$\begin{aligned} \sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^3(S_\lambda + \varphi_\mu) &\geq 2^{2n+2} \\ \sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^4(S_\lambda + \varphi_\mu) &\geq 2^{3n+1} + 2^{2n+3} \end{aligned}$$

avec égalité dans les deux équations quand $\delta(S) = 4$. De plus,

$$\sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^4(S_\lambda + \varphi_\mu) = 2^{3n+1} + 2^{2n+3}$$

si et seulement si le nombre de solutions $x \in \mathbf{F}_{2^n}$ de chacune des équations

$$(x+1)^s + x^s = c$$

est égal à 0 ou à 2 pour tout $c \in \mathbf{F}_2^n \setminus \mathbf{F}_2$.

Ainsi le quatrième moment du spectre de Walsh de la fonction inverse sur $\mathbf{F}_{2^n}$, égal à la borne ci-dessus [CHZ06], a donc la plus petite valeur possible pour une permutation puissance dépendant d'un nombre pair de variables.

La table 8.5 donne la liste des permutations puissances différentiellement 4-uniformes sur $\mathbf{F}_{2^n}$ pour n pair, $n \leq 16$.

n	s	$w_2(s)$	s^{-1}	$w_2(s^{-1})$	δ_1 [a]	$\mathcal{L}(S)$	divisibilité [b]	
$n=8$	127	7	127	7	4	32	2	inverse
$n=10$	5	2	205	5	4	64	6	$\mathcal{Q}(2)$
	17	2	181	5	4	64	6	$\mathcal{Q}(4)$
	13	3	79	5	4	64	6	$\mathcal{K}(2)$
	29	4	247	7	4	144	4	
	103	5	149	4	4	96	4	
	223	7	367	7	4	80	4	
	511	9	511	9	4	64	2	inverse
$n=12$	73	3	731	7	4	128	6	
	2047	11	2047	11	4	128	2	inverse
$n=14$	5	2	3277	7	4	256	8	$\mathcal{Q}(2)$
	17	2	2893	7	4	256	8	$\mathcal{Q}(4)$
	65	2	2773	7	4	256	8	$\mathcal{Q}(6)$
	13	3	1339	7	4	256	8	$\mathcal{K}(2)$
	241	5	205	5	4	256	8	$\mathcal{K}(4)$
	319	7	979	7	4	256	8	$\mathcal{K}(6)$
	8191	13	8191	13	4	256	2	inverse
$n=16$	32767	15	32767	15	4	512	2	inverse

TAB. 8.5 – *Liste des exposants s premiers avec (2^n-1) tels que $S: x \mapsto x^s$ est une permutation différentiellement 4-uniforme sur $\mathbf{F}_{2^n}$, n pair*

[a] $\delta_1(S)$ est le nombre de solutions $x \in \mathbf{F}_{2^n}$ de l'équation $(x+1)^s + x^s = 1$.

[b] On donne ici la valeur de ℓ telle que le spectre de S est exactement 2^ℓ-divisible.

D'après la proposition précédente, les spectres de Walsh de toutes ces fonctions ont le même moment d'ordre 3 :

$$\sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^3(S_\lambda + \varphi_\mu) = 2^{2n+2} .$$

Cette liste contient naturellement la fonction inverse pour tout n pair, de non-linéarité donnée par $\mathcal{L}(S) = 2^{\frac{n}{2}+1}$ et dont le moment d'ordre 4 du spectre de Walsh est égal à la valeur minimale $2^{3n+1} + 2^{2n+3}$. Elle contient les exposants quadratiques $\mathcal{Q}(2i)$ avec i premier avec $n/2$ car la fonction quadratique $S : x \mapsto x^s$ avec $s = 2^i + 1$ sur $\mathbf{F}_{2^n}$ est différentiellement $2^{\mathrm{pgcd}(i,n)}$-uniforme. Cette fonction est une permutation si et seulement si n n'est pas multiple de 4. De même, l'exposant de Kasami $\mathcal{K}(2i)$ avec i premier avec $n/2$ définit une permutation quand n n'est pas multiple de 4. Ces deux types d'exposants correspondent à des fonctions de non-linéarité donnée par

$$\mathcal{L}(S) = 2^{\frac{n}{2}+1}$$

et de spectre de Walsh plateau, ce qui implique

$$\sum_{\mu \in \mathbf{F}_2^n} \mathcal{F}^4(S_\lambda + \varphi_\mu) = 2^{3n+2} .$$

Le spectre de Walsh pour $s = 2^i + 1$ est donné par exemple dans [MS77, page 453]. Dans le cas des exposants de Kasami, le résultat vient du fait que, dans les conditions mentionnées précédemment, on a

$$2^{2i} - 2^i + 1 = \frac{2^{3i} + 1}{2^i + 1}$$

ce qui implique que

$$\begin{aligned} \mathcal{F}(S_\lambda + \varphi_\mu) &= \sum_{x \in \mathbf{F}_{2^n}} (-1)^{\mathrm{Tr}(\lambda x^{2^{3i}+1} + \mu x^{2^i+1})} \\ &= \mathcal{F}((Q_1)_\lambda + (Q_2)_\mu) \end{aligned}$$

où Q_1 et Q_2 sont les deux fonctions puissances quadratiques d'exposants $2^{3i} + 1$ et $2^i + 1$. On constate alors à la vue de la table 8.5 que pour $n = 16$, la fonction inverse est la seule permutation différentiellement 4-uniforme. Afin de trouver une alternative puisque l'inverse peut présenter certaines faiblesses du fait de l'existence de relations quadratiques liant ses entrées et sorties, nous donnons également au tableau 8.6 la liste des exposants correspondant à des permutations puissances différentiellement 6-uniformes. Nous ne fournissons cette liste que pour n multiple de 4 car ces exposants sont en relativement grand nombre quand $n \equiv 2 \bmod 4$.

n	s	$w_2(s)$	s^{-1}	$w_2(s^{-1})$	δ_1	$\mathcal{L}(S)$	divisibilité
$n = 8$	7	3	37	3	4	64	4
$n = 12$	187	6	943	8	4	288	4
	341	5	853	6	4	192	4
$n = 16$	7	3	28087	11	4	1024	6
	1057	3	2047	11	4	1024	6

TAB. 8.6 – *Liste des exposants s premiers avec $(2^n - 1)$ tels que $S : x \mapsto x^s$ est une permutation différentiellement 6-uniforme sur $\mathbf{F}_{2^n}$, n multiple de 4*

Notons que la fonction $x \mapsto x^7$ est différentiellement 6-uniforme comme nous l'avons démontré dans [Can97]. Il s'agit d'une permutation sur $\mathbf{F}_{2^n}$ si et seulement si pgcd$(n,3) = 1$. Les propriétés de cette fonction sont détaillées à la proposition suivante.

Proposition 8.18 *[Can97] Pour tout n, la fonction $S : x \mapsto x^s$ est différentiellement 6-uniforme sur $\mathbf{F}_{2^n}$. Le nombre δ_c de solutions $x \in \mathbf{F}_{2^n}$ de l'équation $(x+1)^7 + x^7 = c$, $c \in \mathbf{F}_{2^n}$, vérifie*

$$\delta_c \in \{0,2,4,6\} ,$$

avec, si n est pair :

$$\begin{aligned} |\{c \in \mathbf{F}_{2^n}, \delta_c = 6\}| &= \frac{2^n - 13}{24} - \frac{1}{2^{n+2}} \sum_{i=0}^{\frac{n}{2}} \binom{n}{2i} (-7)^i \\ |\{c \in \mathbf{F}_{2^n}, \delta_c = 4\}| &= 1 \end{aligned}$$

et si n est impair

$$\begin{aligned} |\{c \in \mathbf{F}_{2^n}, \delta_c = 6\}| &= \frac{2^n + 1}{24} - \frac{1}{2^{n+2}} \sum_{i=0}^{\frac{n-1}{2}} \binom{n}{2i} (-7)^i \\ |\{c \in \mathbf{F}_{2^n}, \delta_c = 4\}| &= 0 . \end{aligned}$$

De manière similaire, nous avons établi aux tables 8.7 et 8.8 la liste des permutations puissances différentiellement 4-uniformes et 6-uniformes sur $\mathbf{F}_{2^n}$ pour n impair, $n \leq 17$.

n	s	$w_2(s)$	s^{-1}	$w_2(s^{-1})$	δ_1	$\mathcal{L}(S)$	divisibilité
$n = 7$	19	3	47	5	2	16	3
$n = 9$	45	4	125	6	2	48	4
$n = 11$	79	5	183	6	2	128	5
	109	5	695	7	2	128	5
	251	7	367	7	2	128	5
	463	7	703	8	2	128	4
$n = 13$	303	6	947	7	2	256	6
$n = 15$	aucun						
$n = 17$	aucun						

TAB. 8.7 – *Liste des exposants s premiers avec $(2^n - 1)$ tels que $S : x \mapsto x^s$ est une permutation différentiellement 4-uniforme sur $\mathbf{F}_{2^n}$, n impair*

8.7 Conclusion

Le fait que les fonctions AB n'existent que pour un nombre impair de variables conduit à utiliser des fonctions sous-optimales au regard de la résistance aux attaques différentielles et linéaires. De plus, ce choix est naturel pour le cryptographe qui forme ainsi l'espoir de

n	s	$w_2(s)$	s^{-1}	$w_2(s^{-1})$	δ_1	$\mathcal{L}(S)$	divisibilité
$n=7$	7	3	55	5	2	40	3
	21	3	31	5	2	40	3
$n=9$	61	5	111	6	2	112	4
	83	4	117	5	2	112	4
$n=15$	73	3	4941	7	2	512	7
	363	6	3701	8	2	640	7
	521	3	4717	7	2	512	7
	1095	5	7871	11	2	736	5
	1809	5	2957	7	2	640	6
	3247	8	5843	8	2	704	6

TAB. 8.8 – *Liste des exposants s premiers avec (2^n-1) tels que $S : x \mapsto x^s$ est une permutation différentiellement 6-uniforme sur $\mathbf{F}_{2^n}$, n impair. Les exposants pour $n = 11$ et $n = 13$ sont trop nombreux pour être mentionnés*

réduire le risque qu'une structure forte fournisse au cryptanalyste une aide inespérée. Mais cette tâche est actuellement extrêmement difficile, car il n'est pas surprenant qu'un objet dont la structure offre moins de prise à un attaquant en offre également moins à qui veut déterminer ses propriétés cryptographiques. En particulier, on constate ici que le lien entre les critères de résistance aux deux types d'attaques semble se relâcher au fur et à mesure que l'on s'éloigne de la situation optimale. Il devient donc beaucoup plus difficile de construire des objets intéressants pour les deux critères. Toutefois, nos résultats numériques semblent ouvrir un certain nombre de perspectives concernant en particulier la recherche de permutations puissances différentiellement 4 et 6-uniformes.

Chapitre 9

Divisibilité et cryptanalyse différentielle d'ordre supérieur

La recherche de fonctions optimales au regard des cryptanalyses différentielle et linéaire nous a amenée à introduire un nouveau paramètre significatif du spectre de Walsh d'une fonction de substitution, important pour en caractériser les qualités cryptographiques. En particulier, la plupart des fonctions connues de haute non-linéarité présentent un spectre de Walsh divisible par une grande puissance de 2, le cas extrémal étant celui des fonctions AB. Nous allons voir maintenant que la divisibilité intervient également dans un autre type d'attaques, les attaques différentielles d'ordre supérieur [Knu95]. Malheureusement, comme c'est souvent le cas en cryptographie, c'est la forte divisibilité du spectre de Walsh, caractéristique des fonctions optimales, qui facilite cette cryptanalyse. Cette propriété, au cœur du travail exposé dans [CV02], permet en particulier d'expliquer l'origine de l'attaque présentée par Tanaka, Hisamatsu et Kaneko [THK99] sur une version réduite de l'algorithme MISTY1.

9.1 Principe général

La cryptanalyse différentielle d'ordre supérieur exploite, comme son nom l'indique, un biais dans la distribution d'une dérivée d'ordre supérieur, notion définie de la manière suivante.

Définition 9.1 (Dérivée d'ordre supérieur) [a] *Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$. Pour tout sous-espace vectoriel V de $\mathbf{F}_2^n$ de dimension k, la dérivée d'ordre k de F par rapport à V est la fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$, notée $D_V F$, définie par*

$$D_V F(x) = D_{a_1} D_{a_2} \dots D_{a_k} F(x) = \sum_{v \in V} F(x+v)$$

où $(a_1, \dots, a_k)$ est une base [b] *de V.*

L'attaque introduite par Knudsen consiste alors à rechercher un espace vectoriel V tel que la dérivée $D_V G$ est constante et égale à $\gamma \in \mathbf{F}_2^n$ pour toute fonction de chiffrement réduit $G \in \mathcal{G}$ — la constante γ étant identique pour toutes les $G \in \mathcal{G}$, c'est-à-dire indépendante de

a. Cette notion relativement naturelle est généralement attribuée à Lai [Lai94], mais elle est déjà mentionnée en 1974 dans la thèse de Dillon [Dil74].

b. On voit naturellement que $D_V F$ est indépendante du choix de la base de V.

la clef. Dans ce cas, on dispose d'un distingueur $\mathcal{T}$ d'ordre $\dim V$ qui s'applique aux images par une permutation π des éléments du sous-espace $x+V$ et qui retourne 1 si et seulement si

$$\sum_{v \in V} \pi(x+v) = \gamma \ .$$

Cette relation étant vérifiée avec probabilité 1 si π est une fonction de chiffrement réduit, le nombre de requêtes à effectuer est très faible et inversement proportionnel à la taille de bloc. La complexité de l'attaque en temps et en données est donc essentiellement déterminée par la dimension de l'espace V car elle est de l'ordre de $2^{\dim V}$.

Le problème principal est alors de trouver un espace V ayant la dimension la plus faible possible tel que $D_V G = \gamma$ pour tout $G \in \mathcal{G}$. Un candidat naturel pour V, mais pas nécessairement de dimension minimale, se déduit du degré multivarié des chiffrements réduits, défini ici[c] au sens du degré maximal des composantes de G

$$\deg G = \max_{\lambda \in \mathbf{F}_2^n} \deg G_\lambda \ .$$

En effet, le degré de la dérivée (d'ordre 1) d'une fonction est strictement inférieur au degré de la fonction, ce qui implique trivialement le résultat suivant.

Proposition 9.2 *[Lai94] Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$. Alors, pour tout sous-espace V de dimension $(\deg F + 1)$, on a*

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

Si l'on souhaite tirer parti de cette constatation pour mener une attaque différentielle d'ordre supérieur, il faut donc majorer le degré du chiffrement réduit. La borne triviale

$$\max_{G \in \mathcal{G}} \deg G \leq \left(\max_K \deg(F_K) \right)^{r-1}$$

où r est le nombre de tours de chiffrement et F_K désigne la fonction interne[d] ne fournit cependant de résultat exploitable que quand le degré de la fonction interne est extrêmement faible. Elle n'est d'ailleurs d'aucun intérêt si le degré de la fonction interne excède la racine carrée de la taille de bloc, auquel cas on obtient uniquement l'inégalité triviale

$$\max_{G \in \mathcal{G}} \deg G \leq n$$

dès que $r \geq 3$. Notons tout de même que cette borne a permis à Jakobsen et Knudsen [JK97] de cryptanalyser l'algorithme proposé dans [NK95] qui reposait sur la fonction de substitution AB $x \mapsto x^3$.

c. Cette définition du degré d'une fonction vectorielle n'est pas la seule dans la littérature car le degré est parfois défini au contraire comme le plus petit degré des fonctions coordonnées [GS03].

d. Comme pour les attaques que nous avons précédemment étudiées, on supposera que le degré de la fonction interne est égal à celui de la fonction de substitution paramétrée par la sous-clef K.

9.2 Divisibilité et degré de la composée de deux fonctions

Afin d'améliorer la borne triviale précédente, il faut étudier le degré de la composée de deux fonctions. Nous nous intéressons donc ici à une fonction composée $F' \circ F$ où F et F' sont deux applications de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$. Comme le degré d'une fonction à n variables dont le spectre de Walsh est 2^ℓ-divisible est majoré par $(n-\ell+1)$ [Car94b], tout résultat sur la divisibilité de $F' \circ F$ permet d'améliorer la borne triviale

$$\deg(F' \circ F) \leq \deg(F') \deg(F) \ .$$

Or, la i-ème fonction coordonnée de $F' \circ F$ s'écrit sous la forme

$$(F' \circ F)_i(x) = F'_i(F_1(x), \ldots, F_n(x))$$

où F'_i est la i-ème fonction coordonnée de F' et $F_1, \ldots, F_n$ désignent les fonctions coordonnées de F. On voit donc que le degré de $(F' \circ F)_i$ est inférieur ou égal au degré maximal du produit de $\deg(F'_i)$ coordonnées de F. Ce dernier peut alors être majoré grâce à la proposition suivante qui lie la transformée de Walsh de la somme de plusieurs fonctions booléennes et celle de leur produit.

Proposition 9.3 *[CV02, lemme 1] Soit $f_1, \ldots, f_k$ k fonctions booléennes à n variables, avec $k > 0$. Alors, on a*

$$\mathcal{F}\left(\sum_{i=1}^{k} f_i\right) = 2^{n-1}\left[(-1)^k + 1\right] + \sum_{I \subset \{1,\ldots,k\}} (-2)^{|I|-1} \mathcal{F}\left(\prod_{i \in I} f_i\right) \ .$$

De plus, pour tout α dans $\mathbf{F}_2^n \setminus \{0\}$,

$$\mathcal{F}\left(\sum_{i=1}^{k} f_i + \varphi_\alpha\right) = \sum_{I \subset \{1,\ldots,k\}} (-2)^{|I|-1} \mathcal{F}\left(\prod_{i \in I} f_i + \varphi_\alpha\right) \ .$$

On en déduit alors la relation suivante sur la divisibilité des coefficients de Walsh du produit de plusieurs fonctions.

Théorème 9.4 *[CV02, Th. 1] Soit $f_1, \ldots, f_k$ k fonctions booléennes à n variables, avec $k > 0$. Supposons que, pour tout sous-ensemble I de $\{1, \ldots, k\}$,*

$$\forall \alpha \in \mathbf{F}_2^n, \ \mathcal{F}\left(\sum_{i \in I} f_i + \varphi_\alpha\right) \equiv 0 \bmod 2^\ell \ .$$

Alors, pour tout $I \subset \{1, \ldots, k\}$ de taille au plus ℓ, on a

$$\forall \alpha \in \mathbf{F}_2^n, \ \mathcal{F}\left(\prod_{i \in I} f_i + \varphi_\alpha\right) \equiv 0 \bmod 2^{\ell+1-|I|} \ . \tag{9.1}$$

En particulier,

$$\deg\left(\prod_{i \in I} f_i\right) \leq n - \ell + |I| \ .$$

En conséquence, on obtient une nouvelle borne sur le degré de la composée $F' \circ F$ qui dépend de la divisibilité du spectre de Walsh de F.

Corollaire 9.5 *Soit F une fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ dont le spectre de Walsh est 2^ℓ-divisible. Alors, pour toute fonction F' de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$, on a*

$$\deg(F' \circ F) \leq n - \ell + \deg(F') \ .$$

En fait, dans le cas où F est une fonction puissance $x \mapsto x^s$ sur $\mathbf{F}_{2^n}$, cette borne peut se déduire directement du théorème de McEliece tel qu'il est énoncé à la proposition 8.9 page 98. En effet, si l'on exprime F' sous forme d'un polynôme de $\mathbf{F}_{2^n}[X]$, $F'(X) = \sum_{u=0}^{2^n-1} a_u X^u$, on a

$$F' \circ F(x) = \sum_{u=0}^{2^n-1} a_u X^{us \bmod (2^n-1)} \ .$$

Son degré vérifie donc

$$\begin{aligned} \deg(F' \circ F) &\leq \max_{u,\, a_u \neq 0} w_2(us \bmod (2^n-1)) \\ &\leq \max_{u,\, a_u \neq 0} w_2(u) + n - \ell \\ &\leq \deg(F') + n - \ell \ . \end{aligned}$$

Cette borne supérieure sur le degré de la composée est donc d'autant plus petite que la divisibilité du spectre de F est élevée. En particulier, quand F est une fonction AB, on a

$$\deg(F' \circ F) \leq \frac{n-1}{2} + \deg(F')$$

ce qui fournit pratiquement toujours un résultat pertinent quand $\deg F' = \deg F$ puisque le degré d'une fonction AB est au plus égal à $(n+1)/2$.

9.3 Application à MISTY1

Les résultats précédents mettent en lumière l'origine d'une attaque différentielle d'ordre 7 due à Tanaka, Hisamatsu et Kaneko [THK99], qui permet de cryptanalyser cinq tours d'une variante de MISTY1. L'algorithme par blocs MISTY1 proposé par Matsui en 1996 [Mat97] est un réseau de Feistel à huit tours opérant sur des blocs de 64 bits. Sa fonction interne FO de $\mathbf{F}_2^{32}$ dans $\mathbf{F}_2^{32}$ est elle-même définie par deux réseaux de Feistel à trois tours imbriqués selon le principe décrit à la figure 9.1. Le niveau le plus interne de MISTY1 utilise deux fonctions de substitution, notées S_7 et S_9, opérant respectivement sur des blocs de 7 et 9 bits. Ces deux fonctions sont des permutations puissances AB composées avec des permutations linéaires :

$$\begin{aligned} S_7 : &\quad x \mapsto L_7(x^{81}) \quad \text{sur } \mathbf{F}_{2^7} \\ S_9 : &\quad x \mapsto L_9(x^5) \quad \text{sur } \mathbf{F}_{2^9} \ . \end{aligned}$$

L'emploi de ces fonctions AB a permis à Matsui de démontrer qu'une version affaiblie de MISTY1 (sans les fonctions linéaires appliquées aux deux moitiés de l'entrée de chaque tour) résistait aux attaques différentielles et linéaires. L'attaque présentée par Tanaka, Hisamatsu et Kaneko sur cette même variante repose sur les propriétés de la dérivée par rapport à l'espace

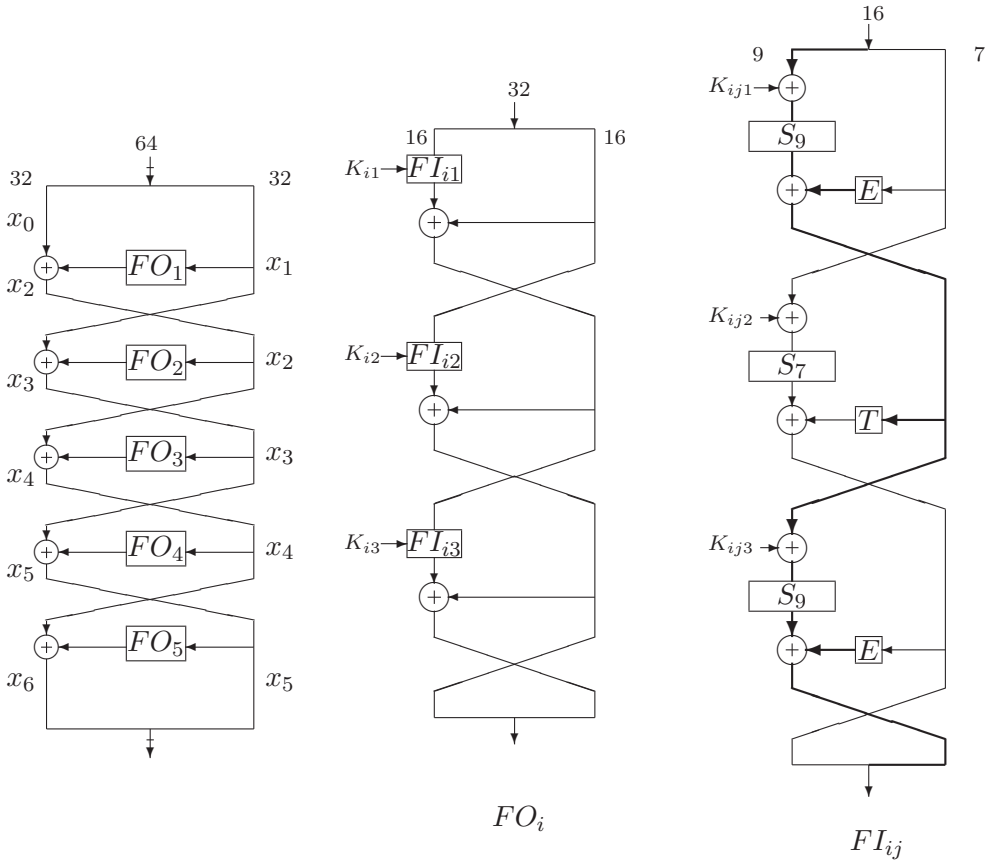

FIG. 9.1 – *Description de cinq tours de MISTY1 (sans les fonctions linéaires FL) et de sa fonction interne*

vectoriel $V \in \mathbf{F}_2^{64}$ composé des vecteurs dont la moitié droite et les 25 bits de poids fort de la moitié gauche s'annulent. Alors, la fonction G_K qui, à tout texte clair, associe les 7 bits de poids fort de x_4, la partie gauche de l'entrée du cinquième tour vérifie

$$D_V G_K(x) = \sum_{v \in V} G_K(x+v) = \text{ constante}, \ \forall x \in \mathbf{F}_2^{64} \ ,$$

où la constante ne dépend pas de la clef utilisée. Comme l'ont montré Babbage et Frisch [BF00], les termes de degré maximal (c'est-à-dire de degré 7) de la fonction G_K résultent d'une composition de la forme

$$S_7\left(S_7(x) + Q(x) + A_K(x)\right)$$

où Q est une fonction quadratique et A_K une fonction affine qui dépend de la clef. Les termes de degré 7 dépendant de la clef ne peuvent donc provenir que du produit de deux composantes de la fonction S_7 et d'une composante de la fonction affine. Tanaka, Hisamatsu et Kaneko avaient alors constaté que tout produit de deux composantes de la fonction $x \mapsto x^{81}$ était de degré au plus 5 et Babbage et Frisch avaient ensuite remarqué, par recherche exhaustive, que

cette propriété était valable pour toutes les fonctions puissances AB de degré 3 sur $\mathbf{F}_{2^7}$. En réalité, cette propriété provient naturellement de la divisibilité élevée du spectre de Walsh des fonctions AB et peut être déduite du corollaire 9.5 précédent.

Comme la propriété exploitée dans cette attaque était plus générale et résultait en fait de l'utilisation de fonctions presque courbes, nous avons pu montrer qu'une attaque différentielle d'ordre supérieur similaire pouvait également être menée sur la plupart des généralisations de MISTY1 à une taille de blocs $16m$ employant des fonctions de substitution presque courbes [CV02]. C'est donc paradoxalement la propriété qui permet de démontrer que le chiffrement résiste aux attaques différentielles et linéaires qui est à l'origine de cette cryptanalyse différentielle d'ordre supérieur.

9.4 Permutations puissances de faible divisibilité

Constatant qu'une haute divisibilité du spectre de Walsh pouvait introduire une faiblesse et que toutes les permutations connues de non-linéarité maximale à l'exception de la fonction inverse présentaient cet inconvénient — les fonctions AB pour un nombre n impair de variables mais aussi, pour n pair, les permutations puissances ayant la meilleure non-linéarité connue (voir tab. 8.2 page 103) — nous nous sommes naturellement intéressées aux permutations puissances dont le spectre de Walsh présente au contraire une divisibilité très faible.

D'après le théorème de Katz (proposition 6.1 page 68), la divisibilité du spectre de Walsh d'une permutation F est supérieure ou égale à

$$2^{\lceil \frac{n-1}{\deg F} \rceil + 1} .$$

Par conséquent, la plus petite divisibilité possible pour le spectre de Walsh d'une permutation est 4. Dans le cas des permutations puissances, cette divisibilité minimale n'est atteignable que par les exposants de poids $(n-1)$, c'est-à-dire par les éléments de la classe cyclotomique de $(2^{n-1}-1)$, ou autrement dit par les fonctions équivalentes à la fonction inverse [Hel76]. Mais, nous pouvons également caractériser les exposants qui correspondent à un spectre de Walsh exactement divisible par 8, et donner des conditions nécessaires à une divisibilité exacte de 16.

Proposition 9.6 *[CCD00b, Prop. 5.3 et 5.4] Soit n et s deux entiers positifs tels que* $\mathrm{pgcd}(2^n-1,s)=1$. *Soit s^{-1} l'unique entier de $\{0,\ldots,2^n-2\}$ tel que $s^{-1}s \equiv 1 \bmod (2^n-1)$.*

- *Le spectre de Walsh de $x \mapsto x^s$ est exactement 4-divisible si et seulement si $w_2(s) = n-1$.*
- *Le spectre de Walsh de $x \mapsto x^s$ est exactement 8-divisible si et seulement si $w_2(s) = n-2$ ou $w_2(s^{-1}) = n-2$.*
- *Si le spectre de Walsh de $x \mapsto x^s$ est exactement 16-divisible, alors $w_2(s^{-1}) = n-3$ ou*

$$\left\lceil \frac{n-2}{2} \right\rceil \leq w_2(s) \leq n-3 .$$

On voit ici que les permutations puissances dont le spectre de Walsh présente une faible divisibilité correspondent à des exposants de poids très élevés. Ces résultats semblent alors plaider en faveur de l'utilisation de la fonction inverse ou, si l'on souhaite disposer d'alternatives, de fonctions au sein de cette famille possédant de bonnes propriétés au regard des attaques différentielles et linéaires.

Chapitre 10

Autres critères de conception

En plus de la résistance aux attaques différentielles, linéaires et différentielles d'ordre supérieur, un certain nombre d'autres critères interviennent dans le choix de la fonction de substitution d'un chiffrement par blocs. Nous avons déjà mentionné le degré multivarié, mais aussi le degré univarié, nécessaire pour se prémunir contre l'attaque par interpolation [JK97]. On peut aussi prendre en compte le degré minimal des relations liant les entrées et les sorties de la fonction, qui détermine la complexité des attaques algébriques. Par ailleurs, un certain nombre d'autres critères qui ne reposent pas sur une attaque précise mais plutôt sur l'intuition du cryptographe apparaissent dans la littérature. Nous en détaillons ici quelques-uns.

10.1 Immunité algébrique

L'existence de relations multivariées de degré faible entre les entrées et les sorties de la fonction interne (et donc de la fonction de substitution) dans un chiffrement par blocs peut également être exploitée dans une attaque algébrique [CP02] reposant sur le même principe que celles étudiées au chapitre 5. Toutefois, le système dérivé de ces relations possède un nombre très important de variables correspondant aux bits de chacune des sous-clefs et aux entrées des différents tours. Sa résolution est donc beaucoup plus coûteuse que dans le cas des attaques sur les chiffrements à flot, et semble actuellement tout à fait hors de portée même si sa complexité reste difficile à évaluer [CMR05, Ars05]. Contrairement à la situation des chiffrements à flot à base de LFSRs, le degré minimal des relations entre les entrées et les sorties de la fonction de substitution ne semble donc pas être un paramètre qu'il faille optimiser lors de la conception d'un algorithme par blocs. Il peut cependant sembler pertinent de vouloir se prémunir de l'existence de relations quadratiques. Par le même raisonnement qu'au chapitre 5, on voit qu'une fonction vectorielle F de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^m$ admet des relations de degré inférieur ou égal à d entre ses entrées et ses sorties dès que

$$\sum_{i=0}^{d} \binom{n+m}{i} > 2^n \ . \tag{10.1}$$

Ainsi, toute fonction de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ admet des relations quadratiques si $n \leq 6$. De même, toute fonction de substitution de $\mathbf{F}_2^8$ dans $\mathbf{F}_2^8$ admet des relations cubiques.

Le problème est donc de trouver des fonctions de substitution qui ne possèdent pas de relation quadratique, ou au moins de minimiser le nombre de ces relations. Différents tra-

vaux [CL04, CDG05] ont montré que toutes les permutations puissances AB connues admettaient des relations quadratiques entre leurs entrées et leurs sorties. De plus, Courtois, Debraize et Garrido prouvent dans [CDG05] que, dans le cas de la fonction inverse, c'est-à-dire pour $x \mapsto x^{2^{n-1}-1}$ sur $\mathbf{F}_{2^n}$, le nombre de relations quadratiques linéairement indépendantes est égal à $5n-1$. Dans le cas, relativement peu intéressant du point de vue du chiffrement par blocs, des exposants quadratiques, le nombre total de relations bilinéaires est étudié dans [DDG+06] car ces relations interviennent dans la cryptanalyse du système à clef publique de Matsumoto-Imai et ses variantes [Pat95]. Par une méthode similaire à celle utilisée dans [CDG05] pour la fonction inverse — mais faisant intervenir un problème combinatoire plus complexe —, nous pouvons déterminer la dimension de l'espace des relations quadratiques entre les entrées et les sorties de chacune des fonctions puissances AB connues. On voit par exemple, que pour $n \geq 8$, les fonctions correspondant aux exposants de Kasami $s = 2^{2i} - 2^i + 1$ sur $\mathbf{F}_{2^n}$ se comportent mieux que la fonction inverse puisqu'elles n'admettent que $2n$ relations quadratiques linéairement indépendantes pour $1 < i \leq \lfloor n/2 \rfloor$.

Par ailleurs, on constate également que toutes les permutations puissances sur $\mathbf{F}_{2^8}$ possèdent des relations quadratiques entre leurs entrées et leur sorties — la seule fonction puissance, à équivalence près, qui n'en possède pas est $x \mapsto x^{27}$, qui n'est pas une permutation. Par contre, on trouve des permutations puissances sur $\mathbf{F}_{2^n}$ qui n'en admettent pas pour $n \geq 9$. Par exemple, la fonction $x \mapsto x^{53}$ possède cette propriété pour $10 \leq n \leq 16$.

10.2 Caractéristiques des permutations puissances

Le tableau 10.1 donne donc la valeur des différents paramètres que nous avons étudiés jusqu'ici, et la totalité du spectre de Walsh de l'ensemble des permutations puissances sur $\mathbf{F}_{2^8}$. Le spectre de Walsh est ici exprimé par l'ensemble des valeurs prises par $\mathcal{F}(S_\lambda + \varphi_\mu)$,

s	$w_2(s)$	s^{-1}	$w_2(s^{-1})$	$\delta(S)$	δ_1 [a]	rel. quad. [b]	$\mathcal{L}(S)$	spectre de Walsh	
7	3	37	3	6	4	24	64	-32 [16], -16 [52], 0 [105], 16 [68], 32 [14], 64 [1]	
11	3	29	4	10	4	24	64	-64 [1], -32 [8], -16 [64], 0 [101], 16 [68], 32 [10], 48 [4]	
13	3	59	5	12	4	16	64	-32 [18], -16 [48], 0 [101], 16 [84], 48 [4], 64 [1]	$\mathcal{K}(2)$
19	3	47	5	16	16	24	48	-16 [88], 0 [88], 16 [64], 32 [8], 48 [8]	Niho
23	4	61	5	16	16	20	64	-16 [88], 0 [90], 16 [56], 32 [20], 64 [2]	Niho
31	5	91	5	16	16	36	32	-16 [80], 0 [120], 16 [16], 32 [40]	$\mathcal{K}(4)$
43	4	43	4	30	28	28	96	-32 [8], -16 [60], 0 [109], 16 [76], 64 [1], 96 [2]	
53	4	53	4	16	16	18	64	-16 [96], 0 [60], 16 [96], 64 [4]	Niho
127	7	127	7	4	4	39	32	-28 [8], -24 [16], -20 [8], -16 [18], -12 [24], -8 [16], -4 [32], 0 [17], 4 [16], 8 [20], 12 [16], 16 [16], 20 [16], 24 [20], 28 [8], 32 [5]	inv.

TAB. 10.1 – *Propriétés des permutations puissances $S : x \mapsto x^s$ sur $\mathbf{F}_{2^8}$*

[a] Nombre de solutions de $(x+1)^s + x^s = 1$.
[b] Dimension de l'espace des relations quadratiques.

chacune d'elles étant suivie entre crochets du nombre de μ tels que $\mathcal{F}(S_\lambda + \varphi_\mu) = i$ pour chacune des valeurs de $\lambda \neq 0$. On distingue en particulier dans cette table les exposants dits de Niho[a], c'est-à-dire les valeurs de s telles que $s \equiv 2^i \bmod (2^{\frac{n}{2}} - 1)$. Nous avons montré dans [CCD00b] que leur spectre de Walsh est $2^{\frac{n}{2}+1}$-divisible et P. Charpin a mené une étude plus poussée [Cha04] qui prouve notamment que la valeur de $\mathcal{L}(S)$ correspondant est supérieure ou égale à $2^{\frac{n}{2}+2}$.

Les tables 10.2 et 10.3 donnent de la même manière — sous une présentation légèrement différente pour des raisons de place — le spectre de Walsh de toutes les permutations puissances S sur $\mathbf{F}_{2^9}$ et $\mathbf{F}_{2^{10}}$, les valeurs de $\delta(S)$ et $\mathcal{L}(S)$ associées, et précisent également s'il existe des relations quadratiques entre les entrées et sorties de la fonction.

TAB. 10.2 – *Propriétés des permutations puissances* $S : x \mapsto x^s$ *sur* $\mathbf{F}_{2^9}$

s	$wt(s)$	s^{-1}	$wt(s^{-1})$	spectre de Walsh	$\mathcal{L}(S)$	δ	rel. quad.
3	2	171	5	-32 [120], 0 [256], 32 [136]	32	2	oui
5	2	103	5	-32 [120], 0 [256], 32 [136]	32	2	oui
9	2	57	4	-64 [28], 0 [448], 64 [36]	64	2	oui
11	3	93	5	-64 [1], -32 [108], 0 [286], 32 [108], 64 [9]	64	8	oui
13	3	59	5	-32 [120], 0 [256], 32 [136]	32	2	oui
15	4	239	7	-48 [3], -40 [27], -32 [36], -24 [18], -16 [63], -8 [54], 0 [73], 8 [100], 16 [45], 24 [39], 32 [18], 40 [18], 48 [9], 64 [9]	64	8	oui
17	2	31	5	-32 [120], 0 [256], 32 [136]	32	2	oui
19	3	27	4	-32 [120], 0 [256], 32 [136]	32	2	oui
23	4	25	3	-64 [1], -32 [108], 0 [286], 32 [108], 64 [9]	64	8	oui
29	4	53	4	-48 [12], -32 [54], -16 [117], 0 [148], 16 [99], 32 [54], 48 [27], 80 [1]	80	8	oui
37	3	183	6	-32 [54], -16 [135], 0 [163], 16 [99], 32 [27], 48 [21], 64 [9], 80 [1], 96 [3]	96	26	oui
39	4	95	6	-64 [1], -48 [3], -32 [72], -16 [108], 0 [145], 16 [99], 32 [54], 48 [30]	64	8	oui
41	3	187	6	-112 [1], -48 [9], -32 [63], -16 [99], 0 [139], 16 [138], 32 [45], 48 [9], 64 [9]	112	8	oui
43	4	107	5	-64 [1], -32 [108], 0 [286], 32 [108], 64 [9]	64	8	oui
45	4	125	6	-48 [12], -32 [81], -16 [63], 0 [154], 16 [138], 32 [37], 48 [27]	48	4	oui
47	5	87	5	-32 [120], 0 [256], 32 [136]	32	2	oui
51	4	191	7	-72 [3], -40 [9], -32 [45], -24 [27], -16 [54], -8 [81], 0 [82], 8 [82], 16 [36], 24 [27], 32 [9], 40 [27], 48 [30]	72	8	oui
55	5	223	7	-40 [18], -32 [27], -24 [36], -16 [63], -8 [72], 0 [100], 8 [72], 16 [36], 24 [27], 32 [9], 40 [27], 48 [21], 88 [3], 104 [1]	104	20	oui
61	5	111	6	-112 [1], -48 [9], -32 [54], -16 [117], 0 [148], 16 [102], 32 [54], 48 [27]	112	6	oui
75	4	75	4	-32 [54], -16 [162], 0 [109], 16 [108], 32 [55], 48 [18], 64 [3], 96 [3]	96	20	oui
79	5	123	6	-48 [12], -32 [54], -16 [117], 0 [148], 16 [99], 32 [54], 48 [27], 80 [1]	80	8	oui
83	4	117	5	-112 [1], -48 [3], -32 [63], -16 [135], 0 [109], 16 [108], 32 [81], 48 [9], 64 [3]	112	6	non

a. Ces exposants sont différents de ceux qui correspondent à des fonctions AB pour un nombre impair de variables, mais introduits par le même auteur dans [Nih72].

s	$wt(s)$	s^{-1}	$wt(s^{-1})$	spectre de Walsh	$\mathcal{L}(S)$	δ	rel. quad.
85	4	127	7	-48 [18], -32 [27], -24 [48], -16 [45], -8 [90], 0 [58], 8 [64], 16 [72], 24 [36], 32 [27], 48 [9], 56 [18]	56	8	oui
109	5	109	5	-64 [1], -32 [108], 0 [286], 32 [108], 64 [9]	64	8	oui
255	8	255	8	-44 [3], -40 [9], -36 [18], -32 [36], -28 [18], -24 [9], -20 [27], -16 [27], -12 [27], -8 [36], -4 [19], 0 [19], 4 [45], 8 [27], 12 [18], 16 [45], 20 [18], 24 [21], 28 [45], 32 [9], 36 [9], 40 [18], 44 [9]	44	2	oui

TAB. 10.3 – *Propriétés des permutations puissances* $S : x \mapsto x^s$ *sur* $\mathbf{F}_{2^{10}}$

s	$wt(s)$	s^{-1}	$wt(s^{-1})$	spectre de Walsh	$\mathcal{L}(S)$	δ	rel. quad.
5	2	205	5	-64 [120], 0 [768], 64 [136]	64	4	oui
7	3	439	7	-96 [1], -80 [10], -64 [20], -48 [70], -32 [135], -16 [160], 0 [196], 16 [200], 32 [115], 48 [70], 64 [40], 80 [2], 96 [5]	96	6	oui
13	3	79	5	-64 [120], 0 [768], 64 [136]	64	4	oui
17	2	181	5	-64 [120], 0 [768], 64 [136]	64	4	oui
19	3	175	6	-64 [70], -32 [210], 0 [428], 32 [260], 64 [46], 96 [10]	96	6	oui
23	4	89	4	-80 [10], -64 [40], -48 [40], -32 [100], -16 [246], 0 [171], 16 [175], 32 [130], 48 [50], 64 [55], 80 [7]	80	6	oui
25	3	41	3	-64 [120], 0 [768], 64 [136]	64	8	oui
29	4	247	7	-80 [10], -64 [10], -48 [70], -32 [180], -16 [155], 0 [173], 16 [135], 32 [150], 48 [125], 64 [15], 144 [1]	144	4	oui
35	3	95	6	-32 [360], 0 [368], 32 [240], 64 [16], 96 [40]	96	34	oui
37	3	83	4	-160 [2], -64 [40], -32 [260], 0 [416], 32 [240], 64 [56], 96 [10]	160	8	oui
43	4	119	6	-96 [10], -64 [20], -32 [300], 0 [368], 32 [250], 64 [76]	96	6	oui
47	5	109	5	-32 [350], 0 [413], 32 [160], 64 [86], 96 [10], 128 [5]	128	34	oui
49	3	107	5	-64 [120], 0 [768], 64 [136]	64	8	oui
53	4	251	7	-96 [1], -80 [2], -64 [5], -48 [50], -32 [170], -16 [220], 0 [186], 16 [180], 32 [110], 48 [50], 64 [25], 80 [10], 96 [10], 160 [5]	160	34	non
59	5	191	7	-80 [10], -64 [30], -48 [50], -32 [175], -16 [110], 0 [183], 16 [240], 32 [135], 48 [60], 64 [11], 80 [10], 96 [10]	96	6	oui
61	5	151	5	-96 [10], -48 [100], -32 [100], -16 [206], 0 [231], 16 [115], 32 [120], 48 [100], 64 [35], 80 [7]	96	6	oui
71	4	245	6	-96 [1], -64 [40], -48 [70], -32 [125], -16 [140], 0 [228], 16 [190], 32 [105], 48 [80], 64 [40], 96 [5]	96	6	oui
73	3	127	7	-96 [1], -80 [2], -64 [30], -48 [90], -32 [115], -16 [140], 0 [196], 16 [220], 32 [135], 48 [50], 64 [30], 80 [10], 96 [5]	96	6	oui
85	4	85	4	-80 [22], -64 [30], -48 [70], -16 [281], 0 [196], 16 [280], 48 [100], 64 [30], 80 [10], 112 [5]	112	10	oui
91	5	215	6	-80 [20], -64 [10], -48 [60], -32 [130], -16 [191], 0 [196], 16 [180], 32 [115], 48 [75], 64 [40], 80 [2], 96 [5]	96	6	oui

s	$wt(s)$	s^{-1}	$wt(s^{-1})$	spectre de Walsh	$\mathcal{L}(S)$	δ	rel. quad.
101	4	157	5	-32 [350], 0 [408], 32 [180], 64 [56], 96 [30]	96	34	oui
103	5	149	4	-96 [1], -80 [10], -64 [30], -48 [30], -32 [160], -16 [200], 0 [188], 16 [150], 32 [115], 48 [90], 64 [50]	96	4	oui
115	5	347	6	-96 [10], -64 [20], -32 [300], 0 [368], 32 [250], 64 [76]	96	6	oui
125	6	221	6	-32 [350], 0 [413], 32 [160], 64 [86], 96 [10], 128 [5]	128	34	oui
167	5	239	7	-96 [1], -80 [12], -64 [40], -48 [60], -32 [70], -16 [190], 0 [236], 16 [180], 32 [145], 48 [60], 64 [20], 112 [10]	112	6	oui
173	5	479	8	-72 [10], -64 [30], -56 [20], -48 [20], -40 [62], -32 [40], -24 [50], -16 [125], -8 [120], 0 [91], 8 [90], 16 [65], 24 [70], 32 [70], 40 [70], 48 [45], 56 [10], 64 [25], 72 [10], 144 [1]	144	6	oui
179	5	383	8	-72 [20], -64 [20], -56 [20], -48 [30], -40 [62], -32 [35], -24 [60], -16 [80], -8 [70], 0 [146], 8 [110], 16 [95], 24 [90], 32 [55], 40 [40], 48 [45], 56 [30], 72 [10], 112 [5], 144 [1]	144	10	oui
223	7	367	7	-80 [10], -64 [30], -48 [40], -32 [175], -16 [140], 0 [193], 16 [210], 32 [105], 48 [70], 64 [41], 80 [10]	80	4	oui
235	6	379	7	-96 [1], -80 [10], -64 [20], -48 [70], -32 [140], -16 [150], 0 [186], 16 [230], 32 [115], 48 [40], 64 [50], 80 [12]	96	6	oui
343	6	343	6	-32 [300], 0 [461], 32 [220], 64 [40], 320 [2], 384 [1]	384	124	oui
511	9	511	9	-60 [12], -56 [11], -52 [30], -48 [20], -44 [30], -40 [40], -36 [40], -32 [20], -28 [40], -24 [60], -20 [20], -16 [45], -12 [40], -8 [20], -4 [40], 0 [61], 4 [30], 8 [40], 12 [50], 16 [45], 20 [50], 24 [20], 28 [20], 32 [40], 36 [50], 40 [35], 44 [20], 48 [20], 52 [20], 56 [30], 60 [20], 64 [5]	64	4	oui

10.3 Utilisation d'une fonction puissance

Il est légitime, au vu des résultats précédents, de se demander si le choix d'une fonction puissance comme fonction de substitution n'est pas source de faiblesse. En effet, on peut être amené à penser que certaines propriétés spécifiques aux permutations puissances influencent la valeur du degré minimal des relations entre les entrées et les sorties d'une permutation. Ainsi, nous avons vu que toutes les permutations puissances de $\mathbf{F}_{2^8}$ admettaient des relations quadratiques alors qu'il ne s'agit pas d'une conséquence de l'argument combinatoire donné par la formule (10.1). Il serait d'ailleurs intéressant d'étudier précisemment les permutations de $\mathbf{F}_{2^8}$ qui n'admettent pas de relations quadratiques et d'analyser leur structure[b].

En particulier, on retrouve ici le problème ouvert posé au chapitre 5 page 65 dans un tout autre contexte : est-ce que le fait que toutes les composantes de la permutation considérée soient affinement équivalentes (et par exemple qu'elles aient toutes le même spectre de Walsh) impose des contraintes particulières sur l'immunité algébrique de cette permutation ?

Le choix de fonctions puissances comme fonction de substitution est naturellement dicté par les contraintes de la réalisation matérielle, mais elles disparaissent complètement si l'on vise les réalisations logicielles. On pouvait jusqu'à très récemment penser que les fonctions puissances étaient des objets remarquables au regard de la résistance aux attaques différentielles et linéaires, mais la découverte récente de fonctions APN et AB échappant à cette construction [BCP06, EKP06, BCFL05] semble plutôt indiquer que cette situation était uniquement justifiée par le fait que les outils dont on dispose pour démontrer le caractère optimal d'une

b. Notons que les techniques développées dans [BCP06] pour trouver des fonctions APN et AB qui ne soient pas des fonctions puissances ne permettent pas de répondre à cette question puisqu'elles construisent, à partir d'une fonction puissance, des fonctions qui possèdent la même immunité algébrique.

fonction ne sont opérationnels que dans le cas des fonctions puissances, la faible proportion de ces objets exceptionnels rendant toute exploration systématique hors de portée.

Enfin, la question de savoir si l'utilisation d'une fonction de substitution qui soit une fonction puissance introduit d'autres faiblesses qui pourraient être exploitées directement dans une attaque demeure ouverte. Gong et Golomb [GG99b] ont même suggéré intuitivement que la distance de la fonction de substitution à l'ensemble des fonctions puissances devait au contraire être la plus élevée possible — ils ont d'ailleurs montré que les boîtes-S du DES possèdent cette propriété.

10.4 Construction par concaténation de fonctions plus petites

Une autre structure très forte présente dans la plupart des algorithmes par blocs provient du fait que la fonction de substitution résulte en fait de la concaténation de plusieurs fonctions plus petites. Ce type de construction semble incontournable si la fonction de substitution est destinée à être mise en table. Le stockage d'une fonction à 16 entrées et 16 sorties nécessite au minimum 128 Koctets, ce qui est encore trop élevé pour certains processeurs super-scalaires, mais sera sans doute envisageable [c] dans un avenir relativement proche [d].

Toutefois, il est important de noter que l'emploi d'une fonction construite par concaténation de fonctions ayant moins d'entrées dégrade notablement la résistance aux attaques classiques, ce qui doit donc être compensé par le nombre de tours effectués par l'algorithme. En effet, si la fonction S de $\mathbf{F}_2^n$ dans $\mathbf{F}_2^n$ est composée de la concaténation de n/k copies de la même fonction S_0 de $\mathbf{F}_2^k$ dans $\mathbf{F}_2^k$, ses coefficients de Walsh sont donnés par

$$\mathcal{F}(S_\lambda + \varphi_\mu) = \prod_{i=1}^{\frac{n}{k}} \mathcal{F}((S_0)_{\lambda_i} + \varphi_{\mu_i})$$

où $\lambda = (\lambda_1, \ldots, \lambda_{n/k})$ et $\mu = (\mu_1, \ldots, \mu_{n/k})$. En particulier, on a

$$\mathcal{L}(S) = 2^{n-k}\mathcal{L}(S_0) \ .$$

De même le spectre de Walsh de S est exactement 2^ℓ-divisible avec

$$\ell = \frac{n\ell_0}{k}$$

où 2^{ℓ_0} est la divisibilité exacte du spectre de Walsh de S_0. Enfin, et peut-être est-ce l'inconvénient majeur de la construction,

$$\deg S \leq k - 1 \ .$$

Ainsi, si l'on considère les paramètres de l'AES, c'est-à-dire une taille de blocs de 128 bits et une fonction de substitution à 8 entrées et 8 sorties, on a

$$\mathcal{L}(S) \geq 2^{n-\frac{k}{2}+1} \geq 2^{125}, \quad \deg S \leq 7$$

c. À titre d'exemple les processeurs Intel Core Duo disposent de 2 Moctets de cache L2.

d. même si la question des attaques par canaux cachés exploitant l'analyse du comportement du cache risque de devenir cruciale car on reste au-delà de la taille du cache de données L1.

et un spectre de Walsh au moins 2^{32}-divisible[e], alors que la fonction inverse sur $\mathbf{F}_{2^{128}}$ vérifie

$$\mathcal{L}(S) = 2^{\frac{n}{2}+1} = 2^{65}, \quad \deg S \leq 127$$

et son spectre de Walsh est exactement 4-divisible. On voit clairement que la construction par concaténation de fonctions dépendant de très peu de variables produit en fait des fonctions relativement mauvaises vis-à-vis des attaques classiques. De même, si l'on s'intéresse aux attaques algébriques, on peut se demander si l'existence de relations quadratiques creuses entre les entrées et les sorties de la fonction de substitution — celles qui ne font intervenir qu'un petit nombre de boîtes — permet d'accélérer la résolution du système algébrique sous-jacent.

e. Ces paramètres optimaux sont d'ailleurs ceux qui sont atteints par la fonction de substitution de l'AES.

Conclusion et perspectives

Après le règne absolu des méthodes de conception empiriques, les années 90 ont vu l'émergence d'un début de théorie de la cryptographie symétrique. La formalisation de la cryptanalyse différentielle et de la cryptanalyse linéaire a notamment conduit à la définition de critères mathématiques qui doivent guider le concepteur : toute nouvelle proposition sérieuse de chiffrement par blocs doit désormais apporter la preuve qu'elle résiste à ces attaques. Toutefois, le cryptographe doit maintenant faire face à un insurmontable dilemme : son algorithme doit reposer sur des objets qui, pour un coût de mise en œuvre donné, lui garantissent la meilleure résistance possible aux cryptanalyses connues ; mais ces objets, par leur caractère exceptionnel, possèdent nécessairement des propriétés structurelles très fortes qui ouvrent la porte à de nouvelles attaques.

Nos travaux ont montré qu'il était désormais indispensable de rechercher des objets « sous-optimaux », au sens où ils répondent de manière relativement satisfaisante aux critères de sécurité mais ne présentent pas les structures particulières propres aux objets optimaux. Le cryptographe est donc paradoxalement confronté à l'étude et à la réalisation d'objets dont on souhaite qu'ils présentent le moins de structure possible.

Animée par la conviction qu'un système dont la conception répondrait uniquement aux préoccupations de sécurité ne peut conduire à une implémentation simple, j'ai donc été tout naturellement amenée à concentrer mes recherches sur des familles d'objets dont la structure est appropriée à une mise en œuvre efficace, dans l'objectif d'y trouver des éléments qui pourraient être à la base des composantes recherchées. L'idée sous-jacente est par exemple que, si l'on peut légitimement craindre que l'existence de relations quadratiques entre les entrées et les sorties de la boîte-S de l'AES puisse, à terme, être exploitée dans une attaque, remplacer cette fonction par une permutation choisie aléatoirement serait une hérésie du point de vue des performances.

À la lumière des travaux présentés dans ce document, les questions et voies de recherche suivantes demandent à être explorées.

Analyser les faiblesses introduites par les contraintes de sécurité et de mise en œuvre

Nos travaux sur les fonctions qui garantissent de manière optimale la résistance aux attaques classiques ont mis en lumière un certain nombre de propriétés algébriques fortes caractéristiques de ces objets. Si nous avons montré qu'une divisibilité élevée du spectre de Walsh, propre aux fonctions presque courbes, pouvait être exploitée dans les attaques différentielles d'ordre supérieur, de nombreux problèmes restent ouverts concernant la possibilité de tirer parti de ces structures dans une attaque. Il serait par exemple intéressant de déterminer si l'on peut accélérer la résolution du système intervenant dans les attaques algébriques (rapides)

quand la fonction de filtrage utilisée est une fonction symétrique, et si la simplification apportée par les fonctions symétriques est du même ordre de grandeur que celle de la réalisation matérielle de la fonction de filtrage. On peut aussi se demander comment la propriété de normalité, propre à la plupart des fonctions utilisées, c'est-à-dire le fait que ces fonctions soient constantes sur un espace de dimension $n/2$, influence la sécurité des systèmes qui les utilisent. Enfin, une question essentielle est bien sûr liée aux faiblesses potentielles introduites par l'emploi de fonctions puissances comme fonctions de substitution d'un chiffrement par blocs. Une piste restée sans suite a été initiée par Golomb et Gong [GG99b] à travers l'identification d'un réseau de Feistel à un registre à décalage à rétroaction non linéaire, mais l'intervention du cadencement de clef semble difficile à intégrer dans ce modèle. Cependant, ce problème mérite clairement que l'on s'y intéresse.

Un autre problème intéressant, lié à des critères que j'ai qualifiés d'« intuitifs », concerne la relation entre les propriétés cryptographiques d'une fonction de substitution et celles de son inverse. En effet, parmi les permutations puissances possédant de bonnes propriétés cryptographiques, on écarte habituellement toutes celles de degré multivarié faible — c'est-à-dire tous les exposants de petit poids de Hamming — ainsi que celles dont l'inverse est de degré faible. Si cette précaution est évidemment motivée dans le cas des réseaux de substitution-permutation par l'existence d'attaques différentielles d'ordre supérieur sur la fonction de chiffrement ou de déchiffrement, il en va autrement des réseaux de Feistel pour lesquels le déchiffrement ne fait jamais intervenir l'inverse de la fonction de substitution (qui n'est d'ailleurs pas nécessairement une permutation). On peut donc par exemple se demander si l'emploi de l'inverse d'une permutation puissance quadratique peut être envisagé dans ce cas, ou si, au contraire, il est possible de concevoir une attaque générique sur les réseaux de Feistel qui exploite certaines propriétés de l'inverse de la fonction de substitution.

Rechercher des fonctions ayant une réalisation matérielle peu coûteuse

L'étude que nous avons menée sur les fonctions puissances et les fonctions symétriques nous amène à penser qu'il est parfois nécessaire de relâcher les contraintes visiblement trop restrictives que nous avons imposées à travers la définition de ces familles de fonctions, et qu'il faut explorer les propriétés d'autres familles de fonctions un peu plus larges, mais qui conservent une faible complexité de mise en œuvre. Parmi les approches qui me semblent naturelles à ce stade figure l'étude des fonctions booléennes dites symétriques par rotation, c'est-à-dire invariantes uniquement par permutation circulaire des entrées. Si ces fonctions semblent parfois présenter de bonnes propriétés cryptographiques [Fon98, SMC04], la possibilité d'en réaliser une mise en œuvre matérielle efficace reste ouverte. Une autre famille de fonctions booléennes dignes d'intérêt est celle constituée des composantes de certains binômes ou trinômes, famille au sein de laquelle se trouvent des fonctions courbes comme nous l'avons montré dans [DLC+06]. Se pose alors la question du choix des exposants afin de minimiser la taille du circuit correspondant et d'obtenir les propriétés cryptographiques souhaitées. La recherche de bonnes fonctions de substitution sous la forme de polynômes de permutation creux est naturellement à explorer. Mais, de manière plus immédiate, la possibilité de construire des fonctions de substitution adéquates à partir de fonctions puissances doit être étudiée plus précisément car de nombreux problèmes restent ouverts, comme l'existence de permutations puissances APN dépendant d'un nombre pair de variables, la valeur de la meilleure non-linéarité qu'elles peuvent atteindre...

Bibliographie

[AB05] F. Arnault et T.P. Berger. « F-FCSR: design of a new class of stream ciphers ». Dans *Fast Software Ecryption - FSE 2003*, volume 3557 de *Lecture Notes in Computer Science*, pages 83–97. Springer-Verlag, 2005.

[ABMV93] G.B. Agnew, T. Beth, R.C. Mullin et S.A. Vanstone. « Arithmetic operations in $GF(2^m)$ ». *Journal of Cryptology*, 6(1):3–13, 1993.

[ACGS88] W. Alexi, B. Chor, O. Goldreich et C.P. Shnorr. « RSA and Rabin functions: certain parts are as hard as the whole ». *SIAM Journal on Computing*, 17, 1988.

[AFI+04] G. Ars, J.-C. Faugère, H. Imai, M. Kawazoe et M. Sugita. « Comparison between XL and Gröbner basis algorithms ». Dans *Advances in Cryptology - ASIACRYPT 2004*, volume 3329 de *Lecture Notes in Computer Science*, pages 338–353. Springer-Verlag, 2004.

[AK03] F. Armknecht et M. Krause. « Algebraic attacks on combiners with memory ». Dans *Advances in Cryptology - CRYPTO 2003*, volume 2729 de *Lecture Notes in Computer Science*, pages 162–176. Springer-Verlag, 2003.

[And95] R. J. Anderson. « Searching for the optimum correlation attack ». Dans *Fast Software Encryption 1994*, volume 1008 de *Lecture Notes in Computer Science*, pages 137–143. Springer-Verlag, 1995.

[AR99] Y. Aumann et M.O. Rabin. « Information Theoretically Secure Communication in the Limited Storage Space Model ». Dans *Advances in Cryptology - CRYPTO'99*, volume 1666 de *Lecture Notes in Computer Science*, pages 65–79. Springer-Verlag, 1999.

[Arm02] F. Armknecht. « A linearization attack on the Bluetooth keystream generator ». `http://th.informatik.uni-mannheim.de/people/armknecht/E0.ps`, 2002.

[Arm04] F. Armknecht. « Improving fast algebraic attacks ». Dans *Fast Software Encryption - FSE 2004*, volume 3017 de *Lecture Notes in Computer Science*, pages 65–82. Springer-Verlag, 2004.

[Arm05] F. Armknecht. « Algebraic attacks and annihilators ». Dans *Proceedings of the Western European Workshop on Research in Cryptology (WEWoRC 2005)*, Lecture Notes in Informatics. Springer-Verlag, 2005. To appear.

[Ars05] G. Ars. « *Application des bases de Gröbner à la cryptographie* ». Thèse de doctorat, Université Rennes 1, 2005.

[Bab95] S. Babbage. « A space/time trade-off in exhaustive search attacks on stream ciphers ». Dans *European Convention on Security and Detection*, numéro 408. IEEE Conference Publication, 1995.

[BAK98] E. Biham, R. Anderson et L. Knudsen. « SERPENT: A New Block Cipher Proposal ». Dans *Fast Software Encryption - FSE'98*, volume 1372 de *Lecture Notes in Computer Science*, pages 222–238. Springer-Verlag, 1998.

[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 ». Dans *Proceedings of SKEW - Symmetric Key Encryption Workshop, Network of Excellence in Cryptology ECRYPT*, Aarhus, Danemark, mai 2005.

[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 software-oriented stream cipher ». Soumission au projet eSTREAM [ECR05], 2005. `http://www.ecrypt.eu.org/stream/`.

[BBC+06] 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. « DecimV2 ». Dans *Proceedings of SASC 2006 - ECRYPT Workshop on stream ciphers*, Leuven, Belgique, février 2006.

[BBS86] L. Blum, M. Blum et M. Shub. « A simple unpredicable pseudo-random number generator ». *SIAM Journal on Computing*, 15, 1986.

[BBS99] E. Biham, A. Biryukov et A. Shamir. « Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials ». Dans *Advances in Cryptology - EUROCRYPT'99*, volume 1592 de *Lecture Notes in Computer Science*, pages 12–23. Springer-Verlag, 1999.

[BCCLC06] T.P. Berger, A. Canteaut, P. Charpin et Y. Laigle-Chapuy. « On Almost Perfect Nonlinear functions ». *IEEE Transactions on Information Theory*, 2006. A paraître.

[BCFL05] L. Budaghyan, C. Carlet, P. Felke et G. Leander. « An infinite class of quadratic APN functions which are not equivalent to power mappings ». Cryptology ePrint Archive, Report 2005/359, 2005. `http://eprint.iacr.org/`.

[BCP05] L. Budaghyan, C. Carlet et A. Pott. « New constructions of Almost Bent and Almost Perfect Nonlinear Polynomials ». Dans *Workshop on Coding and Cryptography – WCC 2005*, pages 306–315, Bergen, Norway, mars 2005.

[BCP06] L. Budaghyan, C. Carlet et A. Pott. « New classes of almost bent and almost perfect nonlinear polynomials ». *IEEE Transactions on Information Theory*, 52(3):1141–1152, 2006.

[BD93] T. Beth et C. Ding. « On almost perfect nonlinear permutations ». Dans *Advances in Cryptology - EUROCRYPT'93*, volume 765 de *Lecture Notes in Computer Science*, pages 65–76. Springer-Verlag, 1993.

[BDQ04] A. Biryukov, C. De Cannière et M. Quisquater. « On Multiple Linear Approximations ». Dans *Advances in Cryptology - CRYPTO 2004*, volume 3152 de *Lecture Notes in Computer Science*, pages 1–22. Springer-Verlag, 2004.

[Ber05] D.J. Bernstein. « Cache-timing attacks on AES », 2005. `http://cr.yp.to/antiforgery/cachetiming-20050414.pdf`.

[BF00] S. Babbage et L. Frisch. « On MISTY1 Higher Order Differential Cryptanalysis ». Dans *ICISC 2000*, volume 2015 de *Lecture Notes in Computer Science*, pages 22–36. Springer-Verlag, 2000.

[BFSY04] M. Bardet, J-C. Faugère, B. Salvy et B-Y. Yang. « On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations ». Dans *Proc. International Conference on Polynomial System Solving (ICPSS'2004)*, 2004.

[BFSY05] M. Bardet, J-C. Faugère, B. Salvy et B-Y. Yang. « Asymptotic Behaviour of the Degree of Regularity of Semi-Regular Polynomial Systems ». Dans *MEGA 2005*, Porto Conte, Italy, May 2005.

[BJV04] T. Baignères, P. Junod et S. Vaudenay. « How far can we go beyond linear cryptanalysis? ». Dans *Advances in Cryptology - ASIACRYPT 2004*, volume 3329 de *Lecture Notes in Computer Science*, pages 432–450. Springer-Verlag, 2004.

[BL06] A. Braeken et J. Lano. « On the (im)possibility of practical and secure nonlinear filters and combiners ». Dans *Selected Areas in Cryptography - SAC 2005*, volume 3897 de *Lecture Notes in Computer Science*, pages 159–174. Springer-Verlag, 2006.

[BLM+05] A. Braeken, J. Lano, N. Mentens, B. Preneel et I. Verbauwhede. « SFINKS: a synchronous stream cipher for restricted hardware environments ». Soumission au projet eSTREAM [ECR05], 2005. `http://www.ecrypt.eu.org/stream/`.

[BM84] M. Blum et S. Micali. « How to generate cryptographically strong sequences of pseudo-random bits ». *SIAM Journal on Computing*, 13, 1984.

[BP05] A. Braeken et B. Preneel. « On the Algebraic Immunity of Symmetric Boolean Functions ». Dans *Progress in Cryptology - INDOCRYPT 2005*, volume 3797 de *Lecture Notes in Computer Science*, pages 35–48. Springer, 2005.

[BPP00] J. Boyar, R. Peralta et D. Pochuev. « On the multiplicative complexity of Boolean functions over the basis $(\wedge, +, 1)$ ». *Theoretical Computer Science*, 235(1):43–57, 2000.

[Bro93] A.E. Brouwer. « The Linear Programming Bound for Binary Linear Codes ». *IEEE Transactions on Information Theory*, 39(2):677–680, 1993.

[BS91] E. Biham et A. Shamir. « Differential Cryptanalysis of DES-like cryptosystems ». *Journal of Cryptology*, 4(1):3–72, 1991.

[BS05] E. Biham et J. Seberry. « Py: a fast and secure stream cipher using rolling arrays ». Soumission au projet eSTREAM [ECR05], 2005. `http://www.ecrypt.eu.org/stream/`.

[BT93] A.E. Brouwer et L.M.G.M. Tolhuizen. « A sharpening of the Johnsson bound for binary linear codes and the nonexistence of linear codes with Preparata parameters ». *Designs, Codes and Cryptography*, 3(2):95–98, 1993.

[BW72] E.R. Berlekamp et L.R. Welch. « Weight distributions of the cosets of the (32,6) Reed-Muller code ». *IEEE Transactions on Information Theory*, 18(1):203–207, 1972.

[Can97] A. Canteaut. « Differential cryptanalysis of Feistel ciphers and differentially uniform mappings ». Dans *Selected Areas on Cryptography, SAC'97*, pages 172–184, Ottawa, Canada, 1997.

[Can01a] A. Canteaut. « Cryptographic Functions and Design Criteria for Block Ciphers ». Dans *INDOCRYPT 2001*, Chennai, India, 2001.

[Can01b] A. CANTEAUT. « On the weight distributions of optimal cosets of the First-Order Reed-Muller Code ». *IEEE Transactions on Information Theory*, 47(1):407–413, janvier 2001.

[Can02] A. CANTEAUT. « On the correlations between a combining function and functions of fewer variables ». Dans *Proceedings of the 2002 IEEE Information Theory Workshop - ITW 2002*, pages 78–81, Bangalore, Inde, octobre 2002. IEEE Press.

[Can06] A. CANTEAUT. « Open problems related to algebraic attacks on stream ciphers ». Dans *Workshop on Coding and Cryptography - WCC 2005*, volume 3969 de *Lecture Notes in Computer Science*, 2006. A paraître.

[Car79] L. CARLITZ. « Explicit evaluation of certain exponential sums ». *Mathematica Scandinavica*, 44:5–16, 1979.

[Car93] C. CARLET. « Partially-bent functions ». *Designs, Codes and Cryptography*, (3):135–145, 1993.

[Car94a] C. CARLET. « Fonctions booléennes en théorie des codes correcteurs d'erreurs et en cryptologie ». Thèse d'habilitation de l'Université de Picardie, 1994.

[Car94b] C. CARLET. « Two new classes of bent functions ». Dans *Advances in Cryptology - EUROCRYPT'93*, volume 765 de *Lecture Notes in Computer Science*, pages 77–101. Springer-Verlag, 1994.

[Car06] C. CARLET. « On the higher-order nonlinearities of algebraic-immune fonctions ». Dans *Advances in Cryptology - CRYPTO 2006*, volume 4117 de *Lecture Notes in Computer Science*. Springer, 2006.

[CC03] A. CANTEAUT et P. CHARPIN. « Decomposing Bent Functions ». *IEEE Transactions on Information Theory*, 49(8):2004–19, août 2003.

[CCCF00] A. CANTEAUT, C. CARLET, P. CHARPIN et C. FONTAINE. « Propagation characteristics and correlation-immunity of highly nonlinear Boolean functions ». Dans *Advances in Cryptology - EUROCRYPT'2000*, volume 1807 de *Lecture Notes in Computer Science*, pages 507–522. Springer-Verlag, 2000.

[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, mai 2001.

[CCD99] A. CANTEAUT, P. CHARPIN et H. DOBBERTIN. « A new characterization of almost bent functions ». Dans *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 conjecture ». *IEEE Transactions on Information Theory*, 46(1):4–8, janvier 2000.

[CCD00b] A. CANTEAUT, P. CHARPIN et H. DOBBERTIN. « Weight divisibility of cyclic codes, highly nonlinear functions on GF(2^m) and crosscorrelation of maximum-length sequences ». *SIAM Journal on Discrete Mathematics*, 13(1):105–138, 2000.

[CCK06] A. CANTEAUT, P. CHARPIN et G. KYUREGHYAN. « A new class of monomial bent functions ». Dans *Proceedings of the 2006 IEEE International Symposium on Information Theory - ISIT 2006*, Seattle, USA, 2006. A paraître.

[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.

[CD96] T. Cusick et H. Dobbertin. « Some new 3-valued crosscorrelation functions of binary m-sequences ». *IEEE Transactions on Information Theory*, 42(4):1238–1240, 1996.

[CDG05] Nicolas Courtois, Blandine Debraize et Eric Garrido. « On Exact Algebraic [Non-]Immunity of S-boxes Based on Power Functions ». IACR ePrint Report 2005/203, juin 2005. `http://eprint.iacr.org/2005/203`.

[CDLD03] A. Canteaut, M. Daum, G. Leander et H. Dobbertin. « Normal and non normal bent functions ». Dans *Workshop on Coding and Cryptography - WCC 2003*, pages 91–100, Versailles, France, mars 2003.

[CDLD06] A. Canteaut, M. Daum, G. Leander et H. Dobbertin. « Normal and non normal bent functions ». *Discrete Applied Mathematics*, 154(2):202–218, février 2006. Special issue in Coding and Cryptology.

[CF01] A. Canteaut et E. Filiol. « Ciphertext only reconstruction of stream ciphers based on combination generators ». Dans *Fast Software Encryption - FSE 2000*, volume 1978 de *Lecture Notes in Computer Science*, pages 165–180. Springer-Verlag, 2001.

[CF02] A. Canteaut et E. Filiol. « On the Influence of the Filtering Function on the Performance of Fast Correlation Attacks on Filter Generators ». Dans *23rd Symposium on Information Theory in the Benelux*, Louvain-la-Neuve, Belgium, mai 2002.

[CG06] C. Carlet et P. Gaborit. « Hyper-bent functions and cyclic codes ». *Journal of Combinatorial Theory, Series A*, (113):466–482, 2006.

[CGH+85] B. Chor, O. Goldreich, J. Hastad, J. Freidmann, S. Rudich et R. Smolensky. « The bit extraction problem or t-resilient functions ». Dans *Proc. 26th IEEE Symposium on Foundations of Computer Science*, pages 396–407, 1985.

[Cha98] P. Charpin. « *Handbook of Coding Theory* », volume I, Chapitre 11 - Open problems on cyclic codes, pages 963–1063. Amsterdam, the Netherlands: Elsevier, 1998.

[Cha04] P. Charpin. « Cyclic codes with few weights and Niho exponents ». *Journal of Combinatorial Theory, Series A*, (108):247–259, 2004.

[CHJ02] D. Coppersmith, S. Halevi et C. Jutla. « Cryptanalysis of stream ciphers with linear masking ». Dans *Advances in Cryptology - CRYPTO 2002*, volume 2442 de *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[CHZ06] P. Charpin, T. Helleseth et V. Zinoviev. « Propagation characteristics of $x \mapsto x^{-1}$ and Kloosterman sums ». *Finite Fields and their Applications*, 2006. A paraître.

[CJM02] P. Chose, A. Joux et M. Mitton. « Fast correlation attacks: an algorithmic point of view ». Dans *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 ». Dans *Fast Software Encryption 2000*, volume 1978 de *Lecture Notes in Computer Science*. Springer-Verlag, 2000.

[CKPS00] N. Courtois, A. Klimov, J. Patarin et A. Shamir. « Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. ». Dans *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 de *Lecture Notes in Computer Science*, pages 392–407. Springer-Verlag, 2000.

[CL04] J.H. Cheon et D.H. Lee. « Resistance of S-boxes against algebraic attacks ». Dans *Fast Software Encryption - FSE 2004*, volume 3017 de *Lecture Notes in Computer Science*, pages 83–94. Springer-Verlag, 2004.

[CL05] C. Cid et G. Leurent. « An analysis of the XSL algorithm ». Dans *Advances in Cryptology - ASIACRYPT 2005*, volume 3788 de *Lecture Notes in Computer Science*, pages 333–352. Springer, 2005.

[CLS06] A. Canteaut, C. Lauradoux et A. Seznec. « Understanding cache attacks ». Rapport de recherche RR-5881, INRIA, avril 2006. `http://www.inria.fr/rrrt/`.

[Clu03] M. Cluzeau. « Reconstruction d'un brasseur linéaire ». Stage de DEA de l'Université de Limoges, 2003.

[Clu04] M. Cluzeau. « Reconstruction of a linear scrambler ». Dans *Proceedings of the 2004 IEEE International Symposium on Information Theory - ISIT'04*, page 230, Chicago, USA, juillet 2004.

[CM03a] N. Courtois et W. Meier. « Algebraic attacks on stream ciphers with linear feedback ». Dans *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 de *Lecture Notes in Computer Science*, pages 345–359. Springer-Verlag, 2003.

[CM03b] N. Courtois et W. Meier. « Algebraic attacks on stream ciphers with linear feedback ». Dans *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 de *Lecture Notes in Computer Science*, pages 345–359. Springer-Verlag, 2003.

[CMR05] C. Cid, S. Murphy et M. Robshaw. « Small Scale Variants of the AES ». Dans *Fast Software Encryption - FSE 2005*, volume 3557 de *Lecture Notes in Computer Science*, pages 145–162. Springer-Verlag, 2005.

[Cou03] N. Courtois. « Fast algebraic attacks on stream ciphers with linear feedback ». Dans *Advances in Cryptology - CRYPTO 2003*, volume 2729 de *Lecture Notes in Computer Science*, pages 176–194. Springer-Verlag, 2003.

[Cou04] N. Courtois. « Feistel schemes and bi-linear cryptanalysis ». Dans *Advances in Cryptology - CRYPTO 2004*, volume 3152 de *Lecture Notes in Computer Science*, pages 23–40. Springer-Verlag, 2004.

[Cou05] N. Courtois. « Algebraic attacks on combiners with memory and several outputs ». Dans *ICISC 2004*, Lecture Notes in Computer Science. Springer-Verlag, 2005. `http://eprint.iacr.org/2003/125/`.

[Cou06] N. Courtois. « Cryptanalysis of SFINKS ». Dans *Information Security and Cryptology - ICISC 2005*, volume 3935 de *Lecture Notes in Computer Science*. Springer, 2006.

[CP02] N. Courtois et J. Pieprzyk. « Cryptanalysis of block ciphers with overdefined systems of equations ». Dans *Advances in Cryptology - Asiacrypt'02*, volume 2501 de *Lecture Notes in Computer Science*, pages 267–287. Springer-Verlag, 2002.

[CP05] P. Charpin et E. Pasalic. « Highly nonlinear resilient functions through disjoint codes in projective spaces ». *Designs, Codes and Cryptography*, (37):319–346, 2005.

[CT91] T. Cover et J. Thomas. *Elements of Information theory*. Wiley, 1991.

[CT00] A. Canteaut et M. Trabbia. « Improved fast correlation attacks using parity-check equations of weight 4 and 5 ». Dans *Advances in Cryptology - EURO-*

CRYPT'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 Inform. Transmission*, 33(4):287–296, 1997.

[CTZ01] P. Charpin, A. Tietäväinen et V. Zinoviev. « On binary cyclic codes with codewords of weight 3 and binary sequences with the trinomial property ». *IEEE Transactions on Information Theory*, 47(1):421–425, 2001.

[CV95] F. Chabaud et S. Vaudenay. « Links between differential and linear cryptanalysis ». Dans *Advances in Cryptology - EUROCRYPT'94*, volume 950 de *Lecture Notes in Computer Science*, pages 356–365. Springer-Verlag, 1995.

[CV02] A. Canteaut et M. Videau. « Degree of composition of highly nonlinear functions and applications to higher order differential cryptanalysis ». Dans *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 de *Lecture Notes in Computer Science*, pages 518–533. Springer-Verlag, 2002.

[CV05] A. Canteaut et M. Videau. « Symmetric Boolean functions ». *IEEE Transactions on Information Theory*, 51(8):2791–2811, août 2005.

[CW90] D. Coppersmith et S. Winograd. « Matrix multiplication via arithmetic programming ». *Journal of Symbolic Computation*, (9):251–280, 1990.

[DD04] J.F. Dillon et H. Dobbertin. « New Cyclic Difference Sets with Singer Parameters ». *Finite Fields and their Applications*, pages 342–389, 2004.

[DDG+06] A. Diene, J. Ding, J.E. Gower, T.J. Hodges et Z. Yin. « Dimension of the linearization equations of the Matsumoto-Imai cryptosystems ». Dans *Workshop on Coding and Cryptography - WCC 2005*, volume 3969 de *Lecture Notes in Computer Science*. Spriner, 2006. A paraître.

[DGM04] D. K. Dalai, K. C. Gupta et S. Maitra. « Results on Algebraic Immunity for Cryptographically Significant Boolean Functions ». Dans *Progress in Cryptology - Indocrypt 2004*, volume 1880 de *Lecture Notes in Computer Science*, pages 92–106. Springer-Verlag, 2004.

[DGM05] D.K. Dalai, K.C. Gupta et S. Maitra. « Cryptographically Significant Boolean Functions: Construction and Analysis in Terms of Algebraic Immunity ». Dans *Fast Software Encryption - FSE 2005*, volume 3357 de *Lecture Notes in Computer Science*, pages 98–111. Springer-Verlag, 2005.

[Did06] F. Didier. « A new bound on the block error probability after decoding over the erasure channel ». *IEEE Transactions on Information Theory*, 2006. A paraître.

[Die04] C. Diem. « The XL algorithm and a conjecture from commutative algebra ». Dans *Advances in Cryptology - ASIACRYPT 2004*, volume 3329 de *Lecture Notes in Computer Science*, pages 323–337. Springer-Verlag, 2004.

[Dil74] J.F. Dillon. « *Elementary Hadamard Difference sets* ». Thèse de doctorat, University of Maryland, 1974.

[DLC+06] H. Dobbertin, G. Leander, A. Canteaut, C. Carlet, P. Felke et P. Gaborit. « Construction of bent functions via Niho power functions ». *Journal of Combinatorial Theory, Series A*, 2006. A paraître.

[Dob] H. Dobbertin. Communication personnelle.

[Dob94] H. Dobbertin. « Construction of bent functions and balanced Boolean functions with high nonlinearity ». Dans *Fast Software Encryption - FSE'94*, volume 1008 de *Lecture Notes in Computer Science*, pages 61–74. Springer-Verlag, 1994.

[Dob98] 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.

[Dob99a] H. Dobbertin. « Almost Perfect Nonlinear power functions on $GF(2^n)$: the Niho case ». *Information and Computation*, 151(1-2):57–72, 1999.

[Dob99b] H. Dobbertin. « Almost Perfect Nonlinear power functions on $GF(2^n)$: the Welch case ». *IEEE Transactions on Information Theory*, 45(4):1271–1275, 1999.

[Dob00] H. Dobbertin. « Almost Perfect Nonlinear Power Functions on $GF(2^n)$: a new class for n divisible by 5 ». Dans *Proceedings of Finite Fields and Applications Fq5*, pages 113–121, Augsburg, Germany, 2000. Springer-Verlag.

[DSM05] D.K. Dalai, S. Sarkar et S. Maitra. « Balanced Boolean functions with maximum possible algebraic immunity ». Preprint, avril 2005.

[DT06] F. Didier et J.-P. Tillich. « Computing the Algebraic Immunity Efficiently ». Dans *Fast Software Encryption - FSE 2006*, Lecture Notes in Computer Science. Springer, 2006. A paraître.

[DZ84] S.M. Dodunekov et V. Zinoviev. « A note on Preparata codes ». Dans *Proceedings of the 6th Intern. Symp. on Information Theory, Moscow-Tashkent Part 2*, pages 78–80, 1984.

[ECR05] ECRYPT - European Network of Excellence in Cryptology. « The eSTREAM Stream Cipher Project ». `http://www.ecrypt.eu.org/stream/`, 2005.

[EJ02] P. Ekdahl et T. Johansson. « A new version of the stream cipher SNOW ». Dans *Selected Areas in Cryptography – SAC 2002*, volume 2295 de *Lecture Notes in Computer Science*, pages 47–61. Springer-Verlag, 2002.

[EJ05] H. Englund et T. Johansson. « A new simple technique to attack filter generators and related ciphers ». Dans *Selected Areas in Cryptography - SAC 2004*, volume 3357 de *Lecture Notes in Computer Science*, pages 39–53. Springer-Verlag, 2005.

[EKP06] Y. Edel, G. Kyureghyan et A. Pott. « A new APN function which is not equivalent to a power mapping ». *IEEE Transactions on Information Theory*, 52(2):744–747, 2006.

[FA03] J.-C. Faugère et G. Ars. « An algebraic cryptanalysis of nonlinear filter generators using Gröbner bases ». Rapport de Recherche RR-4739, INRIA, 2003. `ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-4739.pdf`.

[Fau99] J.-C. Faugère. « A new efficient algorithm for computing Gröbner bases (F_4) ». *Journal of Pure and Applied Algebra*, 139(1-3):61–88, 1999.

[Fau02] J.-C. Faugère. « A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5) ». Dans *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*. ACM, 2002.

[FJ03] J.-C. Faugère et A. Joux. « Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases ». Dans *Advances in Cryptology - CRYPTO 2003*, volume 2729 de *Lecture Notes in Computer Science*. Springer-Verlag, 2003.

[Fon98] C. FONTAINE. « *Contribution à la recherche de fonctions booléennes hautement non-linéaires, et au marquage d'images en vue de la protection des droits d'auteur* ». Thèse de doctorat, Université Paris 6, 1998.

[Fon99] C. FONTAINE. « On some cosets of the First-Order Reed-Muller code with high minimum weight ». *IEEE Transactions on Information Theory*, 45(4):1237–1243, 1999.

[Gal62] R.G. GALLAGER. « Low-Density Parity-check Codes ». *IRE Transactions on Information Theory*, IT-8:21–28, 1962.

[GC91] H. GILBERT et G. CHASSÉ. « A Statistical Attack of the FEAL-8 Cryptosystem ». Dans *Advances in Cryptology - CRYPTO'90*, volume 537 de *Lecture Notes in Computer Science*, pages 22–33. Springer-Verlag, 1991.

[GG99a] S.W. GOLOMB et G. GONG. « Periodic binary sequences with the trinomial property ». *IEEE Transactions on Information Theory*, 45(4):1276–1279, 1999.

[GG99b] G. GONG et S.W. GOLOMB. « Transform domain analysis of DES ». *IEEE Transactions on Information Theory*, 45(6):2065–2073, 1999.

[Gol68a] R. GOLD. « Maximal recursive sequences with 3-valued recursive crosscorrelation functions ». *IEEE Transactions on Information Theory*, 14:154–156, 1968.

[Gol68b] S.W. GOLOMB. « Theory of transformation groups of polynomials over $GF(2)$ with applications to linear shift register sequences ». *Information Sciences*, 1:87–109, 1968.

[Gol96] J. GOLIC. « On the Security of Nonlinear Filter Generators ». Dans *Fast Software Encryption 1996*, volume 1039 de *Lecture Notes in Computer Science*, pages 173–188. Springer-Verlag, 1996.

[Gol97] J. GOLIC. « Cryptanalysis of alleged A5 stream cipher ». Dans *Advances in Cryptology - EUROCRYPT'97*, volume 1233 de *Lecture Notes in Computer Science*, pages 239–255. Springer-Verlag, 1997.

[GRS95] O. GOLDREICH, R. RUBINFELD et M. SUDAN. « Learning polynomials with queries: the highly noisy case ». Dans *36th Annual Symposium on Fondation of Computer Science*, pages 294–303, Milwaukee, Wisconsin, 1995.

[GS03] K.C. GUPTA et P. SARKAR. « Construction of Perfect Nonlinear and Maximally Nonlinear Multi-output Boolean Functions Satisfying Higher Order Strict Avalanche Criteria ». Dans *Progress in Cryptology - Indocrypt 2003*, volume 2904 de *Lecture Notes in Computer Science*, pages 107–120. Springer, 2003.

[GSB+05] A. GOUGET, H. SIBERT, C. BERBAIN, N. COURTOIS, B. DEBRAIZE et C.J. MITCHELL. « Analysis of the Bit-Search Generator and Sequence Compression Techniques ». Dans *Fast Software Encryption - FSE 2005*, volume 3557 de *Lecture Notes in Computer Science*, pages 196–214. Springer, 2005.

[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.

[Hil94] A. HILTGEN. « *Cryptographically relevant contributions to combinatorial complexity theory* ». Thèse de doctorat, ETH Zürich, 1994.

[Hit01] HITACHI, LD. « MUGI Specification ». `http://www.sdl.hitachi.co.jp/crypto/mugi/mugi_spe.pdf`, 2001.

[HJM05] M. Hell, T. Johansson et W. Meier. « Grain: A Stream Cipher for Constrained Environments ». Soumission au projet eSTREAM [ECR05], 2005. `http://www.ecrypt.eu.org/stream/`.

[HK98] T. Helleseth et P. Vijay Kumar. « *Handbook of Coding Theory* », volume II, Chapitre 21 - 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 ». Dans *Advances in Cryptology - EUROCRYPT'95*, volume 921 de *Lecture Notes in Computer Science*, pages 24–38. Springer-Verlag, 1995.

[HLL+01] S. Hong, S. Lee, J. Lim, J. Sung et D. Cheon. « Provable security against differential and linear cryptanalysis for the SPN structure ». Dans *Fast Software Encryption - FSE 2001*, volume 1978 de *Lecture Notes in Computer Science*, pages 273–283. Springer-Verlag, 2001.

[HOP96] J. Hagenauer, E. Offer et L. Papke. « Iterative decoding of binary block and convolutional codes ». *IEEE Transactions on Information Theory*, 42(2):429–445, 1996.

[Hou96a] X.-D. Hou. « The covering radius of $R(1,9)$ in $R(4,9)$ ». *Designs, Codes and Cryptography*, 8:285–292, 1996.

[Hou96b] X.-D. Hou. « Covering Radius of the Reed-Muller code $R(1,7)$ - a simpler proof ». *Journal of Combinatorial Theory, Series A*, (74):337–341, 1996.

[Hou96c] X.-D. Hou. « On the covering radius of $R(1,m)$ in $R(3,m)$ ». *IEEE Transactions on Information Theory*, 42(3):1035–1037, 1996.

[Hou03] X.-D. Hou. « Affinity of permutations of $\mathbf{F}_2^n$ ». Dans *Workshop on Coding and Cryptography – WCC 2003*, pages 273–280, Versailles, France, 2003.

[HR04] P. Hawkes et G. G. Rose. « Rewriting variables: the complexity of fast algebraic attacks on stream ciphers ». Dans *Advances in Cryptology - CRYPTO 2004*, volume 3152 de *Lecture Notes in Computer Science*, pages 390–406. Springer-Verlag, 2004.

[HX01] H.D.L. Hollmann 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.

[ISO05] ISO/IEC 18033-4. « Information technology – Security techniques – Encryption algorithms – Part 4: Stream ciphers ». International Organization for Standardization, juillet 2005.

[JJ99a] T. Johansson et F. Jönsson. « Fast correlation attacks based on turbo code techniques ». Dans *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 ». Dans *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 ». Dans *Advances in Cryptology - CRYPTO'00*, volume 1880 de *Lecture Notes in Computer Science*, pages 300–315. Springer-Verlag, 2000.

[JJ02] F. Jönsson et T. Johansson. « A fast correlation attack on LILI-128 ». *Information Processing Letters*, 81(3):127–132, février 2002.

[JK97] T. Jakobsen et L.R. Knudsen. « The interpolation attack on block ciphers ». Dans *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, Ecole 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.

[Kat71] N. Katz. « On a theorem of Ax ». *American Journal of Mathematics*, 93:485–499, 1971.

[KG97] A. Klapper et M. Goresky. « Feedback shift registers, 2-adic span and combiners with memory ». *Journal of Cryptology*, 10(2), 1997.

[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. A paraître.

[KMT01] L. Keliher, H. Meijer et S. Tavares. « New method for upper bounding the maximum average linear hull probability for SPNs ». Dans *Advances in Cryptology - EUROCRYPT 2001*, volume 2045 de *Lecture Notes in Computer Science*, pages 420–436. Springer-Verlag, 2001.

[KMY06] S. Kavut, S. Maitra et M.D. Yücel. « There exist Boolean functions on n (odd) variables having nonlinearity $> 2^{n-1} - 2^{\frac{n-1}{2}}$ if and only if $n > 7$ », mai 2006.

[Knu95] L. R. Knudsen. « Truncated and higher order differentials ». Dans *Fast Software Encryption - FSE'94*, volume 1008 de *Lecture Notes in Computer Science*, pages 196–211. Springer-Verlag, 1995.

[KS02] A. Klimov et A. Shamir. « A new class of invertible mappings ». Dans *CHES 2002*, volume 2523 de *Lecture Notes in Computer Science*, pages 470–483. Springer-Verlag, 2002.

[KS04] A. Klimov et A. Shamir. « Cryptographic applications of T-functions ». Dans *Selected Areas in Cryptography - SAC 2003*, volume 3006 de *Lecture Notes in Computer Science*. Springer-Verlag, 2004.

[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.

[Lai94] X. Lai. « Higher order derivatives and differential cryptanalysis ». Dans *Proc. "Symposium on Communication, Coding and Cryptography", in honor of J. L. Massey on the occasion of his 60'th birthday*. Kluwer Academic Publishers, 1994.

[Lau05] C. Lauradoux. « The complexity of symmetric Boolean functions ». Dans *Ecole de Jeunes Chercheurs en Algorithmique et Calcul Formel*, Montpellier, avril 2005.

[LC04] S. Lin et D.J. Costello. *Error Control Coding*. Pearson Prentice Hall, 2004. Seconde édition.

[LCPP96] S. Lee, S. Chee, S. Park et S. Park. « Conditional correlation attack on nonlinear filter generators ». Dans *Advances in Cryptology - ASIACRYPT'96*,

volume 1163 de *Lecture Notes in Computer Science*, pages 360–367. Springer-Verlag, 1996.

[Lea04] G. Leander. « *Normality of bent functions, monomial- and binomial-bent functions* ». Thèse de doctorat, Ruhr Universität Bochum, Allemagne, 2004.

[Lea06] G. Leander. « Monomial bent functions ». *IEEE Transactions on Information Theory*, 52(2):738–743, 2006.

[Lev04] S. Leveiller. « *Quelques algorithmes de cryptanalyse du registre filtré* ». Thèse de doctorat, Ecole Nationale Supérieure des Télécommunications, Paris, 2004.

[LMM91] X. Lai, J.L. Massey et S. Murphy. « Markov ciphers and differential cryptanalysis ». Dans *Advances in Cryptology - EUROCRYPT'91*, volume 547 de *Lecture Notes in Computer Science*, pages 17–38. Springer-Verlag, 1991.

[LMV05] Y. Lu, W. Meier et S. Vaudenay. « The Conditional Correlation Attack: A Practical Attack on Bluetooth Encryption ». Dans *Advances in Cryptology - CRYPTO 2005*, volume 3621 de *Lecture Notes in Computer Science*, pages 97–117. Springer-Verlag, 2005.

[LN83] R. Lidl et H. Niederreiter. *Finite fields*. Cambridge University Press, 1983.

[Löh03] B. Löhlein. « Attacks based on Conditional Correlations against the Nonlinear Filter Generator ». IACR ePrint Report 2003/020, 2003. `http://eprint.iacr.org/2003/020/`.

[LQ06] N. Li et W. Qi. « Symmetric Boolean Function with Maximum Algebraic Immunity on Odd Number of Variables ». *IEEE Transactions on Information Theory*, 2006. A paraître.

[Lu06] Y. Lu. « *Applied stream ciphers in mobile communications* ». Thèse de doctorat, Ecole Polytechnique Fédérale de Lausanne, 2006.

[LV04a] Y. Lu et S. Vaudenay. « Cryptanalysis of Bluetooth keystream generator two-level E0 ». Dans *Advances in Cryptology - ASIACRYPT 2004*, volume 3329 de *Lecture Notes in Computer Science*, pages 483–499. Springer-Verlag, 2004.

[LV04b] Y. Lu et S. Vaudenay. « Faster correlation attack on Bluetooth keystream generator E0 ». Dans *Advances in Cryptology - CRYPTO 2004*, volume 3152 de *Lecture Notes in Computer Science*, pages 407–425. Springer-Verlag, 2004.

[LW90] G. Lachaud et J. Wolfmann. « The weights of the orthogonal of the extended quadratic binary Goppa codes ». *IEEE Transactions on Information Theory*, 36(3):686–692, 1990.

[LZGB03] S. Leveiller, G. Zémor, P. Guillot et J. Boutros. « A new cryptanalytic attack for PN-generators filtered by a Boolean function ». Dans *Selected areas in cryptography - SAC 2002*, volume 2595 de *Lecture Notes in Computer Science*, pages 232–249. Springer-Verlag, 2003.

[Mas69] J.L. Massey. « Shift-register synthesis and BCH decoding ». *IEEE Transactions on Information Theory*, 15:122–127, janvier 1969.

[Mat94] M. Matsui. « Linear cryptanalysis method for DES cipher ». Dans *Advances in Cryptology - EUROCRYPT'93*, volume 765 de *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[Mat95] M. Matsui. « The first experimental cryptanalysis of the Data Encryption Standard ». Dans *Advances in Cryptology - CRYPTO'94*, volume 839 de *Lecture Notes in Computer Science*. Springer-Verlag, 1995.

[Mat97] M. Matsui. « New Block Encryption Algorithm MISTY ». Dans *Fast Software Encryption - FSE'97*, volume 1267 de *Lecture Notes in Computer Science*, pages 54–68. Springer-Verlag, 1997.

[Mau92] U. Maurer. « Conditionally-Perfect Secrecy and a Provably-Secure Randomized Cipher ». *Journal of Cryptology*, 5(1):53–66, 1992.

[MC95] G. McGuire et A.R. Calderbank. « Proof of a conjecture of Sarwate and Pursley regarding pairs of binary m-sequences ». *IEEE Transactions on Information Theory*, 41(4):1153–1155, 1995.

[McE72] R.J. McEliece. « Weight congruence for p-ary cyclic codes ». *Discrete Mathematics*, 3:177–192, 1972.

[McF73] R. L. McFarland. « A family of noncyclic difference sets ». *Journal of Combinatorial Theory, Series A*, 15:1–10, 1973.

[MFI00] M. J. Mihaljevic, M. P.C. Fossorier et H. Imai. « A low-complexity and high performance algorithm for the fast correlation attack ». Dans *Fast Software Encryption – FSE 2000*, volume 1978 de *Lecture Notes in Computer Science*. Springer-Verlag, 2000.

[MH04] H. Molland et T. Helleseth. « An improved correlation attack against irregular clocked and filtered generator ». Dans *Advances in Cryptology - CRYPTO 2004*, volume 3152 de *Lecture Notes in Computer Science*, pages 373–389. Springer-Verlag, 2004.

[MH05] H. Molland et T. Helleseth. « A Linear Weakness in T-functions ». Dans *Proceedings of the 2005 IEEE International Symposium on Information Theory - ISIT 2005*. IEEE, 2005.

[Mol04] H. Molland. « Improved Linear Consistency Attack on Irregular Clocked Keystream Generators ». Dans *Fast Software Encryption - FSE 2004*, volume 3017 de *Lecture Notes in Computer Science*, pages 109–126. Springer-Verlag, 2004.

[MP75] D.E. Muller et F.P. Preparata. « Bounds to complexities of networks for sorting and switching ». *Journal of the ACM*, (22):1531–1540, 1975.

[MPC04] W. Meier, E. Pasalic et C. Carlet. « Algebraic attacks and decomposition of Boolean functions ». Dans *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 de *Lecture Notes in Computer Science*, pages 474–491. Springer-Verlag, 2004.

[MS77] F.J. MacWilliams et N.J.A. Sloane. *The theory of error-correcting codes*. North-Holland, 1977.

[MS88] W. Meier et O. Staffelbach. « Fast correlation attacks on stream ciphers ». Dans *Advances in Cryptology - EUROCRYPT'88*, volume 330 de *Lecture Notes in Computer Science*, pages 301–314. Springer-Verlag, 1988.

[MS89] W. Meier et O. Staffelbach. « Fast correlation attack on certain stream ciphers ». *Journal of Cryptology*, pages 159–176, 1989.

[MvOV97] A.J. Menezes, P.C. van Oorshot et S.A. Vanstone. *Handbook of applied cryptography*. CRC Press, 1997. Disponible sur `http://www.cacr.math.uwaterloo.ca/hac/`.

[Myk80] J. Mykkeltveit. « The covering radius of the (128,8) Reed-Muller code is 56 ». *IEEE Transactions on Information Theory*, IT-26(3):359–362, 1980.

[NGG06] Y. Nawaz, G. Gong et K.C. Gupta. « Upper bounds on Algebraic Immunity of Boolean power functions ». Dans *Fast Software Encryption - FSE 2006*, Lecture Notes in Computer Science. Springer, 2006.

[Nih72] Y. Niho. « *Multi-valued cross-correlation functions between two maximal linear recursive sequences* ». Thèse de doctorat, Univ. Southern California, 1972.

[NK93] K. Nyberg et L.R. Knudsen. « Provable security against differential cryptanalysis ». Dans *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 ». Dans *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 ». Dans *Advances in Cryptology - EUROCRYPT'93*, volume 765 de *Lecture Notes in Computer Science*, pages 55–64. Springer-Verlag, 1993.

[Nyb95] K. Nyberg. « S-boxes and Round Functions with Controllable Linearity and Differential Uniformity ». Dans *Fast Software Encryption - FSE'94*, volume 1008 de *Lecture Notes in Computer Science*, pages 111–130. Springer-Verlag, 1995.

[Pat95] J. Patarin. « Cryptoanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88 ». Dans *Advances in Cryptology - CRYPTO'95*, volume 963 de *Lecture Notes in Computer Science*, pages 248–261. Springer-Verlag, 1995.

[Pat96] J. Patarin. « Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms ». Dans *Advances in Cryptology - EUROCRYPT'96*, volume 1070 de *Lecture Notes in Computer Science*, pages 33–48. Springer-Verlag, 1996.

[Ple63] V. Pless. « Power moment identities on weight distributions in error-correcting codes ». *Info. and Control*, 3:147–152, 1963.

[PSLL03] S. Park, S.H. Sung, S. Lee et J. Lim. « Improving the upper bound on the maximum differential and the maximum linear hull probability for SPN structures and AES ». Dans *Fast Software Encryption - FSE 2003*, volume 2887 de *Lecture Notes in Computer Science*, pages 247–260. Springer-Verlag, 2003.

[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.

[PW90] N.J. Patterson et D.H. Wiedemann. « Correction to [PW83] ». *IEEE Transactions on Information Theory*, 36(2):443, 1990.

[Qua02] P. Quanty. « Etude d'une attaque sur les algorithmes de chiffrement à flot ». Stage de DEA de l'Université de Limoges, 2002.

[Rab05] M.O. Rabin. « Provably unbreakable Hyper-Encryption in the limited access model ». Dans *Proceedings of the 2005 IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security*, pages 34–37. IEEE, 2005.

[RB67] J.P. Robinson et A.J. Bernstein. « A class of binary recurrent codes with limited error propagation ». *IEEE Transactions on Information Theory*, 13(1):106–113, 1967.

[rGPP01a] 3rd Generation PartnerShip Project. « 3GPP TS 35.201 - Specification of the 3GPP Confidentiality and Integrity algorithms - Document 1: *f8* and *f9* specification », 2001. `http://www.3gpp.org/ftp/Specs/html-info/35201.htm`.

[rGPP01b] 3rd Generation PartnerShip Project. « 3GPP TS 35.202 - Specification of the 3GPP Confidentiality and Integrity algorithms - Document 2: KASUMI specification », 2001. `http://www.3gpp.org/ftp/Specs/html-info/35202.htm`.

[Rot76] O.S. Rothaus. « On bent functions ». *Journal of Combinatorial Theory, Series A*, (20):300–305, 1976.

[Sha48] C.E. Shannon. « A mathematical theory of communication ». *Bell Syst. Tech. J.*, 27, 1948.

[Sha49] C.E. Shannon. « 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*, IT-30(5):776–780, 1984.

[Sie85] T. Siegenthaler. « Decrypting a class of stream ciphers using ciphertext only ». *IEEE Transactions on Computers*, C-34(1):81–84, 1985.

[Sim94] J. Simonis. « Restrictions on the Weight Distribution of Binary Linear Codes Imposed by the structure of Reed-Muller codes ». *IEEE Transactions on Information Theory*, 40(1):194–196, 1994.

[SK96] B. Schneier et J. Kelsey. « Unbalanced Feistel networks and block cipher design ». Dans *Fast Software Encryption - FSE'96*, volume 1039 de *Lecture Notes in Computer Science*, pages 121–144. Springer-Verlag, 1996.

[SM00a] P. Sarkar et S. Maitra. « Construction of Nonlinear Boolean Functions with Important Cryptographic Properties ». Dans *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 de *Lecture Notes in Computer Science*, pages 485–506. Springer-Verlag, 2000.

[SM00b] P. Sarkar et S. Maitra. « Nonlinearity Bounds and Constructions of Resilient Boolean Functions ». Dans *Advances in Cryptology - CRYPTO 2000*, volume 1880 de *Lecture Notes in Computer Science*, pages 515–532. Springer-Verlag, 2000.

[SMC04] P. Stanica, S. Maitra et J. Clark. « Results on rotation symmetric bent and correlation immune Boolean functions ». Dans *Fast Software Encryption - FSE 2004*, volume 3017 de *Lecture Notes in Computer Science*. Springer-Verlag, 2004.

[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 ». Dans *Advances in Cryptology - CRYPTO'91*, volume 576 de *Lecture Notes in Computer Science*, pages 172–182. Springer-Verlag, 1991.

[THK99] H. Tanaka, K. Hisamatsu et T. Kaneko. « Strength of MISTY1 without FL function for Higher Order Differential Attack ». Dans *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes - AAECC-13*, volume 1719 de *Lecture Notes in Computer Science*, pages 221–230. Springer-Verlag, 1999.

[Val00] A. VALEMBOIS. « Fast soft-decision decoding of linear codes, stochastic resonance in algorithms ». Dans *Proceedings of the 2000 IEEE International Symposium on Information Theory - ISIT 2000*, page 91. IEEE Press, 2000.

[Vau99] S. VAUDENAY. « Vers une théorie du chiffrement symétrique ». Thèse d'Habilitation, Université Paris 7, 1999.

[vzGR97] J. von zur GATHEN et J.R. ROCHE. « Polynomials with two values ». *Combinatorica*, 17(3):345–362, 1997.

[Wag02] D. WAGNER. « A Generalized Birthday Problem ». Dans *Advances in Cryptology - CRYPTO 2002*, volume 2442 de *Lecture Notes in Computer Science*, pages 288–303. Springer-Verlag, 2002.

[Wag04] D. WAGNER. « Towards a Unifying View of Block Cipher Cryptanalysis ». Dans *Fast Software Encryption - FSE 2004*, volume 3017 de *Lecture Notes in Computer Science*, pages 16–33. Springer, 2004.

[Weg87] I. WEGENER. *The complexity of Boolean functions*. Wiley, 1987.

[Wol99] J. WOLFMANN. « Bent functions and coding theory ». Dans A. Pott et AL., éditeur, *Difference sets, sequences and their correlation properties*, pages 393–418. Kluwer Academic Publishers, 1999.

[Yao82] A.C. YAO. « Theory and Applications of trapdoor functions ». Dans *Proceedings of the IEEE 23rd Annual Symposium on Foundations of Computer Science*, pages 80–91. IEEE, 1982.

[ZC00] M. ZHANG et A. CHAN. « Maximum correlation analysis of nonlinear S-boxes in stream ciphers ». Dans *Advances in Cryptology - CRYPTO 2000*, volume 1880 de *Lecture Notes in Computer Science*, pages 501–514. Springer-Verlag, 2000.

[Zha00] M. ZHANG. « Maximum correlation analysis of nonlinear combining functions in stream ciphers ». *Journal of Cryptology*, 13(3):301–313, 2000.

[ZZ95] X.-M. ZHANG et Y. ZHENG. « GAC - the Criterion for Global Avalanche Characterics of Cryptographic Functions ». *Journal of Universal Computer Science*, 1(5):320–337, 1995.

[ZZ99a] X.-M. ZHANG et Y. ZHENG. « The Nonhomomorphicity of Boolean Functions ». Dans *Selected Areas in Cryptography - SAC'98*, volume 1553 de *Lecture Notes in Computer Science*, pages 280–295. Springer-Verlag, 1999.

[ZZ99b] Y. ZHENG et X.-M. ZHANG. « Plateaued functions ». Dans *Information and Communication Security, ICICS'99*, volume 1726 de *Lecture Notes in Computer Science*, pages 224–300. Springer-Verlag, 1999.

Curriculum Vitae

Formation

oct. 1996	doctorat de l'Université Pierre et Marie Curie (Paris 6), spécialité informatique, mention « très honorable avec les félicitations du jury ».
juin 1993	DEA de Systèmes Informatiques de l'Université P. et M. Curie, mention TB.
juin 1993	diplôme d'ingénieur de l'Ecole Nationale Supérieure de Techniques Avancées (ENSTA), spécialité : réseaux et communications.

Cursus professionnel

depuis oct. 1997	Chargée de recherche au projet CODES, INRIA Rocquencourt.
nov. 1996 - sept. 1997	ETH Zürich Séjour post-doctoral sous la direction de James L. Massey : *Critères de sécurité des systèmes de chiffrement à clef secrètes.*
sept. 1993 - oct. 1996	Projet CODES, INRIA Rocquencourt Thèse sous la direction de Paul Camion : *Attaques de cryptosystèmes à mots de poids faible et construction de fonctions t-résilientes.*
mars - août 1993	Laboratoire MASI, Université Paris VI Stage de fin d'études sous la direction de Claude Girault : *Analyse modulaire de réseaux de Petri et application à l'étude des problèmes de partage de ressources pour les systèmes distribués. Implémentation sur système-expert.* Prix du meilleur stage de fin d'études de la promotion 93 de l'ENSTA, décerné par l'AGM-ITA (association des anciens élèves).
sept. 1991 - juin 1992	Centre de Mathématiques de l'École Polytechnique Projet de recherche sous la direction de Christophe Margerin : *Étude du théorème de Bernstein et de ses extensions pour les surfaces minimales.*
janv. - fév. 1991	Institut d'Électronique Fondamentale, Université d'Orsay Stage sous la direction de Claude Chappert : *Modélisation informatique de la croissance de couches minces créées par épitaxie par jets moléculaires.*

Enseignements

- *Cours avancé de programmation de logiciels cryptographiques*, Master Pro 2 « Sécurité de l'Information », Université de Limoges, 12 heures, année 2004-2005.
- *Programmation en C pour le codage et la cryptographie*, DEA « Cryptographie, Calcul, Codage » et DESS « Sécurité de l'Information » de l'Université de Limoges, 30 heures annuelles chacun, de 1999 à 2005. Support de cours de 113 pages disponible sur `http://www-rocq.inria.fr/codes/Anne.Canteaut/COURS_C/`.
- *Cryptographie à clef secrète*, DEA « Cryptographie, Calcul, Codage », Université de Limoges, 10 heures annuelles de 1997 à 2000.
- *Travaux dirigés d'algorithmique et de programmation*, ENSTA (2^e et 3^e année), de 1994 à 1997.
- *Cours de cryptographie*, ENSTA (3e année), 6 heures, année 1994-95.
- *Cours d'UNIX et de C*, ENSTA (Mastère), année 1995-96.
- Diverses formations à la cryptographie (OSSIR, INRIA, association FNET...)

Encadrement d'activités de recherche

Doctorants.

- Marion Videau, *Critères de sécurité des algorithmes de chiffrement à clef secrète*, thèse de l'Université Pierre et Marie Curie, soutenue en novembre 2005, encadrée à 80 % (co-direction avec Pascale Charpin).
- Mathieu Cluzeau, *Reconnaissance de codes*, thèse de l'Ecole Polytechnique, soutenance prévue à l'automne 2006, co-encadrée avec Nicolas Sendrier.

Stagiaires.

- M. Naya, *Cryptanalyse de systèmes de chiffrement à flot*, Master Recherche M2 de l'Université Versailles-St Quentin, avril-sept. 2006.
- D. Heizler, *Etude des propriétés cryptographiques des T-fonctions*, Maîtrise Ingénierie Mathématiques, Université de Cergy-Pontoise, avril-juillet 2005 ;
- R. Triki, *Application de techniques de décodage à la cryptanalyse de systèmes de chiffrement à flot*, DEA Algorithmique, Univ. Paris 6, mars-juillet 2004 (co-direction avec J.P. Tillich) ;
- M. Cluzeau, *Reconstruction d'un brasseur linéaire*, DEA « Cryptographie, Calcul, Codage », Université de Limoges, mars-juillet 2003 (co-direction avec N. Sendrier) ;
- P. Quanty, *Etude d'une attaque sur les algorithmes de chiffrement à flot*, DEA « Cryptographie, Calcul, Codage », Université de Limoges, mars-juillet 2002.
- M. Videau, *Généralisations de la cryptanalyse différentielle*, DEA « Réseaux et Télécommunications », Université de Toulouse, mars-août 2001 ;
- F. Diop, stage de fin d'études de DUT « Informatique et Génie Informatique », IUT de Villetaneuse, Université Paris 13, avril-juillet 2000 ;
- M. Trabbia, stage d'option de l'Ecole Polytechnique, avril-juin 1999 ;
- V. Bétis et L. Brunnengreber, stage de fin d'études de l'Ecole Spéciale Militaire de St-Cyr, avril-juin 1998 ;
- M. Portmann et M. Rennhard, stage d'ingénieur (Studienarbeit), ETH Zürich, mars-juin 1997.

Activités éditoriales

- Editrice associée de la revue *IEEE Transactions on Information Theory*, pour la cryptographie et la complexité, de septembre 2005 à septembre 2008.
- Présidente du comité de programme du colloque *SASC 2006*, Leuven, Belgique, février 2006 `http://www.ecrypt.eu.org/stvl/sasc2006/`.
- Présidente du comité de programme du colloque *Indocrypt 2004*, Chennai, Inde, décembre 2004 (181 soumissions), éditions des actes dans la série *Lecture Notes in Computer Science*.
- Co-présidente du comité d'organisation du colloque *WCC 2007*, Rocquencourt, avril 2007.

Comités de programme

- EUROCRYPT 2007 (Barcelone, Espagne) ;
- ISIT 2007 (IEEE International Symposium on Information Theory), Nice ;
- Fast Software Encryption (FSE): 2003 (Lund, Suède), 2005 (Paris), 2006 (Graz, Autriche) ;
- CRYPTO 2004 (Santa Barbara, USA) ;
- Indocrypt: 2003 (New Delhi, Inde), 2004 (Chennai, Inde), 2006 (Calcutta, Inde) ;
- SETA 2006 (International conference on sequences and their applications), Bejing, Chine ;
- Workshop on Coding Theory and Cryptography (WCC): 1999 (Paris), 2001 (Paris);
- ITW 2003 (2003 IEEE Information Theory Workshop, Paris, France) ;
- YACC (Yet Another Conference on Cryptography, Porquerolles, France) : 2000, 2002 ;
- WEWoRC 2005 (Western European Workshop on Research in Cryptology, Leuven, Belgique) ;
- SKEW (Symmetric Key Encryption Workshop), Aarhus, Danemark, 2005;
- Joint BeNeLuxFra Conference in Mathematics (joint meeting of the Belgian, Dutch, Luxembourg and French mathematical societies), Gand, Belgique, 2005.
- membre du comité de pilotage du projet eSTREAM `http://www.ecrypt.eu.org/stream/`.
- membre du comité d'évaluation de l'appel à projets SetIn (Sécurité et Informatique) de l'Agence Nationale de la Recherche.

Jurys de thèse

- Sylvie Dubuc, Université de Caen, 2001 ;
- Marine Minier, Université de Limoges, 2002 ;
- Emmanuel Prouff, Université de Caen, 2004 ;
- Sabine Leveiller, ENST Paris, 2004 (rapporteure) ;
- Carmen Nedeloaia, Université de Limoges, 2005 ;
- Laurent Poinsot, Université de Toulon, 2005 (présidente) ;
- Marion Videau, Université Paris 6, 2005 (directrice) ;
- Havard Molland, Université de Bergen, Norvège (opponent) ;
- An Braeken, Université de Leuven, Belgique, 2006 (rapporteure) ;
- Yi Lu, EPFL, Suisse, 2006, (rapporteure).

Activités contractuelles

ECRYPT – Réseau d'excellence européen (fév. 2004 - fév. 2008)
`http://www.ecrypt.eu.org/` — 33 partenaires européens

Je suis responsable de la participation de l'INRIA à ce réseau. Mon action scientifique se situe naturellement dans le cadre du laboratoire virtuel STVL (Symmetric Techniques Virtual Labs). Je suis en charge de l'un de ses trois Workpackages, *STVL3: Strategic Research on Symmetric Cryptography*, dans lequel sont impliqués 19 partenaires. Il est consacré aux problèmes de recherche et aux besoins industriels émergents en cryptographie symétrique ; il porte notamment une attention particulière à la conception de systèmes dédiés aux applications à faibles ressources. Dans ce cadre, je suis éditrice d'un rapport intitulé *Open Research Areas in Symmetric Cryptography and Technical Trends in Lightweight Cryptography*. Je participe également au comité d'évaluation des chiffrements à flot proposés au projet eSTREAM.

X-Crypt – projet exploratoire RNRT (début 2004-début 2007) *Outils cryptographiques adaptés aux réseaux de télécommunications à haut débit et aux réseaux sans fil émergents, combinant caractéristiques de sécurité avancées et faible consommation de ressources.*

Partenaires : Axalto, ENS Ulm, France Telecom, Cryptolog International, Université de Versailles, INRIA.

Je suis responsable d'un des 4 sous-projets de X-Crypt, celui sur le chiffrement à flot. L'action marquante de ce sous-projet a été la conception de deux nouveaux algorithmes de chiffrement à flot, Sosemanuk et Decim.

Asphales – Action Concertée Incitative « Sécurité Informatique » (sept. 2004–sept. 2007) *Interactions entre sécurité informatique et sécurité juridique dans les chantiers normatifs de la Société de l'information* `http://www.asphales.cnrs.fr/`

Partenaires : CNRS (labo. CECOJI), Univ. Versailles, Univ. Montpellier, INT, Univ. Lille 2, INRIA.

L'objet de ce projet pluri-disciplinaire est de mettre à plat les textes juridiques liés à la sécurité par une lecture critique commune avec des juristes. Ce travail a aussi un impact en cryptographie : il nous amène à modifier les fonctionnalités de certains protocoles dans la mesure où les hypothèses définies par la législation diffèrent parfois des hypothèses cryptographiques classiques.

ACSION – Action Concertée Incitative « Sécurité Informatique » (sept. 2004–sept. 2007) *Nouvelles applications du codage correcteur d'erreurs à la sécurité de l'information*

Partenaires : ENST, INRIA.

Ma participation à ce projet est focalisée sur les attaques des algorithmes de chiffrements à flot par des techniques de correction d'erreur, en particulier les attaques par corrélation.

Contrat d'expertise CANAL+ Technologies (sept. 2002 - avril 2003) *Expertise d'un algorithme de chiffrement par blocs*

Action Concertée Incitative « Cryptologie » - CrAC (2000-2003) et CrAC II (2002-2005) J'étais en charge de la partie « cryptographie symétrique » du premier projet,

et coordinatrice du second.

Convention DGA « Reconnaissance d'un schéma de codage » (juil. 2003- juil. 2004) L'objet de cette étude, menée avec N. Sendrier, était de déterminer des informations sur les éléments constitutifs d'une chaîne de communication (code correcteur, brasseur...) à partir d'un message bruité intercepté en bout de chaîne.

Contrat Sté Côte Basque Informatique (2000) J'étais responsable de ce contrat dont l'objet était de concevoir un protocole d'authentification et de signature pour la sécurité des paiements (identification, intégrité des transactions...) au moyen d'un porte-monnaie électronique.

Contrat CNET « Etude de nouveaux turbo-codes en blocs : les codes CORTEX » (1999–2001) Cette étude concernait une nouvelle famille de codes en blocs, appelés codes cortex. Dans ce contrat, j'étais notamment en charge de la partie « Étude pratique et Réalisation logicielle », qui a consisté à développer un logiciel permettant d'étudier les propriétés métriques de ces codes.

Contrat DRET « Étude sur les Codes de Reed et Muller et les codes de Goppa » (1996–1998)

Diffusion de l'information scientifique

- « La cryptographie ». *Techniques Avancées*, 53, 2000, numéro spécial *Sécurité des syst. d'information.*
- « La cryptologie moderne». *L'Armement*, 73:76–83 et 74:139–142, mars et juin 2001, avec F. Lévy-dit-Véhel. `http://www-rocq.inria.fr/codes/Anne.Canteaut/crypto_moderne.pdf`.
- « Cryptanalyse de chiffrement par blocs ». *MISC - Le magazine de la sécurité informatique*, 2, 2002.
- « Le chiffrement à la volée ». *Pour la Science*, numéro spécial *cryptographie, l'art du secret*, 2002.
- Rédaction de la partie *générateurs pseudo-aléatoires et le chiffrement à flot* du portail Internet Cryptologie et Sécurité de l'Information, `http://www.picsi.org/`.
- Membre du comité de rédaction du site Interstices `http://interstices.info/` en 2003-2004.
- Interventions dans plusieurs lycées de l'Académie de Versailles sur la cryptographie.

Collaborations inter-disciplinaires

Collaborations avec des industriels et des juristes sur la définition des cadres normatif et législatif de la sécurité de l'information :

- participation au groupe *Conservation électronique des documents* du Forum des Droits sur l'Internet et de la Mission pour l'Economie Numérique (2003-2005) et à la rédaction du rapport de recommandation sur ce sujet ;

- intervention au colloque *Les actes authentiques électroniques*, au Sénat (mai 2002) ;
- participation au groupe *Signature Electronique* du club Cards, Systems and Applications et au *Livre blanc de la signature électronique* (2001).

Liste de publications

Livres et chapitres de livres

[1] A. Canteaut et K. Viswanathan, éditeurs. *Progress in Cryptology - INDOCRYPT 2004*, volume 3348 de *Lecture Notes in Computer Science*. Springer-Verlag, 2004.

[2] A. Canteaut. Articles : A5/1, Berlekamp-Massey algorithm, Combination generator, Correlation attack, Fast correlation attack, Filter generator, Inversion attack, Linear complexity, Linear consistency attack, Linear cryptanalysis for stream ciphers, Linear feedback shift register, Linear syndrome attack, Minimal polynomial, Running-key, Stream cipher. Dans H.C.A. van Tilborg, éditeur, *Encyclopedia of cryptography and security*. Springer, 2005.

Articles dans des revues internationales

[1] A. Canteaut et F. Chabaud. « A new algorithm for finding minimum-weight words in a linear code: application to primitive narrow-sense BCH codes of length 511 ». *IEEE Transactions on Information Theory*, 44(1):367–378, janvier 1998.

[2] P. Camion et A. Canteaut. « Correlation-immune and resilient functions over a finite alphabet and their applications in cryptography ». *Designs, Codes and Cryptography*, 16(2):121–149, février 1999.

[3] A. Canteaut, P. Charpin et H. Dobbertin. « Binary m-sequences with three-valued crosscorrelation: A proof of Welch conjecture ». *IEEE Transactions on Information Theory*, 46(1):4–8, janvier 2000.

[4] A. Canteaut, P. Charpin et H. Dobbertin. « Weight divisibility of cyclic codes, highly nonlinear functions on GF(2^m) and crosscorrelation of maximum-length sequences ». *SIAM Journal on Discrete Mathematics*, 13(1):105–138, 2000.

[5] A. Canteaut. « On the weight distributions of optimal cosets of the First-Order Reed-Muller Code ». *IEEE Transactions on Information Theory*, 47(1):407–413, janvier 2001.

[6] 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, mai 2001.

[7] A. Canteaut et P. Charpin. « Decomposing Bent Functions ». *IEEE Transactions on Information Theory*, 49(8):2004–19, août 2003.

[8] A. Canteaut et M. Videau. « Symmetric Boolean functions ». *IEEE Transactions on Information Theory*, 51(8):2791–2811, août 2005.

[9] A. Canteaut, M. Daum, G. Leander et H. Dobbertin. « Normal and non normal bent functions ». *Discrete Applied Mathematics*, 154(2):202–218, février 2006. Special issue in Coding and Cryptology.

[10] H. Dobbertin, G. Leander, A. Canteaut, C. Carlet, P. Felke et P. Gaborit. « Construction of bent functions via Niho power functions ». *Journal of Combinatorial Theory, Series A*, 2006. A paraître.

[11] T.P. Berger, A. Canteaut, P. Charpin et Y. Laigle-Chapuy. « On Almost Perfect Nonlinear functions ». *IEEE Transactions on Information Theory*, 2006. A paraître.

Comptes-rendus de l'Académie des Sciences

[1] A. Canteaut, P. Charpin et H. Dobbertin. « Couples de suites binaires de longueur maximale ayant une corrélation croisée à trois valeurs: conjecture de Welch ». *Comptes Rendus de l'Académie des Sciences, Série I*, 328(2):173–178, 1999.

Articles longs parus dans des actes de conférences internationales[f]

[1] A. Canteaut. « A new algorithm for finding minimum-weight words in large linear codes ». Dans *Cryptography and Coding - 5th IMA Conference*, volume 1025 de *Lecture Notes in Computer Science*, pages 205–212. Springer-Verlag, 1995.

[2] P. Camion et A. Canteaut. « Construction of t-resilient functions over a finite alphabet ». Dans *Advances in Cryptology - EUROCRYPT'96*, volume 1070 de *Lecture Notes in Computer Science*, pages 283–293. Springer-Verlag, 1996.

[3] P. Camion et A. Canteaut. « Generalization of Siegenthaler inequality and Schnorr-Vaudenay multipermutations ». Dans *Advances in Cryptology - CRYPTO'96*, volume 1109 de *Lecture Notes in Computer Science*, pages 372–386. Springer-Verlag, 1996.

[4] A. Canteaut et N. Sendrier. « Cryptanalysis of the original McEliece cryptosystem ». Dans *Advances in Cryptology - ASIACRYPT'98*, volume 1514 de *Lecture Notes in Computer Science*, pages 187–199. Springer-Verlag, 1998.

[5] A. Canteaut, P. Charpin et H. Dobbertin. « A new characterization of almost bent functions ». Dans *Fast Software Encryption - FSE'99*, volume 1636 de *Lecture Notes in Computer Science*, pages 186–200. Springer-Verlag, 1999.

[6] A. Canteaut, C. Carlet, P. Charpin et C. Fontaine. « Propagation characteristics and correlation-immunity of highly nonlinear Boolean functions ». Dans *Advances in Cryptology - EUROCRYPT'2000*, volume 1807 de *Lecture Notes in Computer Science*, pages 507–522. Springer-Verlag, 2000.

[7] A. Canteaut et M. Trabbia. « Improved fast correlation attacks using parity-check equations of weight 4 and 5 ». Dans *Advances in Cryptology - EUROCRYPT'2000*, volume 1807 de *Lecture Notes in Computer Science*, pages 573–588. Springer-Verlag, 2000.

[8] A. Canteaut. « Cryptographic Functions and Design Criteria for Block Ciphers ». Dans *Progress in Cryptology - INDOCRYPT 2001*, volume 2247 de *Lecture Notes in Computer Science*, pages 1–16. Springer-Verlag, 2001.

f. Il s'agit ici d'*articles longs* sélectionnés par un comité de lecture, sur la base de deux rapports d'experts. Les conférences correspondantes ne sont pas mentionnées dans la rubrique « Conférences internationales », à l'exception de certaines conférences invitées.

[9] A. Canteaut et É. Filiol. « Ciphertext only reconstruction of stream ciphers based on combination generators ». Dans *Fast Software Encryption - FSE 2000*, volume 1978 de *Lecture Notes in Computer Science*, pages 165–180. Springer-Verlag, 2001.

[10] A. Canteaut, P. Charpin et M. Videau. « Cryptanalysis of Block Ciphers and Weight Divisibility of Some Binary Codes ». Dans *Information, Coding and Mathematics (Workshop in honor of Bob McEliece's 60th birthday)*, pages 75–97. Kluwer, 2002.

[11] A. Canteaut et M. Videau. « Degree of composition of highly nonlinear functions and applications to higher order differential cryptanalysis ». Dans *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 de *Lecture Notes in Computer Science*, pages 518–533. Springer-Verlag, 2002.

[12] A. Canteaut. « On the correlations between a combining function and functions of fewer variables ». Dans *Proceedings of the 2002 IEEE Information Theory Workshop - ITW 2002*, pages 78–81, Bangalore, Inde, octobre 2002. IEEE Press.

[13] A. Canteaut. « Open problems related to algebraic attacks on stream ciphers ». Dans *Workshop on Coding and Cryptography - WCC 2005*, Lecture Notes in Computer Science, 2006. A paraître.

[14] A. Canteaut. « Fast Correlation Attacks Against Stream Ciphers and Related Open Problems ». Dans *Proceedings of the 2005 IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security - ITW 2005*, pages 49–54. IEEE Press, 2005.

Conférences invitées

[1] A. Canteaut. « Design and analysis of secret-key ciphers ». Dans *Colloque inter-académique "Network Sciences" (Académie des Sciences - Chinese Academy of Sciences)*, Pékin, Chine, octobre 2000.

[2] A. Canteaut. « Cryptographic Functions and Design Criteria for Block Ciphers ». Dans *INDOCRYPT 2001*, Chennai, India, 2001.

[3] A. Canteaut. « Cryptanalysis of Block Ciphers and Weight Divisibility of Some Binary Codes ». Dans *Workshop in honor of Bob McEliece's 60th birthday*, Pasadena, USA, mai 2002.

[4] A. Canteaut. « Cryptanalysis of Block Ciphers and Related Properties of the Walsh Spectra of S-boxes ». Dans *Yet Another Conference on Cryptography - YACC 2002*, Porquerolles, France, juin 2002.

[5] A. Canteaut. « On the correlations between a combining function and functions of fewer variables ». Dans *2002 IEEE Information Theory Workshop - ITW 2002*, Bangalore, Inde, octobre 2002.

[6] A. Canteaut. « Decoding techniques for correlation attacks on stream ciphers ». Dans *Yet Another Conference on Cryptography - YACC 2004*, Porquerolles, France, juin 2004.

[7] A. Canteaut. « Decoding techniques for correlation attacks on stream ciphers ». Dans *Academy Contact Forum "Coding theory and cryptography"*, The royal Flemish academy of Belgium for science and the arts, Bruxelles, Belgique, octobre 2005. `http://cage.rug.ac.be/~ls/website/contactforum2005.html`.

[8] A. Canteaut. « Fast Correlation Attacks Against Stream Ciphers and Related Open Problems ». Dans *2005 IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security - ITW 2005*, Awaji Island, Japon, octobre 2005.

[9] A. Canteaut. « Open problems related to algebraic attacks on stream ciphers ». Dans *Workshop on Coding and Cryptography - WCC 2005*, pages 1–11, Bergen, Norvège, mars 2005.

Conférences internationales (avec comité de lecture)

[1] A. Canteaut et H. Chabanne. « A further improvement of the work factor in an attempt at breaking McEliece's cryptosystem ». Dans *Proceedings of EUROCODE 94*, La Buissière-sur-Ouche, France, octobre 1994. INRIA.

[2] P. Camion et A. Canteaut. « Characterizations of Correlation-Immune and Resilient functions over any Alphabet ». Dans *Workshop in Cryptography*, Luminy, France, 1995.

[3] A. Canteaut. « True minimum distance of some narrow-sense BCH codes of length 511 ». Dans *Second Mediterranean Workshop on Coding and Information Integrity*, Palma-de-Majorque, Espagne, février 1996.

[4] A. Canteaut et F. Chabaud. « A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to BCH codes of length 511 ». Dans *Proceedings of the 1997 IEEE International Symposium on Information Theory - ISIT'97*, page 327, Ulm, Germany, 1997. IEEE Press.

[5] A. Canteaut. « Differential cryptanalysis of Feistel ciphers and differentially uniform mappings ». Dans *Selected Areas on Cryptography, SAC'97*, pages 172–184, Ottawa, Canada, 1997.

[6] A. Canteaut. « Differential cryptanalysis of Feistel ciphers ». Dans *Third Mediterranean Workshop on Coding and Information Integrity*, Ein Boqeq, Israel, octobre 1997.

[7] A. Canteaut. « On the Hypothesis of stochastic equivalence for Feistel ciphers ». Dans *Proceedings of the 1998 IEEE International Symposium on Information Theory - ISIT'98*, page 81, Boston, USA, août 1998. IEEE Press.

[8] A. Canteaut, C. Carlet, P. Charpin et C. Fontaine. « Fourier Spectrum of Optimal Boolean Functions via Kasami's Identities ». Dans *Proceedings of the 2000 IEEE International Symposium on Information Theory - ISIT'00*, page 183, Sorrente, Italie, juin 2000. IEEE Press.

[9] A. Canteaut et M. Trabbia. « Compared performance of fast correlation attacks on stream ciphers ». Dans *Proceedings of the 2000 IEEE International Symposium on Information Theory - ISIT'00*, page 213, Sorrente, Italie, juin 2000. IEEE Press.

[10] A. Canteaut et P. Charpin. « Decomposing bent functions ». Dans *Workshop on Coding Theory and Data Integrity*, IMS, National University of Singapore, septembre 2001.

[11] A. Canteaut et E. Filiol. « On the Influence of the Filtering Function on the Performance of Fast Correlation Attacks on Filter Generators ». Dans *23rd Symposium on Information Theory in the Benelux*, Louvain-la-Neuve, Belgium, mai 2002.

[12] A. Canteaut et P. Charpin. « Decomposing bent functions ». Dans *Proceedings of the 2002 IEEE International Symposium on Information Theory - ISIT'02*, Lausanne, Suisse, juillet 2002. IEEE Press.

[13] A. Canteaut et M. Videau. « Higher order differential attacks on iterated block ciphers using almost bent round functions ». Dans *Proceedings of the 2002 IEEE International Symposium on Information Theory - ISIT'02*, Lausanne, Suisse, juillet 2002. IEEE Press.

[14] A. Canteaut. « Design criteria for symmetric primitives - position paper ». Dans *STORK Cryptography Workshop - Towards a Roadmap for Future Research*, pages 44–45, Bruges, Belgique, novembre 2002.

[15] A. Canteaut, M. Daum, G. Leander et H. Dobbertin. « Normal and Non Normal Bent Functions ». Dans *Workshop on Coding and Cryptography - WCC 2003*, pages 91–100, Versailles, France, mars 2003.

[16] 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 software-oriented stream cipher ». Dans *Proceedings of SKEW - Symmetric Key Encryption Workshop, Network of Excellence in Cryptology ECRYPT*, Aarhus, Danemark, mai 2005.

[17] 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 ». Dans *Proceedings of SKEW - Symmetric Key Encryption Workshop, Network of Excellence in Cryptology ECRYPT*, Aarhus, Danemark, mai 2005.

[18] T.P. Berger, A. Canteaut, P. Charpin et Y. Laigle-Chapuy. « On Almost Perfect Nonlinear mappings ». Dans *Proceedings of the 2005 IEEE International Symposium on Information Theory - ISIT'05*, Adelaide, Australie, septembre 2005.

[19] 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. « DecimV2 ». Dans *Proceedings of SASC 2006 - ECRYPT Workshop on stream ciphers*, Leuven, Belgique, février 2006.

[20] A. Canteaut, P. Charpin et G. Kyureghyan. « A new class of monomial bent functions ». Dans *Proceedings of the 2006 IEEE International Symposium on Information Theory - ISIT'06*, Seattle, USA, juillet 2006.

Conférences nationales

[1] A. Canteaut. « Fonctions booléennes et cryptographie ». Dans *27e École de printemps d'informatique théorique, Codage et cryptographie*, Batz-sur-Mer, France, juin 1999.

[2] A. Canteaut. « Comment concevoir un chiffrement à clef secrète par blocs: exemple de l'AES ». Dans *Journée cryptographique de Limoges*, Limoges, décembre 2000.

[3] A. Canteaut. « Critères de conception des systèmes de chiffrement à clef secrète ». Dans *Conférence internationale sur les mathématiques dans l'industrie et les services*, École polytechnique, novembre 2000.

[4] A. Canteaut. « Codes correcteurs et cryptographie à clef secrète (tutoriel) ». Dans *École Jeunes Chercheurs Algorithmique et Calcul Formel - EJC 2000*, Caen, mars 2000.

[5] A. Canteaut. « Le chiffrement à flot (tutoriel) ». Dans *Ecole de Jeunes Chercheurs en Algorithmique et Calcul Formel - EJC 2005*, Montpellier, avril 2005.

Spécifications d'algorithmes

[1] 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 software-oriented stream cipher ». Soumission au projet européen eSTREAM, en réponse à *Call for Stream Cipher Primitives, Network of Excellence in Cryptology ECRYPT*, 2005. `http://www.ecrypt.eu.org/stream/`.

[2] 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 ». Soumission au projet européen eSTREAM, en réponse à *Call for Stream Cipher Primitives, Network of Excellence in Cryptology ECRYPT*, 2005. `http://www.ecrypt.eu.org/stream/`.

Rapports de contrat et d'expertise

[1] A. Canteaut. « Expertise d'un protocole d'authentification et de signature par le biais d'un système informatique ». Rapport de contrat "Société Côte Basque Informatique, juin 2000. 121 pages.

[2] T. Berger, E. Cadic, A. Canteaut, J.-C. Carlach, P. Charpin, P. Gaborit, G. Olocco, A. Otmani et J.-P. Tillich. « Nouveaux turbo-codes en bloc : les codes CORTEX - Etude et mise en œuvre ». Rapport de contrat CNET-INRIA-LACO-LRI, février 2001.

[3] A. Canteaut. « Rapport d'expertise de l'algorithme AAC ». Rapport de contrat CANAL+ Technologies, avril 2003. en collaboration avec D. Augot, 244 pages.

[4] M. Minier, A. Canteaut, N. Courtois et H. Gilbert. « Chiffrement à flot - Etat de l'art ». Rapport du projet RNRT X-CRYPT, février 2005. 44 pages.

[5] A. Canteaut (ed.), D. Augot, A. Biryukov, A. Braeken, C. Cid, H. Dobbertin, H. Englund, H. Gilbert, L. Granboulan, H. Handschuh, M. Hell, T. Johansson, A. Maximov, M. Parker, T. Pornin, B. Preneel, M. Robshaw et M. Ward. « D.STVL.3 – Open Research Areas in Symmetric Cryptography and Technical Trends in Lightweight Cryptography ». Rapport du réseau d'excellence européen ECRYPT, 2005. 82 pages.

[6] A. Canteaut (ed.), D. Augot, A. Biryukov, A. Braeken, C. Cid, H. Dobbertin, H. Englund, H. Gilbert, L. Granboulan, H. Handschuh, M. Hell, T. Johansson, A. Maximov, M. Parker, T. Pornin, B. Preneel, M. Robshaw et M. Ward. « D.STVL.4 – Open Research Areas in Symmetric Cryptography and Technical Trends in Lightweight Cryptography ». Rapport du réseau d'excellence européen ECRYPT, 2006. 88 pages.

[7] D. Augot, A. Biryukov, A. Canteaut, C. Cid, N. Courtois, C. De Cannière, H. Gilbert C. Lauradoux, M. Parker, B. Preneel, M. Robshaw et Y. Seurin. « D.STVL.2 – AES Security Report ». Rapport du réseau d'excellence européen ECRYPT, 2006. 73 pages.

[8] Forum des Droits sur l'Internet. « La conservation électronique des documents ». `http://www.foruminternet.org/`, décembre 2005.

Rapports de recherche

[1] A. Canteaut et H. Chabanne. « A further improvement of the work factor in an attempt at breaking McEliece's cryptosystem ». Rapport de recherche RR-2227, INRIA, mars 1994. `http://www.inria.fr/rrrt/rr-2227.html`.

[2] A. Canteaut et F. Chabaud. « Improvements of the attacks on cryptosystems based on error-correcting codes ». Rapport de Recherche LIENS-95-21, Ecole Normale Supérieure, July 1995.

[3] A. Canteaut et F. Chabaud. « A new algorithm for finding minimum-weight words in a linear code: application to primitive narrow-sense BCH codes of length 511 ». Rapport de recherche RR-2685, INRIA, octobre 1995. `http://www.inria.fr/rrrt/rr-2685.html`.

[4] P. Camion et A. Canteaut. « Construction of t-resilient functions over a finite alphabet ». Rapport de recherche RR-2789, INRIA, février 1996. `http://www.inria.fr/rrrt/rr-2789.html`.

[5] A. Canteaut et É. Filiol. « Ciphertext only reconstruction of stream ciphers based on combination generators ». Rapport de recherche RR-3887, INRIA, février 2000. http://www.inria.fr/rrrt/rr-3887.html.

[6] A. Canteaut et M. Videau. « Weakness of Block Ciphers Using Highly Nonlinear Confusion Functions ». Rapport de recherche RR-4367, INRIA, février 2002. `http://www.inria.fr/rrrt/rr-4367.html`.

[7] T.P. Berger, A. Canteaut, P. Charpin et Y. Laigle-Chapuy. « On Almost Perfect Nonlinear functions ». Rapport de recherche RR-5774, INRIA, décembre 2005. `http://www.inria.fr/rrrt/rr-5774.html`.

[8] A. Canteaut, C. Lauradux et A. Seznec. « Understanding cache attacks ». Rapport de recherche RR-5881, INRIA, avril 2006. `http://www.inria.fr/rrrt/rr-5881.html`.

Supports de cours

[1] A. Canteaut. « Programmation en langage C ». Support de cours de DEA et DESS. `http://www-rocq.inria.fr/codes/Anne.Canteaut/COURS_C/`, 133 pages.

Articles de vulgarisation

[1] A. Canteaut. « La cryptographie ». *Techniques Avancées*, 53, septembre 2000. Numéro spécial sur "La sécurité des systèmes d'information".

[2] A. Canteaut et F. Lévy-dit-Véhel. « La cryptologie moderne (1) ». *L'Armement*, 73:76–83, mars 2001. `http://www-rocq.inria.fr/codes/Anne.Canteaut/crypto_moderne.pdf`.

[3] A. Canteaut et F. Lévy-dit-Véhel. « La cryptologie moderne (2) ». *L'Armement*, 74:139–142, juin 2001.

[4] A. Canteaut. « Cryptanalyse de chiffrement à clef secrète par blocs ». *MISC - Le magazine de la sécurité informatique*, 2, mars 2002.

[5] A. Canteaut. « Le chiffrement à la volée ». *Pour la Science*, pages 86–87, juillet 2002. Numéro spécial *La cryptographie, l'art du secret.*

[6] A. Canteaut. « Parcours et fiches sur les générateurs pseudo-aléatoires et le chiffrement à flot ». portail Internet Cryptologie et Sécurité de l'Information, 2006. `http://www.picsi.org/`.

Exposés de vulgarisation

[1] A. Canteaut. « Mathématiques discrètes et cryptographie ». Dans *Journées de la Culture Mathématique*, Cergy-Pontoise, janvier 1999. Conférence organisée par l'Académie de Versailles à destination des professeurs de mathématiques de l'enseignement secondaire.

[2] A. Canteaut. « La cryptographie ou les mathématiques au service de la protection de l'information ». Dans *Ouverture des Olympiades de Mathématiques de l'Académie de Versailles*, Université de Versailles, janvier 2003.

[3] A. Canteaut. « Comment concevoir un algorithme de chiffrement rapide et solide ». Dans *La face cachée des mathématiques*, Paris, mars 2004. Conférence organisée par l'IHES, la Société Mathématique de France, la Société de Mathématiques Appliquées et Industrielles et Pour la Science.

Exposés de formation

[1] D. Augot et A. Canteaut. « Cryptologie pour la sécurité des communications ». Dans *CARI'98 - Journées de formation avancées*, Dakar, Sénégal, octobre 1998.

[2] D. Augot, A. Canteaut et F. Morain. « Cryptosystèmes, algorithmes et longueur de clefs ». Dans *Séminaire X-Aristote - « La sécurité des réseaux »*, École Polytechnique, janvier 2000.

[3] D. Augot, A. Canteaut et F. Morain. « Introduction à la cryptographie ». Dans *Séminaire de l'OSSIR (Observatoire de la Sécurité des Systèmes d'Information et des Réseaux)*, octobre 2000.

[4] A. Canteaut. « Introduction à la cryptographie ». Dans *Séminaire X-Aristote - « Sécurité des transactions »*, octobre 2001.

Index

Notations

Nous donnons ici la liste des principales notations utilisées dans ce document ainsi que la page de leur définition. Les notations spécifiques au chiffrement à flot sont récapitulées page 15.

$\mathbf{F}_q$	corps fini à q éléments
$\|E\|$	cardinal d'un ensemble E
$wt(x)$	poids de Hamming du vecteur binaire x
$w_2(i)$	poids de Hamming de la décomposition binaire de l'entier i
$d_H(x,y)$	distance de Hamming entre x et y
$x \cdot y$	produit scalaire usuel entre x et y
$V^\perp$	orthogonal d'un sous-espace V de $\mathbf{F}_2^n$
$(e_1,\ldots,e_n)$	base canonique de $\mathbf{F}_2^n$
$\langle x_1,\ldots,x_k\rangle$	espace vectoriel engendré par les vecteurs $x_1,\ldots,x_k$
1_I	vecteur de $\mathbf{F}_2^n$ ayant l'ensemble de positions $I \subset \{1,\ldots,n\}$ pour support (page 36)
$\Delta^2(\mathcal{D})$	biais d'une distribution $\mathcal{D}$ proche de la distribution uniforme (page 6)
$\mathcal{B}ool_n$	ensemble des fonctions booléennes à n variables
$\deg(f)$	degré de la fonction booléenne f
$\mathcal{F}(f)$	déséquilibre de la fonction booléenne f (ou coefficient de Walsh en 0 de f) $\mathcal{F}(f) = \sum_{x\in\mathbf{F}_2^n}(-1)^{f(x)}$ (page 7)
F_λ	composante de la fonction vectorielle F : $F_\lambda(x) = \lambda \cdot F(x)$ (page 7)
φ_a	fonction linéaire définie par $x \mapsto a \cdot x$ (page 8)
$\mathcal{F}(f+\varphi_a)$	coefficient de Walsh de f en a (page 8)
$\{\mathcal{F}(f+\varphi_a), a \in \mathbf{F}_2^n\}$	spectre de Walsh de f (page 8)
$\{\mathcal{F}(f+\varphi_a+\varepsilon), a \in \mathbf{F}_2^n, \varepsilon \in \mathbf{F}_2\}$	spectre de Walsh étendu de f (page 8)
$\mathcal{L}(f)$	linéarité de f, *i.e.* amplitude maximale des coefficients de Walsh de f (page 9)
$\mathcal{NL}(f)$	non-linéarité de f, *i.e.* distance de f à l'ensemble des fonctions affines (page 9)
$\nu(f)$	indicateur par somme des carrés de f (page 30)

D_aF	dérivée d'une fonction F définie sur $\mathbf{F}_2^n$ relativement au vecteur $a \in \mathbf{F}_2^n$: $D_aF : x \mapsto F(x) \oplus F(x+a)$ (page 9)
D_Vf	dérivée d'ordre $\dim(V)$ d'une fonction F définie sur $\mathbf{F}_2^n$ relativement au sous-espace vectoriel $V \subset \mathbf{F}_2^n$: $D_VF(x) = \sum_{v \in V} F(x+v)$ (page 111)
F_{a+V}	restriction d'une fonction au sous-espace affine $a+V$: $\forall x \in V$, $F_{a+V}(x) = F(a+x)$ (page 9)
Δ_w	moment normalisé d'ordre w du spectre de Walsh d'une fonction booléenne : $\Delta_w = 2^{-wn} \sum_{\lambda \in \mathbf{F}_2^n} \mathcal{F}^w(f + \varphi_\lambda)$ (page 29)
$AN(f)$	idéal annulateur de la fonction booléenne f (page 58)
$AN_d(f)$	ensemble des annulateurs de degré inférieur ou égal à d de la fonction f (page 58)
$AI(f)$	immunité algébrique de la fonction booléenne f (page 60)
$\delta_F(a,b)$	nombre de solutions x de l'équation $F(x+a)+F(x)=b$ (page 90)
$\delta(F)$	nombre maximum de solutions d'une équation du type $F(x+a)+F(x)=b$, $a \neq 0$ (page 90)
δ_c	nombre de solutions x de l'équation $(x+1)^s + x^s = c$ (page 96)

Liste des tableaux

Table des matières