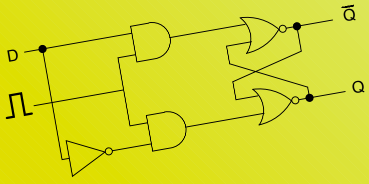
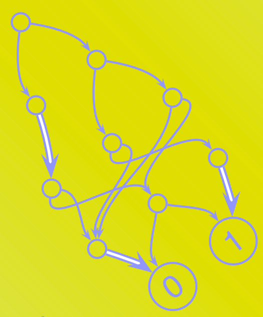
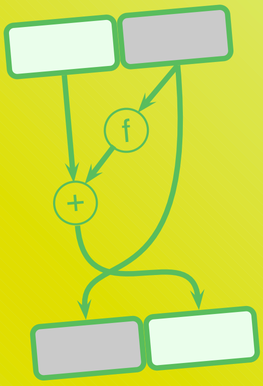
Jean-Francis Michon
Pierre Valarcher
Jean-Baptiste Yunès (Eds.)

Fonct1ons B00léennes: Cryptographie, Appl1cat10ns

Boolean Funct1ons: Crypt0graphy, Appl1cat10ns







$$\left\{\begin{array}{rcl} x_1 + x_2 + x_3 + x_2x_3 & = & y_1 \\ x_2 + x_3 + x_1x_3 & = & y_2 \\ x_2 + x_1x_2 + x_1x_3 & = & y_3 \end{array}\right.$$

$\mathbb{B}^n \longmapsto \mathbb{B}$

10001110100100011101001
01110111110110100010
00011111000111110001111100011111

1st, International Workshop, BFCA'05
Rouen, France, March 2005
Proceedings

Publications des Universités de Rouen et du Havre

BFCA'05

BFCA'05

Boolean Functions: Cryptography and Applications

Edited by

Jean-Francis Michon, Pierre Valarcher and Jean-Baptiste Yunès

Proceedings of the conference organised at the Université de Rouen (march 8-9 2005) by the

Laboratoire d'Informatique Fondamentale et Applications de Rouen (LIFAR)

Publications des Universités de Rouen et du Havre

Contents

PREFACE

Jean-Francis Michon[1], Pierre Valarcher[1] and
Jean-Baptiste Yunès[2]

The Meeting

The "Boolean Functions: Cryptography and Applications" international meeting took place on March 7-8th, 2005, in Rouen, France. It was the first of an expected series in the field of Boolean functions. BFCA'05 was organized by the LIFAR, University of Rouen and the LIAFA, University Denis Diderot of Paris.

The main purpose of the conference was to create contacts between many different scientists working on Boolean functions, and that goal was reached far beyond our expectations. More than 40 participants came from as many as 9 different countries. Authors submitted 20 papers all reviewed by two competent referees, who finally selected 13 of them.

L'Atelier

L'atelier international "Fonctions Booléennes: Cryptographie et Applications" s'est tenu les 7 et 8 mars 2005, à l'Université de Rouen (France). Il s'agissait de la première d'une future série sur le thème des fonctions Booléennes. BFCA'05 a été organisé

[1] Université de Rouen, LIFAR, 75821 Mont Saint Aignan Cedex, France. email: {jean.francis.michon,pierre.valarcher}@univ-rouen.fr
[2] LIAFA - Université Denis Diderot - Paris 7. 175 rue Chevaleret, F-75013 Paris, France. email: Jean-Baptiste.Yunes@liafa.jussieu.fr

conjointement par le LIFAR de l'Université de Rouen et le LIAFA de l'Université Denis Diderot de Paris.

Le but premier de cette conférence était de faire se rencontrer de nombreux chercheurs travaillant sur les fonctions Booléennes et nous pouvons affirmer que cela a été réussi bien au-delà de nos espérances. Nous avons reçu plus de 40 participants venus de 9 pays différents. Les auteurs ont soumis 20 articles tous examinés par deux juges compétents pour n'en retenir que 13.

Thanks/Remerciements

Many thanks to our sponsors:
Un grand merci à nos sponsors:

Le LIFAR
L'Université de Rouen
Le Conseil Régional de Seine-Maritime
Le Ministère délégué à la recherche - programme
"ACI - Cryptologie"
GDR-ALP

Organizing committee/Comité d'organisation

Jean-Francis Michon (Univ. de Rouen, LIFAR)
Pierre Valarcher (Univ. de Rouen, LIFAR)
Jean-Baptiste Yunès (Univ. Paris 7, LIAFA)
Secrétaire: Angélique Daniellou (LIFAR)

Program committee/Comité de programme

Claude Carlet (Univ. Paris 8, INRIA)
Hervé Chabanne (SAGEM)
Pascale Charpin (INRIA)
Jean-Charles Faugère (LIP6, CNRS)
Louis Granboulan (GRECC, ENS)
Étienne Grandjean (Univ. de Caen, GREYC)
Jean-Francis Michon (Univ. de Rouen, LIFAR)
Pierre Valarcher (Univ. de Rouen, LIFAR)

Jean-Baptiste Yunès (Univ. Paris 7, LIAFA)

Support committee/Comité de soutien

Jean-Éric Pin (LIAFA, CNRS)

Referees/Examinateurs

Ali Akhavi
Julien Bringer
Claude Carlet
Hervé Chabanne
Pascale Charpin
Nadia Creignou
Emmanuelle Dottax
Jean-Charles Faugère
Aline Gouget
Louis Grandboulan
Étienne Grandjean
Jean-Marie Le Bars
Jean-Francis Michon
Pierre Valarcher
Jean-Baptiste Yunès

BFCA on the WEB/BFCA sur Internet

`http://www.univ-rouen.fr/LIFAR/bfca/`

Special Thanks/Remerciements particuliers

The organizing committee is thankful to M. Jérôme Segal who accepted to come from Austria and who has talked about his research in the history of sciences. He gave us a good and interesting lecture for the full satisfaction of the participants.

Le comité d'organisation souhaite remercier M. Jérôme Segal qui a accepté de venir d'Autriche nous parler de son travail d'historien des sciences. Sa prestation fut de qualité et riche d'enseignements au grand contentement du public présent.

Jérôme Segal. *Le zéro et le un. Histoire de la Notion Scientifique d'Information au 20e siècle.* 2003. Éditions Syllepse.
ISBN: `2-84797046-0`.

May (Mai), 2005

ON THE CONSTRUCTION OF BALANCED BOOLEAN FUNCTIONS WITH A GOOD ALGEBRAIC IMMUNITY

C. Carlet[1] and Ph. Gaborit[2]

Abstract. In this paper we study the algebraic immunity of Boolean functions and consider in particular the problem of constructing Boolean functions with a good algebraic immunity. We first give heuristic arguments to prove that the algebraic immunity of a random Boolean function on n variables is at least $\lfloor \frac{n}{2} \rfloor$ with a very high probability (while the upper bound is $\lceil \frac{n}{2} \rceil$, the "ceiling" of $\frac{n}{2}$). We give an upper bound, under a reasonable assumption, on the algebraic immunity of Boolean functions constructed through Maiorana-MacFarland construction. We give a construction which strictly increases the algebraic immunity of a Boolean function by adding a certain number of new variables and deduce the first infinite family of functions with a non trivial proven algebraic immunity. At last we give examples of balanced functions with optimal algebraic immunity and a good nonlinearity and of balanced functions with a good algebraic immunity, a good nonlinearity and a good correlation immunity, which can be used for cryptographic purposes.

1. **Introduction**

Boolean ($\{0,1\}$-valued) functions on the set ${F_2}^n$ of binary vectors of a given length n, are used in the pseudo-random generators of stream ciphers and play a central role in their security. The

[1] INRIA, Domaine de Voluceau, Rocquencourt, BP 105 - 78153, Le Chesnay Cedex, FRANCE. email: `claude.carletinria.fr` also member of the University of Paris 8.

[2] LACO, Université de Limoges; 123, av. A. Thomas, 87060 Limoges, France. email: `gaborit@unilim.fr`

generation of the keystream consists, in many stream ciphers, of a linear part, producing a sequence with a large period, usually composed of one or several LFSR's, and a nonlinear combining or filtering function f that produces the output, given the state of the linear part. The main classical cryptographic criteria for designing such function f are balancedness (f is balanced if its Hamming weight equals 2^{n-1}) to prevent the system from leaking statistical information on the plaintext when the ciphertext is known, a high algebraic degree (that is, a high degree of the algebraic normal form of the function) to counter linear synthesis by Berlekamp-Massey algorithm, a high order of correlation immunity (in fact, of resiliency, since the functions must be balanced) to counter correlation attacks (at least in the case of combining functions), and a high nonlinearity (that is, a large Hamming distance to affine functions) to withstand correlation attacks (again) and linear attacks.

Since the introduction of these criteria, the problem of efficiently constructing highly resilient functions with high nonlinearities and algebraic degrees has received much attention. Few constructions are known, giving functions on a sufficient number of variables, achieving or approaching the best possible cryptographic characteristics.

The recent algebraic attacks have dramatically complicated this situation. Algebraic attacks recover the secret key by solving an overdefined system of multivariate algebraic equations. The scenarios found in [9], under which low degree equations can be deduced from the knowledge of the nonlinear combining or filtering function, have led in [15] to a new parameter of the Boolean function: its algebraic immunity, which must be high.

No construction is known, leading to functions whose algebraic immunity can be lower bounded. For instance, the 10-variable Boolean function used in the LILI keystream generator (a submission to NESSIE European call for cryptographic primitives) is built following [19] by using classical constructions. It has algebraic immunity 4 and is responsible for the lack of resistance of LILI to algebraic attacks, as shown in [9].

It is shown in [15] that taking random balanced functions on sufficiently large numbers of variables can suffice to withstand algebraic attacks on the stream ciphers using them. As shown in [17], it would also permit to reach nonlinearities which would not be too far from the optimal ones. But the result of [15] asserts that random n-variable functions have sufficient algebraic immunity for values of n which make the pseudo-random generator slow.

In this paper, after recalling some background in Section 2, we give in Section 3 heuristic arguments showing that the algebraic immunity of a random Boolean function on n variables is almost surely at least $\lfloor \frac{n}{2} \rfloor$; hence, it is almost surely equal to $\frac{n}{2}$ when n is even and it belongs almost surely to the pair $\{\frac{n-1}{2}, \frac{n+1}{2}\}$ when n is odd, because of the upper bound $\lceil \frac{n}{2} \rceil$, given in [9]. We study in Section 4 the algebraic immunity of those Maiorana-MacFarland functions. We show that, under a reasonable assumption (related to a new notion of nonlinearity of S-boxes), it is strictly lower than the maximum possible degree of these functions. In Section 5, we give a construction which permits to strictly increase the algebraic immunity of a Boolean function by adding, similarly to Siegenthaler's construction, a certain number (greater than 1, however) of new variables and deduce the first infinite family of functions with a controlled algebraic immunity. At last, in Section 6, we give examples of balanced functions with optimal algebraic immunity and a good nonlinearity, which are therefore usable as filtering functions in pseudo-random generators, and of balanced functions with a good algebraic immunity, a good nonlinearity and a good correlation immunity, which can be used as combining functions in pseudo-random generators.

2. **Notation and definitions**

Any Boolean function f on n variables admits a unique algebraic normal form (A.N.F.):

$$f(x_1, \dots, x_n) = \sum_{I \subseteq \{1,\dots,n\}} a_I \prod_{i \in I} x_i,$$

where the a_I's are in F_2. The *algebraic degree* of f is the degree of its algebraic normal form. Affine functions are those Boolean functions of degrees whose value is at most 1.

The *Hamming weight* $w_H(f)$ of a Boolean function f on n variables is the size of its support $\{x \in F_2^n;\ f(x) = 1\}$. The *Hamming distance* $d_H(f, g)$ between two Boolean functions f and g is the Hamming weight of the function $f + g$. The *nonlinearity* of f is its minimum distance to all affine functions. Functions used in stream or block ciphers must have high nonlinearities to resist the attacks on these ciphers (correlation and linear attacks, see e.g. [4, 23]). The nonlinearity of f can be expressed by means of the Walsh transform of f, defined as $W_f(a) = \sum_{x \in F_2{}^n} (-1)^{f(x)+x\cdot a}$, where "$\cdot$" denotes the usual inner product in F_2^n:

$$N_f = 2^{n-1} - \frac{1}{2} \max_{a \in F_2^n} |W_f(a)|. \qquad (1)$$

It is upper bounded by $2^{n-1} - 2^{n/2-1}$ because of the so-called Parseval's relation $\sum_{s \in F_2^n} W_f^2(s) = 2^{2n}$.

In the standard model of these ciphers (cf. [22]), the outputs to n linear feedback shift registers are the input to a Boolean function. The output to the function produces the keystream, which is then bitwisely xored with the message to produce the cipher. Some devide-and-conquer attacks exist on this method of encryption (cf. [4, 23]) and lead to criteria which the combining function must satisfy. Two main criteria are the following: the combining function must be balanced (i.e. its output must be uniformly distributed on $\{0, 1\}$); it must also be such that the distribution probability of its output is unaltered when any m of its inputs are fixed [23], with m as large as possible. This property, called *m-th order correlation-immunity* [22], is characterized by the set of zero values in the Walsh spectrum [24]: f is m-th order correlation-immune if and only if $W_f(u) = 0$, for all $u \in F_2^n$ such that $1 \leq w_H(u) \leq m$, where $w_H(u)$ denotes the Hamming weight of the n-bit vector u, (the number of its nonzero components). Balanced m-th order correlation-immune functions are called *m-resilient* functions. They are characterized by the fact that $W_f(u) = 0$ for all $u \in F_2^n$ such that $0 \leq w_H(u) \leq m$.

Siegenthaler's inequality [22] states that any m-th order correlation immune function on n variables has degree at most $n - m$, that any m-resilient function ($0 \leq m < n-1$) has algebraic degree smaller than or equal to $n - m - 1$ and that any $(n-1)$-resilient function has algebraic degree 1. We shall call this property *Siegenthaler's bound*.

Sarkar and Maitra [19] have shown that the Hamming distance between any m-resilient function and any affine function is divisible by 2^{m+1}. This has led to an upper bound on the nonlinearity of m-resilient functions (also partly obtained by Tarannikov and by Zhang and Zheng): the nonlinearity of any m-resilient function is smaller than or equal to $2^{n-1} - 2^{m+1}$ if $\frac{n}{2} - 1 < m + 1$, to $2^{n-1} - 2^{\frac{n}{2}-1} - 2^{m+1}$ if n is even and $\frac{n}{2} - 1 \geq m+1$ and to $2^{n-1} - 2^{m+1} \lceil 2^{n/2-m-2} \rceil$ if n is odd and $\frac{n}{2} - 1 \geq m+1$. We shall call this set of upper bounds *Sarkar et al.'s bound.* A similar bound exists for correlation immune functions, but we do not refer to it since non-balanced correlation immune functions present small cryptographic interest.

Until recently, a high algebraic degree, a high resiliency order and a high nonlinearity were the only requirements needed for the design of the function f used in a stream cipher as a combining function or as a filtering one. The recent algebraic attacks [9] have changed this situation by adding a new criterion of considerable importance to this list. Algebraic attacks recover the secret key by solving an overdefined system of multivariate algebraic equations. These attacks exploit multivariate relations involving key/state bits and output bits of f. If one such relation is found and is of low degree in the key/state bits, algebraic attacks are very efficient [8]. It is suggested in [9] that low degree relations and thus successful algebraic attacks exist for several well known constructions of stream ciphers that are immune to all previously known attacks. These low degree relations are obtained by producing low degree polynomial multiples of f, i.e., by multiplying the Boolean function f by a well chosen low degree function g, such that the product function fg (that is, the function whose support equals the intersection of the supports of f and g) is again of low degree.

The scenarios found in [9], under which low degree multiples of a Boolean function may exist, have been simplified in [15] into two scenarios: (1) there exists a non-zero Boolean function g of low degree whose support is disjoint from the support of f (such a function g is called an annihilator of f); (2) there exists a non-zero Boolean function g of low degree whose support is included in the support of f (i.e. such that g is an annihilator of $f+1$). We write then: $g \preceq f$.

The *algebraic immunity* $AI(f)$ of a Boolean function f is the minimum value of d such that f or $f+1$ admits an annihilator of degree d. It should be high enough (at least equal to 6). Clearly, the algebraic immunity of a Boolean function is upper bounded by its degree. Since the degree (in the case of resilient functions) is upper bounded by Siegenthaler's bound, the best possible situation, with respect to the degree and the algebraic immunity, is when the algebraic degree achieves Siegenthaler's bound and its algebraic immunity equals its algebraic degree.

But it has been proven in [9] that the algebraic immunity of any n-variable Boolean function is upper bounded by $\lceil \frac{n}{2} \rceil$. Hence, if the degree is greater than $\lceil \frac{n}{2} \rceil$, the best possible algebraic immunity is $\lceil \frac{n}{2} \rceil$.

3. Algebraic immunity of random balanced Boolean functions

In [15] the algebraic immunity of random balanced Boolean function is considered. The authors prove by a precise statistical analysis that for sufficiently large n the algebraic immunity of a random balanced Boolean function is almost always at least equal to $0.22n$. They also show by empirical arguments that in fact the algebraic immunity is probably better than $0.27n$.

So a natural question arises on the exact lower bound for the algebraic immunity of random balanced Boolean functions.

In this section we give experimental results which indicate that that the algebraic immunity of a random balanced Boolean function is probably very close to $0.5n$.

We first interpret the algebraic immunity in terms of Reed-Muller codes. The binary representatives of all Boolean functions of degrees less or equal to d on n variables correspond to the codewords of the Reed-Muller code $R(d, n)$. Denote by G_d a generator matrix of the code $R(d, n)$. In terms of matrix vectors, the fact that there exists a non null Boolean function g, of degree up to d, such that $fg = 0$ is equivalent to the fact that if we denote by G_d^f the matrix of length $w_H(f)$ (the Hamming weight of f) obtained by puncturing the matrix G_d, keeping the positions where the codeword associated to f equals 1, then there exists a nonzero vector c of length the dimension of $R(d, n)$, such that $c \times G_d^f = 0$ (where '×' denotes the multiplication between a matrix and a vector). In this case, the word $c \times G_d$ corresponds to the binary representative of an annilihator g of degree up to d of f.

Lemma 3.1. *A Boolean function f has no non null annihilator of degree up to d if and only if the punctured matrix G_d^f with $w_H(f)$ columns and k rows where $k = \sum_{i=0}^{d} \binom{n}{i}$ (the dimension of the code $R(d, n)$) is regular (i.e. has full rank k).*

Now we want to estimate the behaviour, in terms of rank, of such a matrix.

We recall that for large k the probability that a random binary $k \times (k+e)$ matrix has a rank strictly inferior to k is roughly $s2^{-e}$, for s a real number of order $1/2$ ([7], [1], [25]). This result means that adding a new column to a random matrix divides the probability that the matrix has not full rank by 2.

In our case the matrix G_d^f is a submatrix of the generator matrix of a Reed-Muller code. The Reed-Muller codes have been known for a long time and no theoritical properties seem to be known at present to estimate theoritically the behaviour (in terms of rank) of such an extracted matrix. Meanwhile it may seem natural that if one takes a small random submatrix of a structured big matrix like the matrix of a Reed-Muller code, which has good statistical properties, the small matrix behaves more or less like a random matrix.

This idea is confirmed by simulations for $n = 8, 9, 10, 11, 12, 13$. For different Reed-Muller codes $R(d, n)$ of dimension k, we extracted $k+e$ random columns from the 2^n possible columns to obtain a $k \times (k+e)$ matrix M. We then checked whether the rank of M was strictly smaller than k. We repeated this operation between 10000 and 100000 times (depending on the parameters n and d) to have statistically meaningful results (although the results are a little less meaningful for $d = 11$ since $10000/2^{11}$ is not large enough) that we sum up in Table 1.

These results show that for $n = 8$ or 9, adding another column divides the probability by roughly $3/2$, but that for $n \geq 10$ (in our simulations, for $n = 10, 11, 12, 13$), which is the concrete situation in cryptography, adding a new column divides the probability of the matrix not to be of rank k by roughly 2. These experimental results indicate therefore that in terms of rank, the extracted matrix from a Reed-Muller code, for not too small n, seems to behave roughly like a random matrix and that the probability of the extracted matrix not to be of full rank is roughly $s'2^{-e}$ for s' a constant of order $1/2$ (what is really important here is not the constant but the fact that each addition of a column divides the probability by two). These simulations permit us to give heuristic results on the algebraic immunity of random balanced Boolean functions. For odd $n = 2m+1$ the probability that a random balanced Boolean function f (or its complement $1+f$) has a no non null annihilator of degree up to $m-1$ (and hence an algebraic immunity greater than or equal to m) can be approximated via our heuristic by $s'2^{-e}$, where $e = 2^{n-1} - \sum_{i=0}^{m-1} \binom{n}{i} = \binom{n}{m}$, which

R(d,n)	e=0	e=7	e=8	e=10	e=11
(3,8)	0.78	0.04	0.028	$1.5\,10^{-2}$	$1.1\,10^{-2}$
(3,9)	0.72	$1.1\,10^{-2}$	$6.8\,10^{-3}$	$2.5\,10^{-3}$	$1.9\,10^{-3}$
(3,10)	0.71	$7.7\,10^{-3}$	$3.8\,10^{-3}$	$9.3\,10^{-4}$	$4.8\,10^{-4}$
(4,10)	0.71	$8.0\,10^{-3}$	$3.9\,10^{-3}$	$9.1\,10^{-4}$	$5.6\,10^{-4}$
(4,11)	0.71	$7.7\,10^{-3}$	$3.5\,10^{-3}$	$9.1\,10^{-4}$	$5.5\,10^{-4}$
(3,12)	0.71	$7.2\,10^{-3}$	$3.4\,10^{-3}$	$8.0\,10^{-4}$	$3.0\,10^{-4}$
(5,12)	0.72	$7.6\,10^{-3}$	$3.9\,10^{-3}$	$9.0\,10^{-4}$	$4.0\,10^{-4}$
(3,13)	0.72	$6.8\,10^{-3}$	$3.4\,10^{-3}$	$8.9\,10^{-4}$	$4.6\,10^{-4}$
(5,13)	0.71	$7.2\,10^{-3}$	$3.3\,10^{-3}$	$8.9\,10^{-4}$	$4.8\,10^{-4}$

Table 1. Probability for a $k\times(k+e)$ random extracted matrix from the Reed-Muller code $R(d,n)$ (of dimension k) not to be of full rank k

means that the probability that a random balanced Boolean function has algebraic immunity greater or equal to $m = \lfloor n/2 \rfloor$ is of the form $1-2^{-\binom{n}{m}}$ which tends towards 1 very quickly.

In the case where n is even, $n = 2m$, by the same argument, the probability that a random balanced Boolean function has algebraic degree greater or equal to $n/2$ is of order $1-2^{-\frac{\binom{n}{m-1}}{2}}$ and hence this heuristic seems to indicate that for even n almost all random balanced Boolean functions have an optimal algebraic immunity $n/2$.

Our experimental results therefore permit to deduce an heuristic which indicates that very quickly random balanced Boolean functions have almost always an algebraic immunity greater or equal to $\lfloor n/2 \rfloor$.

4. Algebraic immunity and Maiorana-McFarland constructions

One can found in [2] a construction of resilient functions based on the idea of a construction of bent functions due to Maiorana and McFarland:

let m and $n = r+s$ be any integers such that $r > m \geq 0$, $s > 0$, let g be any Boolean function on F_2^s and ϕ any mapping from F_2^s to F_2^r such that every element $\phi(y)$ of $\phi(F_2^s)$ has Hamming weight

$w_H(\phi(y))$ strictly greater than m, then the function:

$$f(x,y) = x \cdot \phi(y) + g(y),\ x \in F_2^r,\ y \in F_2^s \tag{2}$$

(where "$\cdot$" denotes here the usual dot product in F_2^r) is m-resilient, since we have $W_f(a,b) = 2^r \sum_{y \in \phi^{-1}(a)} (-1)^{g(y)+b\cdot y}$.

The highest possible value of the degree of f equals $s+1$, and f achieves such degree if and only if the degree of the mapping ϕ equals s (i.e. if one at least of the coordinate functions of ϕ has odd weight, or equivalently, if the sum $\sum_{y \in F_2^s} \phi(y)$ is non null). This implies that, in order to have a high algebraic degree (hopefully, to achieve Siegenthaler's bound) a Maiorana-McFarland function must be constructed, firstly with a sufficiently high value of s, and secondly such that $\sum_{y \in F_2^s} \phi(y) \neq 0$.

The fact that s must be large is also necessary to achieve high nonlinearity ; indeed, a lower bound on the nonlinearity of f has been obtained in [20] and an upper bound has been obtained in [5]:

$$2^{n-1} - 2^{r-1} \max_{a \in F_2^r} |\phi^{-1}(a)| \leq N_{f_{\phi,g}} \leq 2^{n-1} - 2^{r-1} \left\lceil \sqrt{\max_{a \in F_2^r} |\phi^{-1}(a)|} \right\rceil \tag{3}$$

where $|\phi^{-1}(a)|$ denotes the size of the pre-image of a by ϕ and $\lceil\ \rceil$ denotes the "ceiling". Hence, the nonlinearity of f is large enough (hopefully, it achieves Sarkar-Maitra's bound) under the necessary condition that r is small (i.e. that s is large) and that the values of ϕ are as equally distributed as possible in the set $\{x \in F_2^n \,|\, w_H(x) \geq m+1\}$. Note that, in the case that $m = 0$ (f balanced), f can have nonlinearity close to the maximum (which is unknown, but which is supposed to be near the maximum possible nonlinearity $2^{n-1} - 2^{n/2-1}$ of all Boolean functions) only if s is greater than or equal to $n/2$ or is at least close to $n/2$. Practically, the functions of the form (2), for $\frac{n}{2} - 1 < m+1$, can have nonlinearities approaching Sarkar-Maitra's bound for low values of n or for low values of m; they have been widely used as seeds in secondary constructions to obtain balanced Boolean functions in larger numbers of variables, with high nonlinearities, and (if necessary) with high resiliency orders (their nonlinearities being then upper bounded by Sarkar-Maitra's bound).

We shall see now that their algebraic immunity cannot achieve highest possible value $s+1$ unless s is small (we know that the degree and the nonlinearity of the function are then bad). We first introduce a definition.

Definition 4.1. Let s and r be two positive integers and ϕ a mapping from F_2^s to F_2^r. We say that ϕ is an *ultimate nonlinear* mapping if, for every affine subspace A of F_2^s, whose dimension is strictly positive, the sum $\sum_{y\in A}\phi(y)$ is non null.

In other words, $\phi : F_2^s \rightarrow F_2^r$ is ultimate nonlinear if, for every integer d such that $1 \leq d \leq s$ and every d-dimensional affine subspace A of F_2^s, the restriction of ϕ to A has degree d, exactly. This condition includes the necessary condition seen above for f having degree $s+1$ (which corresponds to $d = s$, that is, to $A = F_2^s$). But it is much stronger. *We conjecture that, unless s is significantly smaller than $n/2$ (we have seen that, in such case, f has bad degree and bad nonlinearity) such mapping does not exist.* Indeed, this condition applied with $d = 1$ and $d = 2$ already means that ϕ is injective and that it is APN (recall that an APN mapping is a mapping such that, for every $b \in F_2^r$ and every $a \in F_2^{s*}$, the equation $\phi(x) + \phi(x+a) = b$ has at most two solutions ; this is equivalent to saying that, for every $a, a' \in F_2^{s*}$ such that $a \neq a'$, the sum $\phi(x) + \phi(x+a) + \phi(x+a') + \phi(x+a+a')$ is non null). Because of obvious combinatorial reasons, no APN mapping exists for $s > r+1$. For $s = r+1$, ϕ is APN if and only if it is perfect nonlinear (i.e. if, for every $a \neq 0$, the derivative $\phi(x) + \phi(x+a)$ is balanced, that is, uniformly distributed) and we know according to [16] that such mapping does not exist either. Very few APN mappings are known for $s = r$ (see e.g. [3]) and none of them is ultimate nonlinear.

Proposition 4.2. *Let f be a Maiorana-McFarland function (2). If ϕ is not ultimate nonlinear, then f has algebraic immunity at most s.*

Proof. Let A be a d-dimensional affine subspace of F_2^s such that $\sum_{y\in A}\phi(y) = 0$. Then, the function g, equal to f multiplied by the indicator of $F_2^r \times A$, has degree at most s, since the restriction of f to this flat has degree at most d and since the indicator of $F_2^r \times A$ has degree $s-d$. And g is clearly an annihilator of $f+1$. □

Remark: Ultimate nonlinear mappings do exist when r is sufficiently larger than s. Let us take for instance $r = 2^s$ and let us choose some total order on F_2^s. For every $i = 1, ..., 2^s$, let us define the ith coordinate function ϕ_i of ϕ as the indicator of the singleton containing the ith vector of F_2^s (say a_i)[1]. For every flat A and every index i, we have $\sum_{y\in A}\phi_i(y)\neq 0$ if and only if $a_i\in A$. The mapping $\phi : F_2^s \to F_2^r$ is then such that $\sum_{x\in A}\phi(x)\neq 0$ for every flat A.

In fact, we can prove that ultimate nonlinear mappings exist for values of r much smaller than 2^s (but still much greater than s). Indeed, the constraint on ϕ given by the equation $\sum_{x\in A}\phi(x)=0$ is F_2^r-linear, and the space of solutions is then a hyperplane of the space of all mappings $\phi : F_2^s \to F_2^r$; hence, there are $(2^r)^{2^s-1}$ solutions for each flat A. If $(2^r)^{2^s-1}$ times the number

$$N_s=\sum_{d=1}^{s} 2^{s-d}\frac{(2^s-1)(2^s-2)...(2^s-2^{d-1})}{(2^d-1)(2^d-2)...(2^d-2^{d-1})}$$

of flats A is smaller than 2^{r2^s}, that is, if $N_s<2^r$, there exist mappings ϕ such that $\sum_{x\in A}\phi(x)\neq 0$ for every flat A of F_2^s. The number $\frac{(2^s-1)...(2^s-2^{d-1})}{(2^d-1)...(2^d-2^{d-1})}$ is equivalent to $2^{d(s-d)}$ times a constant $C\approx 4$ (see e.g. [21]). Hence, $2^{s-d}\frac{(2^s-1)...(2^s-2^{d-1})}{(2^d-1)...(2^d-2^{d-1})}$ is equivalent to $C\,2^{(d+1)(s-d)}$ and N_s is then equivalent to $C\,2^{\frac{(s+1)^2}{4}}$ if s is odd, and to $C\,2^{\frac{s}{2}(\frac{s}{2}+1)}$ if s is even, since the function $h(x)=(x+1)(s-x)$, defined for x an integer, has maximum value for $x=\frac{s-1}{2}$ if s is odd and for $x\in\{\frac{s}{2}-1,\frac{s}{2}\}$ if s is even. Hence, the order of magnitude of r for which ultimate nonlinear mappings $\phi : F_2^s\to F_2^r$ exist is $r\geq\frac{(s+1)^2}{4}$ (that is, is quadratic in s).

5. An extension of Boolean functions with controlled algebraic immunity, and a deduced infinite class of functions with prescribed algebraic immunity

We now give a proposition which permits to strictly increase the algebraic immunity of a function by increasing the number of variables.

Proposition 5.1. *Let f be a Boolean function on m variables $x_1,\cdots,x_m$ and with algebraic immunity a. Then the Boolean function $f+x_{m+1}x_{m+2}..x_{m+a+1}$ has algebraic immunity $a+1$.*

[1] Taking for ϕ_i the ith monomial (a total order being chosen among the 2^s monomials over F_2^s) also works.

Proof. Let:

$$g(x_1,\cdots,x_{m+a+1}) = f(x_1,\cdots,x_m) + x_{m+1}x_{m+2}..x_{m+a+1}.$$

We first remark that if a function h has degree a and is in the annihilator of f (resp. $1+f$) then $h(1+x_{m+1})$ has degree $a+1$ and is in the annihilator of g (resp. $1+g$) and hence g has an algebraic immunity at most $a+1$. Let us now prove that g has algebraic immunity at least $a+1$. We want to prove that there is no non-null Boolean function h with degree less or equal to a such that either $gh=0$ or $(1+g)h=0$.

Suppose first there exists a non-null function h of degree less or equal to a such that $gh=0$. Let us write:

$$h = h_1(x_1,\cdots,x_m) + h_2(x_1,\cdots,x_{m+a+1})$$

where $h_1(x_1,\cdots,x_m) = h(x_1,\cdots,x_m,0,\cdots,0)$. Then all the monomials of h_2 have at least one variable among the variables $x_{m+1},\cdots,x_{m+a+1}$ and $(f + x_{m+1}x_{m+2}..x_{m+a+1})(h_1+h_2) = 0$. We first remark that necessarily $fh_1 = 0$; since f has algebraic immunity a, this implies that either h_1 has degree a (recall that we suppose that the algebraic degree of h is smaller or equal to a) or $h_1 = 0$.

If h_1 has degree a then, in the product

$$(f + x_{m+1}x_{m+2}..x_{m+a+1})(h_1+h_2)$$

there is a monomial in $(x_{m+1}x_{m+2}..x_{m+a+1})h_1$ of degree $2a+1$ which contains the product $x_{m+1}x_{m+2}..x_{m+a+1}$. But such type of monomial cannot exist in fh_2, since h_2 has degree a; it cannot exist in $x_{m+1}x_{m+2}..x_{m+a+1}h_2$ either since this last function has degree at most $2a$, according to the definition of h_2. It is clearly a contradiction.

Suppose now that $h_1 = 0$, then $(f+x_{m+1}x_{m+2}..x_{m+a+1})h_2 = 0$ but since no monomial of fh_2 can contain the term $x_{m+1}x_{m+2}..x_{m+a+1}$ since h_2 has degree less or equal to a, we deduce that $fh_2 = 0$. This implies that any restriction of h_2 obtained by fixing the values of $x_{m+1}, x_{m+2}, \ldots, x_{m+a+1}$ is an annihilator of f, and since such restriction has degree strictly less than a, this implies that any such restriction must be null, a contradiction with the fact that $h_2 \neq 0$.

The case $(1+g)h = 0$ can be proven with similar arguments by noticing that $1+g = (1+f) + x_{m+1}x_{m+2}..x_{m+a+1}$ and the result follows. □

Remark 1: The Walsh transform and the nonlinearity of g can be expressed by means of those of f. It is a simple matter to see that the function

$$\varphi(x_{m+1}, x_{m+2}, \ldots, x_{m+a+1}) = x_{m+1}x_{m+2}\ldots x_{m+a+1}$$

has Walsh transform

$$W_\varphi(u_{m+1}, u_{m+2}, \ldots, u_{m+a+1})$$

equal to $2^{a+1} - 2$ if $u_{m+1} = u_{m+2} = \cdots = u_{m+a+1} = 0$ and to $-2(-1)^{u_{m+1}+u_{m+2}+\cdots+u_{m+a+1}}$ otherwise and has nonlinearity 1. The Walsh transform of g is the direct product of the Walsh transforms of f and φ. Hence $N_g = 2^{m+a} - \frac{1}{2}(2^m - 2N_f)(2^{a+1} - 2N_\varphi) = 2^{a+1}N_f + 2^m N_\varphi - 2N_f N_\varphi$ equals $(2^{a+1} - 2)N_f + 2^m$.

Remark 2: In the case of Boolean functions with a large number of variables, like the function used for Toyocrypt, the previous proposition can be used to construct functions with a lower bound on their algebraic immunity.

Remark 3: The computations we made seem to suggest that, to have an optimal algebraic immunity, a Boolean function has to have many monomials.

The following corollary gives the first family of Boolean functions with a proven non-trivial algebraic immunity (although not optimal). The proof is straightforward from the previous proposition.

Corollary 5.2. *The Boolean function on $n = \frac{a(a+1)}{2}$ variables $f(x_1, \ldots, x_n) = x_1 + x_2x_3 + x_4x_5x_6 + \cdots + x_{n-a+1}x_{n-a+2}\cdots x_n$ has algebraic immunity a.*

Note that this function has also degree a and is balanced (since the variable x_1 is isolated). According to Remark 1 above, the nonlinearity N_f of this function satisfies $2^n - 2N_f = (2^a - 2)(2^{a-1} - 2)\ldots(2^2 - 2)$ (this can be checked by induction). Unfortunately even if this function is the first construction with proven algebraic immunity, the nonlinearity of this function is not very good and further work has to be done in order to construct functions with not only a proven algebraic immunity but also with a good nonlinearity.

6. Examples of balanced Boolean functions with a good algebraic immunity

We now consider the problem of constructing explicit functions for small n which have a good algebraic immunity (AI for short). The usual criteria for Boolean functions include a good nonlinearity, a high algebraic degree and a good order of resiliency. Note that depending of the context in which a Boolean function is used, a Boolean function doesn't have necessarily to meet all these criteria. For instance if a function is used to combine LFSR then a good nonlinearity, a good resiliency order and a good algebraic degree are necessary but in a context of filtered register a good resiliency order seems not necessary. Hence it is of interest to find balanced functions with good AI and good nonlinearity and also functions with good AI, good nonlinearity and good order of resiliency.

6.1. Balanced Boolean functions with a good algebraic immunity and a good nonlinearity

We consider here for $n = 8, 9, 10, 11, 12, 13$ and 14 functions with good nonlinearity for which we compute their AI. These functions include certain families of power functions x^d which are balanced and balanced functions built from bent functions.

Some of the functions we present, achieve a better algebraic immunity, for the same nonlinearity, than the functions presented in [11], which also ask for a good resiliency order.

In Table 2 we compute the algebraic immunity of the inverse function for $7 \leq n \leq 14$. This table shows that this fonction, even if it is good, is not optimal for all n.

In Table 3 we list modified (or not) power functions with constructions * and ** (see below), which achieve optimal value for the algebraic immunity and which have almost optimal nonlinearity. We precise, when it is the case, the general family to which the exponent d, is related to (cf [3]).

Note that this kind of constructions permit to construct optimal balanced functions in terms of their AI but also with a very good nonlinearity.

A '*' in the table means we started from a codeword of weight $2^{n-1} - 2^{n/2-1}$ which was made balanced by replacing the first $2^{n/2-1}$ 0's by 1's. Usually this construction leads to a function with a higher algebraic degree than the starting function.

A '**' in the table means we started from a function which was made balanced by adding 1's and for which we inverted a small number of bits from 0 to 1 and reciprocally from 1 to 0. This small modification does not affect too much the nonlinearity but may increase the AI by 1 in the case when the dimension of the annihilator of the Boolean function f or $1 + f$ is small.

n	d	weight	degree	nonlinearity	alg. immunity
6	-1	32	5	24	3
7	-1	64	6	54	4
8	-1	128	7	112	4
9	-1	256	8	234	4
10	-1	512	9	480	5
11	-1	1024	10	980	5
12	-1	2048	11	1984	5
13	-1	4096	12	4006	6
14	-1	8192	13	8064	6

Table 2. Computation of the nonlinearity and algebraic immunity for the inverse function for $6 \leq n \leq 14$

n	d	weight	degree	nonlin.	alg. immunity
8	31	128	5	112	4
8	39 (Kasami)	128^*	6	114	4
9	57 (Kasami)	256	4	224	4
9	59	256	5	240	5
9	115	256	5	240	5
10	241 (Kasami)	512	5	480	5
10	362	512	5	480	5
10	31 (Dillon)	512^*	9	486	5
10	339 (Dobbertin)	512^*	9	480	5
11	315	1024	6	992	6
12	993 (Kasami)	2048^*	11	2000	6
12	63 (Dillon)	2048^*	11	2000	6
12	636	2048^*	11	2000	6
13	993 (Kasami)	4096	6	4032	6
13	939	4096^{**}	12	4030	7
14	4033 (Kasami)	8192	7	8064	7
14	127 (Dillon)	8192^{**}	13	8088	7

Table 3. Computation of the nonlinearity, algebraic degree and algebraic immunity for certain power functions x^d

6.2. Balanced Boolean functions with a good algebraic immunity, a good order of resiliency and a good nonlinearity

In this section we compute the AI of Boolean functions built by standard constructions derivated from the Maiorana-McFarland construction and which guarantee a certain order of resiliency and we also consider a function based on a variation of the Maiorana-McFarland construction. In particular in the case of Maiorana-McFarland constructions we compare the AI we obtain with the upper bound that we introduce in Section 4.

The results are listed in Table 4. The usual notation is used in the table for n, r, s and d the algebraic degree. The notation 'Const' is for the type of construction used: the classical construction of [2] is denoted by 'a' (when all the images by ϕ in F_2^r of the elements of F_2^s have weight at least w), the double pre-image construction of [6] is denoted by 'b' (when exactly two elements of F_2^s have the same image of weight at least w but with different values for the function g). At last the notations m, nl and ai are respectively for the order of resiliency, the nonlinearity and the algebraic immunity.

We also considered a variation on the Maiorana-McFarland construction which deals with three functions rather than one: $f(x,y) = [x{\cdot}\phi_1(y)][x{\cdot}\phi_2(y)]+[x{\cdot}\phi_1(y)][x{\cdot}\phi_3(y)]+[x{\cdot}\phi_2(y)][x{\cdot}\phi_3(y)]$. We built functions with 14 variables by considering ϕ_1, ϕ_2, ϕ_3 from F_2^6 to F_2^8 such that $\forall\, i \in \{1,2,3\}$ and any $x \in F_2^6$: $w_H(\phi_i(x)) \geq 6$ and such that $w_H(\phi_1(x) + \phi_2(x) + \phi_3(x)) \geq 6$. After several trials for the different images of F_2^6 we were able to construct a balanced Boolean function with 14 variables, algebraic degree 7, nonlinearity 7808, order of resiliency 5 and AI 6.

The results of the table show that the bound of Section 4 is reached for certain parameters s, in particular for $n = 11, 12, 13, 14$ and $s = 5$. If we compare these results and the special function we obtain for 14 variables, with the results of [11] for $n \geq 10$, we see that some of their constructions are optimized in terms of nonlinearity and order of resiliency but not in terms of AI. Since an insufficient AI makes the algebraic attacks very efficient, the constructions of [9] are more interesting theoretically than practically. The constructions of Table 4 and the special function obtained by a variation of the Maiorana-McFarland construction are not so optimal in terms of nonlinearity and resilience but some functions we found have a better AI (which is the important point).

n	r	s	d	Const.	w	m	nl	ai
8	4	4	5	b	2	2	112	3
9	5	4	5	b	3	3	224	3
9	5	4	5	a	3	2	240	4
10	5	5	6	b	3	3	480	4
10	6	4	5	a	4	3	480	4
11	6	5	6	b	4	4	960	4
11	6	5	6	a	3	2	992	5
12	6	6	7	b	4	4	$2^{11}-2^6$	5
12	7	5	6	a	4	3	$2^{11}-2^6$	5
13	7	6	7	a	4	3	$2^{11}-2^6$	5
13	7	6	7	b	4	4	$2^{12}-2^7$	5
13	8	5	6	a	5	4	$2^{12}-2^7$	5
14	7	7	8	b	4	4	$2^{13}-2^7$	5
14	8	6	7	b	6	6	$2^{13}-2^8$	5
14	8	6	7	a	5	4	$2^{13}-2^7$	5
14	8	6	7	a	5	4	$2^{13}-2^7$	5
14	9	5	6	a	7	6	$2^{13}-2^8$	5

TABLE 4. Computation of some characteristics for Boolean functions built by the Maiorana-McFarland construction

7. Conclusion

In this paper we have studied the AI of balanced Boolean functions. We have shown that it is not hard to construct functions with an optimal AI and that there are strong reasons which indicate that, as soon as n is large enough, almost all random balanced Boolean functions are almost optimal. We have given the first construction which strictly increases the AI of a Boolean function by adding a certain number of variables and we have given an upper bound on the AI of Boolean functions built through Maiorana-McFarland construction; this bound which is true under a reasonable assumption is tight for examples when s is a little smaller than $\lfloor n/2 \rfloor$. We have then considered functions optimizing the nonlinearity and the AI, we have showed that at least for n up to 14, functions with a very good nonlinearity and optimized for the AI exist. At last we have exhibited Boolean functions which realize some tradeoff between the nonlinearity, the resiliency order and the AI. Our results combined with those of [11] show that it may be difficult for a function, at least with known constructions, to be optimized, at the same time, in terms of nonlinearity, resiliency order and AI. However there seems to be no reason for Boolean functions which are good in term of nonlinearity and resiliency order to have necessarily a bad AI.

References

[1] J. Blomer, R. Karp and E. Welzl, The rank of sparse random matrices over finite fields, *Random Structures Algorithms* 10 (1997), no. 4, pp. 407–419.

[2] P. Camion, C. Carlet, P. Charpin, N. Sendrier, On correlation-immune functions, *Advances in Cryptology: Crypto '91, Proceedings, Lecture Notes in Computer Science*, V. 576 (1991), pp. 86–100.

[3] A. Canteaut, P. Charpin, and 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), pp. 105–138, 2000.

[4] A. Canteaut and M. Trabbia. Improved fast correlation attacks using parity-check equations of weight 4 and 5, *Advanced in Cryptology-EUROCRYPT 2000. Lecture notes in computer science* 1807, pp. 573-588, 2000.

[5] C. Carlet. A larger Class of Cryptographic Boolean Functions via a Study of the Maiorana-McFarland Construction. *Advances in Cryptology - CRYPT0 2002, no. 2442 in Lecture Notes in Computer Science*, pp. 549-564, 2002.

[6] C. Carlet and E. Prouff. On plateaued functions and their constructions. Proceedings of *Fast Software Encryption 2003, Lecture notes in computer science* 2887, pp. 54-73, 2003.

[7] C. Cooper, On the rank of random matrices, *Random Structures Algorithms* 16 (2000), no. 2, pp. 209–232.

[8] N. Courtois. Fast Algebraic Attacks on Stream Ciphers with Linear Feedback. *Advances in cryptology–CRYPTO 2003, Lecture Notes in Computer Science* 2729, pp. 177-194, Springer, 2003.

[9] N. Courtois and W. Meier. Algebraic Attacks on Stream Ciphers with Linear Feedback. *Advances in cryptology–EUROCRYPT 2003, Lecture Notes in Computer Science* 2656, pp. 346-359, Springer, 2002.

[10] N. Courtois and J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. *Advances in cryptology–ASIACRYPT 2002, Lecture Notes in Computer Science* 2501, pp. 267-287, Springer, 2003.

[11] D. K. Dalai, K. C. Gupta and S. Maitra. Results on Algebraic Immunity for Cryptographically Significant Boolean Functions. proceedings of Indocrypt 2004. To appear in LNCS.

[12] J.-C. Faugère and G. Ars. An Algebraic Cryptanalysis of Nonlinear Filter Generators using Gröbner bases. *Rapport de Recherche INRIA* 4739, 2003.

[13] Mac Williams, F. J. and N. J. Sloane (1977). *The theory of error-correcting codes*, Amsterdam, North Holland.

[14] Meier, W. and O. Staffelbach (1990). Nonlinearity Criteria for Cryptographic Functions, *Advances in Cryptology, EUROCRYPT' 89, Lecture Notes in Computer Science 434, 549-562, Springer Verlag.*

[15] W. Meier, E. Pasalic and C. Carlet. Algebraic attacks and decomposition of Boolean functions. *Advances in Cryptology, EUROCRYPT 2004, Lecture Notes in Computer Science, Springer Verlag* 3027, pp. 474-491, 2004.

[16] K. Nyberg. Perfect non-linear S-boxes, *Advances in Cryptology, EUROCRYPT' 91, Springer Verlag, Lecture Notes in Computer Science* 547, pp. 378-386, 1992.

[17] D. Olejár and M. Stanek. "On cryptographic properties of random Boolean functions." Journal of Universal Computer Science, vol. 4, No. 8, pp. 705-717, 1998.

[18] P. Sarkar and S. Maitra. Construction of nonlinear Boolean functions with important cryptographic properties. *Advances in Cryptology - EUROCRYPT 2000*, no. 1807 in Lecture Notes in Computer Science, Springer Verlag, pp. 485-506, 2000.

[19] Sarkar, P. and Maitra, S. Nonlinearity Bounds and Constructions of Resilient Boolean Functions. *CRYPTO 2000, LNCS* Vol. 1880, ed. Mihir Bellare, pp. 515-532 (2000).

[20] J. Seberry, X.M. Zhang and Y. Zheng. "On constructions and nonlinearity of correlation immune Boolean functions." *Advances in Cryptology - EUROCRYPT'93*, LNCS 765, pp. 181-199 (1994).

[21] N. Sendrier. On the dimension of the hull. *SIAM J. Discrete Math.* Vol. 10, pp. 282-293, 1997.

[22] Siegenthaler, T. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information theory*, V. IT-30, No 5 (1984), pp. 776-780.

[23] Siegenthaler, T. Decrypting a Class of Stream Ciphers Using Ciphertext Only. *IEEE Transactions on Computer, V. C-34*, No. 1, pp. 81-85, 1985.

[24] Xiao Guo-Zhen and Massey, J. L. A Spectral Characterization of Correlation-Immune Combining Functions. *IEEE Trans. Inf. Theory*, Vol IT 34, n° 3 (1988), pp. 569-571.

[25] E. Welzl, Rank of random matrices over GF(2), preprint available at http://www.inf.ethz.ch/personal/emo/SmallPieces.html

ALGEBRAIC IMMUNITIES OF FUNCTIONS OVER FINITE FIELDS

G. Ars[1] and J.-Ch. Faugère[2]

Abstract. A general mathematical definition for a function from $GF(q)^n$ to $GF(q)^m$ to resist to cryptanalytic attacks is developed. It generalizes the definition of Algebraic Immunity for Stream Cipher to any finite field and also Block Cipher. This algebraic immunity corresponds to equations with a low leading term according a monomial ordering. We give the properties of this Algebraic Immunity and also compute explicit and asymptotic bounds. We extend the definitions of Algebraic Immunity to functions with memory but they depend on the number of consecutive outputs we look at. We show that all the results obtained for memoryless function give similarly results on memory functions by a change of variables. And then, we prove that, for a memory function f with memory size l and only one output, if there is no relation which doesn't depend on memory for l consecutive output, then we can construct a polynomial that generates all relations without memories. We apply this theorem to the summation generator and compute explicitly the Algebraic Immunity.

[1] IRMAR, University of Rennes 1
Campus de Beaulieu 35042 Rennes, France
Tel : 02 23 23 58 58 - Fax : 02 23 23 67 90
email: gwenole.ars@math.univ-rennes1.fr
[2] LIP6/CNRS/INRIA, University of Paris VI
8 rue du Capitaine Scott Paris 75015 Paris, France
email: Jean-Charles.Faugere@lip6.fr

1. **Introduction**

Algebraic attacks are among the most efficient attacks for public key cryptosystems, block ciphers and stream ciphers. They try to recover a secret key by solving a system of algebraic equations. Algebraic attacks were first applied to Matsumoto-Imai Public Key Scheme in [13] by Jacques Patarin. Algebraic attacks were also applied to block ciphers in [7], where the complexity of attacking AES and Serpent was evaluated.

For Stream Cipher, the main cryptographic criteria used for boolean functions had previously been a high algebraic degree to counter Berlekamp-Massey algorithm. In [6], it is demonstrated that low degree relations exist. These relations simplify the Algebraic attacks. So a significant step is to find functions resistant against these attacks. In [12], the notion of Algebraic Immunity for boolean functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ is introduced.

Some relations with low degree also exist in Block Cipher, and to construct the polynomial system defined by the Block Cipher, we use this equation. We can also extend the algebraic Immunity to Block Ciphers.

The first objective of this paper is to generalize the notion of Algebraic Immunity to functions over any finite fields $\mathbb{F}_q$ and we give two definitions: Algebraic Immunity for Stream Cipher and Algebraic Immunity for Block Cipher denoted respectively $AI_S(f)$ and $AI_B(f)$.

First, we show that these two notions are linked to Gröbner basis for a specific order on monomials, the DRL order for $AI_B(f)$ and the Elimination order for $AI_S(f)$. We prove that the definition of a function over finite fields gives immediately a Gröbner basis for a lexicographic order. Having a Gröbner basis helps us to find properties on the ideal generated by this basis. These properties give bounds on the notion of Algebraic Immunities. These bounds are the power of the first coefficient of the following series, which is negative:

$$\begin{array}{ll} AI_B(f) & \frac{q^n}{1-t} - \frac{(1-t^q)^{n+m}}{(1-t)^{n+m+1}} \\ AI_S(f) & \frac{q^{n-m}}{1-t} - \frac{(1-t^q)^n}{(1-t)^{n+1}} \end{array}$$

with f a function from $\mathbb{F}_q^n$ to $\mathbb{F}_q^m$.

From these series, we give explicit bounds on Algebraic Immunities. Then we show asymptotic bound on n for $AI_B(f)$ and $m = n$. We remark that the influence of the field $\mathbb{F}_q$ on $AI_B(f)$

and $m = n$ depends only on the $\sqrt{q}$, whereas, the dependence is on q for $AI_S(f)$ and $m = 1$.

The second part is to extend these notions to functions with memories. We notice that Algebraic Immunities can be extended to these functions but depend on the number of consecutive outputs we look at. We show that all the results obtained for memoryless functions give similarly results on memory functions by a change of variables. And then, we prove that, for a memory function f with memory size ℓ and only one output, if there is no relation which doesn't depend on memory for ℓ consecutive output, then we can construct a polynomial that generates all relations without memories. We can apply the theorem to the summation generator and we compute explicitly $AI_S(f)$ for some value of n, it corresponds to n for $n \leq 9$.

Section 2 presents the definition of Algebraic immunities for memoryless functions and some properties. Section 3 is devoted to the computation of explicit and asymptotic bounds of Algebraic Immunities. And section 4 generalizes these notions to function with memory.

2. Definitions of Algebraic Immunities

2.1. Basic Notations and Definitions

Let $\mathbb{F}_q[\mathbf{X}, \mathbf{Z}] = \mathbb{F}_q[x_1, \ldots, x_n, z_1, \ldots, z_m]$ be a polynomial ring with variables $x_1, \ldots, x_n, z_1, \ldots, z_m$ over a finite field $\mathbb{F}_q$ with cardinal q. For a monomial $\mathbf{X}^\alpha \mathbf{Z}^\beta = x_1^{\alpha_1} \cdots x_n^{\alpha_n} z_1^{\beta_1} \cdots z_m^{\beta_m}$, $|(\alpha, \beta)| := \sum_{i=1}^n \alpha_i + \sum_{j=1}^m \beta_j$ is called the *total degree* of this monomial, denoted $\deg(\mathbf{X}^\alpha \mathbf{Z}^\beta)$ and $|\alpha| := \sum_{i=1}^n \alpha_i + \sum_{j=1}^m \beta_j$ is called the *partial degree* of this monomial, denoted $\deg(\mathbf{X}^\alpha \mathbf{Z}^\beta, \mathbf{X})$. In the following, the set of all monomials in variables $x_1, \ldots, x_n, z_1, \ldots, z_m$ is denoted by $M(\mathbf{X}, \mathbf{Z})$, or simply by M. In the theory of Gröbner bases, we need to consider a *monomial ordering* (cf. [8]).

Two of such ordering is the *degree reverse lexicographical order* (DRL) and the *elimination order* defined as follows:

Definition 2.1. For $(\alpha, \beta) = (\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m)$ and $(\alpha', \beta') \in \mathbb{N}^{n+m}$, We say

- $\mathbf{X}^\alpha \mathbf{Z}^\beta \succ_{\mathrm{DRL}} \mathbf{X}^{\alpha'} \mathbf{Z}^{\beta'}$ if $|(\alpha, \beta)| > |(\alpha', \beta')|$, or $|(\alpha, \beta)| = |(\alpha', \beta')|$ and the right-most nonzero entry of the vector $(\alpha, \beta) - (\alpha', \beta') \in \mathbb{Z}^{n+m}$ is negative.

$\succ_{\mathrm{DRL}}$ defined the DRL order of variable $[\mathbf{X},\mathbf{Z}]$.

- $\mathbf{X}^\alpha\mathbf{Z}^\beta \succ_{\mathrm{Elim}} \mathbf{X}^{\alpha'}\mathbf{Z}^{\beta'}$ if $\mathbf{X}^\alpha \succ_{\mathrm{DRL}} \mathbf{X}^{\alpha'}$, or $\mathbf{X}^\alpha =_{\mathrm{DRL}} \mathbf{X}^{\alpha'}$ and $\mathbf{Z}^\beta \succ_{\mathrm{DRL}} \mathbf{Z}^{\beta'}$.

  $\succ_{\mathrm{Elim}}$ defined the Elimination order of variable $[\mathbf{X}],[\mathbf{Z}]$.

There are many other monomial orderings.

A nonzero polynomial g in $k[\mathbf{X}]$ is written as $g = \sum_{\alpha,\beta} c_{\alpha,\beta}\mathbf{X}^\alpha\mathbf{Z}^\beta$, $c_{\alpha,\beta} \neq 0$. We use the following notations:

$T(g) = \{c_{\alpha,\beta}\mathbf{X}^\alpha\mathbf{Z}^\beta \,|\, c_{\alpha,\beta} \neq 0\}$: the set of *terms* of g and $M(g) = \mathbf{X}^\alpha\mathbf{Z}^\beta \,|\, c_{\alpha,\beta} \neq 0\}$: the set of *monomials* of g
We denote the *leading term*, the *leading coefficient* and the *leading term* which respect an order $\prec$, by $\mathrm{LM}(g)$, $\mathrm{LC}(g)$ and $\mathrm{LT}(g)$ respectively. (For each definition, see [8].)

The ideal in $\mathbb{F}_q[\mathbf{X},\mathbf{Z}]$ generated by a subset F is denoted by $\langle F\rangle$.

Under the above notation, a *Gröbner basis* is defined as follows.

Definition 2.2. Let M be the set of all monomial of $\mathbb{F}_q[\mathbf{X},\mathbf{Z}]$ with a fixed ordering. A finite subset $G = \{g_1,\dots,g_m\}$ of an ideal $\mathcal{I}$ is called a *Gröbner basis* if

$$\langle \mathrm{LT}(g_1),\dots,\mathrm{LT}(g_m)\rangle = \langle \mathrm{LT}(\mathcal{I})\rangle.$$

For a given ideal $\mathcal{I}$, its Gröbner basis is not unique. But the *reduced Gröbner basis*, which is defined as follows, is uniquely determined.

Definition 2.3. A Gröbner basis $G = \{f_1,\dots f_m\}$ of an ideal $\mathcal{I}$ is called *reduced Gröbner basis* if for all i, $\mathrm{LC}(f_i) = 1$ and any monomial of f_i is not divisible by any element of $\mathrm{LM}(G\backslash\{f_i\})$.

Proposition 2.4. *Let I be an ideal of $\mathbb{F}_q[x_1,\dots,x_n]$ and $k \in \{1,\dots,n\}$. Assume G a Gröbner basis of I for the Elimination order $[x_1,\dots,x_k]$, $[x_{k+1},\dots,x_n]$ (or the Lexicographical order). Then $G \cap \mathbb{F}_q[x_{k+1},\dots,x_n]$ is a Gröbner basis of $I \cap \mathbb{F}_q[x_{k+1},\dots,x_n]$.*

So if we want to find a polynomial depending on severals variables $x_{k+1},\dots,x_n$, we just need to compute a Gröbner basis with one of these orders.

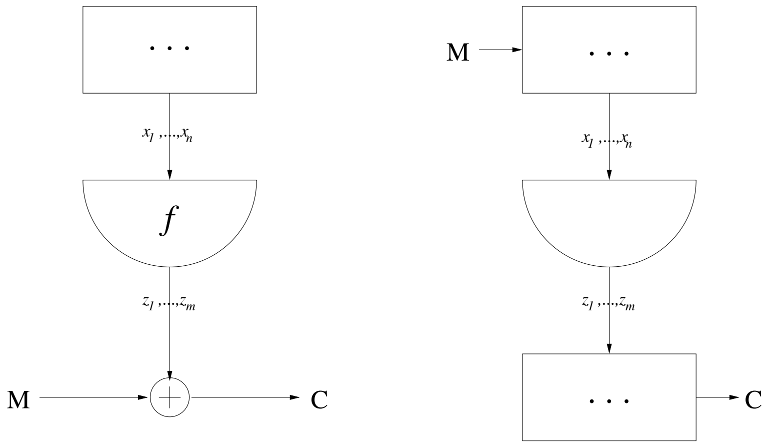
2.2. Algebraic Immunities

Let us consider a function $f : \left\{ \begin{array}{c} \mathbb{F}_q^n \to \mathbb{F}_q^m \\ \mathbf{X} \mapsto \mathbf{Z} \end{array} \right.$, denoted as $z_1 = f_1(\mathbf{X}), \ldots, z_m = f_m(\mathbf{X})$.

Our objective is to study algebraic equations induced by the graph of f which gives solutions on the field $\mathbb{F}_q$ and not on $\overline{\mathbb{F}_q}$, the algebraic closure of $\mathbb{F}_q$. To restrict the study of equation to $\mathbb{F}_q$, we add field equations $x_i^q - x_i$, denoted as the set $\mathbf{X}^q - \mathbf{X}$ and consider elements of the ideal

$$\mathcal{I} = \langle z_1 - f_1(\mathbf{X}), \ldots, z_m - f_m(\mathbf{X}), \mathbf{X}^q - \mathbf{X} \rangle.$$

In symmetric cryptography, there is two important applications: Stream Cipher and Block Cipher.



Stream Cipher

Block Cipher

On Stream Cipher, we use functions to filter elements $(x_1, \ldots, x_n)$. So when we do an attack on this structure, we suppose known elements $(z_1, \ldots, z_m)$ and try to find relations on $(x_1, \ldots, x_n)$. Whereas, in Block Cipher, non-linear functions are used inside the process to create the coded message. A possible way to write algebraic equations is to introduce intermediate variables as N. Courtois has done it in [7].

Now we will present a definition of Algebraic Immunities for both cases.

Definition 2.5. Let us consider a function $f : \mathbb{F}_q^n \to \mathbb{F}_q^m$ and $\mathcal{I} = \langle z_1 - f_1(\mathbf{X}), \ldots, z_m - f_m(\mathbf{X}), \mathbf{X}^q - \mathbf{X} \rangle$.

We define:

- Algebraic Immunity of Block Cipher,

$$AI_B(f) = min\{\deg(P),\ P \in \mathcal{I}\}.$$

- Algebraic Immunity of Stream Cipher,

$$AI_S(f) = min\{\deg(P, \mathbf{X}),\ P \in \mathcal{I}\}.$$

As an ideal with field equations is radical, there is an equivalence between $\mathcal{I}$ and the set of solution of $\mathcal{I}$ which is the graph of f. So $AI(f)$ is a generalization of Algebraic Immunity defined in article [12] and first introduced in article [6].

2.3. **Properties of Algebraic Immunities**

This definition doesn't give us a way to have Algebraic Immunities, we can find them with Gröbner basis according to some orders on monomials. This is resumed in the next theorem.

Theorem 2.6. *Let us consider a function* $f : \mathbb{F}_q^n \to \mathbb{F}_q^m$.

- *A reduced Gröbner basis of* $\mathcal{I}$ *for a DRL order* $[\mathbf{X}, \mathbf{Z}]$ *contains a linear basis of polynomials* P *of* $\mathcal{I}$ *such that* $AI_B(f) = \deg(P)$.
- *A reduced Gröbner basis of* $\mathcal{I}$ *for a elimination order on* $[\mathbf{X}], [\mathbf{Z}]$ *contains a linear basis of polynomials* P *of* $\mathcal{I}$ *such that* $AI_S(f) = \deg(P, \mathbf{X})$.

Proof. First we have noticed that for a DRL order $[\mathbf{X}, \mathbf{Z}]$ and any polynomial g, $\deg(g) = \deg(LM(g))$ and for an Elimination order $[\mathbf{X}, \mathbf{Z}]$ and any polynomial h, $\deg(h, \mathbf{X}) = \deg(LM(h), \mathbf{X})$.

Furthermore to reduce a polynomial g by another one h, we need that $LT(g) \succ LT(h)$.

As all polynomials of $\mathcal{I}$ are reduced to zero by a Gröbner basis, these both remarks prove the theorem. If P is a polynomial of $\mathcal{I}$ such that $\deg(P) = AI_B(f)$, resp. $\deg(P, \mathbf{X}) = AI_S(f)$, then P is reduced by the Gröbner basis G for a DRL order, resp. the Elimination order, so there is $g \in G$, such that $LT(g) \prec LT(P)$. From the first remark, we prove that G contains a linear generated family satisfying the condition of the theorem.

Then the definition of reduced Gröbner basis implies that the linear generated family is a linearly independent family. □

This theorem gives us a way to compute the Algebraic Immunities. To find a Gröbner basis of this ideal $\mathcal{I}$ has a bad theoretical complexity but is very efficient in practice. There is other methods to find $AI_S(f)$ presented in [1,6,12].

Moreover, we have several properties like the comparison between this two notions.

Proposition 2.7. *Let us consider* $f : \mathbb{F}_q^n \mapsto F_q^m$.

Then $AI_S(f) \leq AI_B(f)$.

Proof. Let us consider P a polynomial of $\mathcal{I}$ so that $\deg(P) = AI_B(f)$.

We have $\deg(P, \mathbf{X}) \leq \deg(P)$. Thus $AI_S(f) \leq \deg(P, \mathbf{X}) \leq \deg(P) = AI_B(f)$. □

In the article [4], we want to find algebraic relation g on the graph of f satisfying $\deg(g, \mathbf{X}) = AI_S(f)$ so that g can be written as $g(\mathbf{X}, \mathbf{Z}) = g_1(\mathbf{X}) + g_2(\mathbf{X}, \mathbf{Z})$ with $\deg(g_1, \mathbf{X}) = \deg(g_1) > \deg(g_2, \mathbf{X})$. In fact, we can give with this two Algebraic Immunities a condition of existence of these relations.

Proposition 2.8. *Let us consider* $f : \mathbb{F}_q^n \mapsto F_q^m$.

There exists $g \in \mathcal{I}$ *satisfying* $\deg(g, \mathbf{X}) = AI_S(f)$ *so that* g *can be written as* $g(\mathbf{X}, \mathbf{Z}) = g_1(\mathbf{X}) + g_2(\mathbf{X}, \mathbf{Z})$ *with* $\deg(g_1, \mathbf{X}) = \deg(g_1) > \deg(g_2, \mathbf{X})$.

If $AI_S(f) = AI_B(f)$.

Moreover if $f : \mathbb{F}_2^n \mapsto F_2$, *then this condition is a necessary and sufficient condition.*

Proof. If $AI_S(f) = AI_B(f)$, let g be a polynomial so that $\deg(g) = AI_B(f)$, g can be written as $g_1(\mathbf{X}) + g_2(\mathbf{X}, \mathbf{Z})$ so that g_2 has no monomial which depends only of $\mathbf{X}$. This means that $\deg(g_2) > \deg(g_2, \mathbf{X})$.

We have $\deg(g) = max(\deg(g_1), \deg(g_2)) = \deg(g, \mathbf{X})$ then $\deg(g_1) = \deg(g, \mathbf{X}) \geq \deg(g_2) > \deg(g_2, \mathbf{X})$.

Thus g satisfies the condition of the proposition.

For the case of $f : \mathbb{F}_2^n \mapsto F_2$, a polynomial $g \in \mathcal{I}$ satisfying $\deg(g, \mathbf{X}) = AI_S(f)$ can be written as $g(\mathbf{X}, z_1) = g_1(\mathbf{X}) + z_1 g_2(\mathbf{X})$. As $\deg(g_2, \mathbf{X}) < \deg(g_1)$, $\deg(g) = \deg(g_1) \leq \deg(g, \mathbf{X})$. □

As we have found different properties of these algebraic immunities, we can give bounds of their value.

3. Bound on the value of Algebraic Immunities

In this section, we give bounds on these Algebraic Immunities.

3.1. Properties of the ideal $\mathcal{I}$

Let us consider f a function of $\mathbb{F}_q^n$ on $\mathbb{F}_q^m$. We want to find polynomials with degree $AI_B(f)$, respect $AI_S(f)$ according $\mathbf{X}$, in the ideal $\mathcal{I}$ generated by $z_1 - f_1(\mathbf{X}), \dots, z_m - f_m(\mathbf{X}), \mathbf{X}^q - \mathbf{X}$ and especially, we consider the $\mathbb{F}_q[X_1, \dots, X_n, Z_1, \dots, Z_m]/\mathcal{I}$, denoted by $\mathcal{A}$.

As $\mathcal{I}$ is a zero dimensional ideal, we know that the ring $\mathcal{A}$ is a linear vector space with finite dimension. This subsection gives this dimension.

G is a Gröbner basis of $\mathcal{I}$ for a lexicographic order $z_1 \succ \cdots \succ z_m \succ x_1 \succ \cdots \succ x_n]$ of $\mathcal{I}$.

Thus $\mathcal{A}$ is a linear vector space with $\mathbf{X}^\alpha \mid \alpha \in \mathbb{F}_q^n\}$ as a linear basis.

Then $\mathcal{A}$ is a vector space of dimension q^n.

We deduce that the image of $q^n + 1$ monomials in $\mathcal{A}$ is a linearly dependent family, then there is a linear relation in this family and this relation corresponds to a polynomial of $\mathcal{I}$.

Thus considering the $q^n + 1$ first monomials for the DRL order for $AI_B(f)$ and the Elimination order for $AI_S(f)$ is a sufficient condition to have a relation and find the degree. As the algebraic Immunity corresponds to a degree, we need to count the monomials there are in $\mathbb{F}_q[\mathbf{X}, \mathbf{Z}]/\langle \mathbf{X}^q - \mathbf{X}, \mathbf{Z}^q - \mathbf{Z}\rangle$ for the degree d. Let us denote M_p^d the number of monomial in $\mathbb{F}_q[y_1, \dots, y_p]/\langle y_i^q - y_i$ with degree d. Then the number of monomial m_ℓ satisfying $deg(m_\ell) = d$ in $\mathbb{F}_q[\mathbf{X}, \mathbf{Z}]/\langle \mathbf{X}^q - \mathbf{X}, \mathbf{Z}^q - \mathbf{Z}\rangle$ is M_{n+m}^d and the number of monomial m'_ℓ satisfying $deg(m'_\ell, \mathbf{X}) = d$ in $\mathbb{F}_q[\mathbf{X}, \mathbf{Z}]/\langle \mathbf{X}^q - \mathbf{X}, \mathbf{Z}^q - \mathbf{Z}\rangle$ is $q^m M_n^d$.

We can translate this condition as a series, a bound of the Algebraic Immunity will be the first degree of the series with a negative or zero coefficient. These series are:

$AI_B(f)$	$\frac{q^n}{1-t} - \frac{(1-t^q)^{n+m}}{(1-t)^{n+m+1}}$
$AI_S(f)$	$\frac{q^{n-m}}{1-t} - \frac{(1-t^q)^n}{(1-t)^{n+1}}$

We notice that the difference between the both Algebraic Immunities is solely the following change of variables:

$$\begin{array}{ccc} AI_B(f) & & AI_S(f) \\ n & \longleftrightarrow & n-m \\ n+m & \longleftrightarrow & n \end{array}$$

3.2. Explicit bounds

In this section, we give explicit bounds on the Algebraic Immunities which are only bounds on the first degree of the series with a negative or zero coefficient. Then we compare these bound to the computed degree of the series for given value of n, m and q. A first bound is given by minoring m by 1:

Proposition 3.1. *Let us consider $f : \mathbb{F}_q^n \to \mathbb{F}_q^m$.*

We have

- $AI_B(f) \leq \begin{cases} \lfloor \frac{(n+1)(q-1)}{2} \rfloor & \textit{if } q > 2 \\ \lceil \frac{n+1}{2} \rceil & \textit{if } q = 2 \end{cases}$.
- $AI_S(f) \leq \begin{cases} \lfloor \frac{n(q-1)}{2} \rfloor & \textit{if } q > 2 \\ \lceil \frac{n}{2} \rceil & \textit{if } q = 2 \end{cases}$.

Proof. We can prove it for $AI_B(f)$, the change of variable will give an equivalent bound to $AI_S(f)$.

As M_{n+m}^d denotes the number of monomials with degree d. We have $q^{n+m} = \sum_{i=0}^{(q-1)(n+m)} M_{n+m}^i$.

Furthermore $M_{n+m}^d = M_{n+m}^{(q-1)(n+m)-d}$.

Thus for $m = 1$, $q^{n+1} \leq 2 \sum_{i=0}^{\lfloor \frac{(q-1)(n+1)}{2} \rfloor} M_{n+1}^i$.

As $M_{n+m}^d \geq M_{n+1}^d$, then $\sum_{i=0}^{\lfloor \frac{(q-1)(n+1)}{2} \rfloor} M_{n+m}^i \geq \frac{1}{2} q^{n+1}$.

- If $q > 2$, then $\sum_{i=0}^{\lfloor \frac{(q-1)(n+1)}{2} \rfloor} M_{n+m}^i > q^n$.

  Thus there is a polynomial in $\mathcal{I}$ with a degree lower or equal to $\lfloor \frac{(q-1)(n+1)}{2} \rfloor$.
- If $q = 2$, we do not have a strict inequality as $2^{n+1} = 2 \sum_{i=0}^{\lfloor \frac{n+1}{2} \rfloor} M_{n+1}^i$ for $n+1$ odd.

  But we are sure that $2 \sum_{i=0}^{\lceil \frac{n+1}{2} \rceil} M_{n+1}^i > 2^{n+1}$.

  Then $AI_B(f) \leq \lceil \frac{n+1}{2} \rceil$.

□

In the case of $q = 2$, the bound on $AI_S(f)$ is the bound given in the articles [6, 9].

This proposition does not take into account the number of outputs of f. It is optimal for only one output for f. But in Block Cipher, we use in general inversible function, so $m = n$. We need to give other bounds only for $AI_B(f)$. The next bounds can be extended to $AI_S(f)$ by the change of variables given in the previous section.

Proposition 3.2. *Let us consider $f : \mathbb{F}_q^n \to \mathbb{F}_q^m$.*

Then

If $\binom{m+n+q-1}{q-1} \geq q^n$,

$$AI_B(f) \leq \frac{\sqrt{(m+n-1)^2 + 4\left((m+n)!q^n\right)^{\frac{2}{m+n}}} - (m+n+1)}{2}$$

And if $\binom{m+n+q-1}{q-1} < q^n$, then

$$AI_B(f) \leq \frac{\sqrt{m'^2 - 4(m+n-\Omega\Gamma^2)(1-\Omega)} - m'}{2(1-\Omega)}$$

where $\Gamma = (-q + \frac{m+n-1}{2}) + ((m+n-1)!)^{\frac{1}{m+n}}\, q^{\frac{n}{m+n}})$ and $\Omega = \frac{1}{4}2^{\frac{2}{m+n}}(m+n)^{\frac{2}{m+n}}$ and $m' = m+n+1-2\Omega\Gamma$

Proof. This proof is technical and quite long, we only explain in this article how to find it and refer to [2] for complete proof.

First we lower the bound of the number of monomials with degree lower than d by $\binom{n+m+d}{n+m} - (n+m)\binom{n+m+d-q}{n+m}$. And we find two cases, $d < q$, this means $\binom{m+n+q-1}{q-1} \geq q^n$, and $d > q$.

With bounds on this binomials, we can found a lower bound as a polynomial in d with degree $n+m$. A sufficient condition to find the degree d is to have this lower bound higher than q^n.

Using the Hölder inequality, it give us that a quadratic polynomial in d must be positive and then the result of the proposition. □

This bound is not useful in this form. But it can give us an asymptotic estimation in n of $AI_B(f)$, for $m = n$.

Corollary 3.3. *Let us consider $f : \mathbb{F}_q^n \to \mathbb{F}_q^n$.*

$$\frac{1}{n}AI_B(f) \leq \begin{cases} \frac{5}{6} + o(1) & \text{if } q \leq 7 \\ \frac{2}{3}\left(\frac{4\sqrt{q}}{e} - \frac{11}{4}\right) + o(1) & \text{if } q \geq 8 \end{cases}$$

As we can see, this bound is bad for lower value of q, it is worth than the bound given by proposition 3.1. Then we give a better bound for $q = 2$.

We can compare these explicit bounds for Algebraic Immunity for Bloc Cipher, for exemple. For fixed values of q, n and m, we determinate a bound directely from the series. We have compared this bound with bounds of propositions 3.2 and 3.1 for $q = 16$ and $n = m$.

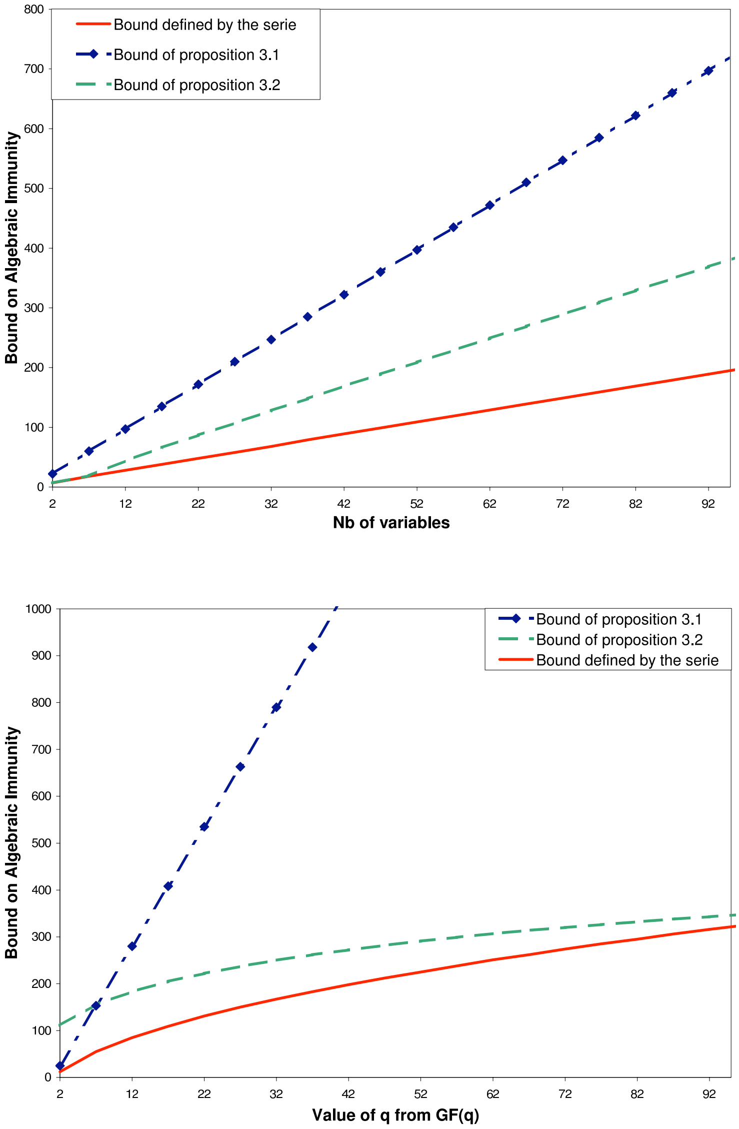


FIGURE 1. Comparison of first bounds

Figure 1 gives us the difference according the value of n, with $m = n$. We see the linear behavior but the asymptotic constant $\frac{1}{n}AI_B(f)$ found is not good.

These bounds depend on the value of q, figure 1 shows that the bound of proposition 3.2 is bad for small field.

Now, we give a better bound on $\mathbb{F}_2$.

Proposition 3.4. *Let us consider* $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$.

$$AI_B(f) \leq \frac{1-\sqrt{1-4\Gamma}}{2}m'.$$

where m'=m+n and

$$\Gamma = \tfrac{1}{8(m'+1)\ln 2}\Big(\sqrt{(\tfrac{1}{12m'+1} - \tfrac{1}{2}\ln 2\pi m' - n\ln 2)^2 + \tfrac{4(m'+1)\ln 2}{3m'}} \\ -(\tfrac{1}{12m'+1} - \tfrac{1}{2}\ln 2\pi - \tfrac{1}{2}\ln m' - n\ln 2)\Big).$$

Proof. As for proposition 3.2, we only explain how we found it and refer to [2] for a complete proof.

First, we bound the number of monomial with degree lower than d by $\binom{n+m}{d}$ and using the double inequality of H. Robbins on $n!$, we have :

$$\ln\binom{k}{d} > -\frac{1}{2}\ln 2\pi k - \left((k\lambda + \frac{1}{2})\ln\lambda + (k(1-\lambda)+\frac{1}{2})\ln(1-\lambda)\right) \\ + \frac{1}{12k+1} - \frac{1}{12}\frac{1}{\lambda(1-\lambda)}$$

with $d = \lambda k$.

By studying the variation of the right term as a function in $\lambda \in [\frac{1}{k}; \frac{1}{2}]$, we find a lower bound of $\ln(\binom{k}{d})$. A sufficient condition to find a bound of d is that this lower bound is higher than q^n.

This gives us a quadratic polynomial in $\lambda(1-\lambda)$, where $d = \lambda(n+m)$, which must be positive. And then we find a minimum value for λ and thus the result of the proposition. □

Corollary 3.5. *Let us consider* $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$.

$$\frac{1}{n}AI_B(f) \leq \frac{1}{8} + o(1).$$

We compare all the bounds together.

The bound on $\mathbb{F}_2$ is close to the bound defined by the serie.

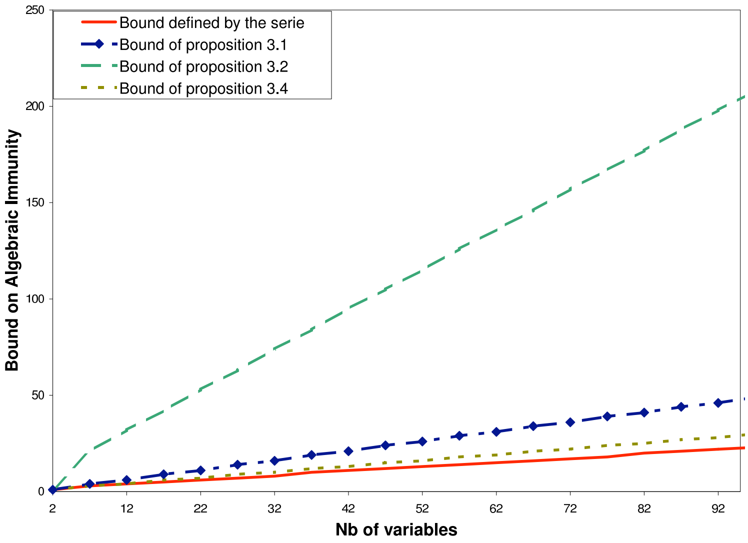


FIGURE 2. Comparison of bounds according the nb of variables

4. Functions with memory

For Stream Cipher, using functions with memories gives good cryptographic criteria. This idea is used in several stream cipher as $E0$, we find it in Bluetooth.

A function with memory is a function where the output depends on the input and a memory defined by previous inputs of size ℓ.

F. Armknecht and M. Krause have proved in article [10] that for a boolean function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ and considering several consecutive outputs, there is a relation between the inputs and the outputs which does not depend on the memories. The articles [5,9] have developed this study for functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$.

In this section, we look at the Algebraic Immunity for Stream Cipher of a function f, this algebraic immunity is defined according the number of consecutive outputs studied.

4.1. Definition

Let us consider a function f with a memory of size ℓ at moment t. For this moment t, the memory is denoted by $\mathbf{C}^{(t)}$ =

$(c_1^{(t)},\dots,c_\ell^{(t)})$, the input by $\mathbf{X}^{(t)} = (x_1^{(t)} \dots x_n^{(t)})$ and the output by $\mathbf{Z}^{(t)} = (z_1^{(t)} \dots z_m^{(t)}$.

The function f can be written at moment t as :

$$f : \left\{ \begin{array}{ccc} z_1^{(t)} & = & f_1(\mathbf{X}^{(t)}, \mathbf{C}^{(t)}) \\ & & \vdots \\ z_m^{(t)} & = & f_m(\mathbf{X}^{(t)}, \mathbf{C}^{(t)}) \\ c_1^{(t+1)} & = & P_1(\mathbf{X}^{(t)}, \mathbf{C}^{(t)}) \\ & & \vdots \\ c_\ell^{(t+1)} & = & P_\ell(\mathbf{X}^{(t)}, \mathbf{C}^{(t)}) \end{array} \right.$$

Let $\mathcal{I}_M$ be the ideal generated by $\left\{ \begin{array}{l} z_j^{(t+k)} - f_j(\mathbf{X}^{(t+k)}, \mathbf{C}^{(t+k)}) \\ c_i^{(t+k+1)} - P_i(\mathbf{X}^{(t+k)}, \mathbf{C}^{(t+k)}) \end{array} \right.$ for $j \in \{1,\dots,m\}$, $i \in \{1,\dots,\ell\}$ and $k \in \{0,\dots,M-1\}$ and field equations $\left(\mathbf{X}^{(t+k)}\right)^q - \mathbf{X}^{(t+k)}$, $k \in \{0,\dots,M-1\}$ and $\left(\mathbf{C}^{(t)}\right)^q - \mathbf{C}^{(t)}$.

Theorem 4.1. *Let us consider f a function from $\mathbb{F}_q^n$ to $\mathbb{F}_q^m$ with a memory of size ℓ and M consecutive outputs of f.*

If $M \geq \lceil \frac{\ell+1}{m} \rceil$,

Then there exists $P \in \mathbb{F}_q[\overline{X_1^{(t)}},\dots,\overline{X_n^{(t+M)}},\overline{Z_1^{(t)}},\dots,\overline{Z_m^{(t+M)}}]$, $P \neq 0$ such that $P \in \mathcal{I}_M$.

Proof. We consider the family that generates the ideal $\mathcal{I}_M$.

This family is a Gröbner basis of $\mathcal{I}_M$ according the lexicographic order $mathbf Z^{(t+M)} \succ \dots \succ \mathbf{Z}^{(t)} \succ \mathbf{C}^{(t+M)} \succ \dots \succ \mathbf{C}^{(t+1)} \succ \mathbf{C}^{(t)} \succ \mathbf{X}^{(t+M)} \succ \dots \succ \mathbf{X}^{(t+M)}$. And the ideal $\mathcal{I}_M$ is a zero dimensional ideal.

Thus $\mathcal{A}_m = \mathbb{F}_q[\mathbf{X}^{(t)},\dots,\mathbf{C}_\ell^{(t+M)}]/\mathcal{I}_M$ defined a vector space of dimension $q^{n\,M+\ell}$.

We have $q^{(n+m)M}$ distinct monomials in $\mathbb{F}_q[\mathbf{X}^{(t)},\dots,\mathbf{X}^{(t+M)},\mathbf{Z}^{(t)},\dots,\mathbf{Z}^{(t+M)}]\,/$ $\langle \left(\mathbf{X}^{(t+k)}\right)^q - \mathbf{X}^{(t+k)}, \left(\mathbf{Z}^{(t+k)}\right)^q - \mathbf{Z}^{(t+k)}, k \in \{0,\dots,M-1\}\rangle$.

We deduced that if $q^{(n+m)M} \geq q^{n\,M+\ell}$, then the set of the image of $q^{(n+m)M}$ distinct monomials in $\mathcal{A}_M$ is a linearly dependent family.

Thus for $M > \frac{\ell}{m}$, there exists $P \in \mathcal{I}_M$, $P \neq 0$. The first integer higher than $\frac{\ell}{m}$ is $\lceil \frac{\ell+1}{m} \rceil$. □

With this theorem, we can adapt the definition of Algebraic Immunity to functions with memories.

Definition 4.2. Let us consider f a function from $\mathbb{F}_q^n$ to $\mathbb{F}_q^m$ with a memory of size ℓ and M consecutive outputs of f and let us denote $\mathcal{J}_M := \mathcal{I}_M \cap \mathbb{F}_q[\mathbf{X}^{(t)}, \dots, \mathbf{X}^{(t+M)}, \mathbf{Z}^{(t)}, \dots, \mathbf{Z}^{(t+M)}]$.

We defined the Algebraic Immunities according M outputs:

$$AI_S(f, M) := \min_{P \in \mathcal{J}_M, P \neq 0} (\deg(P, \mathbf{X})),$$

$$AI_B(f, M) := \min_{P \in \mathcal{J}_M, P \neq 0} (\deg(P)).$$

4.2. **Properties**

We have simple properties on the Algebraic Immunity.

Proposition 4.3. *Let us consider f a function from $\mathbb{F}_q^n$ to $\mathbb{F}_q^m$ with memory of size ℓ and M consecutive outputs of f.*

For all $k \in \mathbb{N}$,
$AI_S(f, M + k) \leq AI_S(f, M)$ and $AI_B(f, M + k) \leq AI_B(f, M)$.

Proof. Let be $k \in \mathbb{N}$. If $P \in \mathcal{J}_M$ then $P \in \mathcal{J}_{M+k}$. □

As we prove in theorem 4.1, $\mathcal{A}_M$ is a vector space with dimension $q^{n\,M+\ell}$. So we can deduce bounds as in the previous section.

A bound of the Algebraic Immunity will be the first degree of the series with a negative or zero coefficient. These series are :

$AI_B(f, M)$	$\frac{q^{n\,M+\ell}}{1-t} - \frac{(1-t^q)^{(n+m)M}}{(1-t)^{(n+m)M+1}}$
$AI_S(f)$	$\frac{q^{(n-m)M+\ell}}{1-t} - \frac{(1-t^q)^{n\,M}}{(1-t)^{n\,M+1}}$

We notice that there are very few differences with Algebraic Immunity for a function without memories. Furthermore, the results of Algebraic Immunity for a function without memories give results for $AI_B(f, M)$ and $AI_S(f, M)$ by the simple change of variable :

$$\begin{array}{cccccc} AI_B(f) & & AI_B(f,M) & AI_S(f) & & AI_S(f,M) \\ n & \longleftrightarrow & n\,M + \ell & n - m & \longleftrightarrow & (n-m)M + \ell \\ n + m & \longleftrightarrow & (n+m)M & n & \longleftrightarrow & n\,M \end{array}$$

All the bounds given in subsection 3.2 give bounds for $AI_B(f, M)$ and $AI_S(f, M)$ by using the change of variable.

With a large number of variables, the computation of a Gröbner basis can be difficult. Knowing generators of $\mathcal{J}_M$ introduced in definition 4.2 simplifies the computation. The following theorem answers partly to this question.

Theorem 4.4. *Let us consider f a function from $\mathbb{F}_q^n$ to $\mathbb{F}_q$ with memory of size ℓ and M consecutive outputs of f.*

Let us consider $\mathcal{J}_M$ introduced in definition 4.2.

If $\mathcal{J}_\ell = \{0\}$ in
$\mathbb{F}_q[\mathbf{X}^{(t)},\ldots,\mathbf{X}^{(t+\ell-1)},z_1^{(t)},\ldots,z_1^{(t+\ell-1)}]\,/$
$\langle (\mathbf{X}^{(t+k)})^q - \mathbf{X}^{(t+k)}, \left(z_1^{(t+k)}\right)^q - z_1^{(t+k)}, k \in \{0,\ldots,M\ell-1\}\rangle$

then $\exists P \in \mathbb{F}_q[\overline{X_1^{(t)}},\ldots,\overline{X_n^{(t+\ell)}},\overline{Z_1^{(t)}},\ldots,\overline{Z_1^{(t+\ell-1)}}]$ so that

$$z_1^{(t+\ell)} = P(\mathbf{X}^{(t)},\ldots,\mathbf{X}^{(t+\ell)},z_1^{(t)},\ldots,z_1^{(t+\ell-1)})$$

And for $M \geq \ell$, $\mathcal{J}_M$ is generated by

$$z_1^{(t+i)} - P(\mathbf{X}^{(t+i-\ell-1)},\ldots,\mathbf{X}^{(t+i)},z_1^{(t+i-\ell-1)},\ldots,z_1^{(t+i-1)})$$

for all $i \in \{\ell,\ldots,M-1\}$ and field equations on variables.

Proof. With notations of the proof of theorem 4.1, we know that $\mathcal{A}_\ell$ is a vector space of dimension $q^{\ell(n+1)}$.

As $\mathbb{F}_q[\mathbf{X}^{(t)},\ldots,\mathbf{X}^{(t+\ell-1)},z_1^{(t)},\ldots,z_1^{(t+\ell-1)}]\,/$ $\langle (\mathbf{X}^{(t+k)})^q - \mathbf{X}^{(t+k)}, \left(z_1^{(t+k)}\right)^q - z_1^{(t+k)}, k \in \{0,\ldots,\ell-1\}\rangle$ have $q^{\ell(n+1)}$ distinct monomials and $\mathcal{J}_\ell = \{O\}$, the image of this monomials in $\mathcal{A}_\ell$ is a linear basis of $\mathcal{A}_\ell$.

So we can express all the memories $c_j^{(t+i)}$ as a polynomial in $\mathbf{X}^{(t+k)}$ and $z_1^{(t+k)}$, $k \in \{0,\ldots,\ell-1\}$, for all $i \in \{0,\ dots,\ell-1\}$ and $j\{1,\ dots,\ell\}$.

Then $z_1^{(t+\ell)}$ can be expressed as a polynomial in $\mathbf{X}^{(t+k)}$ and $z_1^{(t+k')}$, $k \in \{0,\ldots,\ell\}$, $k' \in \{0,\ldots,\ell-1\}$. As well as memory $\mathbf{C}^{t+\ell}$.

With iteration to M, we construct polynomials that are a Gröbner basis for the lexicographic order:
$\mathbf{X}^{(t)} \prec \cdots \prec \mathbf{X}^{(t+M)} \prec z_1^{(t)} \prec \ldots z_1^{(t+M)} \prec \mathbf{C}^{(t)} \prec \cdots \prec \mathbf{X}^{(t+M+1)}$.

Then, from proposition 2.4, the polynomial not depending on memories is a Gröbner basis of $\mathcal{J}_M$ for the deduced lexicographic order. □

An application of this theorem is the summation generator. In this Stream Cipher, the filtering function f has n inputs $\mathbf{X} =$

$(x_1^{(t)}, \ldots, x_n^{(t)})$ in $\mathbb{F}_2$, a memory $C^{(t)} \in \mathbb{Z}/2^\ell\mathbb{Z}$ with $\ell = \lceil \log_2 n \rceil$ the size of the memory and one output $z \in \mathbb{F}_2$ for a moment t. The definition is given by this relation :

$$z^{(t)} = x_1^{(t)} \oplus \cdots \oplus x_n^{(t)} \oplus C^{(t)} \qquad C^{(t+1)} = \left\lfloor \frac{x_1^{(t)} + \cdots + x_n^{(t)} \oplus C^{(t)}}{2} \right\rfloor$$

with $\oplus$, the sum on $\mathbb{F}_2$ and $+$ the sum in the ring $\mathbb{Z}/2^\ell\mathbb{Z}$.

In [11], they construct a polynomial P so that: $z^{(t+\ell)} = P(\mathbf{X}^{(t)}, \ldots, \mathbf{X}^{(t+\ell)}, z^{(t)}, \ldots, z^{(t+\ell-1)})$. In fact in this article, they have proved the hypothesis of theorem 4.4. So, we can compute with this relation the exact value of $AI_S(f, \ell)$ and compares with the bound given by article [11]:

n	2	3	4	5	6	7	8	9
Bound of [11]	2	3	4	6	6	7	8	12
$AI_S(f, \ell)$	2	3	4	5	6	7	8	9

As we can notice, it seems that the $AI_S(f, \ell)$ is equal to n.

5. conclusion

This article generalizes the Algebraic Immunity to all finite fields and also for Block Cipher. All these notions are linked to Gröbner basis with a specific order : the DRL order for Algebraic Immunity in Block Cipher and the Elimination order for Algebraic Immunity in Stream Cipher.

As the definition of a function f directly gives us a Gröbner basis for a lexicographic order, we prove properties of the ideal. Furthermore we give explicit and asymptotic bounds on the Algebraic Immunity.

We extend this notion to function with memories over any finite fields and we give a theorem that helps computing the relations implied by these Algebraic Immunity.

References

[1] F. Armknecht On the Existence of low-degree Equations for Algebraic Attacks In SASC Ecrypt workshop. Avvailable at eprint.iacr.org/2004/185/.

[2] G. Ars Applications des bases de Gröbner en cryptographie Phd Thesis in University of Rennes 1, 2005.

[3] C. Carlet Improving the algebraic immunity of resilient and nonlinear functions and constructing bent functions. On eprint.iacr.org/2004/276/.

[4] N. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In LNCS, editor, *Crypto 2003*, vol. 2729, pp. 177–194, 2003.

[5] N. Courtois. Algebraic Attacks on Combiners with Memory and Several Outputs. In ICISC 2004, LNCS, Springer.

[6] N. Courtois and W. Meier. Algebraic Attacks on Stream Ciphers with Linear Feedback. In Eurocrypt 2003, LNCS 2656, pp. 345-359, Springer.

[7] N. Courtois and J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In *Asiacrypt*, LNCS 2501, pp. 267–287, 2002.

[8] David A. Cox, John B. Little, and Don O'Shea. *Ideals, Varieties, and Algorithms : An introduction to computational algebraic geometry and commutative algebra*. Springer-Verlag, NY, 2nd edition, 1996. 536 pages.

[9] J. Golic Vectorial Boolean functions and induced algebraic equations. On eprint.iacr.org/2004/225/.

[10] M. Krause and F. Armknecht. Algrebraic Attacks on Combiners with Memory. In Crypto 2003, LNCS 2729, pp. 162-176, Springer.

[11] D. H. Lee, J. Kim, J. Hong, J. W. Han and D. Moon. Algebraic Attacks on Summation Generators. In FSE 2004, LNCS, Springer, 2004.

[12] W. Meier, E. Pasalic and C. Carlet. Algebraic Attacks and Decomposition of Boolean Functions. In Eurocrypt 2004, pp. 474-491, LNCS 3027, Springer, 2004.

[13] J. Patarin, Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88, Crypto'95, Springer, LNCS 963, pp. 248-261, 1995.

A STATISTICAL APPROACH ON THE NUMBER OF FUNCTIONS SATISFYING STRICT AVALANCHE CRITERIA

E. YILDIRIM SAYGI[1], Z. SAYGI[1], M. SÖNMEZ TURAN[1] and A. DOĞANAKSOY[2]

Abstract. Boolean functions play an important role in the design of both block and stream ciphers. One of the important criteria that a Boolean function should satisfy is the Strict Avalanche Criteria (SAC). In this study, we concentrated on the number of functions satisfying SAC. A special formula, to find the number of functions with particular difference distribution vector is presented. For $n = 5, 6, 7$ and 8, the number of functions satisfying SAC is estimated using a statistical approach and 99.9% confidence intervals are given. Also, another confidence interval for the number of balanced Boolean functions satisfying SAC is presented.

Keywords: SAC, Difference Distribution Vector, Boolean Functions, Number of functions satisfying SAC.

1. **Introduction**

Boolean functions, as basic elements of most building blocks used in cryptography, play an important role in the design of both block and stream ciphers. For security issues, they are required to

[1] Institute of Applied Mathematics, Middle East Technical University, 06531 Ankara, Turkey
[2] Department of Mathematics, Middle East Technical University, 06531 Ankara, Turkey
The National Research Institute of Electronics and Cryptology - NRIEC (UEKAE), 41470 Gebze, Turkey
email: e110925,saygi,msonmez,aldoks@metu.edu.tr
Phone: +90 (312) 210 53 54 Fax: +90 (312) 210 29 85

satisfy various conflicting criteria such as balancedness, high nonlinearity and strict avalanche criteria (SAC). Balancedness is one of the primary cryptographic criteria. Nonlinearity of a Boolean function is its minimum Hamming distance to the set of affine functions. High nonlinearity is necessary to prevent the attacks that may use affine approximations of the function. It is also natural to expect each input bit to have an effect on the output bits. A function with this property is said to be complete. The concept of avalanche effect, which means an average of one half of the output bits should be changed whenever a single input bit is complemented, was first introduced by Feistel [2]. Webster and Tavares [7] combined avalanche and completeness and introduced the concept of SAC. A cryptographic function is said to satisfy the SAC if whenever a single input bit is complemented, each output bit changes with a probability of one half.

All of these properties are vital for resisting different kind of attacks. Due to the trade-off between these criteria, it is not possible to obtain the best achievable values for each of them separately. Also, it is known that strict fulfillment of some criterion prevents the optimal fulfillment of others.

There are various studies on construction methods for Boolean functions satisfying some of the mentioned properties. Millan et al. [3] concentrated on construction of balanced Boolean functions with high nonlinearity by hill climbing and genetic algorithms. Pasalic et al. [5] proposed algorithms for construction of resilient and correlation immune Boolean functions with high nonlinearity. It is very important to have an idea about the size of the set of functions that satisfy desired properties, before focusing on construction methods. It may happen that a randomly chosen Boolean function satisfies the property with a high probability.

In this study, we give our attention to the number of Boolean functions satisfying SAC. Also, the number of functions with particular difference distribution vectors is studied. The exact formula for a special case is given. Results of some statistical observations are compared to the exact values.

For small values of n, it is easy to calculate the number of functions satisfying SAC by enumeration. But when n gets larger ($n > 5$), enumeration becomes infeasible. In [1], a filtering procedure that identifies the functions that do not satisfy SAC is proposed. However, even considering this procedure it is still infeasible to determine the exact numbers. Rather than using inefficient exact methods, we used a statistical approach which results in a very short time with 99.9% confidence interval for the number of functions satisfying SAC for $n = 6, 7$ and 8. Also, another confidence interval for the number of balanced Boolean functions satisfying SAC is given.

Our paper is organized as follows. The preliminaries and some background knowledge are represented in Section 2. In Section 3, a formula for the number of functions with particular difference distribution vectors is given and some properties are observed. In Section 4, our statistical approach to find the number of functions satisfying SAC is described in detail and the results are tabulated. Finally, we recommend some future studies.

2. **Preliminaries**

Let V_n be the vector space of all n-tuples of elements from $GF(2)$. A Boolean function is a $GF(2)$ valued map defined on V_n and the set of all Boolean functions on V_n is denoted by $\mathcal{F}_n$.

The *Hamming weight* of a function is defined as the number of nonzero entries in the truth table (T_f) of f and is denoted by $w(f)$.

A function is called *balanced* if the number of 1's is equal to the number of 0's in its truth table. Clearly, the number of balanced functions in $\mathcal{F}_n$ is $\binom{2^n}{2^{n-1}}$.

For a given $f \in F_n$, $S_i(f)$ is defined by

$$S_i(f) = \sum_x f(x) \oplus f(x \oplus e_i) \quad i = 1, \dots, n$$

where e_i is the vector having only one nonzero entry in the i-th position.

By $S(f)$, we denote the vector $(S_1(f), S_2(f), \cdots, S_n(f))$ which is called the *difference distribution vector* of f.

It follows that $f \in F_n$ satisfies SAC if and only if $S_i(f) = 2^{n-1}$ for all $i \in \{1, 2, \dots, n\}$. For any fixed $a \in \{1, \dots, 2^n\}$ the number of functions satisfying $S_i(f) = a$ does not depend on the choice of $i \in \{1, \dots, n\}$. That is ,

$$|\{f \in F_n | S_i(f) = a\}| = |\{f \in F_n | S_j(f) = a\}|$$

for any pair of $i, j \in \{1, \dots, n\}$. For $a = 2^{n-1}$, this number is denoted by $S(n,1)$.

This idea can be generalized to define $S(n,k)$ as follows. The number of functions satisfying the condition $S_{i_1}(f) = a_1, S_{i_2}(f) = a_2, \dots, S_{i_k}(f) = a_k$, does not depend on the choice of the subset $\{i_1, \dots, i_k\} \subset \{1, \dots, n\}$. Thus, $S(n,k)$ is given as the number of functions satisfying $S_{i_1}(f) = S_{i_2}(f) = \cdots = S_{i_k}(f) = 2^{n-1}$, where $\{i_1, \dots, i_k\}$ is any subset of $\{1, \dots, n\}$ with cardinality k.

3. Number of functions with a Particular Difference Distribution Vector

In this section, we deal with the number of functions having some particular types of difference distribution vector. In [4], a computation of $S(n,2)$ is given by

$$S(n,2) = \textstyle\sum_{i=0}^{2^{n-3}} \binom{2^{n-2}}{2i} 8^{2^{n-2}-2i} 2^{2i} \sum_{j=0}^{i} \binom{2i}{2j} \binom{2j}{j} \binom{2i-2j}{i-j}. \quad (1)$$

We give a more general, but yet more simple formula which computes the number of functions $f \in F_n$ for which $S_1(f) = \lambda_1$, $S_2(f) = \lambda_2$ for arbitrarily chosen nonnegative even integers λ_1, λ_2.

Theorem 3.1. *Given nonnegative integers $\lambda_1, \lambda_2 \leq 2^n$, the number of functions in $\mathcal{F}_n$ such that $S_1 = \lambda_1$, $S_2 = \lambda_2$ is*

$$\begin{array}{ll} 2^{2^{n-2}} \sum_{t=0}^{N_1} \binom{2^{n-2}}{2t} \binom{2^{n-2}-2t}{\frac{\lambda_1}{4}-t} \binom{2^{n-2}-2t}{\frac{\lambda_2}{4}-t} 2^{4t} & , \ if \ \lambda_i \equiv 0 \ (mod\ 4), \\ 2^{2^{n-2}} \sum_{t=0}^{N_2} \binom{2^{n-2}}{2t+1} \binom{2^{n-2}-2t-1}{\frac{\lambda_1-2}{4}-t} \binom{2^{n-2}-2t-1}{\frac{\lambda_2-2}{4}-t} 2^{4t}, & if \ \lambda_i \equiv 2 \ (mod\ 4), \\ \qquad 0 & , \ otherwise, \end{array}$$

where $i = 1, 2$, $N_1 = min(2^{n-3}, \frac{\lambda_1}{4}, \frac{\lambda_2}{4})$ and
$N_2 = min(2^{n-3} - 1, \frac{\lambda_1-2}{4}, \frac{\lambda_2-2}{4})$.

Proof. We here give just an outline of the proof, for a complete proof one may refer to [9]. Given $f \in \mathcal{F}_n$ there exists a unique quadruple g_1, g_2, g_3, g_4 of functions in F_{n-2} so that the truth table of f is

$$T_{g_4} || (T_{g_2} \oplus T_{g_4}) || (T_{g_1} \oplus T_{g_4}) || (T_{g_1} \oplus T_{g_2} \oplus T_{g_3} \oplus T_{g_4})$$

where $||$ stands for the concatenation of the truth tables. Then,

$$\lambda_1 = 4w(g_1) + 2w(g_3) - 4w(g_1 g_3),$$

$$\lambda_2 = 4w(g_2) + 2w(g_3) - 4w(g_2g_3).$$

After fixing g_3 and considering the possibilities of g_1 and g_2 we obtain the result. □

An immediate consequence of this theorem is

Corollary 3.2.

$$S(n,2) = 2^{2^{n-2}} \sum_{t=0}^{2^{n-3}} \binom{2^{n-2}}{2t} \binom{2^{n-2}-2t}{2^{n-3}-t}^2 2^{4t}.$$

Note that, considering the observation in [8] which states that

$$\sum_{j=0}^{i} \binom{2i}{2j} \binom{2j}{j} \binom{2i-2j}{i-j} = \binom{2i}{i}^2,$$

the result in (1) directly reduces to the expression given in the corollary.

Computed Values vs. Statistical Values

In this study our main aim is to find the number of functions satisfying SAC. Since there is no explicit formula of $S(n,k)$ for $k > 2$, we decided to use a statistical approach. For $n = 8$, the total number of functions is 2^{256}, it is even impossible to choose 5% of the population size as a sample size. However, to verify that a sample size of $N = 100,000,000$ is enough for deriving some conclusions, we perform the following experiment.

For $n = 8$, using the formula given in Theorem 1, we calculate the number of functions having $S_1 = \lambda_1$ and $S_2 = \lambda_2$, and divide them to the total number of functions to obtain the exact proportions given in Table 1. These vectors are chosen such that $100 \leq \lambda_1, \lambda_2 \leq 128$.

Using a sample size of $100,000,000$, we also count the number of functions having $S_1 = \lambda_1$ and $S_2 = \lambda_2$, for $n = 8$. In Table 1, a list of observed proportions and normalized errors are given. Normalized errors are calculated by $|(e_i - o_i)|/e_i$ where e_i and o_i are expected and observed values, respectively. In this table, the maximum, minimum and average absolute errors are 0.019018828, 0.0000042664 and 0.002904854, respectively. Obtaining such small error rates encouraged us to extend the statistical method for finding the number of functions satisfying SAC for $n = 6, 7, 8$.

For $\lambda_1 = 128$ and $\lambda_2 = 128$ case, the normalized error rate is 0.0016786893.

λ_1	λ_2	Exact Proportion	Observed Proportion	Normalized Error Proportion
102	110	0.0002061909	0.0002061900	0.0000042664
118	118	0.0045613921	0.0045614600	0.0000148814
116	128	0.0056626272	0.0056622700	0.0000630780
120	124	0.0072781823	0.0072773900	0.0001088541
102	126	0.0006860166	0.0006861500	0.0001945281
100	120	0.0003576980	0.0003546100	0.0086328875
106	110	0.0004340272	0.0004301900	0.0088410131
102	106	0.0001126250	0.0001108000	0.0162041468
100	100	0.0000239131	0.0000234700	0.0185276455
100	104	0.0000525211	0.0000535200	0.0190188280

Table 1. The best and worst five proportions of the exact and observed proportion of functions having λ_1 and λ_2 in their difference distribution table.

4. Number of Boolean Functions Satisfying SAC

The number of Boolean functions satisfying SAC, $S(n, n)$, is our main concern. For small values of n, it is possible to count the number by enumeration. By using statistics, we mainly tried to estimate two proportions:

P_1: the proportion of SAC satisfying Boolean functions to the number of Boolean functions,

P_2: the proportion of SAC satisfying balanced Boolean functions to the number of balanced Boolean functions.

Exact P_1 and P_2 values for $n \leq 5$ are listed in Table 2.

n	P_1	P_2
2	0.5	0
3	0.25	0.457142857
4	0.062988	0.106293706
5	0.006408	0.011507546

Table 2. Exact P_1 and P_2 values for $n \leq 5$.

n	number of sets	number of functions in each set
5	40	10^6
6	40	10^7
7	40	10^8
8	30	10^9

Table 3. Number of sets and number of functions in each set.

We use a statistical approach to estimate P_1 and P_2, that is to say the number of functions satisfying SAC for $n = 6, 7$ and 8. Although, the desired number is available for $n = 5$, the same statistical calculations are used as a control group. For each value of n, sets containing random functions are generated allowing repetitions. The number of sets and the number of Boolean functions in each set are given in Table 3. These sample sizes are statistically enough to make estimates with small error probabilities. Our approach is as follows: To calculate P_1 and P_2 for a given $n \geq 5$, from each data set, the number of functions satisfying SAC is calculated and is divided to the cardinality of the set. Using the obtained proportion values from each set, a confidence interval is generated for the exact proportion and therefore, the confidence interval for exact the number of functions satisfying SAC is obtained.

To estimate a confidence interval for the proportion values, an experiment is performed to estimate the distribution of proportions for $n = 6$. The total of 100 proportions are distributed over equally length intervals and a statistical goodness-of-fit test is applied to the proportions to test whether they come from a normal distribution. The number of functions satisfying SAC is transformed to standard normal distribution by the transformation, $z = (x - \mu)/\sigma$ where x is the number of functions satisfying SAC from the sets and μ is the average value and σ is the standard deviation obtained from the sample. The z-values are distributed over 8 categories. Expected and observed frequencies for each category are given in Table 4.

Using the chi-square goodness-of-fit test and the values in Table 4, the test statistics is calculated by the formula $\sum(e_i - o_i)^2/e_i$ where e_i and o_i are expected and observed values, respectively and obtained as 4.64. The tabulated chi-square value with $5(= 8-2-1)$ degrees of freedom is $\chi(0.99, 5) = 15.09$ with $\alpha = 0.01$. Since

Category	Boundaries	Expected Probability	Expected number of z-values	Observed number of z-values
1	$x < -1.15$	0.125	12.5	13
2	$-1.15 < x < -0.67$	0.125	12.5	18
3	$-0.67 < x < -0.32$	0.125	12.5	12
4	$-0.32 < x < 0.00$	0.125	12.5	9
5	$0.00 < x < 0.32$	0.125	12.5	12
6	$0.32 < x < 0.67$	0.125	12.5	9
7	$0.67 < x < 1.15$	0.125	12.5	13
8	$1.15 < x$	0.125	12.5	14

TABLE 4. Goodness-of-fit statistical test calculations.

$4.64 < 15.09$, there is no statistical evidence that the data do not come from normal distribution.

By the knowledge that the proportions come from normal distributions, the estimates for the number of Boolean functions satisfying SAC for values $n = 5, 6, 7$ and 8 are tabulated in Table 5. The exact proportion of functions satisfying SAC for $n = 5$ is given in Table 2 as 0.006408, and this proportion lies in the proposed interval.

n	Average proportion	% 99.9 Confidence Interval	Estimate for number functions satisfying SAC
5	0.006421	0.006403 - 0.006438	27577985,01
6	0.000301	0.000299 - 0.000302	5,55247E+15
7	7.10E-06	7.03E-06 - 7.17E-06	2,416E+33
8	8.47E-08	8.22E-08 - 8.72E-08	9,80759E+69

TABLE 5. Estimate Proportion of Boolean functions satisfying SAC and corresponding confidence intervals.

Also, to estimate the number of balanced Boolean functions satisfying SAC, the same procedure is applied and the results are given in Table 6.

n	Average proportion	% 99.9 Confidence Interval	Estimate for number functions satisfying SAC
5	0.011545	0.011518-0.011571	6939473,103
6	0.000555	0.000552-0.000557	1,01711E+15
7	1.38E-05	1.37E-05-1.38E-05	3,31E+32
8	1.67E-07	1.63E-07-1.70E-07	9,63E+68

TABLE 6. Estimate Proportion of balanced Boolean functions satisfying SAC and corresponding confidence intervals.

5. Conclusion

In this study, our interest is to find the number of functions satisfying SAC. First, we presented a formula to find the number of functions with particular difference distribution vector and compared the exact numbers to statistics. We used a statistical approach and obtained 99.9% confidence interval for the number of functions satisfying SAC for $n = 6, 7$ and 8. Also, another confidence interval for the number of balanced Boolean functions satisfying SAC is given. For future work, an explicit formula of $S(n, k)$ for $k \geq 3$ will be studied. Also, statistical estimations will be done for larger values of n.

References

[1] Falkowski B.J. and Kannurao S., *Strict Avalanche Criterion in Boolean Functions A Spectral Approach*, Proceedings of IEEE International Symposium on Circuits and Systems (34th ISCAS), Sydney, Australia, vol. 2, pp. 641-644, May 2001.

[2] Feistel H., *Cryptography and Computer Privacy*, Scienctific American, 228(5): 15-23, 1973.

[3] Millan W., Clark A., Dawson E., *Heuristic Design of Cryptographically Strong Balanced Boolean Functions*, Advances in Cryptology, Eurocrypt'98, 1998.

[4] O'Connor L., *An Upper Bound on the Number of Functions Satisfying the Strict Avalanche Criterion*, Information Processing Letters 52(6): 325-327, 1994.

[5] Pasalic E., Johansson T., Sarkar P., and Maitra S., *New Constructions of Resilient and Correlation Immune Boolean Functions Achieving Upper Bounds on Nonlinearity*, in Proc. Int. Workshop Coding and Cryptography, WCC 2001, pp. 425-434, 2001.

[6] Preneel B., Leekwijck W.V., Linden L.V., Govaerts R., and Vandewalle J., *Propagation Characteristics of Boolean Functions*, Advances in Cryptology - EUROCRYPT90 (Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, New York 1991) 437: 155-165, 1990.

[7] Webster A.F. and Tavares S.E., *On the Design of S-boxes*, Advances in Cryptology - CRYPTO'85 ed. H.C. Williams (Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, Newyork) 218: 523-524, 1985.

[8] Youssef A.M., Cusick T.W., Stanica P. and Tavares S.E., *New Bounds on the Number of Functions Satisfying the Strict Avalanche Criterion*, Workshop on Selected Areas in Cryptography, SAC'96, pp. 49-56, August 1996.

[9] Yıldırım E., *Counting and Constructing Boolean Functions with Particular Difference Distribution Vectors* Master Thesis, Department of Cryptography, Institute of Applied Mathematics, METU, 2004.

The authors would like to thank “A. Kadir Altan” for his help.

ASYMPTOTIC DISTRIBUTION FOR THE NONLINEARITY OF BOOLEAN FUNCTIONS

F. Rodier[1]

Abstract. I recall some properties of the distribution for the nonlinearity of Boolean functions, and I introduce new ones which are related to large deviation theorems in probability.

Keywords: Boolean function, nonlinearity, sum-of-square indicator, large deviation.

1. Introduction

The nonlinearity of a Boolean function $f : \mathbb{F}_2^m \longrightarrow \mathbb{F}_2$ is the distance from f to the set of affine functions with m variables (see § 2.2). It is an important concept. We use it in cryptography (cf. [2,3,5]) to construct strong cryptosystems (symmetric ciphers), and in coding theory with the old problem of the covering radius of the first order Reed-Muller codes (cf. [4, 10]).

The nonlinearity is bounded above by $2^{m-1} - 2^{m/2-1}$. This bound is reached by bent functions [8] which exist only if the number of variables m of the Boolean functions is even. For security reasons in cryptography, and also because Boolean functions also need to have other properties such as balancedness or high algebraic degree, it is important to have the possibility of choosing among many Boolean functions, not only bent functions, but also functions which are almost bent and hence to study the distribution of nonlinearity.

Except for the paper by Chuan-Kun Wu [16] who studies the distribution of Boolean functions with nonlinearity $\leq 2^{m-2}$, the

[1] Institut de Mathématiques de Luminy – C.N.R.S.– Marseille – France
email: rodier@iml.univ-mrs.fr

distribution of nonlinearity was not known until it appeared on papers by Carlet [2,3] and independently by Olejár and Stanek [9] who proved that most of the Boolean functions have a nonlinearity greater than $2^{m-1} - 2^{m/2-1}\sqrt{2m\log 2}$. Then I got more precise results in [12, 13], proving that most of them have indeed a nonlinearity close to $2^{m-1} - 2^{m/2-1}\sqrt{2m\log 2}$.

It is very hard to find results on this distribution. We therefore study the simpler criterion of the "sum of square", linked to the propagation criterion for Boolean functions. This criterion has been studied by Xian-Mo Zhang and Yuliang Zheng [17], or by P. Stănică [15]. His relationship with non-linearity was studied by A. Canteaut et al. [1].

We explore what can be done in this respect, and what kind of result we should obtain by considering large deviation theorems in probabilities.

2. Preliminaries

2.1. Boolean functions

Let m be a positive integer and $q = 2^m$.

Definition 2.1. A Boolean function with m variables is a map from the space $V_m = \mathbb{F}_2^m$ into $\mathbb{F}_2$.

A Boolean function is linear if it is a linear form on the vector space $\mathbb{F}_2^m$. It is affine if it is equal to a linear function up to addition of a constant.

2.2. Nonlinearity

Definition 2.2. We call nonlinearity of a Boolean function $f : V_m \longrightarrow \mathbb{F}_2$ the distance from f to the set of affine functions with m variables:

$$nl(f) = \min_{h \text{ affine}} d(f,h)$$

where d is the Hamming distance.

One can show that the nonlinearity is equal to

$$nl(f) = 2^{m-1} - \frac{1}{2}S(f)$$

where

$$S(f) = \max_{v\in V_m}\Big|\sum_{x\in V_m} \chi(f(x) + v\cdot x)\Big|$$

and $v \cdot x$ denote the usual scalar product in V_m and χ denotes the non trivial character of $\mathbb{F}_2$ with values in the complex numbers: $\chi(x) = (-1)^x$. We call $S(f)$ the *spectral amplitude* of the Boolean function f.

We have by Parseval identity, for $f \in \mathcal{B}_m$:

$$\sqrt{q} \le S(f) \le q.$$

2.3. The sum-of-square indicator

Let f be a Boolean function on V_m. Zhang and Zheng introduced the sum-of-square indicator [17]:

$$\sigma_f = \frac{1}{q}\sum_{x\in V_m} \widehat{f}(x)^4 = \|\widehat{f}\|_4^4$$

where $\widehat{f}$ is defined by $\widehat{f}(u) = \sum_{x\in V_m} \chi(f(x) + u\cdot x)$. We remark that $S(f) = \|\widehat{f}\|_\infty$, hence $2^{2m} = \|\widehat{f}\|_2^4 \le \sigma_f \le S(f)^4$.

2.4. The spaces of Boolean functions with m variables

We define $\mathcal{B}_m$ as the algebra of Boolean functions on V_m.

To study asymptotically Boolean functions, we will need the notion of Boolean functions with an infinity of variables and we will introduce a probability measure on them to be able to state almost sure results.

We define V_∞ as the space of infinite sequences of elements of $\mathbb{F}_2$ which are almost all equal to zero, and $\mathcal{B} = \mathcal{B}_\infty$ as the algebra of Boolean functions on V_∞. We have the restriction mappings

$$\pi_m : \mathcal{B}_\infty \longrightarrow \mathcal{B}_m : f \longmapsto f_m = f\,|_{V_m}\ .$$

We will consider the equiprobability on $\mathcal{B}_m$ and we will endow $\mathcal{B}$ with a probability which will be the Haar measure on it with total mass 1. In other words, for each $f \in \mathcal{B}_m$, the probability of the event $\pi_m^{-1} f = \{g \in \mathcal{B} \mid g|_{V_m} = f\}$ is given by

$$\underline{\mathrm{P}}(\pi_m^{-1} f) = \frac{1}{2^q}$$

where $q = |V_m| = 2^m$.

3. **Known results**

3.1. **Limit of** $S(f)$

The next result shows that in fact $S(f)$ is rather close to $\sqrt{q}$. It is proved in [12–14].

Theorem 3.1. *If f is a boolean function in $\mathcal{B}_\infty$, then almost surely:*

$$\lim_{m\to\infty} \frac{S(f_m)}{\sqrt{2q\log q}} = 1,$$

3.2. **Limit of** σ_f

We have [12]: $q^2 \leq \sigma_f \leq q^3$. Then there is the following results [12, 13].

Proposition 3.2. *If f is a Boolean function on V_m, and t a positive real number,*

$$\underline{P}\left(\left|\frac{\sigma_f}{q^2} - 3\right| \geq t\right) \leq \frac{40}{t^2 q},$$

In [12, 13] we get a slightly better inequality, but we have:

$$\begin{aligned}&\underline{\mathrm{P}}\left(\left|\frac{\sigma_f}{q^2} - 3\right| \geq t\right)\\ &\quad\leq \underline{\mathrm{P}}\left(\left|\frac{\sigma_f}{q^2} - 3 + \frac{2}{q}\right| \geq t + \frac{2}{q}\right) \leq \frac{40}{(t+\frac{2}{q})^2 q} \leq \frac{40}{t^2 q}.\end{aligned} \tag{1}$$

Corollary 3.3. *If $f \in \mathcal{B}$, one has almost surely*

$$\lim_m \frac{\sigma_{f_m}}{2^{2m}} = 3.$$

3.3. **An old conjecture**

Bent functions are such that $S(f) = \sqrt{q} = 2^{m/2}$. In 1983, Patterson and Wiedemann [10] conjectured that

$$\min_f S(f) \sim 2^{m/2} \quad \text{for} \quad f \in \mathcal{B}_m.$$

We can also make a weaker conjecture:

$$\text{if} \quad f \in \mathcal{B}_m, \quad \text{one has} \quad \lim_m \min_{f \in V_m} \frac{\sigma_f}{2^{2m}} = 1. \tag{2}$$

4. Some new conjectures

We define the random variables with values in $\mathbb{R}$ and depending of a in $V_m^\times = V_m - \{0\}$:

$$Y_a = \frac{1}{q}\Bigg(\sum_{x_1+x_2=a} \chi\Big(f(x_1)+f(x_2)\Big)\Bigg)^2.$$

as in [12, 13]. We can write

$$\frac{\sigma_f}{q^2} - 1 = \frac{1}{q}\sum_{a \in V_m^\times} Y_a.$$

As the Y_a have the same distribution, and as their limit is the distribution of density

$$\frac{1}{2\sqrt{\pi x}} e^{-x/4}\mathbf{1}_{(x>0)}$$

we can expect that a large deviation theorem may be applied, to give an estimation of the probability that $\sigma_f = q^2$ (cf. [6]).

As an ingredient of Gärdner-Ellis theorem, let us define

$$\phi_q(u) = \frac{1}{q}\log \mathcal{E}\Big(\exp\big(u \sum_{a \in V_m^\times} Y_a\big)\Big)$$

where $\mathcal{E}$ denotes the expectation of a random variable on $\mathcal{B}_m$ or $\mathcal{B}_\infty$. We will see that the behaviour of this function have consequences on the distribution of the sum-of-square indicator.

4.1. General properties of the function ϕ_q

Proposition 4.1. *We have:*

- $\phi_q(u) < 0$ *for* $u < 0$*;* $\phi_q(0) = 0$*;* $\phi_q(u) > 0$ *for* $u > 0$.
- ϕ_q *is convex in* u.
- $\phi_q'(0) = 2(1 - \frac{1}{q})$.

- *As ϕ_q is convex, we have $\phi_q(u) \geq \phi'_q(0)u$.*
- *ϕ_q is increasing in u.*

Proof. These are well known properties of this function (cf. [6]). □

Proposition 4.2. *For m even, one has $-\log 2 \leq \phi_q(u)$.*

Proof. Let a_i be the different values of the random variable $\frac{\sigma_f}{q^2} - 1$ each occurring with probability $\underline{\mathrm{p}}_i$. Suppose that $a_0 = 0$. We have:

$$\begin{aligned}\phi_q(u) &= \frac{1}{q}\log\left(\sum_i \underline{\mathrm{p}}_i \exp(ua_i)\right)\\ &= \frac{1}{q}\log\left(\sum_i \frac{\underline{\mathrm{p}}_i}{\underline{\mathrm{p}}_0}\exp(u(a_i - a_0))\right) + \frac{1}{q}\log\left(\underline{\mathrm{p}}_0\right).\end{aligned}$$

As m is even there exist bent functions, hence the probability that $\sigma_f = q^2$ (that is $\forall a \neq 0,\ Y_a = 0$) is larger than $1/2^q$, hence $\underline{\mathrm{p}}_0 \geq 1/2^q$. Therefore

$$\phi_q(u) = \frac{1}{q}\log\left(\underline{\mathrm{p}}_0\right) + o(u) \geq -\log 2 + o(u)$$

when $u \to -\infty$. We conclude, as $\phi_q(u)$ is convex. □

4.2. The function ϕ

We conjecture that the functions ϕ_q have a pointwise limit ϕ in $\mathbf{R} \cup \{+\infty\}$. From the properties of the function ϕ_q, we get:

Proposition 4.3.

- *The function $\phi(u)$ is convex in u,*
- *$\phi(0) = 0$,*
- *The function $\phi(u)$ is increasing*
- *For every u, $\phi(u) \geq -\log 2$.*

4.3. Legendre transform of ϕ

The Legendre transform of ϕ is defined by

$$I(x) = \sup_u (ux - \rho(u)).$$

From the preceding properties of $\phi(u)$, we get the following results.

Proposition 4.4.

- *$I(x)$ is a convex nonnegative function on* $\mathbf{R}$.
- $I(2) = 0$
- *the derivative at* 2 *is* $I'(2) = 0$
- $I(0) \leq \log 2$

4.4. Conjecture on the distribution of σ_f for σ_f small

We suppose again that ϕ exists and moreover that it is differentiable on $]-\infty,\ 0[$.

Then we expect to have by Gärtner-Ellis theorem:

Conjecture 4.5. For $a < b \leq 2$, we have

$$\lim_{q\to\infty} \frac{1}{q} \log \underline{\mathrm{P}} \left(\frac{\sigma_f}{q^2} - 1 \in [a,\ b] \right) \quad = \quad -I(b).$$

In particular, as f is bent if and only if $\sigma_f = q^2$, we have:

$$\lim_{q\to\infty} \frac{1}{q} \log \underline{\mathrm{P}} \left(f \text{ is bent} \right) = -I(0).$$

We would deduce from the previous proposition and proposition 4.4 that for given ϵ, for every large q, there exists f such that

$$\frac{1}{q^2} \|\widehat{f}\|_4^4 - 1 < \epsilon$$

and hence it proves the conjecture (2).

5. Bounds for σ_f

5.1. The distribution of σ_f for σ_f large

We can compute the function I for $s > 2$, but unfortunately this gives a trivial result on probabilities.

Proposition 5.1. *There is a subfamily of the functions ϕ_q which tend pointwise to infinity when $q \to \infty$ for $u > 0$.*

Proof. For a given $s \geq 2$ we have, for q large ($q \geq s - 2$)

$$\underline{\mathrm{P}} \left(\frac{\sigma_f}{q^2} - 1 > s \right) \geq 1/2^q,$$

as there is a function such that $\sigma_f = q^3$. Hence

$$\frac{1}{q}\log \underline{\mathrm{P}}\left(\frac{\sigma_f}{q^2} - 1 > s\right) \geq -\log 2.$$

If the functions ϕ_q tend pointwise to infinity when $q \to \infty$ for $u > 0$, there is nothing to prove.

If not, as the ϕ_q are increasing functions, there is $u_0 > 0$ (possibly infinite) such that $\phi_q(u)$ do not tend to infinity for $0 \leq u < u_0$ and $\phi_q(u) \to \infty$ for $u > u_0$. So for any $u_1 < u_0$ there is an infinite family of the functions ϕ_q which are bounded for $0 \leq u \leq u_1$. Consequently, we can choose a subfamily which converges uniformly to a limit ϕ on $[0,\ u_2]$ for any $u_2 < u_1$ (cf [11]). Choosing u_1 and u_2 tending to u_0, and taking subfamilies, we can conclude that there is a family of ϕ_q's which tend pointwise to a function ϕ, for $u \geq 0$, and that $\phi(u) = +\infty$ for $u > u_0$. Let I be the Legendre transform of ϕ.

By Gärtner-Ellis theorem (cf. [6])

$$-I(s) \geq \limsup_{q\to\infty} \frac{1}{q}\log \underline{\mathrm{P}}\left(\frac{\sigma_f}{q^2} - 1 > s\right) \geq -\log 2.$$

Hence $I(s) \leq \log 2$ for any $s \geq 2$. As I is a convex function, we have

$$I(s) = 0$$

for $s > 2$ and the properties of Legendre transform implies that the functions ϕ_q tend to infinity for $u > 0$. □

Corollary 5.2. *If the function ϕ exists as in section 4.2, then $I(x) = 0$ for $x \geq 2$.*

Remark 5.1. *As a consequence of this proposition, we cannot expect such an inequality as:*

$$\underline{P}\left(\frac{\sigma_f}{q^2} - 1 > s\right) \leq A^{-q},$$

for fixed $s \geq 2$ and $A > 1$.

5.2. Bounds on the moments of σ_f

A main ingredient of large deviation theorem is the higher moments. Using them, we get bounds on $\mathcal{E}\Big(\exp\big(u\sum_{a\in V_m^\times} Y_a\big)\Big)$ or on $\mathcal{E}\big(\sum_{a\in V_m^\times} Y_a\big)^s$.

Proposition 5.3. *Let s be an integer such that $2^s < q$. Bounds on $\mathcal{E}(\sum Y_a)^s$ are given by*

$$\begin{aligned}(2(q-1))^s &\leq \mathcal{E}(\sum Y_a)^s \\ &\leq \; 2^s q^s \exp \frac{8s^3 - 3.2^s + 3}{3q} + (q-1)^{s-1}\,(2^s + s - 3)\,\frac{(2s)!}{(s)!}.\end{aligned}$$

Proof. A lower bound is given by Jensen's inequality:

$$\mathcal{E}(\sum_{a \in V_m^\times} Y_a)^s \geq (\mathcal{E} \sum_{a \in V_m^\times} Y_a)^s = (\sum_{a \in V_m^\times} \mathcal{E} Y_a)^s = (2(q-1))^s.$$

The proof of the upper bound is given in section 6. □

5.3. **Pearson's bound**

With these bounds, we can prove better bounds than in proposition 3.2.

Proposition 5.4. *If f is a Boolean function on V_m, t a positive real number, s an integer and $q > 4^s$, we have*

$$\underline{P}\left(\left|\frac{\sigma_f}{q^2} - 3\right| \geq t\right) \leq \frac{\alpha_{2s}}{qt^{2s}}$$

with $\alpha_{2s} = \dfrac{2^{2s+1}(4s)!}{(2s)!}$ *for q large enough.*

Proof. Pearson's inequality (cf. [7]) states that if X is a random variable whose mean is μ and $\mathcal{E}|X - \mu|^r = \beta_r$ then

$$\underline{\mathrm{P}}(|X - \mu|^r \geq \beta_r \lambda^r) \leq \frac{1}{\lambda^r}.$$

We can evaluate the bounds on $\mathcal{E}\left|\frac{\sigma_f}{q^2} - 3 + \frac{2}{q}\right|^r$ by evaluating bounds on $\mathcal{E}(\sum_{a \in V_m^\times} Y_a)^r$. This is done in section 7. Finally we use the relations (1). □

Corollary 5.5. *If moreover $8s\log(2) + 2s\log s \leq \log q$, one has*

$$\frac{1}{s}\log \underline{P}\left(\left|\frac{\sigma_f}{q^2} - 3\right| \geq t\right) \leq -2\log t + o(\frac{1}{s}).$$

Proof. We use Stirling's formula. □

Remark 5.2. *Pearson's bound only works for large t. For instance, the bound for $s = 2$ is better than the one for $s = 1$ if $t \geq 25$ roughly.*

5.4. Bound for σ_f large

The following proposition limits the probability that $\frac{\sigma_f}{q^2} - 1$ is too large.

Proposition 5.6. *One has, for every $t \geq 2$:*

$$\underline{P}\left(\frac{\sigma_f}{q^2} - 1 \geq t\right) \leq \frac{\sqrt{t}}{\sqrt{2}} \exp((2-t)/4)$$

Proof. We have, by the exponential overbound (cf. [6]), for every $u \geq 0$:

$$\begin{aligned}
\underline{\mathrm{P}}\left(\frac{\sigma_f}{q^2} - 1 \geq t\right) &\leq \mathcal{E}\left(\exp(\frac{u}{q}\sum_{a\in V_m^\times} Y_a)\right)\exp(-ut) \\
&\leq \sum_n \left(\frac{u^n}{q^n n!}\mathcal{E}\left(\sum_{a\in V_m^\times} Y_a\right)^n\right)\exp(-ut) \\
&\leq \sum_n \left(\frac{u^n}{n!}\mathcal{E}\left(Y_a^n\right)\right)\exp(-ut) \\
&\qquad\qquad \text{by Holder's inequality} \\
&\leq \sum_n \left(u^n\frac{(2n)!}{n!^2}\right)\exp(-ut) \\
&\leq \sqrt{\frac{1}{1-4t}}e^{-ut}
\end{aligned}$$

By choosing the u conveniently ($u = \frac{t-2}{4t}$), we find the result. □

Remark 5.3. *This rough bound is very inaccurate for small values of t, but is more precise for large values.*

6. Proof of the upper bound on $\mathcal{E}(\sum_{a\in V_m^\times} Y_a)^s$

The upper bound is obtained by expanding $\mathcal{E}(\sum Y_a)^s$:

$$\mathcal{E}(\sum_{a\in V_m^\times} Y_a)^s = \sum \mathcal{E}(Y_{a_1}\dots Y_{a_s})$$

where the sum in the right hand side is on the s-uples $(a_1,\dots,a_s)$ in $V_m^\times \times \cdots \times V_m^\times$.

6.1. Case where $(a_1,\dots,a_s)$ is a family of linearly independent elements of V_m

Proposition 6.1. *When the $(a_1,\dots,a_s)$ is a family of linearly independent elements of V_m one has*

$$\mathcal{E}(Y_{a_1}\dots Y_{a_s}) \le 2^s \exp\frac{8s^3}{3q}.$$

There are at most $q^s\exp(-\frac{2^s-1}{q})$ of such families.

Proof. Let us define

$$E(b_1,b_2,\dots,b_s) = \sum_{x_1,x_2,\dots,x_s} \mathcal{E}\Big(\chi\big(f(x_1)+f(x_1+b_1)+ \\ +f(x_2)+f(x_2+b_2)+\cdots+f(x_s)+f(x_s+b_s)\big)\Big)$$

for $b_i\in V_m^\times$ and

$$E(s) = \sup E(b_1,\dots,b_s)$$

where $b_i = \sum_{j\in B_i} a_j$ and the B_i are distinct subsets of the set $\{1,2,\dots,2s\}$ such that $b_i\neq 0$ and $a_{s+j}=a_j$.

We first note that three of the b_i cannot be equal. Indeed suppose that $b_1=b_2=b_3$. Then, for at least one of the i $(1\le i\le s)$, then $i\in B_1$ and $s+i$ is not in B_2 or not in B_3. Let us suppose that $s+i\notin B_2$, then, the a_j being linearly independent, we have $b_1\neq b_2$.

If two b_i are equal, suppose that they are b_{2s-1} and b_{2s}. Then, by the induction relations given in [12], section 5:

$$\begin{aligned} E(b_1, b_2, \dots, b_{2s-1}, b_{2s}) \leq 2qE(b_1, b_2, \dots, b_{2s-3}, b_{2s-2}) + \\ +2\sum_{i=1}^{2s-2} E(b_1, b_2, \dots, b_i + b_{2s}, \dots b_{2s-1}) \end{aligned}$$

and

$$\begin{aligned} &E(b_1, b_2, \dots, b_i + b_{2s}, \dots b_{2s-1}) \\ &\quad\leq\ 2\sum_{j=1}^{2s-2} E(b_1, b_2, \dots, b_i + b_{2s}, \dots, b_j + b_{2s-1}, \dots, b_{2s-2}) \\ &\quad\leq\ 2(2s-2)E(2s-2) \end{aligned}$$

as $b_i + b_{2s} \neq 0$, $b_j + b_{2s-1} \neq 0$, $b_i + b_{2s} + b_{2s-1} \neq 0$. Therefore

$$\begin{aligned} &E(b_1, b_2, \dots, b_{2s-1}, b_{2s}) \\ &\quad\leq\ 2qE(b_1, b_2, \dots, b_{2s-3}, b_{2s-2}) + 4(2s-2)(2s-2)E(2s-2) \\ &\quad\leq\ (2q + 16(s-1)^2)E(2s-2). \end{aligned}$$

If all the b_i are distinct we have

$$E(b_1, b_2, \dots, b_{2s-1}, b_{2s}) \leq 2\sum_{i=1}^{2s-1} E(b_1, b_2, \dots, b_i + b_{2s}, \dots b_{2s-1})$$

If $s \geq 2$, all the $b_1, b_2, \dots, b_i + b_{2s}, \dots b_{2s-1}$ are distinct, except possibly 2 of them. Suppose that b_{2s-1} are distinct from the others, we have, as previously

$$\begin{aligned} &E(b_1, b_2, \dots, b_i + b_{2s}, \dots b_{2s-1}) \\ &\quad\leq\ 2\sum_{j=1}^{2s-2} E(b_1, b_2, \dots, b_i + b_{2s}, \dots, b_j + b_{2s-1}, \dots, b_{2s-2}) \end{aligned}$$

where $b_i + b_{2s} \neq 0$, $b_j + b_{2s-1} \neq 0$ and $b_i + b_{2s} + b_{2s-1} \neq 0$ as $b_i + b_{2s} \neq b_{2s-1}$. Therefore

$$E(b_1, b_2, \dots, b_{2s-1}, b_{2s}) \leq (2q + 16(s-1)^2)E(2s-2)$$

So we have the following bound on $E(2s)$:

$$\begin{aligned} E(2s) &\leq (2q+16(s-1)^2)(2q+16(s-2)^2)\dots(2q+16)2q \\ &\leq 2^s q^s\Big(1+\frac{8(s-1)^2}{q}\Big)\Big(1+\frac{8(s-2)^2}{q}\Big)\dots\Big(1+\frac{8}{q}\Big). \end{aligned}$$

So we get

$$\log E(2s) \quad\leq\quad s\log 2 + s\log q + \frac{8s^3}{3q}$$

using the inequality

$$\log(1+x) \leq x. \tag{3}$$

Finally, we get:

$$\begin{aligned} \mathcal{E}(Y_{a_1}\dots Y_{a_s}) = \frac{1}{q^s}E(a_1,a_1,a_2,a_2,\dots,a_s,a_s) \leq \\ \leq \frac{1}{q^s}E(2s) \leq 2^s \exp\frac{8s^3}{3q} \end{aligned}$$

if the $a_1\dots a_s$ are linearly independent.

The number N_s of s-uplets $(a_1,\dots,a_s)$ which are linearly independent is

$$N_s = q^s\Big(1-\frac{1}{q}\Big)\Big(1-\frac{2}{q}\Big)\Big(1-\frac{4}{q}\Big)\cdots\Big(1-\frac{2^{s-1}}{q}\Big).$$

Using again the inequality (3) this gives $\log N_s \leq s\log q - \frac{2^s-1}{q}$ and the desired result. □

6.2. **Case of the other terms**

The other terms are bounded by

$$\mathcal{E}(Y_{a_1}\dots Y_{a_s}) \leq \mathcal{E}(Y_a^s) \leq \frac{(2s)!}{(s)!}$$

where we use Holder's inequality and the above mentioned induction relations.

There are at most $(q-1)^{s-1}\,(2^s+s-3)$ of such terms. Indeed we have to find a lower bound for N_s:

$$\begin{aligned}
N_s &= (q-1)^s(1-\frac{1}{q-1})(1-\frac{3}{q-1})\cdots(1-\frac{2^{s-1}-1}{q-1})\\
&\geq (q-1)^s(1-\frac{1}{q-1}-\frac{3}{q-1}-\cdots-\frac{2^{s-1}-1}{q-1})\\
&= (q-1)^s(1-\frac{2^s+s-3}{q-1}).
\end{aligned}$$

Whence

$$(q-1)^s - N_s \leq (q-1)^s\frac{2^s+s-3}{q-1}.$$

7. Bound on $\mathcal{E}|X-\mu|^{2s}$

We use the relation

$$\mathcal{E}|X-\mu|^{2s} = \mathcal{E}(X-\mu)^{2s} = \sum_{i=0}^{2s}(-1)^i\mathcal{E}(X^i)\mu^{2s-i}$$

where $X = \frac{\sigma_f}{q^2}-1$, and $\mu = \mathcal{E}(\frac{\sigma_f}{q^2}-1) = \mathcal{E}(\frac{1}{q}\sum_{a\in V_m^\times} Y_a) = 2-2/q$ and proposition 5.3.

For even i, we get:

$$\begin{aligned}
\mathcal{E}X^i &= \frac{1}{q^i}\mathcal{E}(\sum Y_a)^i\\
&\leq 2^i\exp\frac{8i^3-3\times 2^i+3}{3q} + (q-1)^{i-1}q^{-i}\left(2^i+i-3\right)\frac{(2i)!}{(i)!}\\
&\leq 2^i\Big(1+O(\frac{1}{q}) + \frac{1.1}{q}\frac{(2i)!}{(i)!}\Big).
\end{aligned}$$

For odd i:

$$\mathcal{E}X^i \geq 2^i\frac{(q-1)^i}{q^i} \geq 2^i\left(1-\frac{i}{q}+O(i^2/q^2)\right).$$

Hence, as $i \leq 2s = O(\log q)$, and remarking that the constant terms cancel:

$$\begin{aligned}
&\mathcal{E}|X-\mu|^{2s} \\
&\quad\leq \sum_{i \text{ even}} \binom{2s}{i} 2^{2s}\left(1+O(1/q)+\frac{1.1}{q}\frac{(2i)!}{(i)!})\right) \\
&\qquad -\sum_{i \text{ odd}} \binom{2s}{i} 2^{2s}\left(1-\frac{i}{q}+O\Big(\frac{\log^2 q}{q^2}\Big)\right)\left(1-\frac{1}{q}\right)^{2s-i} \\
&\quad\leq \frac{2^{2s}}{q}\sum_{i \text{ even}} \binom{2s}{i}\left(O(1)+1.1\frac{(2i)!}{(i)!}\right) \\
&\qquad +\frac{2^{2s}}{q}\sum_{i \text{ odd}} \binom{2s}{i}\big(2s+i+O(\log^2 q/q)\big) \\
&\quad\leq 1.1\frac{2^{2s}}{q}\sum_{i \text{ even}} \binom{2s}{i}\frac{(2i)!}{(i)!}+\frac{2^{2s}}{q}\sum_{i}\binom{2s}{i}(4s+O(1)).
\end{aligned}$$

We have

$$\sum_{i \text{ even}} \binom{2s}{i}\frac{(2i)!}{(i)!} = (2s)!\sum_{i \text{ even}} \frac{(2i)!}{i!^2(2s-i)!}.$$

Going from i to $i+2$, we multiply the term of the sum by

$$\frac{(2i+4)(2i+3)(2i+2)(2i+1)(2s-i)(2s-i-1)}{(i+2)^2(i+1)^2} \geq 20$$

Hence

$$\sum_{i \text{ even}} \binom{2s}{i}\frac{(2i)!}{(i)!} \leq \frac{20}{19}\frac{(4s)!}{(2s)!}.$$

The other terms are negligible with respect to this one. So, we have, for q large enough:

$$\beta_{2s} \leq \frac{2^{2s+1}}{q}\frac{(4s)!}{(2s)!}.$$

References

[1] A. Canteaut, C. Carlet, P. Charpin, C. Fontaine *Propagation characteristics and correlation-immunity of highly nonlinear Boolean functions,* Advances in cryptology, EUROCRYPT 2000 (Bruges), 507–522, Lecture Notes in Comput. Sci., Vol. 1807, Springer, Berlin, 2000.

[2] C. Carlet, *On cryptographic complexity of Boolean functions*, Proceedings of the Sixth Conference on Finite Fields with Applications to Coding Theory, Cryptography and Related Areas (G.L. Mullen, H. Stichtenoth and H. Tapia-Recillas Eds), Springer (2002) pp. 53-69.

[3] C. Carlet, *On the algebraic thickness and non-normality of Boolean functions, with developments on symmetric functions*, submitted to IEEE Trans. Inform. Theory.

[4] G. Cohen, I. Honkala, S. Litsyn, A. Lobstein, *Covering codes.* North-Holland Mathematical Library, 54, North-Holland Publishing Co., Amsterdam (1997).

[5] 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, Université Paris VI (1998).

[6] A. Dembo, O. Zeitouni, *Large deviations techniques and applications* Applications of Mathematics, 38. Springer-Verlag, New York, 1998.

[7] S. Karlin, W. Studden *Tchebycheff systems: With applications in analysis and statistics,* Pure and Applied Mathematics, Vol. XV, Interscience Publishers, John Wiley & Sons, New York-London-Sydney 1966

[8] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes,* North-Holland, Amsterdam (1977).

[9] D. Olejár and M. Stanek, *On cryptographic properties of random Boolean functions*, J.UCS 4, no. 8, (1998) 705-717.

[10] N. Patterson and D. Wiedemann, *The covering radius of the* $(2^{15}, 16)$ *Reed-Muller code is at least* 16 276, IEEE Trans. Inform. Theory 29, no. 3 (1983), 354-356.

[11] R. Rockafellar, *Convex analysis*, second printing, Princeton University Press, Princeton, 1972

[12] F. Rodier, *Sur la non-linéarité des fonctions booléennes*, Acta Arithmetica, vol 115, (2004), 1-22, preprint: arXiv: math.NT/0306395.

[13] F. Rodier, *On the nonlinearity of Boolean functions*, Proceedings of WCC2003, Workshop on coding and cryptography 2003 (D. Augot, P. Charpin, G. Kabatianski eds), INRIA (2003), pp. 397-405.

[14] F. Rodier, *Asymptotic nonlinearity of Boolean functions*, prétirage de l'IML n^o 2003-10.

[15] P. Stănică, *Nonlinearity, local and global avalanche characteristics of balanced Boolean functions*, Discrete Math. 248 (2002), no. 1-3, 181–193.

[16] Wu, Chuan-Kun *On distribution of Boolean functions with nonlinearity* $\leq 2^{n-2}$, Australas. J. Combin. 17 (1998), 51-59.

[17] Xian-Mo Zhang and Yuliang Zheng, *GAC —the Criterion for Global Avalanche Characteristics of Cryptographic Functions*, Journal of Universal Computer Science, vol. 1, no. 5 (1995), 316-333

ON THE SUPPORTS OF THE WALSH TRANSFORMS OF BOOLEAN FUNCTIONS

C. Carlet[1, 2] and S. Mesnager[1]

Abstract. In this paper, we study, in relationship with covering sequences, the structure of those subsets of $\mathbb{F}_2^n$ which can be the Walsh supports of Boolean functions.

1. **Introduction**

Cryptographic Boolean functions play an important role in the design of hash functions and of stream and block ciphers. Various criteria related to cryptographically desirable Boolean functions have been proposed, such as balancedness, high nonlinearity, high correlation immunity order, high degree of the propagation criterion and inexistence of linear structure. The most important mathematical tool for the study of cryptographic properties of Boolean functions is the Walsh (or Hadamard) transform, the characteristic 2 special case of the discrete Fourier transform. The Walsh transform permits to measure the correlation between a Boolean function and all linear Boolean functions. The knowledge of the Walsh transform of a Boolean function uniquely determines the function and hence it is possible to work entirely with the Walsh transform. In particular, its systematic use leads to uniform, elegant and efficient treatments and statements of the main cryptographic criteria. Resiliency and inexistence of linear structures are directly related to the properties of the support of the

[1] MAATICAH, Université de Paris VIII, Département de Mathématiques, 2, rue de la Liberté, 93526 Saint-Denis Cedex - France
[2] INRIA - Projet CODES, Bâtiment 25, Domaine de Voluceau - Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex - France

Walsh transform of a Boolean function (i.e. its Walsh support). The other essential criteria - degree, non-linearity, propagation criterion - are also connected, in a more distant way, to the Walsh support of a Boolean function.

However, little is known on the possible structure of the Walsh supports of Boolean functions. We know only few generic examples of subsets of $\mathbb{F}_2^n$ which can be the Walsh supports of some Boolean functions on n variables. We know even less examples of subsets of $\mathbb{F}_2^n$ which cannot be such supports.

The interest of studying the structure of Walsh supports of Boolean functions is still strengthened after the introduction of covering sequences. The notion of covering sequence of a Boolean function, related to the derivatives of the function, was introduced in [7]; it enables a complete characterization of the balancedness and of the resiliency of Boolean functions, and there exists a characterization of those Boolean functions which admit some given covering sequence by means of their Walsh spectra. We shall see that the existence of non-constant covering sequences for a balanced Boolean function depends on a property of its Walsh support.

In this paper, we summarize what is known on this subject, we introduce several general results, and we study the forms of the Walsh supports of all functions on at most 6 variables (for which a classification is known).

In Section 2, we first introduce the notation, the definitions and preliminary results on covering sequences. We study subsequently the Walsh supports of those balanced Boolean functions whose covering sequences are indicators of flats. We show (Proposition 2.6) that, for any Boolean function f on $\mathbb{F}_2^n$ which admits no derivative equal to the constant function 1 and any flat $a + E$ of $\mathbb{F}_2^n$, there is an equivalence between the fact that f admits the indicator of $a + E$ of $\mathbb{F}_2^n$ as non-trivial covering sequence and the fact that the Walsh support of f is disjoint from the orthogonal space of E. We characterize then those Boolean functions on $\mathbb{F}_2^n$ whose Walsh support is disjoint from the orthogonal of a given vector subspace of $\mathbb{F}_2^n$. Next, in Section 3, we study the possible structures of the Walsh supports of Boolean functions. Along the way, we recall what are the Walsh supports of classical Boolean functions : affine, quadratic, and more generally partially bent. It is well known that, for every n, many kinds of Boolean functions (including the classical Maiorana-McFarland's functions) can have

Walsh support equal to the whole space $\mathbb{F}_2^n$, and that the empty set cannot be such support. Also, any singleton is the support of an affine function. The next natural step is to ask whether the difference $\mathbb{F}_2^n \setminus \{a\}$ (where a denotes any vector of $\mathbb{F}_2^n$) can be or not the Walsh support of a Boolean function, that is, whether Walsh supports of Boolean functions can have size $2^n - 1$. We remark that adding a linear function moves a to 0; this brings us to be interested in finding balanced Boolean functions whose Walsh support is $\mathbb{F}_2^n \setminus \{0\}$ (that is, which are the only balanced function in the coset of the Reed-Muller code of order 1 that they generate). For small values of the number of variables, it is easy to see that such functions do not exist. For $n \geq 10$, we give a construction of a class of balanced Boolean functions whose Walsh support has size $2^n - 1$ (cf. Construction 1). Such functions admit only one kind of covering sequences: the sequences which are constant on $\mathbb{F}_2^n \setminus \{0\}$. The question of knowing whether such functions are exceptional arises then; indeed, there are examples of characteristics of n-variable Boolean functions (e.g. non-normality) which are impossible for small values of n, and which become the common case for high values of n. We prove in Proposition 3.3 that such functions are rare among the balanced Boolean functions.

2. **Notation and Preliminaries**

We shall have to distinguish in the whole paper between the additions of integers in $\mathbb{Z}$, denoted by $+$ and $\sum_i$, and the additions mod 2, denoted by $\oplus$ and $\bigoplus_i$. For simplicity and because there will be no ambiguity, we shall denote by $+$ the addition of vectors of $\mathbb{F}_2^n$ (words). If x and b are two vectors in $\mathbb{F}_2^n$, we denote by $x \cdot b$ their usual inner product $x \cdot b = \bigoplus_{i=1}^n x_i b_i$. We recall the basic facts about Boolean functions. A Boolean function f is an $\mathbb{F}_2$-valued function on the vector-space $\mathbb{F}_2^n$ of n-tuples of elements from $\mathbb{F}_2$. Any Boolean function f on n variables admits a unique *algebraic normal form* (A.N.F.) :

$$f(x_1, \dots, x_n) = \bigoplus_{u \in \mathbb{F}_2^n} a_u \left(\prod_{i=1}^n x_i^{u_i} \right) = \bigoplus_{u \in \mathbb{F}_2^n} a_u x^u.$$

We call the degree of the algebraic normal form of a Boolean function its *algebraic degree*. The *Hamming weight* $\mathrm{wt}(f)$ of f is the number of vectors x in $\mathbb{F}_2^n$ such that $f(x) = 1$. A function f is *balanced* if $\mathrm{wt}(f) = \mathrm{wt}(f \oplus 1)$, i.e. if $\mathrm{wt}(f) = 2^{n-1}$. The *"sign" function* of f is the integer-valued function $\chi_f(x) = (-1)^{f(x)}$. The *Walsh transform* of f, that is, the discrete Fourier transform of χ_f, whose value at $b \in \mathbb{F}_2^n$ equals by definition $\widehat{\chi_f}(b) = \sum\limits_{x\in\mathbb{F}_2^n} (-1)^{f(x)+x\cdot b}$, is related to the Hamming weight of the function $f \oplus l_b$ (where $l_b(x) = b\cdot x$) via the relation: $\widehat{\chi_f}(b) = 2^n - 2\mathrm{wt}(f \oplus l_b)$. It satisfies Parseval's relation:

$$\sum_{b\in\mathbb{F}_2^n} \widehat{\chi_f}^2(b) = 2^{2n} \tag{1}$$

and the inverse formula relation:

$$\sum_{b\in\mathbb{F}_2^n} \widehat{\chi_f}(b)(-1)^{b\cdot x} = 2^n \chi_f(x) \tag{2}$$

The *Hamming distance* between two Boolean functions f_1 and f_2 on $\mathbb{F}_2^n$ is equal to the weight of $f_1 \oplus f_2$. The minimum distance between f and the set of all affine functions $l_b \oplus \epsilon$ ($b \in \mathbb{F}_2^n, \epsilon \in \mathbb{F}_2$), called the *nonlinearity* of f, is denoted by N_f and satisfies the relation:

$$N_f = 2^{n-1} - \frac{1}{2} \max_{b\in\mathbb{F}_2^n} |\widehat{\chi_f}(b)|\,. \tag{3}$$

Because of Parseval's relation, it is upper bounded by $2^{n-1} - 2^{n/2-1}$. This bound is tight for n even. The functions which achieve it are called *bent*. But these functions are never balanced. The maximum nonlinearity of balanced functions is unknown for every $n \geq 8$.

The auto-correlation function of the sign function of a Boolean function f is $\hat{r}(a) = \sum_{x\in\mathbb{F}_2^n} (-1)^{f(x)\oplus f(x+a)}$. The function $D_a f(x) = f(x) \oplus f(x+a)$ is called a *derivative* of f. As shown in [3], for every Boolean function, we have $(2^n - N_{\hat{r}})(2^n - N_{\widehat{\chi_f}}) \geq 2^n$, where $N_{\hat{r}}$ and $N_{\widehat{\chi_f}}$ are the numbers of zeros of respectively $\hat{r}$ and $\widehat{\chi_f}$. If f satisfies the equality $(2^n - N_{\hat{r}})(2^n - N_{\widehat{\chi_f}}) = 2^n$, then f is called partially-bent.

Notation: Throughout this paper, S_f denotes the Walsh support of f, *i.e.* $S_f := \{\omega \in \mathbb{F}_2^n \mid \widehat{\chi_f}(\omega) \neq 0\}$.

2.1. Covering sequences of balanced functions

Definition 2.1. A *covering sequence* of a Boolean function f on $\mathbb{F}_2^n$ is any real-valued sequence $\lambda = (\lambda_a)_{a \in \mathbb{F}_2^n}$ such that $\sum_{a \in \mathbb{F}_2^n} \lambda_a D_a f$ is a constant function ρ. The value of ρ is called the *level* of this sequence. If $\rho \neq 0$, then we say that the covering sequence is *non-trivial.*

The following characterization of balanced Boolean functions is shown in [7] :

Proposition 2.2. *[7] If a Boolean function on $\mathbb{F}_2^n$ admits a non-trivial covering sequence, then it is balanced. Conversely, any balanced function admits the constant sequence 1 as non-trivial covering sequence (with level 2^{n-1}). Thus, any Boolean function is balanced if and only if it admits a non-trivial covering sequence.*

Note that we can change the value λ_0 of any (non-trivial) covering sequence without changing its property of being a (non-trivial) covering sequence. A question arises: does there exist balanced functions admitting as only non-trivial covering sequences those which are constant on $\mathbb{F}_2^n \setminus \{0\}$? We shall see that this question is related to a question on the Walsh supports. Note that any balanced quadratic function, and more generally any balanced partially-bent function (cf. [3]), admits a non-trivial atomic covering sequence (i.e. with one coefficient λ_a equal to 1 and all the others null). Equivalently, it has a constant derivative equal to 1. Such function is affinely equivalent to the sum of a Boolean function on $n-1$ variables and of the function x_n. It is (weakly) degenerate (see [8]). In this paper, we are interested in the functions which admit no derivative $D_a f$ equal to the constant function 1.

Recall that we denote by $\widehat{\chi_f}(b)$ the value $\sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+x \cdot b}$. We denote similarly by $\widehat{\lambda}(b)$ the value $\sum_{a \in \mathbb{F}_2^n} \lambda_a (-1)^{a \cdot b}$, i.e. the value at b of the Fourier transform of the sequence λ. Recall also that the support of λ is $\{a \in \mathbb{F}_2^n \mid \lambda_a \neq 0\}$. The following characterization is shown in [7] :

Theorem 2.3. *[7] Let f be any Boolean function on $\mathbb{F}_2^n$ and $\lambda = (\lambda_a)_{a \in \mathbb{F}_2^n}$ any real-valued sequence.*

f admits λ as covering sequence if and only if $\widehat{\lambda}$ takes constant value on the support $S_f = \{b \in \mathbb{F}_2^n \mid \widehat{\chi_f}(b) \neq 0\}$ of $\widehat{\chi_f}$. Let r be

this constant value, then the level of this covering sequence is the number $\frac{1}{2}[(\sum_{a\in\mathbb{F}_2^n}\lambda_a) - r]$*.*

Notice that if $\widehat{\lambda}$ takes value r on the support S_f of $\widehat{\chi_f}$ then, replacing its coefficient λ_0 by $\lambda_0 - r$, we obtain a covering sequence λ' such that $\widehat{\lambda'}$ takes value 0 on S_f.

Thanks to Theorem 2.3, we can characterize by their Walsh supports those balanced functions whose non-trivial covering sequences are all constant on $\mathbb{F}_2^n \setminus \{0\}$.

Corollary 2.4. *Let f be any balanced n-variable Boolean function. The Walsh support of f equals $\mathbb{F}_2^n \setminus \{0\}$ if and only if the only non-trivial covering sequences of f are those sequences which are constant on $\mathbb{F}_2^n \setminus \{0\}$.*

Proof. If S_f equals $\mathbb{F}_2^n \setminus \{0\}$ then the only non-trivial covering sequences of f are those sequences which are constant on $\mathbb{F}_2^n \setminus \{0\}$, according to Theorem 2.3 and to the fact that a sequence is constant on $\mathbb{F}_2^n \setminus \{0\}$ if and only if its Fourier transform has the same property (this is a direct consequence of the bijectivity of the Fourier transform). Conversely, if S_f of f does not equal $\mathbb{F}_2^n \setminus \{0\}$, then, by inverse Fourier transform, there exists a sequence whose Fourier transform equals the indicator of S_f, and this sequence cannot be constant on $\mathbb{F}_2^n \setminus \{0\}$ since its Fourier transform is not. □

2.2. Walsh support of balanced Boolean functions whose covering sequences are indicators of flats

Since every balanced function admits the constant covering sequence 1, we focus now on the covering sequences whose coefficients are equal to 0 or 1. In the sequel, we shall always exclude, as we said already, the possibility that a function admits a derivative equal to the constant 1, because it is an extremal case (it is the simplest case of balancedness for a Boolean function) and because the functions admitting constant derivatives are degenerate (see [8]).

We first make an observation on those Boolean functions which admit a covering sequence whose support is included in a vector subspace of $\mathbb{F}_2^n$.

Proposition 2.5. *Let E be any vector subspace of $\mathbb{F}_2^n$. Let f be any Boolean function on $\mathbb{F}_2^n$. Then f admits a covering sequence*

λ with support $S \subseteq E$ if and only if the restriction of f to any coset of E admits the same covering sequence λ.

Proof. The condition is clearly necessary and sufficient since the integer-valued function $\sum_{a \in E} \lambda_a D_a f$ is equal to a constant function ρ if and only if its restriction to any coset of E equals ρ. □

Proposition 2.6. *Let E be any vector subspace of $\mathbb{F}_2^n$ and $u + E$ any of its cosets. Let f be any Boolean function on $\mathbb{F}_2^n$. Assume it admits no derivative $D_a f$ equal to the constant function 1. Then f admits the indicator of $u+E$ as non-trivial covering sequence if and only if S_f is disjoint from $E^\perp = \{x \in \mathbb{F}_2^n \mid x \cdot v = 0,\ \forall v \in E\}$. This is equivalent to the fact that the restriction of f to any coset of E is balanced. The level of this covering sequence is then equal to $|E|/2$ and the indicator of every coset of E is also a covering sequence of f with the same level. More generally, any sequence λ such that $\lambda_{a+u} = \lambda_u$ for all $a \in E$ and all $u \in \mathbb{F}_2^n$, is also a covering sequence of f.*

Proof. Denote by $\lambda = (\lambda_a)_{a \in \mathbb{F}_2^n}$ the indicator of $u + E$. For every $b \in \mathbb{F}_2^n$, $\widehat{\lambda}(b) = \sum_{a \in E} (-1)^{(u+a)\cdot b}$ equals $(-1)^{u \cdot b}\ |E|$ if $b \in E^\perp$ and 0 otherwise. Thus, according to Theorem 2.3, alinea 2, λ is a covering sequence of f if and only if S_f is either included in $E^\perp \cap u^\perp$ (but in such case, the covering sequence is trivial, since we have then $r = \sum_{a \in \mathbb{F}_2^n} \lambda_a = |E|$ in Theorem 2.3; this is excluded by the hypothesis) or included in $E^\perp \setminus u^\perp$ (but in such case, for every element a of $u+E$, the function $D_a f$ is equal to the constant function 1, since we have then $r = -|E|$ in Theorem 2.3; this is also excluded by the hypothesis), or disjoint from $E^\perp$ (in which case the level of the sequence is equal to $|E|/2$). This latter case is the only one satisfying the hypothesis. Its equivalence with the fact that the restriction of f to any coset of E is balanced is a consequence of Proposition 2.5 applied to the sequence equal to the indicator of E (whose Fourier transform equals $|E|$ times the indicator of $E^\perp$) and of Proposition 2.2.

The indicator of every coset of E (whose Fourier transform equals $\pm|E|$ times the indicator of $E^\perp$) is then clearly also a covering sequence of such function f with the same level.

Any sequence λ such that $\lambda_{a+u} = \lambda_u$ for every $a \in E$ and every $u \in \mathbb{F}_2^n$ is the linear combination of the indicators of cosets of E. Therefore, it is also a covering sequence of f. □

Remark. If a balanced function f is such that $\widehat{\chi_f}^{-1}(0)$ contains a non-zero vector b, then we can apply Propositions 2.6 and 2.5 to the vector-subspace $E^\perp = \{0, b\}$. Hence, Proposition 2.6 shows (again) that a balanced function admits a non-trivial non-constant covering sequence if and only if its Walsh support is different from $\mathbb{F}_2^n \setminus \{0\}$.

Remark. Let f be any Boolean function on $\mathbb{F}_2^n$ and $\lambda = (\lambda_a)_{a\in\mathbb{F}_2^n}$ a covering sequence of f. Let r be the constant value of $\widehat{\lambda}$ on S_f. Then the nonlinearity of f satisfies:

$$N_f \leq 2^{n-1} - \frac{2^{n-1}}{\sqrt{|\widehat{\lambda}^{-1}(r)|}}$$

where $\widehat{\lambda}^{-1}(r) = \{a \in \mathbb{F}_2^n \mid \widehat{\lambda}(a) = r\}$ and $|\widehat{\lambda}^{-1}(r)|$ denotes the cardinality of $\widehat{\lambda}^{-1}(r)$. Indeed, according to Parseval's relation (1) and since S_f is included in $\widehat{\lambda}^{-1}(r)$, we have

$$\sum_{b\in\widehat{\lambda}^{-1}(r)} \widehat{\chi_f}^2(b) = 2^{2n}.$$

Thus, we have

$$\max_{b\in\mathbb{F}_2^n}\left(\widehat{\chi_f}^2(b)\right) = \max_{b\in\widehat{\lambda}^{-1}(r)}\left(\widehat{\chi_f}^2(b)\right) \geq \frac{2^{2n}}{|\widehat{\lambda}^{-1}(r)|}$$

and the result follows from relation (3).

3. The Walsh supports of Boolean functions

We denote by $\mathcal{S}_n$ the set of all the Walsh supports of Boolean functions on $\mathbb{F}_2^n$. We begin with some general elementary remarks on $\mathcal{S}_n$. We subsequently study the possible structures of Walsh supports.

3.1. Generalities

For every n, $\mathcal{S}_n$ is globally invariant under any affine automorphism of $\mathbb{F}_2^n$. Indeed, it is clearly invariant under translations since if $g(x) = f(x) \oplus a \cdot x$ then $S_g = a + S_f$, and it is also invariant under linear isomorphisms: let f be any Boolean function on $\mathbb{F}_2^n$, and L any linear automorphism of $\mathbb{F}_2^n$; let $L^\star$ be the unique linear automorphism of $\mathbb{F}_2^n$ such that, for every x and y in $\mathbb{F}_2^n$, we have: $y \cdot L^\star(x) = L(y) \cdot x$ (the matrices of these two automorphisms are transposed one of each other and we have $(L^\star)^{-1} = (L^{-1})^\star$). Then for every b in $\mathbb{F}_2^n$, we have $\widehat{\chi_{f \circ L^\star}}(b) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f \circ L^\star(x) + x \cdot b} = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) + L^{\star -1}(x) \cdot b} = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) + x \cdot L^{-1}(b)} = \widehat{\chi_f}(L^{-1}(b))$. Thus, the Walsh support of $f \circ L^\star$ is equal to $L(S_f)$. These remarks means that two Boolean functions f and g which are affinely equivalent, namely if there exists a linear automorphism L on $\mathbb{F}_2^n$, two n-dimensional binary vectors a and b and a binary scalar c such that, for all $x \in \mathbb{F}_2^n$, $g(x) = f(L(x) + a) \oplus b \cdot x \oplus c$, then their Walsh support are linked by the relation : $S_g = b + L^\star(S_f)$.

If f is a Boolean function on $\mathbb{F}_2^n$ and g a Boolean function on $\mathbb{F}_2^m$, then $S_f \times S_g$ is the Walsh support of the function $h(x, y) = f(x) \oplus g(y)$ on $\mathbb{F}_2^{n+m}$. In particular, taking g affine, S_g is then a singleton and $S_h = S_f \times \{a\}$.

We do not know any other example of an operation on sets, under which $\mathcal{S}_n$ would be globally invariant. In particular, $\mathcal{S}_n$ is not invariant under intersection; indeed, it contains all singletons (it is well-known that if f is affine, say $f(x) = a \cdot x \oplus \epsilon$, then S_f equals the singleton $\{a\}$, and the converse is true according to Parseval's relation and to Relation (2)) and it does not contain the empty set. It is not invariant under union or symmetric difference either; indeed, it does not contain pairs: let us suppose that a pair $\{a, b\}$, $a \neq b$, is the Walsh support of a Boolean function f; let us denote by λ_a and λ_b the values of the Walsh transform of f at a and b; then, we have $|\lambda_a| < 2^n$ and $|\lambda_b| < 2^n$ according to Parseval's relation; and according to Relation (2), we have $\lambda_a + \lambda_b = \pm 2^n$ and $\lambda_a - \lambda_b = \pm 2^n$, which is clearly impossible.

Many secondary constructions of Boolean functions permit to express the Walsh transform of the constructed function f by means of those of the functions taken in input; but the Walsh support S_f of f depends on the values of the these Walsh transforms - not only on their supports. This is the case, for instance, of Siegenthaler's construction $f(x, x_{n+1}) = (x_{n+1} \oplus 1) f_1(x) \oplus x_{n+1} f_2(x)$, for which we have $\widehat{\chi_f}(a, a_{n+1}) = \widehat{\chi_{f_1}}(a) + (-1)^{a_{n+1}} \widehat{\chi_{f_2}}(a)$.

3.2. The whole space $\mathbb{F}_2^n$ as a Walsh support

For every n, $\mathcal{S}_n$ contains $\mathbb{F}_2^n$ as an element, i.e. there exist functions f whose Walsh support is equal to $\mathbb{F}_2^n$. These functions, which are such that no function $f(x) \oplus b \cdot x \oplus \epsilon$ (where $b \in \mathbb{F}_2^n$, $\epsilon \in \mathbb{F}_2$) is balanced, can be constructed in many different ways. A first class of examples of such functions is that of Boolean functions of odd weights. A second class, valid for every even n, is that of *bent* functions, which are characterized by the fact that, for every $b \in \mathbb{F}_2^n$, the number $\widehat{\chi_f}(b)$ has magnitude $2^{n/2}$. A third example can be found in the general class of Maiorana-McFarland functions. The following proposition is well-known (see for instance [2, 4]).

Proposition 3.1. *Let s and t be any positive integers, g any Boolean function on $\mathbb{F}_2^t$ and ϕ any mapping from $\mathbb{F}_2^t$ to $\mathbb{F}_2^s$. Define for every $x \in \mathbb{F}_2^s$ and every $y \in \mathbb{F}_2^t$: $f(x,y) = x \cdot \phi(y) \oplus g(y)$. Then*

$$\widehat{\chi_f}(a,b) = 2^s \sum_{y \in \phi^{-1}(a)} (-1)^{g(y) \oplus b \cdot y}, \forall a \in \mathbb{F}_2^s,\ b \in \mathbb{F}_2^t.$$

Thus, if for every $a \in \mathbb{F}_2^s$ the set $\phi^{-1}(a)$ has odd size (such ϕ exists if and only if $s \leq t$) then S_f is equal to $\mathbb{F}_2^{s+t}$.

3.3. The other flats of $\mathbb{F}_2^n$ as Walsh supports

3.3.1. Even-dimensional flats

For every n, the Walsh support of any quadratic function on $\mathbb{F}_2^n$ is a flat of $\mathbb{F}_2^n$ of even dimension. Conversely any flat of $\mathbb{F}_2^n$ of even dimension is the Walsh support of a quadratic function.

Indeed, any quadratic function f on $\mathbb{F}_2^n$ may be written (see [9]) as $f = q^{(t)} \circ A \oplus \ell_a \oplus \epsilon$, where $q^{(t)}$ denotes the canonical quadratic function: $q^{(t)}(x_1, ..., x_n) = \bigoplus_{i=1}^t x_i x_{t+i}$, $\epsilon \in \mathbb{F}_2$, A is a linear automorphism of $\mathbb{F}_2^n$ and ℓ_a, $a \in \mathbb{F}_2^n$, is the linear Boolean function $\ell_a(x) := a \cdot x$. According to Subsection 3.1, we have $S_f = a + A^\star(S_{q^{(t)}})$. It is well known that $S_{q^{(t)}} = \mathbb{F}_2^{2t} \times \{0\}$ and so S_f is a flat of $\mathbb{F}_2^n$ of even dimension. Conversely, let $a+V$ be any flat of $\mathbb{F}_2^n$ of even dimension (V being a vector subspace of $\mathbb{F}_2^n$); there exists a linear automorphism A of $\mathbb{F}_2^n$ such that $A(V) = \mathbb{F}_2^{2t} \times \{0\}$. Set $f := q^{(t)} \circ A^{\star -1} \oplus \ell_a$. Then $S_f = a + A^{-1}(\mathbb{F}_2^{2t} \times \{0\}) = a + V$.

More generally, the Walsh support of any partially-bent function on $\mathbb{F}_2^n$, that is, of any function $f = g \circ A \oplus \ell_a \oplus \epsilon$, where g is a bent function on $2t$ variables and A is a linear mapping from

$\mathbb{F}_2^n$ to $\mathbb{F}_2^{2t}$, is a flat of $\mathbb{F}_2^n$ of even dimension. Conversely any flat of $\mathbb{F}_2^n$ of even dimension is the Walsh support of a partially-bent function, in which the choice of the bent function g is arbitrary.

3.3.2. Odd-dimensional flats

For every n, there also exist functions whose Walsh supports are any odd-dimensional flats $a + E$ of $\mathbb{F}_2^n$ of dimensions at least 3 (E being a vector subspace of $\mathbb{F}_2^n$), and in fact any flats of dimensions at least 2: take for instance $f := \delta_{E^\perp} \oplus \ell_a$ where ℓ_a denotes the linear Boolean function $\ell_a(x) := a{\cdot}x$ and $\delta_{E^\perp}$ denotes the indicator of $E^\perp := \{x \in \mathbb{F}_2^n \mid \forall y \in E,\, x \cdot y = 0\}$; according to Subsection 3.1, $S_f = a + S_{\delta_{E^\perp}}$; now, straightforward calculation yields

$$\widehat{\chi_{\delta_{E^\perp}}}(\omega) = \begin{cases} 2^n - 2\left|E^\perp\right| & \text{if } \omega = 0 \\ -2\left|E^\perp\right| & \text{if } \omega \in E \setminus \{0\} \\ 0 & \text{otherwise} \end{cases}$$

Therefore $S_{\delta_{E^\perp}} = E$.

Remark. We have had to exclude the case of 1-dimensional flats in the construction above. Actually, this case is peculiar, since we have seen that a pair (that is, a 1-dimensional flat) cannot be the Walsh support of a Boolean function..

3.4. **Complements of singletons**

As seen in the introduction, if a Boolean function f is such that $b \notin S_f$, then changing $f(x)$ into $f(x) \oplus b \cdot x$ (a function belonging to the same coset of the Reed-Muller code of order 1) permits to assume that f is balanced. Thus we are brought to study the Walsh supports of balanced functions. We show now that there exist balanced functions f such that $\widehat{\chi_f}^{-1}(0)$ contains no non-zero vector, by giving a construction of a new class of Boolean functions on any number $n \geq 10$ of variables, and whose Walsh supports equal $\mathbb{F}_2^n \setminus \{0\}$.

Construction 1. Let k and m be two positive integers such that $m \geq k + 2$ and $2^{k-1} \geq m + 1$ (this is possible only with $m \geq 6$ and $k \geq 4$, and for every $n \geq 10$, there exist such m and k for which $n = m + k$). Then there exists a mapping ϕ from $\mathbb{F}_2^m$ to $\mathbb{F}_2^k$ such that the size of $\phi^{-1}(0)$ is equal to 1 and, for any nonzero vector $a \in \mathbb{F}_2^k$, the size of $\phi^{-1}(a)$ is an odd integer greater than or equal to 3. There also exists a subset E of $\mathbb{F}_2^k \times \mathbb{F}_2^m$ such that $E \subseteq \{(x,y) \in \mathbb{F}_2^k \times \mathbb{F}_2^m / \, x \cdot \phi(y) = 0\}$, such that $|E| = 2^{k-1}$ and which contains an element $(0,v)$ of even Hamming weight as well as all the elements of the form $(0,u^i)$, where the vector u^i $(1 \leq i \leq m)$ is defined as $u^i_j = 1$ if $j = i$ and $u^i_j = 0$ otherwise (such a subset E exists because we suppose that $|E| = 2^{k-1} \geq m+1$). We denote by δ_E the indicator of E: $\delta_E(x,y) = 1$ if $(x,y) \in E$ and $\delta_E(x,y) = 0$ otherwise. Define then the following Boolean function f on $\mathbb{F}_2^{k+m}$:

$$\forall (x,y) \in \mathbb{F}_2^k \times \mathbb{F}_2^m, \quad f(x,y) := \phi(y) \cdot x \oplus \delta_E(x,y).$$

Proposition 3.2. *Let f be defined as in construction 1. Then f is balanced and the Walsh support S_f of f equals $\mathbb{F}_2^n \setminus \{0\}$.*

Proof. A straightforward calculation yields: $\forall (a,b) \in \mathbb{F}_2^k \times \mathbb{F}_2^m$,

$$\widehat{\chi_f}(a,b) = 2^k \sum_{y \in \phi^{-1}(a)} (-1)^{b \cdot y} - 2 \sum_{(x,y) \in E} (-1)^{a \cdot x \oplus b \cdot y}.$$

In particular, when $(a,b) = (0,0)$, we have :

$$\widehat{\chi_f}(0,0) = 2^k |\phi^{-1}(0)| - 2|E| = 0$$

which ensures that f is balanced. Let $(a,b) \in \mathbb{F}_2^k \times \mathbb{F}_2^m$ be a non zero word i.e. $(a,b) \neq (0,0)$. If $b = 0$ then

$$\widehat{\chi_f}(a,0) \geq 2^k |\phi^{-1}(a)| - 2^k > 0$$

since $|\phi^{-1}(a)| > 1$. Assume now that $b \neq 0$. Since $|\phi^{-1}(a)|$ is odd, it holds

$$\sum_{y \in \phi^{-1}(a)} (-1)^{b \cdot y} \equiv |\phi^{-1}(a)| \equiv 1 \pmod 2$$

which implies that

$$2^k \left| \sum_{y \in \phi^{-1}(a)} (-1)^{b \cdot y} \right| \geq 2^k.$$

Therefore it suffices to show that

$$\left| \sum_{(x,y) \in E} (-1)^{a \cdot x \oplus b \cdot y} \right| < 2^{k-1} = |E|$$

to ensure that $\widehat{\chi_f}(a,b) \neq 0$. To this end, we show that we can find two elements z_1 and z_2 in E such that $(a,b) \cdot z_1 = 0$ and $(a,b) \cdot z_2 = 1$. Suppose that b is not the all-one vector. There exists then at least two indices i and j such that $b_i = 0$ and $b_j = 1$, and it suffices to take $z_1 = (0, u^i)$ and $z_2 = (0, u^j)$. If b is the all-one vector, it suffices to take $z_1 = (0, v)$ and $z_2 = (0, u^1)$. □

Concerning the values of n smaller than 10, we know that for $n = 1$, the Boolean function $f : x \in \mathbb{F}_2 \mapsto x$ is such that $S_f = \mathbb{F}_2 \setminus \{0\}$. By computer search, we know that there is no Boolean function f such that $S_f = \mathbb{F}_2^n \setminus \{0\}$ when $n \in \{2, 3, 4\}$. We present in appendix B some results about the Walsh supports of Boolean functions on 5 variables. In particular, we have checked with a computer program that there is no Boolean function on 5 variables whose Walsh support is equal to $\mathbb{F}_2^5 \setminus \{0\}$. To this end, we have used the classification of Boolean functions on 5 variables obtained by Berlekamp and Welsh [1]. In the case $n = 6$, the same method can be used with a classification of Boolean functions on 6 variables: such classification was obtained for the first time by Maioarana [10]; it is precisely listed on the web page maintained by Fuller (`http://www.isrc.qut.edu.au/people/fuller`), with the indication, for every class, of the algebraic degree, the nonlinearity and the maximum value in autocorrelation spectrum. But it is possible to avoid visiting all the classes. Indeed, assume there exists a Boolean function f on 6 variables such that $S_f = \mathbb{F}_2^6 \setminus \{0\}$. Let d be the algebraic degree of f (we assume $d \geq 2$ since the Walsh support of affine functions are singletons). It is known that the values of a balanced Boolean function f on n variables of algebraic degree d are divisible by $2^{2+\lfloor \frac{n-2}{d} \rfloor}$. Therefore, for $d \in \{2, 3, 4\}$, the Walsh spectrum of f is of the form $\{\pm 8k,\, k = 0 \dots 7\}$. Let n_k be the number of words $\omega \in \mathbb{F}_2^6$ such that $\widehat{\chi_f}(\omega) = \pm 8k$. Clearly $n_0 = 1$. Parseval's relation requires that $(\star)$ $\sum_{k=1}^{7} k^2 n_k = 64$. Moreover the condition $S_f = \mathbb{F}_2^6 \setminus \{0\}$ implies that $(\star\star)$ $\sum_{k=1}^{7} n_k = 63$. One easily sees that there is no solution for the diophantine system formed with $(\star)$ and $(\star\star)$. This shows that d must be equal to 5. We checked with a computer program that no Walsh support of any balanced Boolean function on 6 variables of algebraic degree equal to 5 is equal to $\mathbb{F}_2^n \setminus \{0\}$.

Concerning the other values $n \in \{7, 8, 9\}$, the question remains completely open since all the arguments exposed above fail for these values of n.

The question then arises of knowing if there are few or many balanced Boolean functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$ such that $S_f = \mathbb{F}_2^n \setminus \{0\}$. We answer in the proposition below that only a small number of balanced Boolean functions are such that $S_f = \mathbb{F}_2^n \setminus \{0\}$. We denote below by $\mathcal{E}_n$ the set of all balanced Boolean functions on $\mathbb{F}_2^n$.

Proposition 3.3. *For every positive integer n, the density in $\mathcal{E}_n$ of the set $\{f \in \mathcal{E}_n \mid S_f = \mathbb{F}_2^n \setminus \{0\}\}$ is less than $\sqrt{\frac{\pi}{2}}\, e^{\frac{3}{2^{n+3}}}\, 2^{-\frac{n}{2}}$.*

Proof. Let us introduce the following family of subsets of the set $\mathcal{B}_n$ of all Boolean functions on $\mathbb{F}_2^n$:

$$F_a \;=\; \{f : \mathbb{F}_2^n \to \mathbb{F}_2 \mid S_f = \mathbb{F}_2^n \setminus \{a\}\},$$

where $a \in \mathbb{F}_2^n$.

It is easily shown that all the subsets F_a have the same cardinality: fix $a \in \mathbb{F}_2^n \setminus \{0\}$ and define the mapping φ_a from $\mathcal{B}_n$ to $\mathcal{B}_n$ which maps $f \in \mathcal{B}_n$ to $f \oplus \ell_a$ (where ℓ_a denotes the linear mapping on $\mathbb{F}_2^n$ defined as $\ell_a(x) := a \cdot x$ for every $x \in \mathbb{F}_2^n$). Given $f \in F_a$, we have $S_{\varphi_a(f)} = a + S_f = \mathbb{F}_2^n \setminus \{0\}$ (see Subsection 3.1). Hence φ_a is a bijection between F_a and F_0.

We deduce from the inclusion $\displaystyle\bigcup_{a\in\mathbb{F}_2^n} F_a \subseteq \mathcal{B}_n$ and from the fact that the sets F_a are pairwise disjoint that $|\mathcal{B}_n| \geq 2^n|F_0|$. Hence $|F_0| \leq 2^{2^n-n}$.

Finally, the density in $\mathcal{E}_n$ of F_0 is equal to $\frac{|F_0|}{|\mathcal{E}_n|}$. It is well-known that $|\mathcal{E}_n| = \binom{2^n}{2^{n-1}}$. Moreover Lemma A.2 provides the following lower bound on $|\mathcal{E}_n|$: $\binom{2^n}{2^{n-1}} \geq \sqrt{\frac{2}{\pi}}\, 2^{2^n-\frac{n}{2}}\, e^{-\frac{3}{2^{n+3}}}$. This lower bound together with the upper bound $|F_0| \leq 2^{2^n-n}$ yields to the result. □

Appendix A. Lower bounds on binomial coefficients

Lemma A.1 (Robbins, [11])**.** *For $n \geq 1$,*

$$n! \;=\; \sqrt{2\pi n}\,\left(\frac{n}{e}\right)^n\, e^{r(n)}$$

where $r(n)$ satisfies $\frac{1}{12n+1} < r(n) < \frac{1}{12n}$.

Lemma A.2. *For $n \geq 1$,*

$$\binom{2^n}{2^{n-1}} \geq \sqrt{\frac{2}{\pi}}\, 2^{2^n - \frac{n}{2}}\, e^{-\frac{3}{2^{n+3}}}$$

Proof. By definition , $\binom{2^n}{2^{n-1}} = \frac{2^n!}{\left(2^{n-1}!\right)^2}$. If we use Lemma A.1, then we get

$$\binom{2^n}{2^{n-1}} \geq \frac{\sqrt{\pi 2^{n+1}}\left(\frac{2^n}{e}\right)^{2^n} e^{r(2^n)}}{\pi 2^n \left(\frac{2^{n-1}}{e}\right)^{2^n} e^{2r(2^{n-1})}} = \sqrt{\frac{2}{\pi}} 2^{2^n - \frac{n}{2}} e^{r(2^n) - 2r(2^{n-1})}$$

Now $r(2^n) - 2r(2^{n-1}) \geq \frac{1}{12\,2^n + 1} - \frac{2}{12\,2^{n-1}} \geq -\frac{3}{2^{n+3}}$. □

Appendix B. On the Walsh supports of Boolean functions in five variables

Berlekamp and Welsh [1] shown that the set of all Boolean functions on 5 variables can be reduced to 48 equivalence classes with 29 equivalence classes of even Hamming weight and 19 equivalence classes of odd Hamming weight.

The equivalence class of 0 is simply formed by affine functions on $\mathbb{F}_2^5$ (whose Walsh supports are the singletons of $\mathbb{F}_2^5$). The Walsh support of Boolean functions of odd Hamming weights is the whole space $\mathbb{F}_2^5$. These equivalence classes are:

$$x_1x_2x_3x_4x_5$$
$$x_1x_2 \oplus x_1x_2x_3x_4x_5$$
$$x_1x_2 \oplus x_3x_4 \oplus x_1x_2x_3x_4x_5$$
$$x_1x_2x_3 \oplus x_1x_2x_3x_4x_5$$
$$x_1x_2 \oplus x_1x_2x_3 \oplus x_1x_2x_3x_4x_5$$
$$x_1x_4 \oplus x_1x_2x_3 \oplus x_1x_2x_3x_4x_5$$
$$x_4x_5 \oplus x_1x_2x_3 \oplus x_1x_2x_3x_4x_5$$
$$x_1x_4 \oplus x_2x_5 \oplus x_1x_2x_3 \oplus x_1x_2x_3x_4x_5$$
$$x_1x_2 \oplus x_4x_5 \oplus x_1x_2x_3 \oplus x_1x_2x_3x_4x_5$$
$$x_1x_2 \oplus x_3x_4 \oplus x_1x_2x_3 \oplus x_1x_2x_3x_4x_5$$
$$x_1x_2x_3 \oplus x_1x_4x_5 \oplus x_1x_2x_3x_4x_5$$
$$x_1x_2 \oplus x_1x_2x_3 \oplus x_1x_4x_5 \oplus x_1x_2x_3x_4x_5$$
$$x_2x_3 \oplus x_1x_2x_3 \oplus x_1x_4x_5 \oplus x_1x_2x_3x_4x_5$$
$$x_2x_3 \oplus x_4x_5 \oplus x_1x_2x_3 \oplus x_1x_4x_5 \oplus x_1x_2x_3x_4x_5$$

$$\begin{gathered}
x_2x_4 \oplus x_1x_2x_3 \oplus x_1x_4x_5 \oplus x_1x_2x_3x_4x_5\\
x_2x_3 \oplus x_2x_4 \oplus x_1x_2x_3 \oplus x_1x_4x_5 \oplus x_1x_2x_3x_4x_5\\
x_2x_4 \oplus x_3x_5 \oplus x_1x_2x_3 \oplus x_1x_4x_5 \oplus x_1x_2x_3x_4x_5\\
x_2x_3 \oplus x_2x_4 \oplus x_3x_5 \oplus x_1x_2x_3 \oplus x_1x_4x_5 \oplus x_1x_2x_3x_4x_5\\
x_2x_3 \oplus x_2x_4 \oplus x_3x_5 \oplus x_4x_5 \oplus x_1x_2x_3 \oplus x_1x_4x_5 \oplus x_1x_2x_3x_4x_5
\end{gathered}$$

By computer search, we also found Boolean functions of even weights whose Walsh support is the whole space $\mathbb{F}_2^5$. These equivalence classes are:

$$\begin{gathered}
x_4x_5 \oplus x_1x_2x_3\\
x_1x_2x_3 \oplus x_1x_4x_5\\
x_2x_4 \oplus x_1x_2x_3 \oplus x_1x_4x_5
\end{gathered}$$

Concerning the other equivalence classes of even Hamming weights, we begin with Boolean functions whose Walsh support is a flat or the complement of a flat with respect to $\mathbb{F}_2^5$ (clearly these properties are invariant under equivalence and addition of affine function). We adopt the following convention to write the Boolean functions and Walsh supports: the Boolean functions are written in abbreviated notation; for example, we simply write $12+2345$ for $x_1x_2 \oplus x_2x_3x_4x_5$. We also write the flats in abbreviated notation. For example, $1 + (234)$ denotes the flat $e_1 + \text{span}\{e_2, e_3, e_4\}$ where $(e_1, e_2, e_3, e_4, e_5)$ denotes the canonical basis of $\mathbb{F}_2^5$ while $\overline{(234)}$ denotes the complement of $\text{span}\{e_2, e_3, e_4\}$ with respect to $\mathbb{F}_2^5$.

Equivalence class	Walsh support
2345	(2345)
23 + 2345	(2345)
23 + 45 + 2345	(2345)
24 + 35 + 123 + 2345	$\overline{4+5+(23)}$
12 + 34 + 2345	$\overline{5+(234)}$
12 + 34	(1234)
123	(123)
12	(12)

For the other equivalence classes of even Hamming weights, we can write the Walsh supports of 9 equivalences classes as a symmetric difference of two flats or the complement of a symmetric difference of two flats with respect to $\mathbb{F}_2^5$.

Equivalence class	Walsh support
12 + 2345	$1 + (2345)\,\Delta\,(2)$
123 + 2345	$1 + (2345)\,\Delta\,(23)$
12 + 123 + 2345	$(2345)\,\Delta\,1 + (23)$
24 + 123 + 2345	$\overline{5 + (234)\,\Delta\,3 + (2)}$
45 + 123 + 2345	$(2345)\,\Delta\,1 + (23)$
12 + 34 + 123 + 2345	$(2345)\,\Delta\,1 + (23)$
14 + 123	$4 + (123)\,\Delta\,(1)$
14 + 35 + 123 + 2345	$\overline{1 + 2 + (345)\,\Delta\,1 + 4 + 5 + (23)}$
14 + 25 + 123	$(1245)\,\Delta\,4 + 5 + (123)$

Concerning the remaing equivalence classes, their Walsh supports can not be written in a simple form as above. We introduce the rank and affine rank of a subset of $\mathbb{F}_2^5$. The rank of a subset E of $\mathbb{F}_2^5$ is the dimension of the subspace of $\mathbb{F}_2^5$ generated by E while the affine rank of a subset E of $\mathbb{F}_2^5$ is the dimension of the smallest flat containing E. These notions have been used by Carlet and Charpin [5] to classify the cubic Boolean functions on n variables which are $(n-4)$-resilient. Note that the rank and the affine rank are constant on an equivalence class. For all the remaining equivalence classes, the rank and the affine rank are equal to 5 but not the rank and the affine rank of the complement of their Walsh supports with respect to $\mathbb{F}_2^5$. Therefore, for each remaining equivalence class, we give the cardinal $|S_f|$ of its Walsh support, the rank k and the affine rank $\mathbf{k}$ of the complement of the Walsh support with respect to $\mathbb{F}_2^5$.

Equivalence class	$\lvert S_f\rvert$	$\mathbf{k}$	k
23 + 24 + 35 + 123 + 145	16	5	5
23 + 123 + 145	13	5	5
123 + 145 + 2345	23	4	4
12 + 45 + 123 + 2345	25	4	4
14 + 123 + 2345	21	4	5
24 + 35 + 123 + 145 + 2345	26	4	4
24 + 45 + 123 + 145 + 2345	22	4	5
45 + 123 + 145 + 2345	22	4	5

References

[1] E. R. Berlekamp, L. R. Welch. Weight Distributions of the Cosets of the $(32, 6)$ Reed-Muller Code. *IEEE Transactions on Information Theory* **18**, 1972.

[2] P. Camion, C. Carlet, P. Charpin, N. Sendrier, On correlation-immune functions, Advances in Cryptology: Crypto '91, Proceedings, Lecture Notes in Computer Science, V. 576, 1991, pp. 86–100.

[3] C. Carlet. Partially-bent functions. *Designs, Codes and Cryptography*, **3**:135–145, 1993.

[4] C. Carlet. More correlation-immune and resilient functions over Galois fields and Galois rings. *Advances in Cryptology, EUROCRYPT' 97, Lecture Notes in Computer Science* 1233, pp. 422-433, Springer Verlag (1997)

[5] C. Carlet and P. Charpin. Cubic Boolean functions with highest resiliency, to appear in *IEEE Trans. on Inf. Theory*, vol. 51, 2005.

[6] C. Carlet and P. Sarkar. Spectral Domain Analysis of Correlation Immune and Resilient Boolean Functions. *Finite fields and Applications* 8, pp. 120-130, 2002.

[7] C. Carlet and Y. Tarannikov. Covering sequences of Boolean functions and their cryptographic significance. *Designs, Codes and Cryptography*, **25**:263–279, 2002.

[8] S. Dubuc. Characterization of linear structures. *Designs, Codes and Cryptography* vol. 22, pp. 33-45, 2001.

[9] F. J. MacWilliams and N. J. A. Sloane, The theory of error-correcting codes, North–Holland, Amsterdam, 1977.

[10] J. Maiorana, A Classification of the Cosets of the Reee-Muller Code R(1,6), *Math. of computation*, vol. 57, No. 195, July 1991, pp. 403-414.

[11] H. Robbins. A Remark on Stirling Formula. *Amer. Math. Monthly*, 62: 26–29, 1955.

PLATEAUED ROTATION SYMMETRIC BOOLEAN FUNCTIONS ON ODD NUMBER OF VARIABLES

A. Maximov[1], M. Hell[2] and S. Maitra[3]

Abstract. The class of Rotation Symmetric Boolean Functions (RSBFs) has received serious attention in searching functions of cryptographic significance. These functions are invariant under circular translation of indices. In this paper we study such functions on odd number of variables and interesting combinatorial properties related to Walsh spectra of such functions are revealed. In particular we concentrate on plateaued functions (functions with three valued Walsh spectra) in this class and derive necessary conditions for existence of balanced rotation symmetric plateaued functions. As application of our result we theoretically show the non existence of 9-variable, 3-resilient RSBF with nonlinearity 240 that has been posed as an open question in FSE 2004. Further we show how one can make efficient search in the space of RSBFs based on our theoretical results and as example we complete the search for unbalanced 9-variable, 3rd order correlation immune plateaued RSBFs with nonlinearity 240.

Keywords: Boolean Functions, Balancedness, Combinatorial Cryptography, Correlation Immunity, Nonlinearity, Walsh Transform.

[1] Department of Information Technology, Lund University
P.O. Box 118, 221 00 Lund, Sweden
email: movax@it.lth.se

[2] Department of Information Technology, Lund University
P.O. Box 118, 221 00 Lund, Sweden
email: martin@it.lth.se

[3] Applied Statistics Unit, Indian Statistical Institute,
203, B T Road, Kolkata 700 108, INDIA
email: subho@isical.ac.in

1. Introduction

In designing cryptographically significant Boolean functions, many requirements have to be fulfilled, such as balancedness, nonlinearity, algebraic degree, correlation immunity, resistance from algebraic attacks etc. Some of them may contradict each other, e.g., bent functions, which have highest possible nonlinearity, can not be balanced. Getting the best possible trade-off among these parameters has always been a challenging task as we can see in many papers (see [13, 14, 16] and the references in these papers). The class of Rotation symmetric Boolean functions (RSBFs) is a class of functions that are invariant under circular translation of indices. It has been shown that many functions in this class are rich in terms of cryptographic properties [2,5,7,10,15,16]. Further the RSBF class is much smaller ($\approx 2^{\frac{2^n}{n}}$) compared to the space of n-variable Boolean functions (2^{2^n}) and hence search techniques work much better in this smaller class. Given Boolean functions on even number of input variables, the best possible nonlinearity can be achieved when the magnitude of all the Walsh spectra values are the same. However, this is not possible when the number of input variables are odd. In such a scenario, the functions with three valued Walsh spectra $0, \pm\lambda$ may be investigated [1, 18], which are known as plateaued functions. It has been noted that there are functions with very good cryptographic properties in this class [1, 18].

In [16], two data structures, the matrices ${}_n\mathcal{A}$ and ${}_n\mathcal{B}$, were presented and made the search for RSBFs more efficient. The matrix ${}_n\mathcal{B}$ is used for fast generation of the truth table from its algebraic normal form, and ${}_n\mathcal{A}$ is used for fast calculation of the Walsh transform for the RSBF. In this paper we investigate the matrix ${}_n\mathcal{A}$ in detail. We introduce a new matrix, ${}_n\mathcal{H}$, which is a sub matrix of ${}_n\mathcal{A}$, for *odd* n, after some permutation. This allows us to improve the calculation of the Walsh transform for RSBFs and provides much better combinatorial insight to the problem. Our matrix structure can be used to make a concrete study on plateaued RSBFs on odd number of variables and we could provide necessary conditions on existence of balanced plateaued RSBFs. The construction of 9-variable, 3-resilient Boolean function with nonlinearity 240 is still an unsolved open question in literature [13, 14].

In [16] an estimate to search such functions in rotation symmetric class has been presented which needed search of 2^{43} Boolean functions and could not be completed in [16]. Since such functions are plateaued functions, we apply our results to theoretically show the nonexistence of 9-variable, 3-resilient, nonlinearity-240 functions in the rotation symmetric class. Furthermore, using the matrix ${}_n\mathcal{H}$, we found efficient search strategies for plateaued RSBFs which are much faster than what presented in [16]. We also use efficient implementation strategy in software to make the search faster. As an example of our search efficiency we exhaustively searched for unbalanced 9-variable, 3rd order correlation immune, algebraic degree 5 and nonlinearity-240 RSBFs and found $2 \cdot 8406$ many such functions. The search took only 6064 seconds against the estimated time of 3 years[1] as presented in [16].

2. **Preliminaries**

A Boolean function on n variables may be viewed as a mapping from $V_n = \{0,1\}^n$ into $V_1 = \{0,1\}$. We interpret a Boolean function $f(x_1, \ldots, x_n)$ as the output column of its *truth table*, i.e., a binary string of length 2^n,

$$f = [f(0,0,\ldots,0), f(1,0,\ldots,0), f(0,1,\ldots,0), \ldots, f(1,1,\ldots,1)].$$

We say that a Boolean function f is *balanced* if the truth table contains an equal number of 1's and 0's.

The *Hamming weight* of a binary string S is the number of ones in the string. This number is denoted by $wt(S)$. The *Hamming distance* between two strings, S_1 and S_2 is denoted $d_H(S_1, S_2)$ and is the number of places where S_1 and S_2 differ. Note that $d_H(S_1, S_2) = wt(S_1 \oplus S_2)$.

Any Boolean function $f(x_1, \ldots, x_n)$ has a unique representation as a polynomial over F_2, called the *algebraic normal form* (ANF),

[1] Note that, we have attempted to make the search (as explained in [16]) faster using efficient software implementation and found that it is possible to implement optimized code that can search the complete space in 470 hours using a Pentium M 1.6 GHz computer with 512 MB RAM. We have also parallelized the effort over a few computers and searched the complete space as explained in [8].

as

$$a_0 \oplus \bigoplus_{1\le i\le n} a_i x_i \oplus \bigoplus_{1\le i<j\le n} a_{ij}x_ix_j \oplus \ldots \oplus a_{12\ldots n}x_1x_2\ldots x_n,$$

where the coefficients $a_0, a_{ij}, \ldots,$ $a_{12\ldots n} \in \{0,1\}$. The *algebraic degree*, $\deg(f)$, is the number of variables in the highest order term with non-zero coefficient. A Boolean function is *affine* if there exists no term of degree > 1 in the ANF and the set of all affine functions is denoted $A(n)$. An affine function with constant term equal to zero is a *linear* function. The *nonlinearity* of an n-variable function f is the minimum distance from the set of all n-variable affine functions,

$$nl(f) = \min_{g\in A(n)} (d_H(f,g)).$$

Boolean functions used in ciphers must have high nonlinearity to prevent linear attacks [6, 9].

Many properties of Boolean functions can be described by the *Walsh transform.* Let $x = (x_1, \ldots, x_n)$ and $\omega = (\omega_1, \ldots, \omega_n)$ both belonging to $\{0,1\}^n$ and $x \cdot \omega = x_1\omega_1 \oplus \ldots \oplus x_n\omega_n$. Let $f(x)$ be a Boolean function on n variables. Then the *Walsh transform* of $f(x)$ is a real valued function over $\{0,1\}^n$ which is defined as

$$W_f(\omega) = \sum_{x\in\{0,1\}^n} (-1)^{f(x)\oplus x\cdot\omega}.$$

A Boolean function f is balanced iff $W_f(0) = 0$. The nonlinearity of f is given by $nl(f) = 2^{n-1} - \frac{1}{2}\max_{\omega\in\{0,1\}^n} |W_f(\omega)|$. Correlation immune functions and resilient functions are two important classes of Boolean functions. A function is m-resilient (respectively mth order correlation immune) iff its Walsh transform satisfies

$$W_f(\omega) = 0, \text{ for } 0 \le wt(\omega) \le m \text{ (respectively } 1 \le wt(\omega) \le m).$$

Following the same notation as in [13, 14, 16] we use (n, m, d, σ) to denote an n-variable, m-resilient function with degree d and nonlinearity σ. Furthermore, by $[n, m, d, \sigma]$ we denote an unbalanced n-variable, mth order correlation immune function with degree d and nonlinearity σ.

2.1. Rotation Symmetric Boolean Functions

The rotation symmetric Boolean functions are invariant under cyclic rotation of inputs. Let $x_i \in \{0,1\}$ for $1 \le i \le n$. For $1 \le k \le n$, we define the permutation $\rho_n^k(x_i)$ as

$$\rho_n^k(x_i) = \begin{cases} x_{i+k}, & \text{if } i+k \le n \\ x_{i+k-n}, & \text{if } i+k > n \end{cases}$$

Let $(x_1, x_2, \ldots, x_{n-1}, x_n) \in V_n$. Then we extend the definition as $\rho_n^k(x_1, x_2, \ldots, x_{n-1}, x_n) = (\rho_n^k(x_1), \rho_n^k(x_2), \ldots, \rho_n^k(x_{n-1}), \rho_n^k(x_n))$. Hence, ρ_n^k acts as k cyclic rotation on an n-bit vector.

Definition 2.1. A Boolean function f is called *Rotation Symmetric* if for each input $(x_1, \ldots, x_n) \in \{0,1\}^n$, $f(\rho_n^k(x_1, \ldots, x_n)) = f(x_1, \ldots, x_n)$ for $1 \le k \le n$.

The inputs to a rotation symmetric Boolean function can be divided into partitions so that each partition consists of all cyclic shifts of one input. A partition is generated by

$$G_n(x_1, x_2, \ldots, x_n) = \{\rho_n^k(x_1, x_2, \ldots, x_n) | 1 \le k \le n\}$$

and the number of such partitions is denoted by g_n. Thus the number of n-variable RSBFs is 2^{g_n}. Let $\phi(k)$ be Euler's *phi*-function, then it can be shown by Burnside's lemma that (see also [15])

$$g_n = \frac{1}{n} \sum_{k|n} \phi(k)\, 2^{\frac{n}{k}}.$$

By $g_{n,w}$ we denote the number of partitions with weight w. It can also be checked that the number of degree w RSBFs is $(2^{g_{n,w}} - 1)2^{\sum_{i=0}^{w-1} g_{n,i}}$. For the formula of how to calculate $g_{n,w}$ for arbitrary n and w, we refer to [15].

A *partition*, or *group*, can be represented by its *representative element* $\Lambda_{n,i}$. This is the lexicographically first element belonging to the group. The representative elements are again arranged lexicographically. *The rotation symmetric truth table* (RSTT) is defined as the g_n-bit string

$$[f(\Lambda_{n,0}), f(\Lambda_{n,1}), \ldots, f(\Lambda_{n,g_n-1})].$$

In [16] it was shown that the Walsh transform takes the same value for all elements belonging to the same group, i.e., $W_f(u) = W_f(v)$ if $u \in G_n(v)$.

In [16], two matrices were introduced, ${}_n\mathcal{A}$ and ${}_n\mathcal{B}$, for efficient search of RSBFs. The matrix ${}_n\mathcal{A}$ is defined as

$${}_n\mathcal{A}_{i,j} = \sum_{x \in G_n(\Lambda_{n,i})} (-1)^{x \cdot \Lambda_{n,j}},$$

for an n-variable RSBF. Using this $g_n \times g_n$ matrix, the Walsh spectra for an RSBF can be calculated from the RSTT as

$$W_f(\Lambda_{n,j}) = \sum_{i=0}^{g_n - 1} (-1)^{f(\Lambda_{n,i})} \, {}_n\mathcal{A}_{i,j}.$$

The notation of ρ_n^k can be extended, in a similar fashion, to monomials. For example, if we have a 4 variable rotation symmetric Boolean function and the term $x_1x_2x_3$ is present in the ANF, then the terms $x_2x_3x_4, x_3x_4x_1$ and $x_4x_1x_2$ must also be present in the ANF. We can associate n-bit pattern $(x_1, x_2, \ldots, x_n)$ of $\Lambda_{n,i}$ with a monomial as well. If there is a '1' in the corresponding position we say that the variable is present in the monomial. Considering this, the $g_n \times g_n$ matrix ${}_n\mathcal{B}$ is defined as [16]

$${}_n\mathcal{B}_{i,j} = \bigoplus_{e \in G_n(\Lambda_{n,j})} e|_{\Lambda_{n,i}}.$$

That is, we take a function with all monomials coming from one group, represented by $\Lambda_{n,j}$. Then we check the value of the function when the input is $\Lambda_{n,i}$. This value is put in the location ${}_n\mathcal{B}_{i,j}$. With this matrix, one can get the RSTT of the function from the ANF.

Note that the ANF of the RSBFs are such that if one monomial from a rotational symmetric group is present in the ANF then all the other monomials of that rotational symmetric group are also present [7, 16]. Thus the algebraic normal from of any RSBF possesses a very nice and regular form. The algebraic attack (see [3,11] and the references in these papers) is getting a lot of attention recently. To resist algebraic attacks, the Boolean functions used in the cryptosystems should be chosen properly. It is shown [3] that given any n-variable Boolean function f, it is always possible to get a Boolean function g with degree at most $\lceil \frac{n}{2} \rceil$ such that $f * g$ is of degree at most $\lceil \frac{n}{2} \rceil$. Here the functions are considered to be multivariate polynomials over GF(2) and $f * g$ is the polynomial multiplication. Thus while choosing an f, the cryptosystem designer should be careful that it should not happen that degree of $f * g < \lceil \frac{n}{2} \rceil$ where g is also a low degree function. Recently there are experimental evidences that the RSBFs are good in terms of algebraic immunity [4] and this gives a good motivation to study the RSBFs for cryptographic purposes.

3. Walsh Spectra of RSBFs

In this section we derive combinatorial results related to RSBFs and their Walsh spectra. We first start with a technical result that counts the number of groups of t elements when $t|n$. This result will be used later to analyse the Walsh spectra of balanced plateaued RSBFs. In fact, the result is true for classes of cyclically shift-invariant binary sequences irrespective of their usage in RSBFs.

Theorem 3.1. *For an n-variable RSBF the number of groups with t elements is $d_{n,t} = \frac{1}{t}\sum_{k|t} \mu(\frac{t}{k}) 2^{\gcd(n,k)}$,* for $t = 1, 2, \ldots, n$, *where $\mu(t)$ is the Möbius function, i.e., $\mu(t) = 1$, if $t = 1$, $\mu(t) = 0$, if $e_i \geq 2$ and $\mu(t) = (-1)^m$, otherwise, when $t = p_1^{e_1} p_2^{e_2} \ldots p_m^{e_m}$ is factorized in powers of m distinct primes, $p_1, p_2 \ldots p_m$.*

Proof. Let $S = \{0,1\}^n$ and $x \in S$. Denote by p_t the number of elements for which $\rho_n^t(x) = x$. Since the number of orbits for the permutor ρ_n^t is $\gcd(n,t)$, and each orbit must contain all 0's or all 1's to fulfill the condition $\rho_n^t(x) = x$, the number of combinations must be $p_t = |\{x \in S : \rho_n^t(x) = x\}| = 2^{\gcd(n,t)}$. A recursive expression for $d_{n,t}$ can be derived as

$d_{n,1} = 2$ and $d_{n,t} = (p_t - \sum_{k|t,k<t} k \cdot d_{n,k})/t$.

Each element $x \in S$ must be counted once in some group t. First we count how many elements will be counted in groups of size t, and then divide this number by t, in order to get the number of such groups $d_{n,t}$. Hence, $t \cdot d_{n,t} = 2^{\gcd(n,t)} - \sum_{\substack{k|t \\ k<t}} k \cdot d_{n,k} \Rightarrow \sum_{k|t} k \cdot d_{n,k} = 2^{\gcd(n,t)}$. We use Möbius function $\mu(t)$ to invert the expression. Hence, $d_{n,t} = \frac{1}{t}\sum_{k|t} \mu(\frac{t}{k}) 2^{\gcd(n,k)}$. □

Corollary 3.2. $g_n = \sum_{t=1}^{n} d_{n,t}$ and $|S| = \sum_{t=1}^{n} t \cdot d_{n,t} = 2^n$.

3.1. **Investigation of ${}_n\mathcal{A}$ Matrix for n Odd**

We consider ${}_n\mathcal{A}$ when n is an *odd* number and note that the number of groups with *even* $wt(\Lambda_{n,i})$ is the same as the number of groups with *odd* $wt(\Lambda_{n,i})$. Moreover, if we consider all $\Lambda_{n,i}$ with *even* Hamming weights and denote by $\overline{\Lambda}_{n,i}$ the representative element for the group containing the complement of $\Lambda_{n,i}$, it is easy to note that $G_n(\Lambda_{n,i}) \neq G_n(\overline{\Lambda}_{n,j})$ for any i, j. Hence, the set of groups can be divided into two equal parts containing representative elements of even weight and odd weight, respectively.

Permute the matrix ${}_n\mathcal{A}$ using a permutation π such that the first $g_n/2$ rows correspond to the representative elements, $\Lambda_{n,i}$, of even weight and the second $g_n/2$ rows correspond to the complements of them. That is we first list the representative elements $\lambda_{n,i}$ with even weights in lexicographical order for $i = 0$ to $\frac{g_n}{2} - 1$. Then we put the elements (these are of odd weights) in the order such that $\Lambda_{n,i} = \overline{\Lambda}_{n,i-\frac{g_n}{2}}$ for $i = \frac{g_n}{2}$ to $g_n - 1$. In the permutation we swap rows and the corresponding columns of ${}_n\mathcal{A}$. We denote the resulting matrix by ${}_n\mathcal{A}^\pi$ and show that ${}_n\mathcal{A}^\pi$ is of the form

$$ {}_n\mathcal{A}^\pi = \left(\begin{array}{c|c} {}_n\mathcal{H} & {}_n\mathcal{H} \\ \hline {}_n\mathcal{H} & -{}_n\mathcal{H} \end{array}\right), $$

where ${}_n\mathcal{H}$ is a sub matrix of ${}_n\mathcal{A}^\pi$.

Let us consider $n = 5$, for which $g_n = 8$. In [16], the group representatives are ordered lexicographically, i.e., $(0,0,0,0,0)$, $(0,0,0,0,1)$, $(0,0,0,1,1)$, $(0,0,1,0,1)$, $(0,0,1,1,1)$, $(0,1,0,1,1)$, $(0,1,1,1,1)$, $(1,1,1,1,1)$. We get the matrix ${}_5\mathcal{A}$. On the other hand if we permute them as $(0,0,0,0,0)$, $(0,0,0,1,1)$, $(0,0,1,0,1)$, $(0,1,1,1,1)$, $(1,1,1,1,1)$, $(0,0,1,1,1)$, $(0,1,0,1,1)$, $(0,0,0,0,1)$, i.e., even weight elements and then the corresponding odd weight elements, we get the matrix ${}_5\mathcal{A}^\pi$ which is of a nice sub matrix structure.

$$ {}_5\mathcal{A} = \left(\begin{array}{rrrr|rrrr} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 5 & 3 & 1 & 1 & -1 & -1 & -3 & -5 \\ 5 & 1 & 1 & -3 & 1 & -3 & 1 & 5 \\ 5 & 1 & -3 & 1 & -3 & 1 & 1 & 5 \\ \hline 5 & -1 & 1 & -3 & -1 & 3 & 1 & -5 \\ 5 & -1 & -3 & 1 & 3 & -1 & 1 & -5 \\ 5 & -3 & 1 & 1 & 1 & 1 & -3 & 5 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \end{array}\right), $$

$$
{}_5\mathcal{A}^\pi = \left(\begin{array}{cccc|cccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
5 & 1 & -3 & 1 & 5 & 1 & -3 & 1 \\
5 & -3 & 1 & 1 & 5 & -3 & 1 & 1 \\
5 & 1 & 1 & -3 & 5 & 1 & 1 & -3 \\
\hline
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
5 & 1 & -3 & 1 & -5 & -1 & 3 & -1 \\
5 & -3 & 1 & 1 & -5 & 3 & -1 & -1 \\
5 & 1 & 1 & -3 & -5 & -1 & -1 & 3
\end{array}\right).
$$

We now present the proof with the following results. Let $X \wedge Y$ and $X \oplus Y$ denote bitwise AND respectively XOR for the vectors X and Y.

Proposition 3.3. *Let $A = (a_1, a_2, \ldots, a_n) \in \{0,1\}^n$ and $B = (b_1, b_2, \ldots, b_n) \in \{0,1\}^n$. If $wt(A)$ and $wt(B)$ is an even number and if n is odd, then*

$$\bigoplus_{i=1}^{n}(a_i \wedge b_i) = \bigoplus_{i=1}^{n}(\overline{a}_i \wedge b_i) = \bigoplus_{i=1}^{n}(a_i \wedge \overline{b}_i) = 1 \oplus \bigoplus_{i=1}^{n}(\overline{a}_i \wedge \overline{b}_i). \quad (1)$$

Proof. We have $(X \wedge Y) \oplus (\overline{X} \wedge Y) = (X \oplus \overline{X}) \wedge Y = 1 \wedge Y = Y$. Since $\bigoplus_{i=1}^n \Big((a_i \wedge b_i) \oplus (\overline{a}_i \wedge b_i)\Big) = \bigoplus_{i=1}^n b_i = 0$, it follows that $\bigoplus_{i=1}^n (a_i \wedge b_i) = \bigoplus_{i=1}^n (\overline{a}_i \wedge b_i)$. The second equality in (1) also follows immediately. Similarly, we can write $(X \wedge \overline{Y}) \oplus (\overline{X} \wedge \overline{Y}) = (X \oplus \overline{X}) \wedge \overline{Y} = 1 \wedge \overline{Y} = \overline{Y}$. Since $\bigoplus_{i=1}^n \Big((a_i \wedge \overline{b}_i) \oplus (\overline{a}_i \wedge \overline{b}_i)\Big) = \bigoplus_{i=1}^n \overline{b}_i = 1$, it follows that $\bigoplus_{i=1}^n (a_i \wedge \overline{b}_i) = 1 \oplus \bigoplus_{i=1}^n (\overline{a}_i \wedge \overline{b}_i)$ □

Theorem 3.4. *When n is odd, the matrix ${}_n\mathcal{A}^\pi$ is of the form*

$${}_n\mathcal{A}^\pi = \left(\begin{array}{c|c} {}_n\mathcal{H} & {}_n\mathcal{H} \\ \hline {}_n\mathcal{H} & -{}_n\mathcal{H} \end{array}\right),$$

where ${}_n\mathcal{H}$ is a $\frac{g_n}{2} \times \frac{g_n}{2}$ matrix.

Proof. Since the matrix ${}_n\mathcal{A}^\pi$ is written such that $\Lambda_{n,i}$ corresponds to row/column i and $\overline{\Lambda}_{n,i}$ corresponds to row/column $g_n/2 + i$, we can write the following. For $0 \le r, c < g_n/2$ we have

$$\begin{aligned}
{}_n\mathcal{A}^{\pi}_{r,c} &= \sum_{x\in G_n(\Lambda_{n,r})} (-1)^{x\cdot\Lambda_{n,c}}\\
&= \sum_{x\in G_n(\Lambda_{n,r})} (-1)^{\bigoplus_{i=1}^{n}(x_i\wedge\Lambda_{(n,c)_i})}\\
{}_n\mathcal{A}^{\pi}_{r,c+\frac{g_n}{2}} &= \sum_{x\in G_n(\Lambda_{n,r})} (-1)^{x\cdot\Lambda_{n,c+\frac{g_n}{2}}}\\
&= \sum_{x\in G_n(\Lambda_{n,r})} (-1)^{\bigoplus_{i=1}^{n}(x_i\wedge\overline{\Lambda}_{(n,c)_i})}\\
{}_n\mathcal{A}^{\pi}_{r+\frac{g_n}{2},c} &= \sum_{x\in G_n(\Lambda_{n,r+\frac{g_n}{2}})} (-1)^{x\cdot\Lambda_{n,c}}\\
&= \sum_{x\in G_n(\Lambda_{n,r})} (-1)^{\bigoplus_{i=1}^{n}(\overline{x}_i\wedge\Lambda_{(n,c)_i})}\\
{}_n\mathcal{A}^{\pi}_{r+\frac{g_n}{2},c+\frac{g_n}{2}} &= \sum_{x\in G_n(\Lambda_{n,r+\frac{g_n}{2}})} (-1)^{x\cdot\Lambda_{n,c+\frac{g_n}{2}}}\\
&= \sum_{x\in G_n(\Lambda_{n,r})} (-1)^{\bigoplus_{i=1}^{n}(\overline{x}_i\wedge\overline{\Lambda}_{(n,c)_i})}
\end{aligned}$$

Since the number of 1's in $\Lambda_{n,i}$ is even, $0 \leq i < g_n/2$, it follows from Proposition 3.3 that ${}_n\mathcal{A}^{\pi}_{r,c} = {}_n\mathcal{A}^{\pi}_{r,c+\frac{g_n}{2}} = {}_n\mathcal{A}^{\pi}_{r+\frac{g_n}{2},c} = -\,{}_n\mathcal{A}^{\pi}_{r+\frac{g_n}{2},c+\frac{g_n}{2}}$. □

Corollary 3.5. *The first column of the matrix ${}_n\mathcal{A}$ contains exactly $d_{n,t}$ values of t, for $t = 1, 2, \ldots, n$. Also, for n odd, $d_{n,t}$ is an even number.*

Proof. The first column ${}_n\mathcal{A}_{i,0}$ is constructed as

$${}_n\mathcal{A}_{i,0} = \sum_{x\in G_n(\Lambda_{n,i})} (-1)^{x\cdot\vec{0}} = |G_n(\Lambda_{n,i})|,$$

since we know that there are $d_{n,t}$ groups with $|G_n(\Lambda_{n,i})| = t$, the first part of the corollary follows.

We have proved that for odd n, ${}_n\mathcal{A}$ can be constructed through the matrix ${}_n\mathcal{H}$ which must contain $\frac{d_{n,t}}{2}$ groups of size t in the first column. Hence, $d_{n,t}$ is even. □

In Subsection 2.1 we defined the RSTT of an RSBF as the g_n-bit string

$$[f(\Lambda_{n,0}), f(\Lambda_{n,1}), \ldots, f(\Lambda_{n,g_{n-1}})],$$

where $\Lambda_{n,0}, \Lambda_{n,1}, \ldots, \Lambda_{n,g_n-1}$ are ordered lexicographically. Given Theorem 3.4, from now on, we consider the RSTT$^\pi$, where we first list the representative elements $\lambda_{n,i}$ with even weights in lexicographical order for $i = 0$ to $\frac{g_n}{2} - 1$. Then we put the elements in the order such that $\Lambda_{n,i} = \overline{\Lambda}_{n,i-\frac{g_n}{2}}$ for $i = \frac{g_n}{2}$ to $g_n - 1$. In the rest of the document, we will use only this ordering (permutation) and by abuse of notations, apply (RSTT, RSTT$^\pi$) and $({}_n\mathcal{A}, {}_n\mathcal{A}^\pi)$ as the same thing unless specifically mentioned.

3.2. Improved Walsh Transform Computation

The fact that ${}_n\mathcal{A}^\pi$ is of this form reduces the number of operations needed to calculate the Walsh spectra for an RSBF. For notation purposes, divide the RSTT into two partitions, σ_1 and σ_2, such that RSTT $= \{0,1\}^{g_n} = \{0,1\}^{g_n/2} \parallel \{0,1\}^{g_n/2} = \sigma_1 \parallel \sigma_2$. We define a one-to-one mapping

$$\mu_\sigma : \sigma_1 \parallel \sigma_2 = \{0,1\}^{\frac{g_n}{2}} \parallel \{0,1\}^{\frac{g_n}{2}}$$

$$\longrightarrow \sigma_1^* \parallel \sigma_2^* = (-1)^{\{0,1\}^{\frac{g_n}{2}}} \parallel (-1)^{\{0,1\}^{\frac{g_n}{2}}},$$

i.e., if $\sigma_{1_i} = 0$ then $\sigma_{1_i}^* = 1^0 = +1$, otherwise $\sigma_{1_i}^* = (-1)^1 = -1$.

Then we can define

$$w_1 = \sigma_1^* \; {}_n\mathcal{H}, w_2 = \sigma_2^* \; {}_n\mathcal{H} \tag{2}$$

and $W_f(\omega) = ((w_1 + w_2) \parallel (w_1 - w_2))$. In the following, we will sometimes refer to w_1 and w_2 as *partial Walsh transform*, or just *pWT*. To compute the Walsh transform using the matrix ${}_n\mathcal{A}$, g_n^2 operations must be done. In the case when ${}_n\mathcal{H}$ is used, the number of operations is instead $2 \cdot \left(\frac{g_n}{2}\right)^2 + g_n = \frac{g_n^2}{2} + g_n \leq g_n^2$.

3.3. Plateaued RSBFs

A Boolean function on odd number of variables is said to be plateaued [1,18] if its Walsh transform takes only the three values 0 and ±λ, where λ is some positive integer. We call λ the *amplitude* of the function.

Following the notation (2) from Subsection 3.2, for plateaued RSBFs we get,

$$w_{1_i} + w_{2_i} = 0 \text{ or } \pm\lambda, w_{1_i} - w_{2_i} = 0 \text{ or } \pm\lambda. \tag{3}$$

$w_{1_i} + w_{2_i}$	$w_{1_i} - w_{2_i}$	w_{1_i}	w_{2_i}
0	0	0	0
0	$+\lambda$	$+\lambda/2$	$-\lambda/2$
0	$-\lambda$	$-\lambda/2$	$+\lambda/2$
$+\lambda$	0	$+\lambda/2$	$+\lambda/2$
$+\lambda$	$+\lambda$	$+\lambda$	0
$+\lambda$	$-\lambda$	0	$+\lambda$
$-\lambda$	0	$-\lambda/2$	$-\lambda/2$
$-\lambda$	$+\lambda$	0	$-\lambda$
$-\lambda$	$-\lambda$	$-\lambda$	0

Table 1. Possible values for w_{1_i} and w_{2_i} when searching for plateaued RSBFs.

There are only 9 valid pairs (w_{1_i}, w_{2_i}) fulfilling (3) and they are listed in Table 1. This means that w_{1_i} and $w_{2_i} \in \{0,\ \pm\lambda/2,\ \pm\lambda\}$, i.e., they can only take 5 values. The partition of the matrix ${}_n\mathcal{A}^{\pi}$ as in Theorem 3.4 and Table 1 give us the following result.

Proposition 3.6. *Consider an RSBF on odd number of variables represented by the RSTT* $(\sigma_1 \parallel \sigma_2)$.

(1) *If it is plateaued then the functions with RSTT* $(\sigma_2 \parallel \sigma_1)$, $(\overline{\sigma_1} \parallel \overline{\sigma_2})$, $(\overline{\sigma_2} \parallel \overline{\sigma_1})$, $(\sigma_1 \parallel \overline{\sigma_2})$, $(\overline{\sigma_2} \parallel \sigma_1)$, $(\overline{\sigma_1} \parallel \sigma_2)$ *and* $(\sigma_2 \parallel \overline{\sigma_1})$ *are also plateaued.*

(2) *If it is correlation immune (respectively resilient) then the functions with RSTT* $(\sigma_2 \parallel \sigma_1)$, $(\overline{\sigma_1} \parallel \overline{\sigma_2})$, *and* $(\overline{\sigma_2} \parallel \overline{\sigma_1})$ *are also correlation immune (respectively resilient).*

3.4. Necessary condition for balanced plateaued RSBFs

Based on the above discussion, we now present concrete results on necessary conditions for existence of balanced plateaued RSBFs.

Theorem 3.7. *For n odd, if there exist an n-variable balanced plateaued RSBF with amplitude $\lambda = 2^k$, then the following condition must be satisfied:*

There exist $k'_1 \dots k'_n$ and $k''_1 \dots k''_n$, $k^*_i \in [0 \dots \frac{d_{n,i}}{2}]$, and $\tau \in \{0,1\}$, such that $\sum_{t=1}^{n} t \cdot k'_t = \frac{\tau\lambda + 2^n}{4}, \sum_{t=1}^{n} t \cdot k''_t = \frac{-\tau\lambda + 2^n}{4}$.

Proof. If the function $(\sigma_1 \parallel \sigma_2)$ is balanced, then from Table 1, the partial Walsh transform (pWT) for the first column must be $\{0, \frac{\pm\lambda}{2}\}$, i.e., $(\sigma_1^* \cdot {}_n\mathcal{H})[0] = \tau \cdot \frac{\lambda}{2}, (\sigma_2^* \cdot {}_n\mathcal{H})[0] = -\tau \cdot \frac{\lambda}{2}$, for $\tau \in \{0,1\}$. In the first column there are $\frac{d_{n,t}}{2}$ groups of size t. Let for k'_t

of them σ_1^* get $(+1)$, and for the rest $(\frac{d_{n,t}}{2} - k'_t)$ it will be (-1). Then pWT for the first column is expressed as $(\sigma_1^* \cdot {}_n\mathcal{H})[0] = \sum_{t=1}^{n}[t \cdot k'_t - (\frac{d_{n,t}}{2} - k'_t) \cdot t] = \tau \cdot \frac{\lambda}{2} \Rightarrow 2\sum_{t=1}^{n} k'_t \cdot t = \frac{\tau\lambda}{2} + \sum_{t=1}^{n} t \cdot \frac{d_{n,t}}{2} \Rightarrow \sum_{t=1}^{n} t \cdot k'_t = \frac{\tau\lambda + 2^n}{4}$, for $k'_t = [0 \dots \frac{d_{n,t}}{2}]$.

The similar expression for $(\sigma_2^* \cdot {}_n\mathcal{H})[0]$ is $\sum_{t=1}^{n} t \cdot k''_t = \frac{-\tau\lambda + 2^n}{4}$, for $k''_t = [0 \dots \frac{d_{n,t}}{2}]$. □

Now we present the result for non existence of $(9, 3, 5, 240)$ RSBF, which has been posed as an open question in [16].

Theorem 3.8. *A* $(9, 3, 5, 240)$ *RSBF can not exist.*

Proof. Note that this function is plateaued [14]. Thus we analyze 9-variable balanced plateaued functions for $\lambda = 2^5$ and for this we need to study the ${}_9\mathcal{H}$ matrix. Since $2^9 = 512$, for a balanced function to exist, it must be that $1 \cdot k_1 + 3 \cdot k_3 + 9 \cdot k_9 = \frac{\pm\tau \cdot 2^5 + 512}{4}$ following Theorem 3.7 and we get the only solution $k'_1 = 1, k'_3 = 0, k'_9 = 15$ and $k''_1 = 0, k''_3 = 1, k''_9 = 13$ for $\tau = 1$.

Let us now consider the value of $W_f(011011011)$, which must be any one of $0, \pm 32$. Let $W_f(011011011) = w_{1_i} + w_{2_i}$. From Table 1 we get that w_{1_i}, w_{2_i} can take values $0, \pm 16$. To get the exact values of w_{1_i}, w_{2_i} one needs to look at the last but one column of the matrix ${}_9\mathcal{H}$. The matrix ${}_9\mathcal{H}$ can be seen as $[A|B]$ where A, B are respectively the following two matrices.

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
9 & 5 & 1 & 1 & 5 & 1 & 1 & 1 & 1 & 1 & -3 & -3 & 1 & -3 & 1 \\
9 & 1 & 5 & 1 & 1 & 1 & 1 & -3 & 1 & -3 & 1 & 1 & -7 & 1 & -3 \\
9 & 1 & 1 & 5 & -3 & 1 & -3 & 1 & -3 & -3 & -3 & 1 & -3 & -3 & -3 \\
9 & 5 & 1 & -3 & 1 & -7 & -3 & 1 & -3 & -3 & 1 & -3 & 5 & 1 & -3 \\
9 & 1 & 1 & 1 & -7 & 5 & -3 & -3 & -3 & 1 & 1 & -3 & 5 & 1 & 1 \\
9 & 1 & 1 & -3 & -3 & -3 & 1 & -3 & 1 & -3 & -3 & 5 & 1 & 5 & 5 \\
9 & 1 & -3 & 1 & 1 & -3 & -3 & 1 & -3 & 5 & 1 & -3 & -3 & 1 & 5 \\
9 & 1 & 1 & -3 & -3 & -3 & 1 & -3 & 1 & 5 & 5 & 5 & 1 & -3 & -3 \\
9 & 1 & -3 & -3 & -3 & 1 & -3 & 5 & 5 & 1 & 5 & 1 & -3 & -3 & 1 \\
9 & -3 & 1 & -3 & 1 & 1 & -3 & 1 & 5 & 5 & 1 & -3 & -3 & 1 & -3 \\
9 & -3 & 1 & 1 & -3 & -3 & 5 & -3 & 5 & 1 & -3 & 1 & 5 & -3 & 1 \\
9 & 1 & -7 & -3 & 5 & 5 & 1 & -3 & 1 & -3 & -3 & 5 & 1 & -3 & -3 \\
9 & -3 & 1 & -3 & 1 & 1 & 5 & 1 & -3 & -3 & 1 & -3 & -3 & 1 & 5 \\
9 & 1 & -3 & -3 & -3 & 1 & 5 & 5 & -3 & 1 & -3 & 1 & -3 & 5 & 1 \\
9 & 5 & 1 & -3 & 1 & -3 & 1 & 5 & 1 & -3 & 1 & 1 & 1 & 1 & -3 \\
9 & -3 & -3 & 5 & 1 & -3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
9 & -3 & -3 & 5 & 1 & -3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
9 & -3 & -3 & 1 & 5 & 1 & 1 & -3 & 1 & -3 & 5 & -3 & 1 & 5 & -3 \\
9 & -7 & 5 & -3 & 5 & 1 & -3 & 5 & -3 & 1 & -3 & 1 & 5 & -3 & 1 \\
9 & 1 & 1 & 1 & 1 & -3 & 5 & -3 & -3 & 1 & 1 & -3 & -3 & -7 & 1 \\
9 & 1 & -3 & 1 & 1 & 1 & 1 & -3 & -7 & 5 & 1 & 1 & 1 & 1 & -3 \\
9 & 1 & -3 & -3 & -3 & 5 & 1 & 1 & 1 & 1 & -3 & -3 & 1 & -3 & 1 \\
9 & 1 & -3 & 1 & 1 & 1 & -7 & -3 & 1 & -3 & 1 & 1 & 1 & 1 & 5 \\
9 & 1 & 1 & 1 & 1 & -3 & -3 & -3 & 5 & 1 & -7 & -3 & -3 & 1 & 1 \\
9 & -3 & 5 & -3 & 1 & 1 & -3 & -3 & -3 & 1 & 1 & 5 & -3 & 1 & 1 \\
9 & -3 & 1 & 1 & -3 & 1 & 1 & 1 & 1 & 1 & -3 & -3 & 1 & 5 & -7 \\
9 & -3 & 1 & 1 & -3 & 1 & 1 & 1 & 1 & -7 & 5 & -3 & 1 & -3 & 1 \\
3 & -1 & -1 & 3 & -1 & -1 & -1 & 3 & -1 & -1 & -1 & 3 & -1 & -1 & -1 \\
9 & 5 & 5 & 5 & 1 & 5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}$$

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
5 & -3 & -3 & -3 & -7 & 1 & 1 & 1 & 1 & 1 & -3 & -3 & -3 & -3 & 5 \\
1 & -3 & -3 & -3 & 5 & 1 & -3 & -3 & -3 & 1 & 5 & 1 & 1 & -3 & 5 \\
-3 & 5 & 5 & 1 & -3 & 1 & 1 & -3 & 1 & 1 & -3 & 1 & 1 & 9 & 5 \\
1 & 1 & 1 & 5 & 5 & 1 & 1 & -3 & 1 & 1 & 1 & -3 & -3 & -3 & 1 \\
-3 & -3 & -3 & 1 & 1 & -3 & 1 & 5 & 1 & -3 & 1 & 1 & 1 & -3 & 5 \\
1 & 1 & 1 & 1 & -3 & 5 & 1 & 1 & -7 & -3 & -3 & 1 & 1 & -3 & 1 \\
5 & 1 & 1 & -3 & 5 & -3 & -3 & 1 & -3 & -3 & -3 & 1 & 1 & 9 & 1 \\
1 & 1 & 1 & 1 & -3 & -3 & -7 & 1 & 1 & 5 & -3 & 1 & 1 & -3 & 1 \\
-3 & 1 & 1 & -3 & 1 & 1 & 5 & 1 & -3 & 1 & 1 & 1 & -7 & -3 & 1 \\
1 & 1 & 1 & 5 & -3 & 1 & 1 & -3 & 1 & -7 & 1 & -3 & 5 & -3 & 1 \\
1 & 1 & 1 & -3 & 1 & -3 & 1 & -3 & 1 & -3 & 5 & -3 & -3 & 9 & 1 \\
1 & 1 & 1 & 1 & 5 & -3 & 1 & 1 & 1 & -3 & -3 & 1 & 1 & -3 & 1 \\
1 & 1 & 1 & 5 & -3 & -7 & 1 & -3 & 1 & 1 & 1 & 5 & -3 & -3 & 1 \\
-3 & 1 & 1 & -3 & 1 & 1 & -3 & 1 & 5 & 1 & 1 & -7 & 1 & -3 & 1 \\
-3 & -3 & -3 & -3 & -3 & -3 & 1 & 1 & 1 & -3 & 1 & 5 & 5 & 9 & -3 \\
-3 & -7 & 1 & 1 & 1 & 5 & -3 & -3 & 5 & -3 & -3 & 5 & -3 & -3 & 1 \\
-3 & 1 & -7 & 1 & 1 & -3 & 5 & -3 & -3 & 5 & -3 & -3 & 5 & -3 & 1 \\
-3 & 1 & 1 & 1 & -3 & 1 & -3 & 5 & -3 & 1 & 1 & -3 & -3 & 9 & 1 \\
-3 & 1 & 1 & -3 & 1 & 1 & -3 & 1 & -3 & 1 & 1 & 1 & 1 & -3 & 1 \\
-3 & 5 & -3 & 1 & 1 & 5 & 1 & 5 & 1 & -3 & 1 & 1 & 1 & -3 & -3 \\
1 & -3 & 5 & -3 & -3 & 1 & 5 & -3 & -3 & 1 & 5 & 1 & 1 & -3 & -3 \\
1 & -3 & -3 & 5 & 1 & 5 & -3 & -3 & -3 & 5 & 1 & 1 & 1 & 9 & -3 \\
1 & 5 & -3 & -3 & -3 & 1 & -3 & -3 & 5 & 1 & 5 & 1 & 1 & -3 & -3 \\
-3 & -3 & 5 & 1 & 1 & -3 & 1 & 5 & 1 & 5 & 1 & 1 & 1 & -3 & -3 \\
1 & -3 & -3 & 1 & 1 & 1 & 5 & 1 & 5 & 1 & -3 & -3 & -3 & 9 & -3 \\
5 & 5 & -3 & -3 & 1 & 1 & 1 & 1 & 1 & 1 & -3 & 5 & -3 & -3 & -3 \\
5 & -3 & 5 & -3 & 1 & 1 & 1 & 1 & 1 & 1 & -3 & -3 & 5 & -3 & -3 \\
3 & -1 & -1 & 3 & -1 & -1 & -1 & 3 & -1 & -1 & 3 & -1 & -1 & -1 & -1 \\
-3 & 1 & 1 & 1 & 1 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -7
\end{pmatrix}$$

Let us represent the last but one column of the matrix ${}_9\mathcal{H}$ as a column matrix C. Thus $w_{1_i} = \sigma_1^* C$ and $w_{2_i} = \sigma_2^* C$, where σ_1^*, σ_2^* are as given in Subsection 3.2. The values k_1', k_1'' correspond to the top most element of C, which is 1 and the values k_3', k_3'' correspond to the last but one element of C, which is -1. The values of k_9', k_9'' correspond to the other 28 values in the column matrix C, where twenty one many values are -3 and seven many values are 9. Let $k_9' = a' + b'$ and $k_9'' = a'' + b''$, where a', a'' correspond to the values -3 and b', b'' correspond to the values 9. Now $w_{1_i} = \sigma_1^* C = 1 \times 1 + (-1) \times (-1) + (2a' - 21) \times (-3) + (2b' - 7) \times 9 = 2 - 6a' + 18b'$. Also, we have $a' + b' = k_9' = 15$. Thus the only possible solution is $a' = 12, b' = 3$ and in that case $w_{1_i} = -16$. Similarly, $w_{2_i} = \sigma_2^* C = (-1) \times 1 + 1 \times (-1) + (2a'' - 21) \times (-3) + (2b'' - 7) \times 9 = -2 - 6a'' + 18b''$. Also, we have $a'' + b'' = k_9'' = 13$. Thus the only possible solution is $a'' = 9, b'' = 4$ and in that case $w_{2_i} = 16$. Hence $W_f(011011011) = w_{1_i} + w_{2_i} = 0$. From Theorem 3.4, it follows that if $W_f(011011011) = w_{1_i} + w_{2_i}$ then $W_f(001001001) = w_{1_i} - w_{2_i}$. Thus, $W_f(001001001) = -32 \neq 0$. Hence, from definition, the function can not be 3-resilient. This proves that there can not be any $(9, 3, 5, 240)$ RSBF. □

We have checked the necessary condition is satisfied for $\lambda = 2^{\frac{n+1}{2}}$ for odd composite $n = 15, 21$ and 25. For $n = 15$, the solutions are $k_1' = 1, k_3' = 0, k_5' = 1, k_{15}' = 550$ and $k_1'' = 0, k_3'' = 1, k_5'' = 2, k_{15}'' = 541$ when $\tau = 1$. For $n = 21$, the solutions are $k_1' = 0, k_3' = 1, k_7' = 1, k_{21}' = 24990$ or $k_1' = 0, k_3' = 1, k_7' = 4, k_{21}' = 24989$ or $k_1' = 0, k_3' = 1, k_7' = 7, k_{21}' = 24988$ and $k_1'' = 1, k_3'' = 0, k_7'' = 2, k_{21}'' = 24941$ or $k_1'' = 1, k_3'' = 0, k_7'' = 5, k_{21}'' = 24940$ or $k_1'' = 1, k_3'' = 0, k_7'' = 8, k_{21}'' = 24939$ when $\tau = 1$. For $n = 25$, the solutions are $k_1' = 1, k_5' = 1, k_{25}' = 335626$ and $k_1'' = 0, k_5'' = 2, k_{25}'' = 335462$ when $\tau = 1$. Note that there is no solution with $\tau = 0$. It will be interesting to find out some general solution pattern for odd composite n's from the necessary condition of Theorem 3.7, which is done for odd prime n's in Corollary 3.9 below. Further we need to study the other columns of the matrix ${}_n\mathcal{H}$ as in the proof of Theorem 3.8 if we like to prove the non existence results for these cases.

Corollary 3.9. *For a balanced plateaued RSBF on $n \geq 3$ variables, n prime, τ can only be $(+1)$, i.e., pWT must take the value $\pm\lambda/2$. Further, the necessary condition of Theorem 3.7 is always satisfied for n prime and $\lambda = 2^{\frac{n+1}{2}}$.*

Proof. For n prime, in the first column of ${}_n\mathcal{H}$ we have 1 row with $(+1)$ and $\frac{2^{n-1}-1}{n}$ rows with values $(+n)$. With $\tau = 0$ we require σ_1^* such that $pWT = 0$, i.e., $(k \cdot n \pm 1)$ must be 0, for some k. For prime $n \geq 3$ there is no such k.

Now we prove the second part. For n prime, $d_{n,1} = 2$ and $d_{n,n} = \frac{2^n-2}{n}$. Thus we get an equation of the form $1 \cdot k_1 + n \cdot k_n = \frac{\tau\lambda+2^n}{4} = \pm 2^{\frac{n-3}{2}} + 2^{n-2}$, where $k_1 \in [0,1]$ and $k_n \in [0,\dots,\frac{2^{n-1}-1}{n}]$. We show that it is always possible to get an integer solution for k_1, k_n.

Note that for $n > 3$ prime, $n | 2^{n-1} - 1$, i.e., $n | (2^{\frac{n-1}{2}} + 1)(2^{\frac{n-1}{2}} - 1)$.

If $n | (2^{\frac{n-1}{2}} + 1)$, then $n | 2^{\frac{n-3}{2}}(2^{\frac{n-1}{2}} + 1)$, i.e., $n | 2^{\frac{n-3}{2}} + 2^{n-2}$. Thus for $\tau = 1$, we take $k_1' = 0$. Also, $n | 2^{\frac{n-3}{2}} + 2^{n-2} - (2^{\frac{n-1}{2}} + 1)$, i.e., $n | -2^{\frac{n-3}{2}} + 2^{n-2} - 1$. Thus for $\tau = -1$, we take $k_1'' = 1$.

If $n | (2^{\frac{n-1}{2}} - 1)$, then $n | 2^{\frac{n-3}{2}}(2^{\frac{n-1}{2}} - 1)$, i.e., $n | -2^{\frac{n-3}{2}} + 2^{n-2}$. Thus for $\tau = -1$, we take $k_1'' = 0$. Also, $n | -2^{\frac{n-3}{2}} + 2^{n-2} + (2^{\frac{n-1}{2}} - 1)$, i.e., $n | 2^{\frac{n-3}{2}} + 2^{n-2} - 1$. Thus for $\tau = 1$, we take $k_1' = 1$. □

Existence of $(n, \frac{n-3}{2}, \frac{n+1}{2}, 2^{n-1} - 2^{\frac{n-1}{2}})$ functions for odd n is an important open question in Boolean function literature [13,14,16]. These functions are plateaued with $\lambda = 2^{\frac{n+1}{2}}$. The only results available are for $n = 5, 7$ as described in [12]. Corollary 3.9 shows that the necessary condition is satisfied for any odd prime n when we search in the class of RSBFs. This gives a partial theoretical justification why such functions were available in the RSBF class for $n = 5, 7$ as observed in [15]. Thus it will be interesting to target the problem for $n = 11$ also.

4. Search Strategy

Based on the theoretical results discussed so far, we present how these results can be used for actual search for RSBFs with certain cryptographic properties. It has been observed in [16] that to search for $(9, 3, 5, 240)$ one needs to check for 2^{43} many RSBFs. Though we have already proved theoretically that such RSBF does not exist, we now show that the search can be reduced to 2^{34} only. This search also produces the $[9, 3, 5, 240]$ functions and we implement the search to get the complete list of $[9, 3, 5, 240]$ RSBFs. Apart from the theoretical results, we exploit nontrivial software implementation to make the search much faster. This is important since the search space becomes larger for higher number of variables and best possible software implementation is required for actual search.

The algorithm uses only the matrix ${}_n\mathcal{H}$ in the search. The idea behind the algorithm is very simple and it can be used to find plateaued RSBFs for a *desired* Walsh transform, e.g., m-resilient or mth order correlation immune.

The first step of the algorithm is to search the complete set of σ_1's such that $w_1 = \sigma_1^* \cdot {}_n\mathcal{H}$ only take values from the set $w_{1_i} \in \{0, \pm\lambda/2, \pm\lambda\}$. Note that in the positions where the Walsh transform must be zero, the corresponding values of the pWT must be $w_{1_i} \in \{0, \pm\lambda/2\}$, three valued only. Let us denote this set of σ_1's by $\mathcal{S}_{\sigma_1}$. From (2) and Table 1 we see that $w_2 = \sigma_2 \cdot {}_n\mathcal{H}$ is calculated in the same way and has the same restrictions, so it means that $\mathcal{S}_{\sigma_2} = \mathcal{S}_{\sigma_1}$.

The second step of the algorithm is to calculate the Walsh transform for $(\sigma_1 \parallel \sigma_2)$ in the space $\mathcal{S}_{\sigma_1} \times \mathcal{S}_{\sigma_2}$. It means that we need to save $\mathcal{S}_{\sigma_1}$ in a list or in a file.

The time complexity for the first step to find $\mathcal{S}_{\sigma_1}$ is $O(2^{g_n/2})$ and the second step has the complexity $O(|\mathcal{S}_{\sigma_1}|^2)$, so the total time complexity is $O(2^{g_n/2})+O(|\mathcal{S}_{\sigma_1}|^2)$. Note that in this strategy we do not care about what degree we have on the functions, all functions with desired Walsh spectra will be found.

Now we describe how to use the proposed search strategy to implement an exhaustive search for $[9, 3, 5, 240]$ functions. For RSBFs on 9 variables there are $g_9 = 60$ groups and, hence, the total search space for these functions is 2^{60}. However, in the ANF there can not be terms of degree 6, 7, 8 or 9 and, at least one term of degree 5 must be present. Therefore, the search space does not include all RSBFs on 9 variables, instead the search space is of size $2^{\sum_{i=1}^{4} g_{9,i}}(2^{g_{9,5}} - 1) = 2^{29}(2^{14} - 1) \approx 2^{43}$. This is the complexity of the algorithm when one first uses the ${}_n\mathcal{B}$ matrix and then the ${}_n\mathcal{A}$ matrix in the search [16], without considering ${}_n\mathcal{H}$. The term of degree 0 is not considered in the search space.

The restrictions on Walsh spectra for a $[9, 3, 5, 240]$ function are $W_f(\omega) = 0$, for $1 \le wt(\omega) \le 3$ and $W_f(\omega) = 0$ or $\pm\, 32$, for $wt(\omega) = 0, wt(\omega) > 3$. We do not use the restriction that the function has a certain degree, instead we only use the matrix ${}_n\mathcal{H}$ to reduce the time complexity. Since $g_9 = 60$, the matrix ${}_n\mathcal{H}$ is of

Boolean functions on 9 variables	2^{512}
RSBFs on 9 variables	2^{60}
Finding [9,3,5,240] using matrices ${}_n\mathcal{A}$, ${}_n\mathcal{B}$ [16]	2^{43}
Finding [9,3,5,240] using our strategy	2^{34}

TABLE 2. Different search strategy complexities.

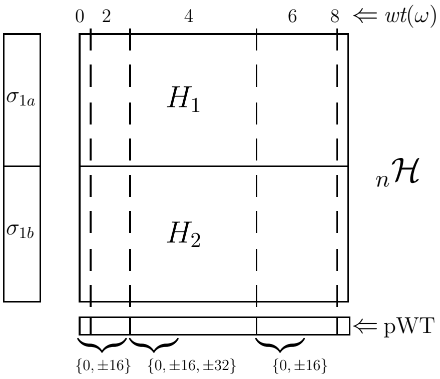


FIGURE 1. For fast implementation purposes, the matrix ${}_n\mathcal{H}$ is divided into sections.

size 30 × 30. We divide the RSTT into 2 parts, σ_1 and σ_2, each of 30 bits, and generate the set $\mathcal{S}_{\sigma_1}$. By simulation we found that this set is of size $|\mathcal{S}_{\sigma_1}| \approx 2^{17}$ so there is no memory problem with storing the complete set in memory. *This will give us the total search of* 2^{34}*, which is* 2^9 *times faster than only using* ${}_n\mathcal{A}$ *and* ${}_n\mathcal{B}$ *as done in [16].*

Although the complexity is reduced it is important to minimize the constant time needed to check each candidate pair. For fast implementation purposes we divide the matrix ${}_n\mathcal{H}$ into two sections, H_1 and H_2 as shown in Figure 1, each containing 15 rows. We divide σ_1 in the same way and denote the two parts $\sigma_1 = (\sigma_{1a} \parallel \sigma_{1b})$. For each section, the sum of the rows is precomputed for each of the 2^{15} possible inputs, and these sums are stored in the table $H_{fast}[2][2^{15}][30]$, having 2 sections with 2^{15} possible inputs for each, and the result is a vector of 30 integers. Now to calculate the partial Walsh transform we only need 2 table look ups and a maximum of 30 integer summations. Unnecessary computation can be avoided by calculating the values of the pWT one by one. If one value is not valid, then we stop and select the next σ_1. Since $W_f(\omega)$ must be 0 for $wt(\omega) \leq 3$, the pWT in these positions must be in $\{0, \pm 16\}$. Note that the complement of the representative elements of weight 6 and 8 have weights 1 and 3, so in these positions pWT must also be in $\{0, \pm 16\}$. In the rest positions, pWT $\in \{0, \pm 16, \pm 32\}$. These restrictions can be seen in Table 1. When $\mathcal{S}_{\sigma_1}$ is found, we try all combinations for the cartesian product $(\mathcal{S}_{\sigma_1} \times \mathcal{S}_{\sigma_1})$ and check if the Walsh transform is valid for a $[9, 3, 5, 240]$ function. Since $\mathcal{S}_{\sigma_2} = \mathcal{S}_{\sigma_1}$, we can use the same precomputed tables for fast calculation of $w_2 = \sigma_2^* \cdot {}_n\mathcal{H}$.

The exact search time required is 6064 seconds on a computer with Pentium M 1.6 GHz processor and 512MB RAM using Windows XP operating system. In [16], it was estimated that the search will take almost 3 years to complete the search on a single Pentium 1.6 GHz computer with 256 MB RAM using Linux 7.2 operating system.

Using our strategy we could check that there is no resilient RSBFs with parameters $(9, 3, 5, 240)$ (already proved theoretically) and there are 8406 correlation immune functions with the same parameters $[9, 3, 5, 240]$, when the term of degree 0 is not considered. That is if we also consider the complement of the functions then there are 2 × 8406 many functions.

Let us denote the autocorrelation value of an n-variable Boolean function f with respect to the vector α as

$$\Delta_f(\alpha) = \sum_{x \in \{0,1\}^n} (-1)^{f(x) \oplus f(x \oplus \alpha)},$$

and the absolute indicator

$$\Delta_f = \max_{\alpha \in \{0,1\}^n, \alpha \neq \bar{0}} |\Delta_f(\alpha)|.$$

Low autocorrelation value is important for functions in cryptographic applications [17]. Thus we also check the Δ_f value for these $[9, 3, 5, 240]$ functions.

The Δ_f values of the functions are 80 (4956 many out of 8406), 96 (1020), 112 (312), 136 (180), 152 (1734) and 224 (204). A few correlation immune RSBFs with these parameters have been reported using simulated annealing based heuristic search [2]. We execute the search completely and show that the search space can be exhaustively analysed implying that the heuristic method is not required in this case.

It should be noted that we have only exploited the ${}_n\mathcal{H}$ matrix but not used the degree restrictions on the functions. The ${}_n\mathcal{B}$ matrix may also be used for faster search with ${}_n\mathcal{H}$.

Motivated by Corollary 3.9 and the discussion after it, we also attempted the search for $(11, 4, 6, 992)$ functions. Note that these functions are plateaued. Existence of these functions is not yet known. Since $g_{11} = 188$, the ${}_{11}\mathcal{H}$ matrix is 94×94 and the method of search that we attempt here will not work. Even if using the degree restriction and use of ${}_n\mathcal{B}$ matrix does not come to much help. We attempted some heuristic search and found an $(11, 1, 6, 992)$ plateaued RSBF with Δ_f value 240. Heuristic search, as attempted in [2] may come to help in such a scenario.

5. Conclusion

In this paper we studied the Walsh spectra of rotation symmetric Boolean functions. The set of rotation symmetric Boolean functions is much smaller than the complete space of Boolean functions. Even then complete search of RSBFs is not practical for $n \geq 9$. Our results provide combinatorial insight to the Walsh spectra of the functions and we show that some necessary conditions on existence of certain kinds of functions can be derived from them. In particular, we studied the plateaued RSBFs in this paper. The central result here is to show that the ${}_n\mathcal{A}$ matrix can be written as

$$\left(\begin{array}{c|c} {}_n\mathcal{H} & {}_n\mathcal{H} \\ \hline {}_n\mathcal{H} & -{}_n\mathcal{H} \end{array}\right)$$

after certain permutations when n is odd. Further research in this direction is to study these matrices in more details and to see whether some methods can be explored to analyse functions on higher number of variables. It should also be noted that the matrix structure we present here cannot be extended for n even and studying that case is also an interesting research area.

References

[1] C. Carlet and E. Prouff. On plateaued functions and their constructions. In *Fast Software Encryption 2003*, number 2887 in Lecture Notes in Computer Science, pages 54–73. Springer Verlag, 2003.

[2] J. Clark, J. Jacob, S. Maitra and P. Stanica. Almost Boolean Functions: The Design of Boolean Functions by Spectral Inversion. In *CEC 2003, the 2003 Congress on Evolutionary Computation*, Volume 3 in the proceedings, pages 2173–2180, IEEE Press, December 8–12, 2003, Canberra, Australia.

[3] N. Courtois and W. Meier. Algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology - EUROCRYPT 2003*, number 2656 in Lecture Notes in Computer Science, pages 345–359. Springer Verlag, 2003.

[4] D. K. Dalai, K. C. Gupta and S. Maitra. Results on Algebraic Immunity for Cryptographically Significant Boolean Functions. In *INDOCRYPT 2004*, pages 92–106, number 3348, Lecture Notes in Computer Science, Springer-Verlag.

[5] D. K. Dalai, S. Maitra and P. Stănică. Results on Rotation Symmetric Bent Functions, IACR eprint archive, eprint.iacr.org, no. 2005/118, 21 April 2005.

[6] C. Ding, G. Xiao, and W. Shan. *The Stability Theory of Stream Ciphers.* Number 561 in Lecture Notes in Computer Science. Springer-Verlag, 1991.

[7] E. Filiol and C. Fontaine. Highly nonlinear balanced Boolean functions with a good correlation-immunity. In *Advances in Cryptology - EUROCRYPT'98*. Springer-Verlag, 1998.

[8] M. Hell, A. Maximov. S. Maitra. On efficient implementation of search strategy for rotation symmetric Boolean functions. In *Ninth International Workshop on Algebraic and Combinatoral Coding Theory, ACCT 2004*, June 19–25, 2004, Black Sea Coast, Bulgaria.

[9] M. Matsui. Cryptanalysis method for DES cipher. In *Advances in Cryptology, Eurocrypt 1993*, Lecture Notes in Computer Science, Number 765, Pages 386–397, Springer-Verlag, 1994.

[10] A. Maximov. Classes of Plateaued Rotation Symmetric Boolean functions under Transformation of Walsh Spectra. In *WCC 2005*, Pages 325–334. Also available at IACR eprint server, no. 2004/354.

[11] W. Meier, E. Pasalic and C. Carlet. Algebraic attacks and decomposition of Boolean functions. In *Advances in Cryptology - EUROCRYPT 2004*, to be published in Lecture Notes in Computer Science. Springer Verlag, 2004.

[12] E. Pasalic, S. Maitra, T. Johansson and P. Sarkar. New constructions of resilient and correlation immune Boolean functions achieving upper bound on nonlinearity. In *Workshop on Coding and Cryptography - WCC 2001*, Paris, January 8–12, 2001. Electronic Notes in Discrete Mathematics, Volume 6, Elsevier Science, 2001.

[13] P. Sarkar and S. Maitra. Construction of nonlinear Boolean functions with important cryptographic properties. In *Advances in Cryptology - EUROCRYPT 2000*, number 1807 in Lecture Notes in Computer Science, pages 485–506. Springer Verlag, May 2000.

[14] P. Sarkar and S. Maitra. Nonlinearity bounds and construction of resilient Boolean functions. In *Advances in Cryptology - Crypto 2000*, number 1880 in Lecture Notes in Computer Science, pages 515–532, Springer-Verlag, 2000.

[15] P. Stănică and S. Maitra. Rotation Symmetric Boolean Functions – Count and Cryptographic Properties. In *R. C. Bose Centenary Symposium on Discrete Mathematics and Applications*, December 2002. Electronic Notes in Discrete Mathematics, Elsevier, Volume 15.

[16] P. Stănică, S. Maitra and J. Clark. Results on Rotation Symmetric Bent and Correlation Immune Boolean Functions. In *Fast Software Encryption 2004*, to be published in volume 3017 in Lecture Notes in Computer Science, Springer-Verlag, 2004.

[17] X-M. Zhang and Y. Zheng. GAC – the criterion for global avalanche characteristics of cryptographic functions. *Journal of Universal Computer Science*, 1(5):316–333, 1995.

[18] Y. Zheng and X. M. Zhang. Plateaued Functions. In *ICICS'99*, pages 284-300, volume 1726 in Lecture notes in Computer Science, Springer Verlag.

ANALYSIS OF AFFINELY EQUIVALENT BOOLEAN FUNCTIONS

Qingshu Meng[1], Min Yang[1], Huanguo Zhang[1] and Yuzhen Liu[1]

Abstract. By walsh transform, autocorrelation function, decomposition, derivation and modification of truth table, some new invariants are obtained. Based on invariant theory, we get two results: first a general algorithm which can be used to judge if two boolean functions are affinely equivalent and to obtain the affine equivalence relationship if they are equivalent; second, the classification of the Reed-Muller code $R(4,6)/R(1,6), R(3,7)/R(1,7)$.

1. **Introduction**

Boolean functions are used widely in science and engineering, like in circuit design, cryptography and error-correction coding. The affine classification of boolean functions is meaningful at least for the following two reasons: first, equivalent functions have similar properties (like Hamming weight distribution in error-correction coding, same nonlinearity property in cryptography). second, the number of representatives is much less than the number of boolean functions. Out of the need of circuit design, the classification of boolean functions under the action of general affine group was discussed much often in 60s in the 20th century [1–3]. Recently the analysis of affine equivalence of Boolean functions was discussed in several papers [4–8]. Fuller and Millan disclosed the affine equivalence between the output functions of the AES s-box by getting the affine equivalence relationship, but the method is not efficient in the case of bent functions. Method in paper [8]

[1] Comp. school, Wuhan Univ. Hubei China. email: `mqseagle@sohu.com`

is not efficient too in the bent functions case though it improves the efficiency of Fuller-Millan algorithm. In eurocrypt'03, a toolbox was developed to analyze affine equivalence between bijective s-box or s-box with small $n-m$, where n, m are numbers of inputs and outputs respectively, and thus the toolbox can't deal with Boolean functions, where $m=1$. In attacking HFE problem(hidden fields equation), Geiseleman gave an collum-wise method, but the method is not efficient in Boolean function with uneven truth table. Other papers on classifying Boolean functions can be found in papers [9–11]. To authors' knowledge, how to judge if two functions are equivalent and how to get the equivalent relationship if they are equivalent is not known in general case.

In this paper, first an algorithm is given which can efficiently solve the two above problems in general case. Second, we classify the Reed-Muller codes $R(4,6)/R(1,6), R(3,7)/R(1,7)$. The basic tools we use are Walsh transform, autocorrelation function, derivation function, decomposition, and modification of truth table.

2. Preliminary

For each subset $s \subseteq \{1, 2, \cdots, n\}$, there exists a corresponding vector $(s_1, s_2, \cdots, s_n)$ of dimension n by letting $s_i = 1$ if element i is in s else letting $s_i = 0$. And the vector $(s_1, s_2, \cdots, s_n), s_i \in \{0, 1\}$ for $i = 1, 2, \cdots, n$ can be denoted by an integer s whose 2-adic expansion is just the vector $(s_1, s_2, \cdots, s_n)$. Obviously, the set, the vector and the integer are isomorphic. In this paper, if confusion is not caused, we will use the three notations for description convenience. Denote by F_2 the Galois field with two elements $\{0, 1\}$ and denote by F_2^n the vector space over F_2. Denote by $p_n = F_2[x_1, x_2, \cdots, x_n]/(x_1^2 - x_1, \cdots, x_n^2 - x_n)$ the algebra of all functions $F_2^n \to F_2$. For each subset $s \subseteq \{1, 2, \cdots, n\}$, denote $\prod_{i \in s} x_i \in p_n$ by x^s. The algebraic normal form of a Boolean function $F_2^n \to F_2$ can be written as $f(x) = \sum_{s=0}^{2^n-1} a_s x^s$, where $a_s \in F_2$. The degree of $f(x)$ is defined as

$$deg(f) = \max_{s \in \{0,1,\cdots,2^n-1\}, a_s \neq 0} H(s),$$

and the low degree of $f(x)$ is defined as

$$ldeg(f) = \min_{s \in \{0,1,\cdots,2^n-1\}, a_s \neq 0} H(s),$$

where $H(s)$ is the Hamming weight of vector s. The set $\{f(x)|deg(f) \leq r\}$ is denoted by $R(r,n)$. Denote by $R(r,n)/R(s,n)$ the set $\{f(x)+R(s,n)|s < ldeg(f), deg(f) \leq r\}$.

Denote by $GL(n,2)$ the set of all nonsingular matrices of order n, i.e. the general linear group. Denote by $AGL(n,2)$ the general affine group $\{(A,b)|A \in GL(n,2), b \in F_2^n\}$. The group operation is defined as

$$(A,u)(B,w) = (AB, A(w)+u)$$
$$(A,u)^{-1} = (A^{-1}, A^{-1}(u)),$$

where $(A,u),(B,w) \in AGL(n,2)$.

The action of group $AGL(n,2)$ on Boolean functions is defined as:

$$\begin{array}{ll} c: & p_n \to p_n \\ by: & f(x) \to f \circ c = f(xA+b) \end{array},$$

where $c = (A,b) \in AGL(n,2)$.

Two functions $f(x), g(x) \in R(r,n)/R(s,n)$ are called equivalent if there exists $(A,b) \in AGL(n,2)$ such that $f(x) = g(xA+b) \mod R(s,n)$. An invariant of $R(r,n)/R(s,n)$ is a mapping M from $R(r,n)/R(s,n)$ to a set such that for any two equivalent functions $f(x), g(x) \in R(r,n)/R(s,n)$, $M(f) = M(g)$ holds.

3. Basic Transforms

3.1. Walsh Transform and Autocorrelation Function

Definition 3.1. Define

$$s_{(f)}(w) = \sum_{x \in F_2^n} (-1)^{f(x)}(-1)^{w \cdot x}$$

as the Walsh spectrum of $f(x)$ at vector w, where $f(x) \in p_n, w \in F_2^n$.

The transform is called the Walsh transform.

Definition 3.2. Define $c_f(s) = \sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{f(x+s)}$ be the autocorrelation function of $f(x)$, where $f(x) \in p_n, s \in F_2^n$.

The following two propositions are well known. And the fact that the distribution of absolute Walsh spectra and autocorrelation function are invariant under affine transform is known due to Preneel's work [12].

Proposition 3.3. *Let $f(x), g(x) \in p_n$ be two functions such that $g(x) = f(xA + b) + lx$, then for any $w \in F_2^n$,*

$$s_{(g)}(w) = (-1)^{(l+w)\cdot bA^{-1}} s_{(f)}((l + w)A^{-1T}),$$

where A^{-1T} is the transpose of A^{-1}.

Corollary 3.4. *The absolute Walsh spectrum of $f(x)$ at i is equal to the Walsh spectrum of $g(x)$ at j, where $j = l + iA^T$. Therefore the deficiency of the rank of vectors with the same absolute spectrum between two equivalent functions is at most 1. The distribution of absolute value of Walsh spectra of $f(x)$ is the same as that of $g(x)$.*

Proposition 3.5. *Let $f(x), g(x) \in p_n$ be two functions such that $g(x) = f(xA + b) + lx$, then for any given $s \in F_2^n$,*

$$c_g(s) = (-1)^{l\cdot s} c_f(sA).$$

Corollary 3.6. *The absolute autocorrelation function of $f(x)$ at j is equal to the absoulte autocorrelation function of $g(x)$ at i, where $j = iA$. Therefore the ranks of vectors with the same absolute autocorrelation function value are the same between two equivalent functions. The distribution of absolute value of autocorrelation function of $f(x)$ is the same as that of $g(x)$.*

3.2. **Derivation**

For any Boolean function $f(x) \in R(r, n)$, define its derivative function as $D_a(f) = f(x) + f(x + a)$. Similarly we can define two-order derivative function as $D_{a,b}(f) = f(x) + f(x+a) + f(x+b) + f(x + a + b)$. By the definition, it is easy to get the following two properties [13]:

(1) $D_{a,b}(f) = D_a(f) + D_b(f) + D_{a+b}(f)$.
(2) $D_a(f \circ B) = D_{aA}(f) \circ B$, where $B = (A, c) \in AGL(n, 2)$. similarly, $D_{a,b}(f \circ B) = D_{aA,bA}(f) \circ B$, where $B = (A, c) \in AGL(n, 2)$.

Proposition 3.7. *If $f(x) \in R(r,n)/R(s,n)$, then $D_a(f \circ B) = (D_{aA}(f)) \circ B$ mod $R(s-1,n)$, where $B = (A,b) \in AGL(n,2)$. If M is an invariant of $R(r-1,n)/R(s-1,n)$, then $M(D_a(f \circ B)) = M((D_{aA}(f)) \circ B)$, so $\{M(D_a(f))|a \in F_2^n\}$ is an invariant of $R(r,n)/R(s,n)$.*

Remark The derivative function was used by Hou [10] in classification of $R(3,7)/R(2,7)$ and by Brier [13] in classification of $R(3,9)/R(2,9)$. Proposition 3.7 is an extension of their result. Here we use the value $\{M(D_a(f)|a \in F_2^n\}$ of the invariant $\{M \circ D_a|a \in F_2^n\}$ instead of the invariant itself. In the following part, we use the value of an invariant instead of the invariant itself for convenience on several occasions.

3.3. **Decomposition**

Proposition 3.8. *Let $f(x), g(x) \in R(r,n)$ be two functions such that $g(x) = f(xA+b)$ mod $R(s,n)$. If $f(x) = (x_n+1)f_0(x') + x_n f_1(x')$, where $x' = (x_{n-1}, \cdots, x_1)$, then $g(x) = (x \cdot c_n + b_n + 1)f_0(x'') + (x \cdot c_n + b_n)f_1(x'')$ where $c_1, c_2, \cdots, c_n$ are the columns of the matrix A, and $x'' = (x \cdot c_{n-1} + b_{n-1}, \cdots, x \cdot c_1 + b_1)$. Obviously, $f_0(x'), f_1(x')$ are affinely equivalent to $f_0(x''), f_1(x'')$ mod $R(s,n-1)$ respectively. Similar result holds for two-vector based decomposition.*

By proposition 3.8, if $f(x)$ is decomposed into two subfunctions at vector a (like $a = (1,0,\cdots,0)$), then $g(x)$ can be decomposed into two subfunctions at vector $b = aA$(like the $b = aA = r_1$, the first row) such that the two subfunctions of $f(x)$ are equivalent to those of $g(x)$.

Proposition 3.9. *If M is an invariant of $R(r,n-1)/R(s,n-1)$, then the set $\{\{M(f_{ax=0}), M(f_{ax=1})\}|a \in F_2^n\}$ is an invariant of $R(r,n)/R(s,n)$.*

Remarks The basic idea of the decomposition of a function can be found early in Maiorana's paper [9], which made the classification of R(6,6)/R(1,6) possible early in the 90s. And recently it is used by Brier [13] to classify R(3,9)/R(2,9).

3.4. The Modification of Truth Table

Definition 3.10. For a function $f(x)$, define its 1-local connection functions as

$$f_i(x) = \{ \begin{array}{cc} f(x) & x \neq i \\ f(x)+1 & x = i \end{array} , i = 0, 1, \cdots , 2^n - 1.$$

similarly 2-local connection functions can be defined.

Proposition 3.11. *Let $f(x), g(x) \in R(r, n)$ be such that $g(x) = f(xA + b) + lx$, then $g_j(x) = f_i(xA + b) + lx$, where $jA = (i + b), i = 0, 1, \cdots , 2^n - 1$. Similar result holds for two-local connection functions.*

The above definition 3.10 can be found in [14] and proposition 3.11 can be found in [7].

Proposition 3.12. *Let $f(x) \in R(r, n)$. If M is an invariant of $R(n, n)/R(1, n)$, then $\{M(f_i(x))|i \in F_2^n\}$ is an invariant of $R(r, n)/R(1, n)$.*

4. The Analysis of Affinely Equivalent Boolean Functions

Algorithm 4.1. *input: two functions $f(x), g(x) \in R(n, n)$,*
output: A, b, and l, if $g(x) = f(xA + b) + lx$ else the functions are not equivalent.

(1) *Calculate the Walsh spectra and autocorrelation function of $f(x), g(x)$ respectively. Compare the distribution of absolute Walsh spectra and absolute autocorrelation function of $f(x)$ with those of $g(x)$ respectively. If the two functions have two same distributions, then go to step 2 else they are not affinely equivalent, exit.*

(2) *Denote the autocorrelation value of $g(x)$ at unit vector e_i by $c_g(e_i)$. By corollary 3.6, there exists at least one element $v \in \{v|abs(c_f(v)) = abs(c_g(e_i))\}$ such that $v = e_iA$ holds. Let $i = 1, 2, \cdots , n$, there are n equations.*

(3) *Decompose $f(x)$ at unit vector e_i, and calculate the invariant of the two subfunctions, denote it by $de_{e_i}(f)$. By proposition 3.9, there exists at least one element $v \in \{v|de_v(g) = de_{e_i}(f)\}$ such that $v = e_iA$ holds. Let $i = 1, 2, \cdots , n$, we get another n equations. These n equations should be consistent to the n equations obtained in step 2, else the two functions are not equivalent.*

(4) *Calculate the invariant of the derivative function of $g(x)$ at unit vector e_i, and denote it by $d_{e_i}(g)$. By proposition 3.7, there exists at least one element $v \in \{v|d_v(f) = d_{e_i}(g)\}$ such that $v = e_iA$ holds. Let $i = 1, 2, \cdots, n$, we get another n equations. These n equations should be consistent to the n equations obtained in step 2 and 3, else the two functions are not equivalent.*

(5) *Denote by $g_{e_i}(x)$ the local connection function of $g(x)$ at unit vector e_i, and denote its invariant by $lc_{e_i}(g)$. By proposition 3.12, there exists at least one element $v \in \{v|lc_v(f) = lc_{e_i}(g)\}$ such that $v = e_iA + b$ holds. Let $i = 1, 2, \cdots, n$, we get another n equations.*

(6) *Denote by $s_{(f)}(e_i)$ the Walsh spectrum of $f(x)$ at unit vector e_i. By corollary 3.4, there exists at least one element $v \in \{v|abs(s_{(g)}(v)) = s_{(f)}(e_i)\}$ such that $v = e_iA^T + l$. Let $i = 1, 2, \cdots, n$, we get n equations.*

(7) *By step $2 \sim 4$, we get matrix A. By step 5, we can obtain b. By step 6, we can get l. With all these parameters(usually there are many choice for some parameters), we can verify them by checking if the equation $g(x) = f(xA + b) + lx$ holds.*

4.1. **Analysis of the Algorithm**

Walsh transform, autocorrelation function, derivation, decomposition and modification of truth table are the basic transforms to Boolean functions. Walsh transform and autocorrelation functions can be done by fast Hadamard transform. Derivation transform lowers the degree of the two functions, and decomposition transform lowers the number of variables. Thus these two transforms lower the complexity of our problem. Modification of truth table gives us more equations with same affine equivalence and thus it is more possible to obtain the affine equivalence. By step 5 and 6, it is unnecessary to enumerate parameters b, l. By above analysis, we say our algorithm is more efficient. However it is not easy to analyze the computation complexity.

By step 3 and 4, it is easy to address the bent functions case, and by step 2,3 and 4, we can deal with functions with uneven truth table. Therefore we say our algorithm is more general.

5. Classification of Reed-Muller Code

Invariant is a good tool to classify set. If we know N, the number of equivalent classes under some equivalent relationship, and an invariant just takes N different values, then the set is already classified.

5.1. Classification of $R(4,6)/R(1,6)$

The number of orbits of $R(4,6)/R(1,6)$ under the action of $AGL(6,2)$ is 2499 by Hou's work [11]. The classification of $R(4,6)/R(1,6)$ can be done as follows:

1. It is easy to get the four orbits of $R(2,6)/R(1,6)$. By hou's work [10], their complementary functions are the four orbits of $R(4,6)/R(3,6)$, denoted by $f_i + R(3,6), i = 0,1,2,3$, where

(1) $f_0(x) = 0$,
(2) $f_1(x) = x_3x_4x_5x_6$,
(3) $f_2(x) = x_1x_2x_5x_6 + x_3x_4x_5x_6$,
(4) $f_3(x) = x_1x_2x_3x_4 + x_1x_2x_5x_6 + x_3x_4x_5x_6$.

2. By proposition 3.7, classify the four cosets $f_i + R(3,6), i = 0,\cdots,3$ into 6,10,12,6 cosets of form $g_j + R(2,6)$, $2 < ldeg(g_j(x))$, $deg(g_j(x)) \leq 4$ respectively. The invariant of $R(3,6)/R(1,6)$ used in proposition 3.7 is the distribution of absolute Walsh spectra. The basic time complexity of this step is $O(4 \times 2^{20})$.

3. By proposition 3.9 and 3.12, classify the 34 cosets $g_i + R(2,6), i = 0,1,\cdots,33$ into 2499 cosets of form $h_i(x) + R(1,6)$, $1 < ldeg(h_i(x))$, $deg(h_i(x)) \leq 4, i = 0,1,\cdots,2498$. The invariant of $R(4,5)/R(1,5)$ used in proposition 3.9 is the distribution of absolute Walsh spectra and absolute autocorrelation function. The invariant of R(6,6)/R(1,6) used in proposition 3.12 is the distribution of absolute Walsh spectra and absolute autocorrelation function. For any combination of invariants given in this paper except the invariant in proposition 3.12, we can't get 2499 orbits. The basic complexity is $O(34 \times 2^{15})$.

5.2. Classification of $R(3,7)/R(1,7)$

The number of orbits of $R(3,7)/R(1,7)$ under the action of $AGL(7,2)$ is 179 by Hou's work [11]. All these 179 orbits can be obtained as follows:

1. By [10], we can get 12 representatives of $R(3,7)/R(2,7)$: $f_i(x) + R(2,7), i = 0,1,\cdots,11$.

2. By proposition 3.9 the coset $f_i(x) + R(2,7), i = 0, 1, \cdots, 11$ can be classifed into 4,8,19,10,20,6,7,29,12,39,10,15 cosets of form $g_i(x) + R(1,7)$ respectively. these are all 179 representatives. The invariant of $R(3,6)/R(1,6)$ used in proposition 3.9 is the distribution of absolute Walsh spectra and absolute autocorrelation function.

By the above two examples, it is very efficient to classify Reed-Muller code for some parameters by invariant theory.

6. Conclusion

Based on some basic transforms, we give an algorithm which can be used to judge if two functions are equivalent and to get the equivalent relationship if they are equivalent in general case. This result also can be used for IP(isomorphism of polynomials) problem with one secret over F_2; Second, $R(4,6)/R(1,6)$ and $R(3,7)/R(1,7)$ are classified efficiently by invariant theory. The direct application of this classification is the semi-enumeration of 8-variable bent functions [15].

Except transforms in this paper, finding other transforms is useful.

References

[1] M.A. Harrison, Counting Theorems and Their Applications to Classifications of Switching Functions, in A. Mukhopadhyay ed. Recent Developments in Switching Theory, Acdemic Press, New York, London,1971, pp. 85-120.

[2] M.A. Harrison, On the Classification of Boolean Functions by the General Linear and Affine Groups, J. Soc. Indust. Appl. Math. 12, 1964, 285-299.

[3] Berlekamp.E.R, Welch.L.R, Weight Distributions of the Cosets of the (32,6) Reed-Muller Code, IEEE Trans. Inform. Theory V.18, 1972, 203-207.

[4] A. Biryukov, C.D. Canniere, A. Braeken, B. Preneel, Toolbox for Cryptanalysis: Linear and Affine Equivalence Algorithms, Eurocrypt'03, LNCS 2656, 33-50.

[5] W. Geiselmann,W. Meier,R. Steinwandt. An Attack on the Isomorphisms of Polynomials Problem with One Secret. International Journal of Information Security, 2003, 2(1): 59-64.

[6] A. Braeken, Y. Borissov,S. Nikova, B. Preneel, Classification of Boolean Functions of 6 Variables or Less with Respect to Cryptographic Properties, http://eprint.iacr.org, 2004/248.

[7] J. Fuller,W. Millan. Linear Redundancy in S-Box. In: Fast Software Encryption, LNCS 2887, Springer-Verlag, 2003, 74-86.

[8] Q. Meng,H. Zhang. The Analysis of Linear Equivalence of Boolean Functions and Its Applications, Chinese Journal of Computers, 2004, 11. 1528-1532. (in Chinese)

[9] J.A. Maiorana, A Classification of the Cosets of the Reed-Muller Code $R(1,6)$, Math. Comp. 57, 403-414, 1991.

[10] X. Hou, $GL(m,2)$ Acting on $R(r,m)/R(r-1,m)$, Discrete Mathematics, 149(1996),pp. 99-122.

[11] X. Hou, $AGL(m,2)$ Acting on $R(r,m)/R(s,m)$, Journal of Algebra, 171(1995)921-938.

[12] B. Preneel, Analysis and Design of Cryptographic Hash Functions, Ph.D Thesis, KU Leuven(Belgium),February 1993.

[13] E. Brier, Philippe Langevin, Classification of Boolean Cubic Forms in Nine Variables, 2003 IEEE Information Theory Workshop, 179-182.

[14] W. Millan,A. Clark,E. Dawson. Smart Hill Climbing Finds Better Boolean Functions. in Proceeding of the Workshop on Selected Areas in Cryptology 1997, Ottawa,Canada, 1997, 50–63.

[15] Q. Meng, H. Zhang, J. Cui, et al. Semi-Enumeration of 8-Variable Bent Functions. http://eprint.iacr.org. 2005/100.

This work is supported by the National Natural Science Foundation of China (60373087, 90104005, 60473023) and the Ph.D Programs Foundation of Ministry of Education of China (20020486046).

ON THE DECOMPOSITION OF BOOLEAN FUNCTIONS

Gérard H. E. Duchamp[1], Hatem Hadj Kacem[2] and Éric Laugerotte[2]

Abstract. The minimization of a weighted automaton given by its linear representation (λ, μ, γ) taking its letters in an alphabet A and its multiplicities in a (commutative or not) field k, due to Schützenberger, provides the construction of a suffix set P such that the orbit $(\mu(p)\gamma)_{p\in P}$ is a basis of the k-space $\mu(k\langle A\rangle)\gamma$. This allows to study algorithmically the $\mathfrak{S}_n$-module $\mathbb{Z}/2\mathbb{Z}[\mathfrak{S}_n].f$ where $\mathfrak{S}_n$ is the symmetric group which acts on the unknowns $x_1, \ldots, x_n$ by change of variables, and $f(x_1, \ldots, x_n)$ is a boolean function. In this work, we present an algorithm which computes the possible decompositions of f with respect to this action. In case the function f is indecomposable the algorithm gives a proof of indecomposability.

1. Introduction

This contribution is intended to tackle the multifaceted problem of decomposing the Boolean Functions (BF in the sequel). By boolean function we here mean any function $\{0,1\}^n \mapsto \{0,1\}$ which, in the language of Computer Science, is just any function taking a n-bits word as argument and returning a boolean value. These functions are efficiently represented by a BDD (a *Binary Decision Diagram*). This representation can be traced back as far as in the late fifties [14] and was exploited extensively (for the first

[1] LIPN, Université de Paris-Nord, 99, avenue Jean-Baptiste Clément, 93430 Villetaneuse, France. email: gerard.duchamp@lipn.univ-paris13.fr

[2] LIFAR, Université de Rouen, place Émile Blondel, 76821 Mont-Saint-Aignan, France. email: eric.laugerotte@univ-rouen.fr

developments see [1, 2]). The great merit of this coding is that it is extremely concise and also compatible with boolean automata theory [12] to such a point that a measure of hardness has been derived from the consideration of a minimal automaton associated to the BDD of a boolean function [5].

As a BDD is variable-order dependant, we would like here to study the orbit of a boolean function under the action of the (algebra of the) symmetric group on the variables. The set of boolean functions of n-variables is naturally a $\mathbb{Z}_2$ $(= \mathbb{Z}/2\mathbb{Z})$ vector space. Thus, the action of the symmetric group given by permutation of variables can be at once extended by linearity to the algebra $\mathbb{Z}_2[\mathfrak{S}_n] = \mathfrak{A}_n$ and, by Krull-Schmitt's theorem, we get that the orbit of f can be split (uniquely, up to isomorphism) as a direct sum of $\mathfrak{A}_n$ indecomposable submodules. The interest of such a splitting is that the components are monogenous (i.e. generated by a single element). The decomposition reads $\mathfrak{A}_n.f = \oplus \mathfrak{A}_n f_i$ and this yields a decomposition of f using $\mathfrak{End}_{\mathfrak{A}_n}(\mathfrak{A}_n.f)$ idempotents

$$f = \pi_1.f_1 + \pi_2.f_2 \cdots \pi_k.f_k \tag{1}$$

Surprizingly, a suitable adaptation of Schützenberger's algorithm [16] for the minimization of automata with multiplicities (here with coefficients in $\mathbb{Z}_2$) makes all this computable. We use here half of the minimization process, keeping a note of the relators appearing and then getting a minimal presentation of the module $\mathfrak{A}_n.f$. This process is reminiscent of the theory of non-commutative Gröbner bases [11], but here we need more. We need also to compute idempotents in the transfer algebra $\mathfrak{End}_{\mathfrak{A}_n}(\mathfrak{A}_n.f)$, which can be done using the reduced basis of the module $\mathfrak{A}_n.f$ previously computed. All the process has been implemented in MuPAD.

The structure of the contribution is the following. In Section 2, we present the main aspects of weighted automaton minimization. In Section 3, we deal with the splitting of modules. After, in Section 4, we present the algorithmic of the decomposition of boolean functions. At the end, in Section 5, an example is given with the numbers of decomposable functions for the first values of n.

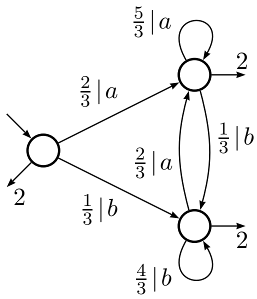


FIGURE 1. A $\mathbb{Q}$-automaton.

2. Minimization of weighted automata

Let us give here a short review of the minimization algorithm from the theory of automata with multiplicities (see also [4, 16] for fields and domains and [10] for a detailed algorithm and an extension to skew fields). An *automaton with multiplicities* $\mathcal{A}$ is a structure equivalent to a triplet (λ, μ, γ) called *linear representation* which is defined by:

- an alphabet (of commands, say) A
- a (finite) set of states Q
- a (semi)ring k of scalars
- an input vector $\lambda \in k^{1\times Q}$
- an output vector $\gamma \in k^{Q\times 1}$
- a mapping $\mu : A \to k^{Q\times Q}$

These data are usually represented as a valued graph (see Figure 1). The mapping μ is at once extended to a morphism from (A^*, conc) to $(k^{Q\times Q}, \cdot)$ where conc stands for the binary operator of concatenation of words and $\cdot$ for the usual matrix multiplication. The number of states of the weighted automaton $\mathcal{A}$ is its *dimension* noted dimension$(\mathcal{A})$. Therefore $\mathcal{A}$ is a finite state machine taking words and providing coefficients (called also costs or weights) which are provided by $\lambda\mu(w)\gamma$ for a word $w \in A^*$. The function $A^* \to k$, given by $w \mapsto \lambda\mu(w)\gamma$, can more conveniently be written as a noncommutative series

$$\text{behaviour}(\mathcal{A}) = \sum_{w\in A^*} \lambda\mu(w)\gamma w \qquad (2)$$

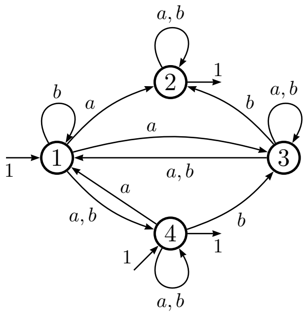


FIGURE 2. A $\mathbb{Z}_2$-automaton

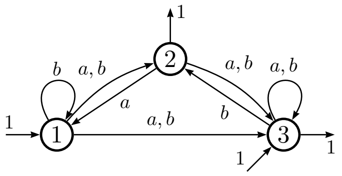


FIGURE 3. A $\mathbb{Z}_2$-automaton

which is called the *behaviour of* $\mathcal{A}$. Such series are just functions $A^* \mapsto k$ called *rational* [4, 8, 15, 17].

The whole set of functions k^{A^*} (noncommutative series) is often denoted $k\langle\langle A\rangle\rangle$ and a function $S \in k\langle\langle A\rangle\rangle$, written in the style of (2) reads

$$S = \sum_{w \in A^*} \langle S|w\rangle w \tag{3}$$

so that $S(w)$ (i.e. the coefficient of w in S) will be denoted as the scalar product $\langle S|w\rangle$. The *behaviour* of $\mathcal{A}$ thus determines the weight of w for the automaton $\mathcal{A}$.

The aim of minimization is to construct an automaton

$$\mathcal{A}_{\min} = (\lambda_{\min}, \mu_{\min}, \gamma_{\min})$$

with the same behaviour and of smallest dimension.

From now on, we set once for all $\mathbb{Z}_2 = \mathbb{Z}/\mathbb{Z}_2$.

The $\mathbb{Z}_2$-automaton given in Figure 2 is minimized in Figure 3.

Minimization is obtained by a left and a right *reduction.* In fact, let $\circ$ be the left action defined for all formal series $S \in k\langle\langle A\rangle\rangle$ and all word $w \in A^*$ by $w \circ S = \sum_{x \in A^*} \langle S|xw\rangle x$. If S is rational, there exists a finitely generated submodule of $k\langle\langle A\rangle\rangle$ stable for $\circ$ which contains the formal series behaviour($\mathcal{A}$) (this is even a criterium of rationality, see [4,9]). The generators S_i ($i = 1, \dots,$ dimension($\mathcal{A}$)) may be explicitly given by

$$S_i = \sum_{w\in A^*} (\lambda_i \mu_i(w)\gamma_i)w$$

but in general it is not a family of smallest rank.
Finding algorithmically such a minimal family goes as follows [10]. Call *suffix set* a subset P of the free monoid A^* such that, if a word w belongs to P then every suffix of w belongs to P.
Left reduction of $\mathcal{A}$ allows to construct a suffix set P such that $(\mu(p)\gamma)_{p\in P}$ is a basis of the space of columns $\mu(k\langle A\rangle)\gamma$. The family $(p \circ \text{behaviour}(\mathcal{A}))_{p\in P}$ generates a stable submodule of $k\langle\langle A\rangle\rangle$ which contains behaviour($\mathcal{A}$) and whose the dimension is smaller or equal to dimension($\mathcal{A}$). Indeed, it is the smallest possible among the stable submodules containing behaviour($\mathcal{A}$). More precisely, let $p \in P$ and $a \in A$,

$$a \circ (p \circ S) = \begin{cases} ap \circ S & \text{if } ap \in P, \\ \sum_{q\in P} \alpha^a_{pq}\ q \circ S & \text{if } ap \notin P. \end{cases}$$

To each formal series $p \circ S$ is associated a state in the reduced automaton. The weight of a transition $p \to q$ is the scalar α^a_{pq}, the transition label being a. After left reduction, right reduction is applied and returns the minimized automaton $\mathcal{A}_{\min}$ because $\dim(\lambda_{\min}\mu_{\min}(k\langle A\rangle)) = \dim(\mu_{\min}(k\langle A\rangle)\gamma_{\min})$.

3. Splitting modules

In what follows, we consider the algebra $\mathfrak{A}_n = \mathbb{Z}_2[S_n]$ (we omit the subscript as it is fixed once for all) of the symmetric group S_n over $\mathbb{Z}_2$ [13]. It is generated by the simple transpositions $\sigma_1, \dots, \sigma_{n-1}$ (σ_i is the transposition of i and $i+1$) and therefore can be presented by generators $(s_i)_{1\le i\le n-1}$ and the relations (the symbol s_i standing for σ_i)

$$\begin{cases} {s_i}^2 = 1, & \\ s_i s_j = s_j s_i & \text{if } |i-j| > 1, \\ s_i s_{i+1} s_i = s_{i+1} s_i s_{i+1}. & \end{cases}$$

(called Moore-Coxeter relations [7, 13]). The algebra $\mathfrak{A}_n$ acts on the left on $\mathbb{Z}_2\langle x_1, \ldots, x_n\rangle$ by change of variables which are morphisms $s_i : A^* \mapsto A^*$ defined on the letters by

$$\begin{cases} s_i x_i = x_{i+1}, \\ s_i x_{i+1} = x_i, \\ s_i x_j = x_j \qquad \text{if } j \neq i, i+1. \end{cases}$$

Let $S = \{s_1, \ldots, s_{n-1}\}$ be the set of symbols of these (simple) transpositions. The morphism $\mathbb{Z}_2\langle S\rangle \to \mathfrak{A}_n$ is onto and then the notions of submodule and decomposition are the same for the action of $\mathfrak{A}_n$ and the action of $\mathbb{Z}_2\langle S\rangle$. For this reason, we will denote similarly (and with no risk of confusion) the two actions. Let $\mathcal{F}_n$ be the left $\mathfrak{A}_n$-module of boolean functions with n variables (it is a finite dimensional $\mathbb{Z}_2$-vector space). We consider the submodule $\mathfrak{A}_n.f$ where $f \in \mathcal{F}_n$ is a single generator. Krull-Schmidt's theorem [6] implies that there exists (unique up to isomorphisms) a splitting of the module $\mathfrak{A}_n.f$ into a direct sum $\mathfrak{A}_n.f = M_1 \oplus \cdots \oplus M_l$ of indecomposable $\mathfrak{A}_n$-submodules M_i. The aim of the algorithm below is to compute a splitting of $\mathfrak{A}_n.f$ by the knowledge of a complete family of orthogonal projectors $\pi_i \in \mathfrak{End}_{\mathfrak{A}_n}(\mathfrak{A}_n.f)$ $(i = 1, \ldots, l)$ i.e. which satisfy:

$$\begin{cases} \pi_i \circ \pi_i = \pi_i, \\ \pi_i \circ \pi_j = 0 \quad \text{if } i \neq j, \\ \pi_1 \oplus \cdots \oplus \pi_l = 1_{\mathfrak{End}_{\mathfrak{A}_n}(\mathfrak{A}_n.f)}. \end{cases}$$

Therefore, the module $\mathfrak{A}_n.f$ is the direct sum of submodules given by $\pi_i(\mathfrak{A}_n.f)$ which are generated by a single element. If π_i is the projector which carries out $\mathfrak{A}_n.f$ to M_i $(M_i = \pi_i(\mathfrak{A}_n.f))$ then π_i must be $\mathfrak{A}_n$-linear.
Let then $\varphi \in \mathfrak{End}_k(\mathfrak{A}_n.f)$. Whether $\varphi \in \mathfrak{End}_{\mathfrak{A}_n}(\mathfrak{A}_n.f)$ is algorithmically decidable thanks to the fact that the ideal of annihilators of $\mathrm{ann}(f)$ is finitely generated. We explain now how this can be done.

One can construct a suffix set (see below or [10]) $P \subset S^*$ such that $P.f$ is a basis of $\mathfrak{A}_n.f$. Let $E := \{(\sigma_i, \sigma) \in S \times P \mid \sigma_i\sigma \notin P\}$. For $(\sigma_i, \sigma) \in E$, one has:

$$\sigma_i\sigma.f = \sum_{\sigma' \in P} \alpha_{\sigma_i\sigma, \sigma'}\sigma'.f \tag{4}$$

and the differences $R := \left\{\sigma_i\sigma - \sum_{\sigma'\in P}\alpha_{\sigma_i\sigma,\sigma'}\sigma'\right\}_{(\sigma_i,\sigma)\in E}$ are a complete set of generators of $\mathrm{ann}(f)$. The construction of idempotents will rely on the following lemma:

Lemma 3.1. *Let $\varphi \in \mathfrak{End}_k(\mathfrak{A}_n.f)$ and set $f_\varphi = \varphi(f)$. Then the linear transformation φ belongs to $\mathfrak{End}_{\mathfrak{A}_n}(\mathfrak{A}_n.f)$ iff:*
i) for all $\sigma \in P$, one has $\varphi(\sigma.f) = \sigma.\varphi(f)$
ii) for all $(\sigma_i, \sigma) \in E$, the difference $\left(\sigma_i\sigma - \sum_{\sigma'\in P}\alpha_{\sigma_i\sigma,\sigma'}\right)$ annihilates f_φ.

Thus, it suffices to compute a basis of $\mathfrak{A}_n.f$, keeping track of the relators appearing, to obtain a test which allows to select the idempotents of $\mathfrak{End}_{\mathfrak{A}_n}(\mathfrak{A}_n.f)$ among the projectors of $\mathfrak{End}_k(\mathfrak{A}_n.f)$.

4. Computation of endomorphisms and projectors

We can transfer the half minimization process to $\mathfrak{A}_n.f$ and also take care of keeping trace of the relators appearing. The following algorithm allows us to find a suffix set of S^* and the corresponding set of relators:

```
algorithm suffix
input     the set $S$
          a boolean function $f \in \mathcal{F}_n$
output    a suffix set $P \subset S^*$
          a set of relators $R$
$(P, Y, R) := (\emptyset, \{\varepsilon\}, \emptyset)$
while $Y \neq \emptyset$
do take $y \in Y$
   if $ym \notin \mathrm{span}(mp : p \in P)$
   then $(P, Y) := (P \cup \{y\}, (Y - \{y\}) \cup yS)$
   else there exits a relation $ym = \sum_{p\in P}\alpha_p mp$
        $(P, Y, R) := (P, (Y - \{y\}), R \cup \{y - \sum_{p\in P}\alpha_p p\})$
   end_if
end_while
return$(P, R)$
end
```

The set P is a suffix set and the algorithm terminates. In fact, we show that the set P is suffix at each step of the algorithm. This is clear from the beginning when $P = \{\varepsilon\}$. Now, if $y \in Y \subseteq S^*$ is accepted it must have been so of every suffix of it before. Let $|\sigma|$ denote, as usual, the length of a word $\sigma \in S^*$. As the space $\mathfrak{A}_n.f$ has a finite dimension, it exists a non-negative integer l such that:

$$\text{span}(\sigma.f : \sigma \in S^*) = \text{span}(\sigma.f : \sigma \in S^* \text{ and } |\sigma| < l).$$

One has $P \subseteq \{\sigma \in S^* : |\sigma| < l\}$ and $Y \subseteq \{\sigma \in S^* : |\sigma| < l+1\}$. Then the set Y becomes empty during the algorithm and then the algorithm terminates.

Lemma 4.1. *The set $P.f = \{\sigma.f : \sigma \in P\}$ is a basis of the space $\mathfrak{A}_n.f$.*

Proof. Let $C = SP\backslash P$ be the complete suffix code associated to P [3]. One has the decomposition $S^* = P \sqcup S^*CP$ of the free monoid. Let $\sigma_i\sigma \in C$. Then there exists a relator $\sigma_i\sigma - \sum_{\sigma' \in P} \alpha_{\sigma'}\sigma' \in R$. Now let $\sigma_i \in S$ and $\sigma_i\sigma \in CS^*S$. One has by induction:

$$\begin{aligned}\sigma_i\sigma f &= \sum_{\sigma' \in P} \alpha_{\sigma'}\sigma_i\sigma' f \\ &= \sum_{\sigma_i\sigma' \in P} \alpha_{\sigma'}\sigma_i\sigma' f + \sum_{\sigma_i\sigma \notin P}\sum_{\sigma'' \in P} \alpha_{\sigma'}\beta^{\sigma''}_{\sigma_i\sigma'}\sigma'' f.\end{aligned}$$

And then $\sigma_i\sigma f \in \text{span}(\sigma f : \sigma \in P)$ which ends the proof. □

By Lemma 4.1, Algorithm *suffix* computes a complete description of the space $\mathfrak{A}_n f$. We can observe that the set R of relators depends on the choice of words $y \in Y$. Let $f = x_1x_2 + x_1 \in \mathcal{F}_3$. The set of relators are different if the words are choosen with the graded or with the usual lexicographic order. In fact, the element $\sigma_2\sigma_2 + \varepsilon \in S^*$ is a relator with the use of the second order but not with the first. See Figure 4 where a full transition means an action giving an new element of the basis, a dotted transition giving a relator.

Lemma 4.2. *The ideal generated by the set of relators R is then* $\text{ann}(f)$.

In [11], the tools for the proof of Lemma 4.2 are presented. Therefore we associate at $\varphi \in \mathfrak{End}(\mathfrak{A}.f)$ a unique element $f_\varphi = (\sum_{\sigma \in P} \alpha_{\sigma,\varepsilon}\sigma f)$ [6]. By Lemma 4.1, it is easy to determine algorithmically the endomorphism φ in the basis $(\sigma f)_{\sigma \in P}$ of $\mathfrak{A}f$. In fact,

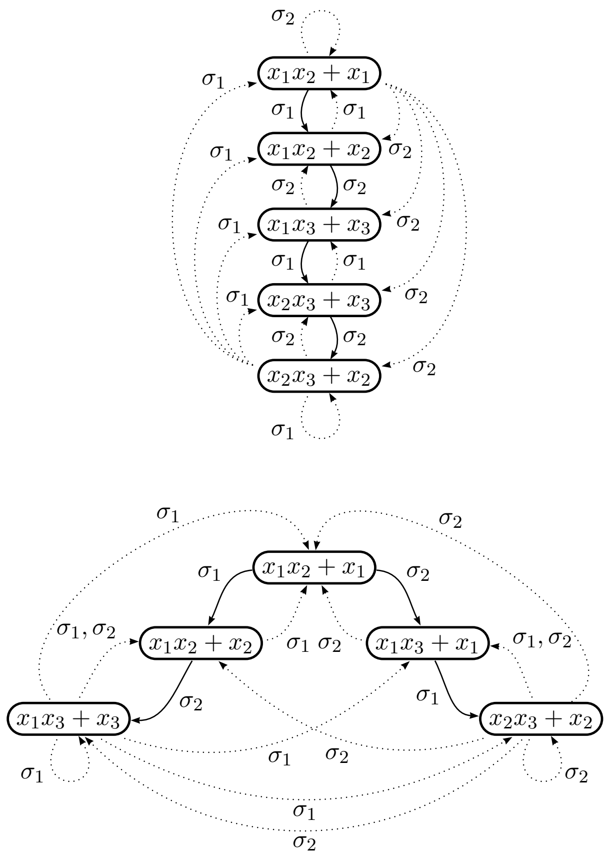


FIGURE 4. Suffix sets and relators for $f = x_1x_2 + x_1$

Algorithm *suffix* allows to compute the suffix set P and the relator set R. By linearity, for any element $\sigma' \in P$, the endomorphism φ depends only to the unknowns $\alpha_{\sigma,\varepsilon}$. In fact, one has:

$$\varphi(\sigma' f) = \sum_{\sigma \in P} \alpha_{\sigma,\varepsilon}\sigma'\sigma f = \sum_{\sigma'\sigma \in P} \alpha_{\sigma,\varepsilon}\sigma'\sigma f + \sum_{\sigma'\sigma \notin P} \alpha_{\sigma,\varepsilon} \sum_{\sigma'' \in P} \beta^{\sigma''}_{\sigma'\sigma}\sigma''.$$

The scalars $\beta^{\sigma''}_{\sigma'\sigma}$ are given by the relators. The order of computation of vectors $\varphi(\sigma f)$ is given by the entry of σ in the set P. If $\sigma = \sigma_i\sigma'$ then $\varphi(\sigma' f)$ will be known. The morphism φ is computed by the following algorithm:

algorithm *cons_φ*

```
input   the set of simple transitions $S$
        the suffix set $P$
        the set of relators $R$
output  the morphism $\varphi$
$Y := S$
$\varphi(\varepsilon f) := \sum_{\sigma\in P} \alpha_{\sigma,\varepsilon}\sigma f$
while $Y \neq \emptyset$
do take $\sigma_i\sigma \in Y$
    if $\sigma_i\sigma \notin P$
    then $Y := Y - \{\sigma_i\sigma\}$
    else $\varphi(\sigma_i\sigma f) := \sigma_i\varphi(\sigma f)$
         $Y := (Y \cup S\sigma_i\sigma) - \{\sigma_i\sigma\}$
    end_if
end_while
return($\varphi$)
end
```

In order to find projectors, we must study the relation of idempotence $\varphi^2 = \varphi$. Moreover we must verify that $\varphi(\sigma f) = \sigma\varphi(f)$ for all element $\sigma \in P$ and $r\varphi(f) = 0$ for all relator $r \in R$ as $\varphi \in \mathfrak{End}_{\mathfrak{A}_n}(\mathfrak{A}_n.f)$. Therefore we obtain a system of $n \times |P|^2 + |P|$ equations in the unknowns $\alpha_{\sigma,\varepsilon}$ for all $\sigma \in P$. Each non-trivial solution φ gives a decomposition $\mathfrak{A}_n.f = \mathfrak{A}_n.f_\varphi \oplus \mathfrak{A}_n.f_{1-\varphi}$. In this case, we restart the algorithm on $\mathfrak{A}_n.f_\varphi$ and $\mathfrak{A}_n.f_{1-\varphi}$. We get by repetition a direct sum of indecomposable submodules. Otherwise, if no non-trivial solutions exists, we deduce that the module $\mathfrak{A}_n.f$ can not be written in a direct sum of submodules non-zero submodules.

Theorem 4.3. *Let $f \in \mathcal{F}_n$. A finite repetition of Algorithm suffix and cons_φ decides if there exits a decomposition of $\mathfrak{A}_n.f$ in direct sum of indecomposable submodules. If the algorithm finds no non-trivial decomposition at the first step, then the module $\mathfrak{A}_n.f$ is indecomposable.*

The preceding process can be applied mutatis mutandis with $\mathfrak{A}$ any finitely generated associative algebra with unit over a field k acting on a finite dimensional (as a k-vector space) $\mathfrak{A}$-module.

5. Example of splitting

We consider now the boolean function $f = x_1x_2 + x_1 \in \mathcal{F}_3$. A basis of the module $\mathfrak{A}_3.f$ is presented in Figure 4 when the

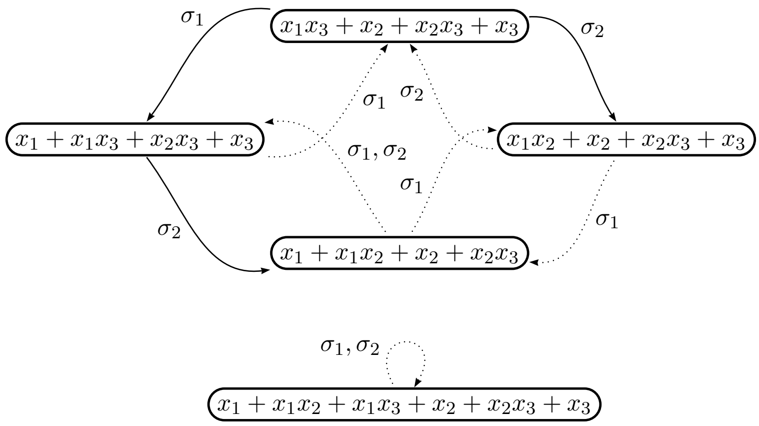

σ_1
$x_1x_3 + x_2 + x_2x_3 + x_3$
σ_2
σ_1
σ_2
$x_1 + x_1x_3 + x_2x_3 + x_3$
σ_1, σ_2
$x_1x_2 + x_2 + x_2x_3 + x_3$
σ_1
σ_2
σ_1
$x_1 + x_1x_2 + x_2 + x_2x_3$
σ_1, σ_2
$x_1 + x_1x_2 + x_1x_3 + x_2 + x_2x_3 + x_3$

FIGURE 5. Indecomposable modules

graded lexicographic order is choosen for the computation of the suffix set P. Let $\varphi \in \mathfrak{End}_k(\mathfrak{A}_3.f)$ and $f_\varphi \in \mathfrak{A}_3.f$ such that $f_\varphi = \alpha_\epsilon \epsilon.f + \alpha_{\sigma_1}\sigma_1.f + \alpha_{\sigma_2}\sigma_2.f + \alpha_{\sigma_2\sigma_1}\sigma_2\sigma_1.f + \alpha_{\sigma_1\sigma_2}\sigma_1\sigma_2.f$. The matrix corresponding to φ is:

$$\begin{pmatrix} \alpha_\varepsilon & \alpha_{\sigma_1}+\alpha_{\sigma_1\sigma_2} & \alpha_{\sigma_2}+\alpha_{\sigma_2\sigma_1} & \alpha_{\sigma_2}+\alpha_{\sigma_2\sigma_1} & \alpha_{\sigma_1}+\alpha_{\sigma_1\sigma_2} \\ \alpha_{\sigma_1} & \alpha_\varepsilon+\alpha_{\sigma_1\sigma_2} & \alpha_{\sigma_1\sigma_2}+\alpha_{\sigma_2\sigma_1} & \alpha_{\sigma_2} & \alpha_{\sigma_1}+\alpha_{\sigma_2} \\ \alpha_{\sigma_2} & \alpha_{\sigma_1\sigma_2}+\alpha_{\sigma_2\sigma_1} & \alpha_\varepsilon+\alpha_{\sigma_2\sigma_1} & \alpha_{\sigma_1}+\alpha_{\sigma_2} & \alpha_{\sigma_1} \\ \alpha_{\sigma_1\sigma_2} & \alpha_{\sigma_1\sigma_2} & \alpha_{\sigma_1}+\alpha_{\sigma_2\sigma_1} & \alpha_\varepsilon+\alpha_{\sigma_2} & \alpha_{\sigma_1}+\alpha_{\sigma_2\sigma_1} \\ \alpha_{\sigma_2\sigma_1} & \alpha_{\sigma_2}+\alpha_{\sigma_1\sigma_2} & \alpha_{\sigma_2\sigma_1} & \alpha_{\sigma_2}+\alpha_{\sigma_1\sigma_2} & \alpha_\varepsilon+\alpha_{\sigma_2} \end{pmatrix}$$

Non-trivial solutions of the system given by $\varphi^2 = \varphi$ and $\sigma\varphi(f) = \varphi(\sigma.f)$ for all $\sigma \in P$ are $f_\varphi = \sigma_2\sigma_1.f + \sigma_1\sigma_2.f$ and $f_{1+\varphi} = \varepsilon.f + \sigma_2\sigma_1.f + \sigma_1\sigma_2.f$. Or else, $f_\varphi = x_1x_3 + x_2 + x_2x_3 + x_3$ and $f_{1+\varphi} = x_1 + x_1x_2 + x_1x_3 + x_2 + x_2x_3 + x_3$. In fact, one has $\mathfrak{A}_3.f = \mathfrak{A}_3.f_\varphi + \mathfrak{A}_3.f_{1+\varphi}$, and the indecomposable modules $\mathfrak{A}_3.f_\varphi$ and $\mathfrak{A}_3.f_{1+\varphi}$ are expressed by Figure 5.

First results of experiments are presented in the following table:

Nb of unknowns	1	2	3	4
Nb of functions	1	16	256	65536
Nb of dec. functions	0	0	82	683
% of dec. functions	0	0	32.03	1.04

6. Concluding remarks

The linear representation of the action of the symmetric group by change of variables of a boolean function has been studied with respect to indecomposability and using $\mathbb{Z}_2$ coefficients. We have got a presentation of the module generated by a boolean function by means of an algorithm designed by Schützenberger for the minimization of automata with multiplicities and a suited recording of the relators appearing during the computation. The whole process has been implemented in MuPAD.

References

[1] S. B. Akers, *Binary decision diagrams*, IEEE Transactions on Computers, **27**, 1978.

[2] R. E. Bryant, *Graph-based algorithms for boolean function manipulation*, IEEE Transactions on Computers, **35**, 1986.

[3] J. Berstel, D. Perrin, *Theory of codes*, Academic Press, 1985.

[4] J. Berstel, C. Reutenauer, *Rational series and their languages*, Springer, 1988.

[5] J.-M. Champarnaud, J.-F. Michon, *Automata and binary decision diagrams*, Workshop on Implementing Automata, 178-182, 1998.

[6] C.W. Curtis, I. Reiner, *Methods of representation theory*, Wiley Interscience, 1990.

[7] G. Duchamp, D. Krob, A. Lascoux, B. Leclerc, T. Scharf, J.Y. Thibon, *Euler-Poincaré characteristic and polynomial representations of Iwahori-Hecke algebras*, Pub. of the RIMS, **31**, 1995.

[8] S. Eilenberg, *Automata, languages and machines*, vol A, Academic Press, 1974.

[9] M. Fliess, *Matrices de Hankel*, Journal of Mathematics Pures and Applied, **53**, 197-222, 1974.

[10] M. Flouret, É. Laugerotte, *Noncommutative minimization algorithms*, Information Processing Letters, **64**, 123-126, 1997.

[11] G. Melançon, *Réécritures dans l'algèbre de Lie libre, le groupe libre et l'algèbre associative libre*, Monographies du LACIM, 1991.

[12] S. Fortune, J. E. Hopcroft, E. M. Schmidt, *The complexity of equivalence and containment for free single variable program schemes*, LNCS, Proceedings of the Fifth Colloquium on Automata, Languages and Programming, **62**, 227-240, 1978.

[13] J. E. Humphreys, *Reflection groups and Coxeter groups*, Cambridge University Press, 1990.

[14] C. Y. Lee, *Representation of switching circuits by binary-decision programs*, Bell Systems Technical Journal, 38:985999, 1959.

[15] J. Sakarovitch, *Éléments de théorie des automates*, Vuibert, 2003.

[16] M. P. Schützenberger, *On the definition of a family of automata*, Inform. and Contr. **4**, 245-270, 1961.

[17] R. P. Stanley, *Enumerative combinatorics*, Cambridge University Press, 1999.

GREY-BOX IMPLEMENTATION OF BLOCK CIPHERS PRESERVING THE CONFIDENTIALITY OF THEIR DESIGN [*]

V. Carlier[1], H. Chabanne[2] and E. Dottax[3]

Abstract. In 1997, Patarin and Goubin introduce new asymmetric cryptosystems based on the difficulty of recovering two systems of multivariate polynomials from their composition. We make a different use of this difficult algorithmic problem to obtain a way of representing block ciphers concealing their design but leaving them executable. We show how to implement our solution giving a compact representation with Binary Decision Diagrams.

1. Introduction

Protection of the design of ciphering algorithms is not a new problem. The situation is a bit paradoxical, as usually confidentiality is addressed by encryption. At one hand, algorithms can be enclosed into a tamper resistant circuit. A famous example is given by the Skipjack algorithm history. Its specifications were once classified and it had to stay into devices such as the Capstone or Clipper chips, known to implement specific protections against reverse engineering [18]. We talk in this case of *black-box* environment, where confidentiality of design relies on physical resistance. At the other hand, the new concept of *white-box* cryptography emerges [6, 7, 14]. Here, the whole source code is supposed to

[*] *This work has been supported by the MEDEA+ programme under the CryptoSoC project.*

[1] SAGEM SA, France. email: `vincent.carlier@sagem.com`
[2] SAGEM SA, France. email: `herve.chabanne@sagem.com`
[3] SAGEM SA, France. email: `emmanuelle.dottax@sagem.com`

be known to the attacker, and the security is provided by logical ways. Note, however, that in these articles, the sole keys used by the algorithm are to be hidden, the design of the algorithm being known. We place ourselves in-between and call our solution *grey-box* cryptography. We propose a solution using well-known tamper response techniques where volatile memories are zeroized whenever an intrusion is detected, and we accept that some information is recovered by an intruder. This hypothesis is confirmed by experiments [20] and seems quite reasonable to assume. We then find ourselves with an instance related to a well-known algorithmic problem, introduced for cryptographic purposes by Goubin and Patarin in the *two rounds schemes* with partial revelation of the polynomials, noted $2\mathbf{R}^-$ schemes [12]. In this, we follow works of Sander *et al.* [15,19] where the Quadratic Residue Hypothesis [11] is used to hide polynomials and subsequently programs, and more recently of Billet and Gilbert [3] who utilize the Isomorphism of Polynomials problem [8, 16, 17] to implement a concealed block cipher with a traceability property.

The remainder of this paper is organized as follows. Section 2 explains the setting of our solution. Section 3 goes further in details giving some concrete examples and explaining the method used. Section 4 concludes.

2. A New Way to Implement a Block Cipher Protecting the Confidentiality of its Design

2.1. $2\mathbf{R}^-$ Schemes

Goubin and Patarin introduce in [12] new asymmetric cryptosystems based on the idea of hiding one or two rounds of small S-box computations with secret functions of degree one or two. The public key is given by multivariate polynomials of small degree. In the following we recall the so-called *two-rounds schemes*, designed to be more secure than one-round schemes.

Let $\mathbf{K}$ be a finite field with $q = p^m$ elements. Plaintexts and ciphertexts are elements of $\mathbf{K}^n$. The secret key consists of three affine bijections $r, s, t : \mathbf{K}^n \to \mathbf{K}^n$, and two applications f, g, each given by n quadratic equations over $\mathbf{K}$. The public key consists of the polynomials $P_1, \ldots, P_n$ of degree 4 in n variables that describe the composed mapping $H = t \circ g \circ s \circ f \circ r$. When all these polynomials are given, the scheme is called a $2\mathbf{R}$ scheme. When only some of them are given, the scheme is called a $2\mathbf{R}^-$ scheme. The public-key side computation is just an application of the mapping H. For the secret-key computations, we need to invert the functions f and g. The authors propose to choose them among the following classes of functions:

- $C^\star$-functions: monomials over an extension of degree n over $\mathbf{K}$;
- triangular functions:

$$(a_1, \dots, a_n) \mapsto (a_1, a_2 + q_1(a_1), \dots, a_n + q_{n-1}(a_1, \dots, a_{n-1}))$$

  where the q_i are quadratic;
- S-boxes functions, which map $(a_1, \dots, a_n) \in \mathbf{K}^n$ to:

$$\begin{aligned}&(S_1(a_1, \dots, a_{n_1}),\ S_2(a_{n_1+1}, \dots, a_{n_1+n_2}), \dots\\ &\dots,\ S_d(a_{n_1+n_2+\dots+n_{d-1}+1}, \dots, a_{n_1+\dots+n_d}))\end{aligned}$$

  where $n = \sum n_i$ and the $S_i : \mathbf{K}^{n_i} \to \mathbf{K}^{n_i}$ are quadratic;
- Combinations of S-boxes with triangular functions;
- $D^{\star\star}$-functions: squaring in an extension of $\mathbf{K}$ of degree n.

These schemes are based on the difficulty of decomposing compositions of multivariate polynomials, *i.e.* given $h = f \circ g$, recover f and g. Note that if we drop t and g in above description, we get the one-round schemes, and they have all been shown to be insecure [12]. The two-rounds schemes have also been shown to be insecure when g lies in the first two classes [12]. The variant that we are interested in is $2\mathbf{R}$ with S-boxes, where both f and g are S-boxes functions.

So far, there exist two different attacks against $2\mathbf{R}$ with S-boxes. In [2], Biham succeeds in cryptanalysing the scheme. Note that this attack is not based on functionnal decomposition. Another attack has been published [10], based on the algebraic structure of the scheme and with the intention of decomposing the composition. However, the attack imposes restrictions on the scheme:

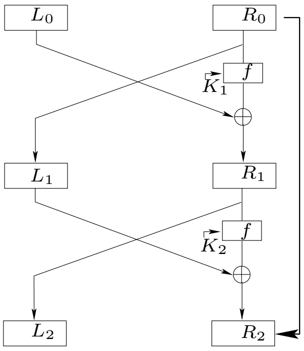
(1) the field $\mathbf{K}$ should have more than 4 elements;
(2) the attack would not work if the S-box functions are not quadratic.

Note as well that the $2\mathbf{R}^-$ schemes, *i.e.* when some of the polynomials describing the composition are kept secret, are not subject to these atttacks.

2.2. Our Idea and a Way to Implement It

Our idea is to use the same problem as in $2\mathbf{R}^-$ schemes for protecting the confidentiality of the design of block ciphers.

A block cipher is usually composed of several rounds, and a round itself is composed of different operations. The description of these operations constitutes the design of the algorithm: they have been chosen by the designer and they are an evidence of its know-how. Our method aims at keeping these design secret by composing rounds. For a given cipher acting on n-bit blocks, let $x_1, \dots, x_n$ be the boolean input variables and $y_1, \dots, y_n$ the output bits after the first round. Each y_i can be expressed as a boolean function $p_{1,i}$ in the variables $x_1, \dots, x_n$ and obtained by combination of its component functions, including an S-box function. We compute as well the boolean functions $(p_{2,i})_{1\leq i\leq n}$ corresponding to the second round of the cipher. Let $(q_i)_{1\leq i\leq n}$ be the boolean functions that implement these two rounds (an example with DES is shown on Fig. 1). This system of boolean functions allows us to describe the two rounds of the cipher in an executable way, but without revealing information about the design of the algorithm.



$$\begin{cases} R_2(1) = q_1(L_0(1), \dots, L_0(32), R_0(1), \dots, R_0(32)) \\ R_2(2) = q_2(L_0(1), \dots, L_0(32), R_0(1), \dots, R_0(32)) \\ \vdots \\ R_2(32) = q_{32}(L_0(1), \dots, L_0(32), R_0(1), \dots, R_0(32)) \end{cases}$$

FIGURE 1. Two rounds of DES.

We further subtract some equations from attackers analysis by a physical mean. The design of the algorithm is stored in a volatile memory which is zeroized when an intrusion is detected. Such techniques known as *tamper response* can be implemented following various ways [21]. The simplest one is a quick drop of the power line of the memory (see Fig. 2). Due to data remanence phenomena [13] and external conditions [20], it is hard to exactly predict how many equations will be erased. Now the confidentiality of the design is based on the same problem as the $2\mathbf{R}^-$ scheme.

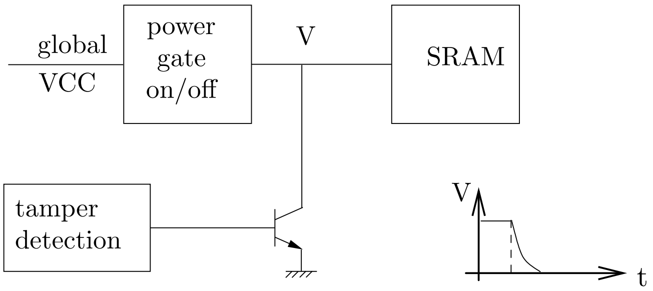


Figure 2. Physical implementation of the "–" of $2\mathbf{R}^-$.

There is one point that we have not tackled yet but that is worthy of attention: the treatment of the secret key. As the key is usually diversified into several subkeys (one for each round), we have several possibilities to implement the block cipher, among them:

(1) The boolean functions have additionnal variables k_i so that for each round, the corresponding subkey can be input, the key schedule being performed separately. At one hand, this allows flexibility in key injection, and has the advantage that each round can be represented by the same boolean functions, so we can implement the whole cipher with only one composition of two rounds. On the other hand, it adds a lot of variables to the boolean functions we have to compute.
(2) The secret key is integrated into the boolean functions, *i.e.* we perform the key schedule before the implementation of the cipher and we compute the boolean functions with only the text variables as input. This has the drawback that every round will have a different expression, so we have to compute and implement every composition of 2 rounds separately.
(3) If the cipher permits it, we can envisage an intermediate solution. When the block cipher has a very simple key-schedule, it is possible to integrate the main key and the key-schedule into the boolean functions. We can think for

instance about the block cipher 3-Way [9] of which key-schedule is reduced to bitwise XOR-ing a short round constant to the main key. This allows us to implement all the cipher with only one composition of two rounds, but without adding too many variables to the boolean functions.

These different solutions may lead to different levels of security.

3. An example

3.1. BDDs

We choose binary decision diagrams (BDDs) for representing boolean circuits. They were introduced in 1986 by Bryant [4] and are known to give a compact representation of logical functions. Some operations are defined on BDDs. For instance, we can use their composition for step by step computing the BDDs standing for one or many rounds. As well, the algorithm for evaluating BDDs can be considered as a trivial way to implement our solution with a network of multiplexers.

BDDs are data structures used to represent boolean functions. Here we shortly present their properties, the interested reader is referred to [1, 4, 5].

Let f be a boolean function of n variables. If $f|_{x_i=b}$ denotes the function resulting when the i-th variable is replaced by the constant b, the *Shannon expansion* of the function f around variable x_i is given by:

$$f = x_i \cdot f|_{x_i=1} + \overline{x_i} \cdot f|_{x_i=0}$$

This simple relation is used to represent boolean functions as particular graphs in an if-then-else notation.

Definition 3.1. A *binary decision diagram (BDD)* is a rooted, directed acyclic graph with two types of nodes. A *non-terminal node* N is labelled $i \in \{0, \ldots, n\}$ and has two children noted $low(N)$ and $high(N)$. A *terminal node* is labelled 0 or 1 and has no child.

A graph having root node labelled i denotes the function f_i

$$f_i(x_1, \ldots, x_n) = x_i \cdot f_{high(N)}(x_1, \ldots, x_n) + \overline{x_i} \cdot f_{low(N)}(x_1, \ldots, x_n)$$

The set of values $\{x_1, \ldots, x_n\}$ describes a path in the graph starting from the root : at each node labelled i, we follow the high child if $x_i = 1$ (“`THEN`”) and the low child otherwise (“`ELSE`”).

Definition 3.2. A BDD is *ordered (OBDD)* if, on all paths through the graph, the labels respect a given order. An OBDD is *reduced (ROBDD)* if the following conditions are satified:

uniqueness: two nodes having the same label and children are equal;

non-redondant tests: there are no node with both children leading to the same node.

These three conditions for constructing an ROBDD are illustrated on Fig. 3. Figure 4 shows an example on the function $f(x, y, z) = x \cdot y + z$.

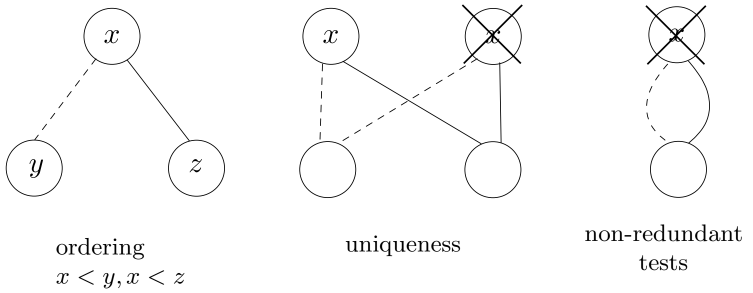


Figure 3. The conditions of ROBDDs.

Note that ROBBDs depend only on the order of the variables, so they are canonical representations of functions. In other words, for a given variables order, any way of computing an ROBBD leads to the same result. There exist various types of BDDs. Here we use *signed* BDDs, where a tag is added on each link for if we have to complement the result. This leads to more compact representations, as a function and its complementary can be represented by the same BDDs.

3.2. Grouping together two rounds of DES

To fix ideas, we here give some figures (see Tab. 1) on the number of nodes needed to represent the right part of the composition of the first two rounds of DES with signed ROBDDs. What we exactly compute is illustrated by Fig. 1. Each one of the 32 bits here stands as a logical function of 34 variables. Note that when represented by polynomials, each of these logical functions has degree 25 and more than 150000 terms. We used the BDD library [22] to compute the composition of two rounds. The following table

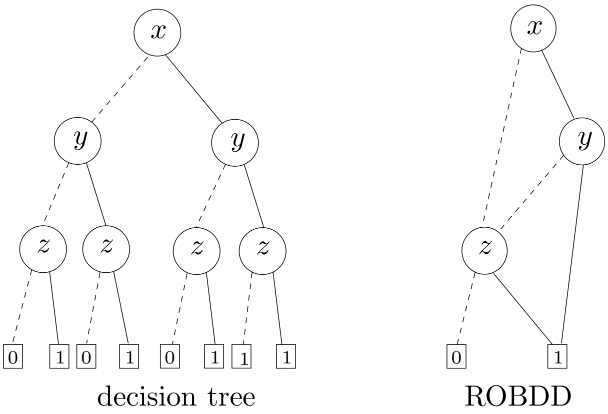


FIGURE 4. Representations of the function $f(x, y, z) = x \cdot y + z$.

gives the exact complexity of the BDDs, which varies from 6667 to 34947 nodes (13750 on average), R(i, j) standing for the j-th bit of the right block after i rounds of DES. Note that in this experiment the key is fixed to a random value. As for the variable ordering, it has a great influence on the size of the BDDs, for instance the size of the BDD for one bit can be more than 1.5 millions of nodes with some orders. The figures given here were obtained with an order that gives acceptable size for all BDDs. However, we think that the complexity can be further reduced, for instance by using a specific order for each output bit. Further research is needed to explicit the relation between the input variable ordering and the size of the resulting BDD.

bit	#nodes	bit	#nodes	bit	#nodes	bit	#nodes
R(2,1)	13448	R(2, 9)	16536	R(2,17)	17256	R(2,25)	9402
R(2,2)	30741	R(2,10)	13564	R(2,18)	34947	R(2,26)	13449
R(2,3)	9322	R(2,11)	6667	R(2,19)	7240	R(2,27)	6944
R(2,4)	7095	R(2,12)	7067	R(2,20)	13436	R(2,28)	25947
R(2,5)	6938	R(2,13)	32393	R(2,21)	7002	R(2,29)	6813
R(2,6)	19294	R(2,14)	9285	R(2,22)	7057	R(2,30)	20057
R(2,7)	7057	R(2,15)	6914	R(2,23)	16337	R(2,31)	17070
R(2,8)	9592	R(2,16)	16076	R(2,24)	18064	R(2,32)	7070

TABLE 1. Complexity of BDDs for the right block of DES after 2 rounds.

3.3. An implementation

In a straightforward implementation of the complete DES, 8 sets of such BDDs (one for each pair of rounds) are placed in a memory and we run through them, according to the value of the 64 input bits. Each node (except the last one) is coded on 40 bits and consists of its variable number, 2 tags for the signs of the "THEN" and "ELSE" links (we use signed ROBBDs), and the addresses of its "THEN" and "ELSE" nodes. All the BDDs necessary to represent the 16 DES rounds need approximately 18 Mo of memory to be stored using this representation. We used a RAM memory, which is accessed by an FPGA (Field Programmable Gate Array). The FPGA is programmed to take as input the 64 plaintext bits, and to run through the BDDs in memory according to these values. When the FPGA reaches the leaves of the last set of BDDs, it gets the 64 output bits. The throughput of this implementation is 152 Kbits/s.

As a comparison, the white-box DES implementation of [14], which is a software implementation, occupies 4.5 MB and encrypts one block in 30ms.

4. Conclusion

We introduce a new way of implementing cryptographic algorithms preserving their confidentiality. Our technique demands some tamper response in case of intrusion to obtain an instance of a hard algorithmic problem. There is still avenues to improve this. For instance, and as usual with BDDs, variable ordering should have a great importance for size optimization of manipulated graphs [5]. A very simple implementation of this scheme consists in storing BDDs in an external but tamper responsive SRAM, and to add some logic to run through this memory. From another point of view, note that some cryptographic algorithms are more difficult to represent this way as they rely on primitives for which BDDs are not so efficient such as multiplicators or rotators. Finally, we try to place ourselves outside known attacks but we are dealing with special instances where a sparse polynomial is composed with approximately itself. We invite readers to carefully analysis our solution before using it.

References

[1] Henrik Reif Andersen. An Introduction to Binary Decision Diagram. Lecture notes, October 1997.

[2] Eli Biham. Cryptanalysis of Patarin's 2-Round Public Key System with S-Boxes (2R). In *Proceedings of EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 408–416. Springer-Verlag, 2000.

[3] Olivier Billet and Henri Gilbert. A Traceable Block Cipher. In Chi-Sung Lailh, editor, *Proceedings of ASIACRYPT'03*, volume 2894 of *Lecture Notes in Computer Science*, pages 331–346. Springer-Verlag, 2003.

[4] Randal E. Bryant. Graph-Based Algorithms for Boolean Functions Manipulation. *IEEE Transactions on Computer*, 8(C-35):677–691, 1986.

[5] Randal E. Bryant. Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.

[6] S. Chow, P. Eisen, H. Johnson, and P.C. van Oorschot. A White-Box DES Implementation for DRM Applications. In *Proceedings of ACM CCS-9 Workshop DRM 2002*, volume 2595 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 2003.

[7] S. Chow, P. Eisen, H. Johnson, and P.C. van Oorschot. White-Box Cryptography and an AES Implementation. In *Proccedings of SAC'02*, Lecture Notes in Computer Science, pages 250–270. Springer-Verlag, 2003.

[8] Nicolas T. Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In Bart Preneel, editor, *Proceedings of EUROCRYPT'00*, volume 1807 of *Lecture Notes in Computer Science*, pages 392–407. Springer-Verlag, 2000.

[9] Joan Daemen, René Govaerts, and Joos Vandewalle. A New Approach Towards Block Cipher Design. In R. Anderson, editor, *Proceedings of FSE'93*, volume 809 of *Lecture Notes in Computer Science*, pages 18–32. Springer-Verlag, 1994.

[10] Ye Ding-Feng, Lam Kwok-Yan, and Dai Zong-Duo. Cryptanalysis of "2R" Schemes. In *Proceedings of CRYPTO'99*. Springer-Verlag, 1999.

[11] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[12] Louis Goubin and Jacques Patarin. Asymmetric Cryptography with S-Boxes. In *Proceedings of 1st International Information and Communications Security Conference*, pages 369–380, 1997.

[13] Peter Gutmann. Data Remanance in Semiconductor Devices. In *10th USENIX Security Symposium*, pages 39–54, 2001.

[14] Hamilton E. Link and William D. Neumann. Clarifying Obfuscation: Improving the Security of White-Box Encoding. Cryptology ePrint Archive, Report 2004/025, 2004. `http://eprint.iacr.org/2004/025`.

[15] Richard Lipton and Tomas Sander. An additively homomorphic encryption scheme or how to introduce a partial trapdoor in the discrete log, 1997.

[16] Jacques Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In

Ueli M. Maurer, editor, *Proceedings of EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48. Springer-Verlag, 1996.

[17] Jacques Patarin, Louis Goubin, and Nicolas T. Courtois. Improved Algorithms for Isomorphisms of Polynomials. In Kaisa Nyberg, editor, *Proceedings of EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 184–200. Springer-Verlag, 1998.

[18] Michael Roe. How to Reverse Engineer an EES Device. In Bart Preneel, editor, *Proceedings of HFE'94*, volume 1008 of *Lecture Notes in Computer Science*, pages 305–328. Springer-Verlag, 1995.

[19] Tomas Sander and Christin F. Tschudin. On Software Protection Via Function Hiding. In *Proceeding of Information Hiding, Second International Workshop*, volume 1525 of *Lecture Notes in Computer Science*, pages 111–123. Springer-Verlag, 1998.

[20] Sergei Skorobogatov. Low temperature data remanence in static RAM. Technical report, University of Cambridge Computer Laboratory, 2002.

[21] Steve H. Weingart. Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defense. In Çetin Kaya Koç and Christof Paar, editors, *Proceedings of CHES'00*, volume 1965 of *Lecture Notes in Computer Science*, pages 302–317. Springer-Verlag, 2000.

[22] Jean-Baptiste Yunès. BDD Library. Available from `http://www.liafa.jussieu.fr/web9/outils/outils_scientifiques_fr.php`.

We thank Jean-Francis Michon, Pierre Valarcher and Jean-Baptiste Yunès for profitable discussions and permitting us to use their BDD library.

CRYPTOGRAPHICAL BOOLEAN FUNCTIONS CONSTRUCTION FROM LINEAR CODES

Philippe Guillot[1]

Abstract. This paper presents an extension of the Maiorana-McFarland method for building Boolean functions with good cryptographic properties.

The original Maiorana-McFarland construction was proposed to design bent functions. Then, it was extended in [1] to build highly nonlinear resilient functions.

The classical construction splits the set of variables into two separate subsets. There, is proposed a decomposition of the whole working space into two complementary vector spaces. One of these spaces is considered as a linear code and its parameters assign cryptographic properties to the constructed Boolean function.

The cryptographical properties we are interested in are nonlinearity, resiliency and propagation properties.

The obtained functions are linearly equivalent to those constructed by the traditional way. Thus, no improvement for affine invariant parameters such as nonlinearity is expected. On the other hand, for non affine invariant cryptographic parameters such as resiliency order or propagation order, better values are obtained.

1. **Motivation**

Cryptographic algorithms design is still based on confusion and diffusion principles stated by Shannon in 1949 (see [6]). Diffusion means that a bit change in the key is propagated in the whole ciphertext. It is performed by linear transformations. Confusion means that the relationship between the key, the plaintext and the

[1] Université Paris 8. email: philippe.guillot@univ-paris8.fr

ciphertext is complex and involved. It is performed by nonlinear transformations and mostly implemented as Boolean functions.

The nonlinearity may be defined in at least four ways.

First, a Boolean function is nonlinear if it is not correlated to any affine function. This is the correlation criterion. It means that the function is far from the set of affine functions.

Nonlinearity may also be defined through propagation properties. If some variables are changed, is the value changed too ? If the function has a linear structure, then the answer is always predictable : yes or no depending on which variables are changed. For cryptographic oriented functions, the answer should be unpredictable.

Third, a linear function is expressed as a n-variable polynomial of degree one. A nonlinear function should be expressed as a polynomial of degree as high as possible.

Finally, a linear function is simple. A nonlinear function is expected to be complex. The complexity may be measured by several ways : number of gates to implement it, number of nodes in a Binary Decision Diagram, and so on.

The designer has to deal with all these criteria together. It is rarely possible to optimize all of them. We are mainly interested in the sequel in a compromise between correlation and propagation criteria.

2. Spectral Analysis

The mathematical tool to explore nonlinearity of boolean functions consists in two objects: the Walsh transform and the autocorrelation function. In this section, we recall basic results and definitions which will be used in the sequel.

Let n be any integer ≥ 2 and $\mathbb{F}_2^n$ be the n-dimensional vector space over the field $\mathbb{F}_2$. For any vectors $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ in $\mathbb{F}_2^n$, the inner product of x and y is $x \cdot y = x_1y_1 + \cdots + x_ny_n \in \mathbb{F}_2$.

A Boolean function over $\mathbb{F}_2^n$ is a mapping $\mathbb{F}_2^n \to \mathbb{F}_2$.

The Fourier transform of a Boolean function f is by definition the real valued function $\widehat{f}$ defined as

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

The Fourier transform of the sign function $f_\chi = (-1)^f = 1-2f$ is called the Walsh transform of f:

$$\forall u \in \mathbb{F}_2^n \quad \widehat{f_\chi}(u) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+u\cdot x}.$$

For all $u \in \mathbb{F}_2^n$, the Wash transform value $\widehat{f_\chi}(u)$ is the number of times $f(x)$ equals $u \cdot x$ minus the number of time it differs. Thus $\widehat{f_\chi}(u)$ measures the correlation between f and the linear form $\lambda_u : x \mapsto u \cdot x$. The function f is statistically independent from λ_u if and only if $\widehat{f_\chi}(u) = 0$. In particular, f is balanced if and only if $\widehat{f_\chi}(0) = 0$.

The power of this tool is based on the orthogonality relation of the so called Walsh functions $\chi_u : x \mapsto (-1)^{u\cdot x}$:

$$\begin{aligned} \forall (u,v) \in \mathbb{F}_2^n \times \mathbb{F}_2^n \quad \sum_{x \in \mathbb{F}_2^n} \chi_u(x)\chi_v(x) &= \sum_{x \in \mathbb{F}_2^n} (-1)^{(u+v)\cdot x} \\ &= \begin{cases} 2^n & \text{if } u = v; \\ 0 & \text{elsewhere.} \end{cases} \end{aligned}$$

For any p-dimensional vector subspace E of $\mathbb{F}_2^n$, the dual of E, denoted $E^\perp$, is the $(n-p)$-dimensional vector space of linear forms that vanish on E.

$$E^\perp = \{u \in \mathbb{F}_2^n \mid \forall x \in E,\ u \cdot x = 0\}.$$

If f is defined on a vector subspace E of $\mathbb{F}_2^n$, the expression of the Fourier transform of f is given by

$$\widehat{f}(u) = \sum_{x \in E} f(x)(-1)^{u\cdot x}.$$

A first glance, $\widehat{f}$ is defined over the whole space $\mathbb{F}_2^n$, but in fact $\widehat{f}(u)$ remains unchanged when u is added to any element of $E^\perp$. In other word, $\widehat{f}$ is constant on any coset of $E^\perp$. Thus, $\widehat{f}$ may be considered as defined over the quotient space $\mathbb{F}_2^n / E^\perp$.

For convenience and easier computation, it may be useful to consider a complementary space F of E, *i.e.* such that $\mathbb{F}_2^n = E \oplus F$. The dual spaces $E^\perp$ and $F^\perp$ are complementary too and the quotient space $\mathbb{F}_2^n / E^\perp$ is isomorphic to $F^\perp$. Thus, $\widehat{f}$ is considered as defined on $F^\perp$.

The second object of the spectral analysis is the autocorrelation function. For any Boolean function f over $\mathbb{F}_2^n$, the autocorrelation function of f, denoted r_f is by definition:

$$r_f : \begin{array}{ccc} \mathbb{F}_2^n & \to & \mathbb{R} \\ u & \mapsto & \displaystyle\sum_{x\in\mathbb{F}_2^n} (-1)^{f(x)+f(x+u)} \end{array} .$$

The value $r_f(u)$ is the number of time $f(x)$ equals $f(x+u)$ minus the number of times it differs. Thus it measure the avalanche effect of vector u.

If $r_f(u) = 0$ then the value of f is unpredictable when the variables x_i such that $u_i = 1$ are changed.

If $r_f(u) = \pm 2^n$ then the function $x \mapsto f(x) + f(x+u)$ is constant. In this case, the vector u is called a linear structure for f. The set of linear structures over $\mathbb{F}_2^n$ is the subset of Boolean function that do have a linear structure. The set of affine functions over $\mathbb{F}_2^n$ is a subset of the set of linear structures (see [4]).

3. Cryptographic criteria

In a symmetric algorithm, the Boolean function is in charge of the confusion property. Thus, it has to be highly nonlinear. The nonlinearity is measured by the distance $\delta(f)$ of the Boolean function f from the set of affine functions. It can be expressed by mean of the Walsh transform (see [4]) :

$$\delta(f) = 2^{n-1} - \frac{1}{2}\max_{u\in\mathbb{F}_2^n}\big(|\widehat{f_\chi}(u)|\big).$$

The lower is the greatest magnitude of the Walsh transform, the further is the function from the set of affine functions.

Another nonlinearity measure is given by the distance $\sigma(f)$ of the Boolean function f from the set of linear structures. It can be expressed by mean of the autocorrelation function (see [4]) :

$$\sigma(f) = 2^{n-2} - \frac{1}{4}\max_{u\in\mathbb{F}_2^n\setminus\{0\}}\big(|r_f(u)|\big).$$

Similarly, the lower is the greatest magnitude of the autocorrelation function on nonzero vectors, the further is the function from the set of linear structures.

The cryptographer should minimize maximum magnitude of both the Walsh transform and the autocorrelation function in order to design non linear functions with good cryptographic properties.

Both δ and σ are affine invariants, *i.e.* the values $\delta(f)$ and $\sigma(f)$ remain unchanged if f is composed with any invertible affine mapping on $\mathbb{F}_2^n$.

A Boolean function is said to be k-resilient, if the knowledge of any k variables does not provide any statistical information on the value of f. A function is 0-resilient means that it is balanced. Resiliency has a nice characterization by mean of the Walsh transform (see [7]).

Proposition 3.1. *A Boolean function f over $\mathbb{F}_2^n$ is k-resilient if and only if for any vector $u \in \mathbb{F}_2^n$ of weight less than or equal to k, its Walsh transform vanishes at vector u, i.e. $\widehat{f_\chi}(u) = 0$.*

A Boolean function satisfies the propagation criterion at order k, which is denoted $PC(k)$, if changing any k variables does not allow to guess if the value of f changes or not. Similarly, propagation criterion has a characterization by mean of the autocorrelation function. The following proposition is a straightforward consequence of the definition.

Proposition 3.2. *A Boolean function f over $\mathbb{F}_2^n$ satisfies $PC(k)$ if and only if for any nonzero vector $u \in \mathbb{F}_2^n \setminus \{0\}$ of weight less than or equal to k, its autocorrelation function vanishes at vector u, i.e. $r_f(u) = 0$.*

4. The Maiorana-McFarland Construction

The Maiorana-McFarland construction was originally designed to build bent function (see [3]). It has been extended in [1] to build resilient functions. Here, we extend it again according to a technique similar to those proposed in [2].

Let $n \geq 2$ be an integer and $\mathbb{F}_2^n = E \oplus F$ a decomposition into two complementary vector subspaces: E of dimension p and F of dimension $q = n - p$.

For any mapping $\pi : E \to \mathbb{F}_2^n$ and any mapping $h : E \to \mathbb{F}_2$ the Maiorana-McFarland construction defines a Boolean function f as follows:

$$f: \begin{array}{ccc} E \oplus F & \rightarrow & \mathbb{F}_2 \\ x+y & \mapsto & \pi(x)\cdot y + h(x) \end{array} .$$

The mapping π is defined onto $\mathbb{F}_2^n$, but as $\pi(x)$ is only involved by an inner product with an element of F, the value of f is unchanged when $\pi(x)$ is translated by any vector in $F^\perp$. Thus, π may be considered to be defined onto the quotient space $\mathbb{F}_2^n/F^\perp$, which is isomorphic to $E^\perp$.

The traditional definition appears to be a particular case of this definition by considering $E = \{(x_1,\dots,x_n) \in \mathbb{F}_2^n \mid x_{p+1} = 0,\dots,x_n = 0\}$ and $F = \{(x_1,\dots,x_n) \in \mathbb{F}_2^n \mid x_1 = 0,\dots,x_p = 0\}$.

Conversely, any linear equivalent of the classical Maiorana-McFarland construction may be obtained by the way presented here.

In order to establish the correlation properties of the function f, the following proposition expresses the Walsh transform.

Proposition 4.1. *For any $w \in \mathbb{F}_2^n$, let $w = u+v$ be the unique decomposition of w in the direct sum $E^\perp \oplus F^\perp$ with $u \in E^\perp$ and $v \in F^\perp$.*

$$\widehat{f_\chi}(u+v) = 2^q \sum_{x\in\pi^{-1}(u)} (-1)^{h(x)+v\cdot x} \tag{1}$$

Proof. By definition, for any $w \in \mathbb{F}_2^n$,

$$\begin{aligned} \widehat{f_\chi}(w) &= \sum_{(x,y)\in E\times F} (-1)^{\pi(x)\cdot y + h(x) + w\cdot(x+y)} \\ &= \sum_{x\in E} (-1)^{h(x)+w\cdot x} \sum_{y\in F} (-1)^{(\pi(x)+w)\cdot y} \end{aligned}$$

The latter sum equals $|F| = 2^q$ if $\pi(x)+w \in F^\perp$ and 0 elsewhere. Thus, the only nonzero terms in the above sum are those such as $\pi(x) \in w + F^\perp$. As, $x \in E$ in the sum, $w\cdot x = u\cdot x + v\cdot x = v\cdot x$. Furthermore, $\pi(x) \in w + F^\perp \Leftrightarrow x \in \pi^{-1}(u)$ and the result holds. □

In order to study resiliency, we are interested in the case where the Walsh transform vanishes. This occurs in two cases : either if $\pi^{-1}(u)$ is empty or if the function $x \mapsto h(x) + v\cdot x$ is balanced on the subset $\pi^{-1}(u)$ of E. This latter property is not so easy to check in general. An interesting particular case is when $\pi^{-1}(u)$ is an affine subspace of E.

Proposition 4.2. *Let u be an element of $E^{\perp}$, if the preimage $\pi^{-1}(u)$ is the affine subspace of E defined by direction V_u and element x_u, then, for all $v \in F^{\perp}$,*

$$\widehat{f_{\chi}}(u+v) = 2^q(-1)^{v\cdot x_u}\widehat{(h_u)_{\chi}}(v),$$

where h_u denotes the Boolean function on V_n defined by $t \mapsto h(t+x_u)$.

Proof. Set $x = t + x_u$ in the sum of expression (1) and the result holds. □

In order to establish propagation properties of the function f, the following proposition expresses the autocorrelation function.

Proposition 4.3. *For any $z \in \mathbb{F}_2^n$, let $z = x + y$ the unique decomposition of z in the direct sum $E \oplus F$ with $x \in E$ and $y \in F$.*

$$r_f(x+y) = 2^q \sum_{t\in E|\pi(t)+\pi(t+x)\in F^{\perp}} (-1)^{h(t)+h(t+x)+\pi(t)\cdot y}.$$

Proof.

$$\begin{aligned} r_f(x+y) &= \sum_{(t,s)\in E\times F} (-1)^{\pi(t)\cdot s+h(t)+\pi(t+x)\cdot(s+y)+h(t+x)} \\ &= \sum_{t\in E}(-1)^{h(t)+h(t+x)+\pi(t+x)\cdot y}\sum_{s\in F}(-1)^{(\pi(t)+\pi(t+x))\cdot s} \end{aligned}$$

The latter sum equals $|F| = 2^q$ if $\pi(t)$ and $\pi(t+x)$ belong to the same $F^{\perp}$–coset, and equals 0 elsewhere. Thus, the only nonzero terms are those for which $\pi(t) + \pi(t+x) \in F^{\perp}$. □

If $x = 0$, then any t in E satisfies the condition $\pi(t)+\pi(t+x) = 0 \in F^{\perp}$. Thus for any $y \in F$,

$$r_f(y) = 2^q \sum_{t\in E}(-1)^{\pi(t)\cdot y}.$$

Let $u = \pi(t)$. For any y in F,

$$r_f(y) = 2^q \sum_{u\in E^{\perp}} \psi(u)(-1)^{u\cdot y} = 2^q\widehat{\psi}(y), \tag{2}$$

where, for any $u \in E^{\perp}$, the value $\psi(u)$ is the number of elements of the preimage $\pi^{-1}(u)$.

5. Practical constructions

5.1. π is one-to-one

We assume in this section, that for any $u \in E^\perp$, the preimage $\pi^{-1}(u)$ contains at most one element. This is possible only if $p \leq q$. If this preimage is nonempty, then the vector space V_u of proposition 4.1 is always the null vector space and $\widehat{(h_u)_\chi}(u) = \pm 1$. Consequently, for all $(u, v) \in E^\perp \times F^\perp$,

$$\widehat{f_\chi}(u+v) = \begin{cases} \pm 2^q & \text{if } \pi^{-1}(u) \neq \emptyset; \\ 0 & \text{elsewhere.} \end{cases} \tag{3}$$

The assumption on π implies that for all t and x in E,

$$\pi(t) + \pi(t+x) \in F^\perp \iff x = 0.$$

From proposition 4.3, if $x \neq 0$ then $r_f(x+y) = 0$. Finally, from relation (2), for all $(x, y) \in E \times F$,

$$r_f(x+y) = \begin{cases} 2^q \widehat{\varphi_{\pi(E)}}(y) & \text{if } x = 0; \\ 0 & \text{elsewhere,} \end{cases} \tag{4}$$

where $\varphi_{\pi(E)}$ denotes the indicator of the image $\pi(E)$ in $E^\perp$.

From relation (3), the following correlation properties of f are deduced:

- f is balanced if and only if $\widehat{f_\chi}(0) = 0$, *i.e.* $0 \notin \pi(E)$. This requires in particular $p < q$.
- If for all $x \in E$, the coset leaders of $\pi(x) + F^\perp$, which are by definition the element of lowest weight, have weight at least k, then $\widehat{f_\chi}$ vanishes for all vectors of weight $< k$. Therefore, f is $(k-1)$-resilient.
- As $\widehat{f_\chi}$ has constant magnitude equal to 2^q, the nonlinearity of f is $\delta(f) = 2^{n-1} - 2^{q-1}$.

From relation (4), the following propagation properties of f are deduced:

- As $r_f(z)$ is nonzero only for $z \in F$, if F has minimum distance d, then f satisfies the propagation criterion $PC(d-1)$.

- The distance from f to the set of linear structures depends on the nonlinearity of the $\pi(E)$ indicator $\varphi_{\pi(E)}$. Namely,

$$\sigma(f) = 2^{n-2} - 2^{q-2} \max_{u \in E^{\perp} \setminus \{0\}} |\widehat{\varphi_{\pi(E)}}(u)|.$$

In particular, if $\pi(E)$ spans the whole space $\mathbb{F}_2^n$, then no nonzero linear form is constant over $\pi(E)$ and f is non degenerate in the sense that it is not affinely equivalent to a Boolean function of strictly less variables.

Relation (4) shows that the propagation order may be increased if $\varphi_{\pi(E)}$ is chosen resilient. But on the other hand, due to the Sarkar-Maitra's bound (see [5]), this increases the greatest magnitude of the autocorrelation function and thus decreases the distance from the set of linear structures.

Note that the cryptographic properties of f only depend on the properties of the vector space F and of the image $\pi(E)$. Once the image $\pi(E)$ and the vector space chosen, choice of permutation π and Boolean function h lead to $2^p! \times 2^{2^p}$ different functions with similar cryptographic properties.

By an appropriate choice of π or h, the algebraic degree of f can be increased to the maximum value, which equals p.

Unfortunately, the following proposition states that, when π is one-to-one, no better resiliency order than the classical construction can be expected.

Proposition 5.1. *If π is one-to-one, then the maximum resiliency order k reached by such a construction satisfies*

$$2^p \leq \binom{q}{k+1} + \binom{q}{k+2} + \cdots + \binom{q}{q} \qquad (5)$$

This is based on the following result:

Lemma 5.2. *Let C be any d-dimensional vector subspace of $\mathbb{F}_2^n$. For any integer k such that $0 \leq k \leq n-d$, Then there exists at least*

$$N = 1 + \binom{n-d}{1} + \cdots + \binom{n-d}{k} \qquad (6)$$

coset leaders of C of weight less than or equal to k.

Proof. Without a loss of generality, one may assume that a generator matrix of C is of the systematic form

$$G = \begin{pmatrix} 1 & & & \\ & \ddots & & A \\ & & 1 & \end{pmatrix}$$

with information positions on the d first components. Each coset admits a unique element that vanishes on the information set, namely of the form $x = (0, \ldots, 0, x_{d+1}, \ldots, x_n)$. The corresponding coset leader has necessarily lower weight than x. The value of N in (6) is the number of such vector x of weight $\leq k$. □

Proof. (*of proposition 5.1*) Inequality (5) in proposition 5.1 states that the number of coset leaders of weight $> k$ given by lemma 5.2, is greater than or equal to the number of vectors in E. □

Example. Let $p = 4$ and $q = 5$ and F be the 5-dimensional vector space given by the generator matrix

$$G_F = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

This vector space has minimum distance $d = 3$, thus the function f satisfies $PC(2)$. A generator matrix of the dual space $F^\perp$ is

$$G_{F^\perp} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The $2^p = 16$ cosets defined by vectors $u = (u_1, \ldots, u_5, 0, 0, 0, 0)$ with $(u_1, \ldots, x_u) \in \{00011, 00101, 00110, 00111, 1001, 1011, 1100, 1110, 1111, 10001, 10010, 10101, 10110, 10111, 11000, 11001\}$ only contain vectors of weight ≥ 2. Consequently, the function f is 1-resilient.

As $\max_{u \in \mathbb{F}_2^n} |\widehat{f_\chi}(u)| = 32$, the nonlinearity of f is $\delta(f) = 2^8 - 2^4 = 240$.

For all $x \in \mathbb{F}_2^5 \setminus \{0\}$, the indicator of $\pi(E)$ satisfies, $|\widehat{\varphi_{\pi(E)}}(x)| \leq 4$, thus $\max_{z \in \mathbb{F}_2^n \setminus \{0\}} |r_f(z)| = 128$ and the distance from f to the set of linear structures is $\sigma(f) = 128 - 32 = 96$.

5.2. π is two-to-one

In this section, we assume that π is a two-to-one mapping, that is to say, for any u in $\pi(E)$, the preimage $\pi^{-1}(u)$ contains exactly two elements, namely x_u and x'_u. This implies $p + 1 \geq q$.

We first examine the Walsh transform of f in such a case. As any pair is a one-dimensional affine subspace, the proposition 4.2 is applicable. With the notations of proposition 4.2, V_u is the vector space $\{0, x_u + x'_u\}$ and for any $v \in F^\perp$,

$$\begin{aligned} \widehat{(h_u)_\chi}(v) &= (-1)^{h(x_u)+v\cdot x_u} + (-1)^{h(x'_u)+v\cdot x'_u} \\ &= \begin{cases} 0 & \text{if } h(x_u) + h(x'_u) \neq v \cdot (x_u + x'_u); \\ \pm 2 & \text{elsewhere.} \end{cases} \end{aligned}$$

For convenience, let H denote the Boolean mapping on $F^\perp$ defined by $H : x \mapsto h(x_u) + h(x'_u)$. The Walsh transform of f is expressed, for any $u \in E^\perp$ and any $v \in F^\perp$:

$$\widehat{f_\chi}(u+v) = \begin{cases} 0 & \text{if either } \pi^{-1}(u) = \emptyset \text{ or } H(u) \neq v \cdot (x_u + x'_u) \\ \pm 2^{q+1} & \text{elsewhere} \end{cases} \tag{7}$$

In particular, f is balanced if either no vector of E maps to 0 by π or $h(x_0) \neq h(x'_0)$.

We study now the autocorrelation function of f.

For any t and x in E, if $x \neq 0$, then stating that $\pi(t)$ and $\pi(t + x)$ belong to the same $F^\perp$–coset defined by vector $u \in E^\perp$ means that $\{t, t+x\}$ is precisely the pair $\{x_u, x'_u\}$ for this coset, and $x_u + x'_u = x$. Thus, from proposition 4.3, and as the pair $\{x_u, x'_u\}$ appears for both $x_u = t$ and $x_u = t + x$, for any $x \in E \setminus \{0\}$ and any $y \in F$,

$$r_f(x + y) = 2^{q+1} \sum_{u \in E^\perp | x_u + x'_u = x} (-1)^{H(u)+u\cdot y}. \tag{8}$$

Maximizing the propagation order requires that the autocorrelation function has as many zero values as possible. If for any

$u \in E^{\perp}$ the sum $x_u + x'_u$ is a constant x_0 independent of u, then the sum (8) is nonzero only for $x = 0$ or $x = x_0$. Let us study this particular case now.

Relation (7) becomes, for any $u \in E^{\perp}$ and $v \in F^{\perp}$:

$$\widehat{f_\chi}(u+v) = \begin{cases} 0 & \text{if either } \pi^{-1}(u) = \emptyset \text{ or } H(u) \neq v \cdot x_0; \\ \pm 2^{q+1} & \text{elsewhere.} \end{cases} \tag{9}$$

Let F' be the vector subspace of $F^{\perp}$ defined by $F' = \{v \in F^{\perp} \mid v \cdot x_0 = 0\}$. As $x_0 \in E$, then $x_0 \notin F$. Therefore, F' is a hyperplane of $F^{\perp}$. Each $F^{\perp}$-coset is the union of two F'-cosets defined by the value ε of the linear form $v \mapsto v \cdot x_0$. Thus, each F'-coset is characterized by a vector $u \in E^{\perp}$ that defines a $F^{\perp}$-coset, and a value $\varepsilon \in \mathbb{F}_2$. If any coset defined by $u \in E^{\perp}$ and $\varepsilon_u \in \mathbb{F}_2$ such that $H(u) = \varepsilon_u$ only contains vectors of weight $\geq k$, then, from relation (9), the function f is $(k-1)$-resilient.

To maximize the resiliency order, for any $F^{\perp}$–coset F_u defined by vector $u \in E^{\perp}$, one may choose $h(x_u)$ at random in $\mathbb{F}_2$ and define $h(x'_u)$ such that $h(x_u) + h(x'_u) = \varepsilon_u$, where ε_u defines the F'–coset in F_u which has the greatest minimum weight.

For the propagation point of view, the autocorrelation function has to be considered. Let G be the real valued function defined for any $u \in E^{\perp}$ by

$$G(u) = \varphi_{\pi(E)}(u) H_\chi(u) = \begin{cases} 0 & \text{if } u \notin \pi(E); \\ 1 & \text{if } u \in \pi(E) \text{ and } H(u) = 0; \\ -1 & \text{if } u \in \pi(E) \text{ and } H(u) = 1. \end{cases} \tag{10}$$

Relation (8) becomes, for any $x \in E$ and $y \in F$:

$$r_f(x+y) = \begin{cases} 2^{q+1}\widehat{\varphi_{\pi(E)}}(y) & \text{if } x = 0,\ \textit{i.e. } x+y \in F; \\ 2^{q+1}\widehat{G}(y) & \text{if } x = x_0,\ \textit{i.e. } x+y \in x_0 + F; \\ 0 & \text{elsewhere.} \end{cases} \tag{11}$$

It results from this relation that, if vector space F and the coset $x_0 + F$ have minimum nonzero weight k, then f satisfies $PC(k-1)$, and also, let M be the maximum of $\max_{y \in F} |\widehat{G}(y)|$ and $\max_{y \in F \setminus \{0\}} |\widehat{\varphi_{\pi(E)}}(y)|$, the distance of f from the set of linear structures is

$$\sigma(f) = 2^{n-2} - 2^{q-1}M.$$

Example. The following example shows the construction of a 10 variable 2-resilient and $PC(2)$ Boolean function f with $\delta(f) = 480$ and $\sigma(f) = 96$.

Let $p = 5$ and $q = 5$ and F be the 5-dimensional vector space given by the following generator matrix:

$$G_F = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

The vector space F has minimum distance $d = 4$. Let E be the complementary space of F of vectors whose 5 first components are null. Let x_0 be the element of E equal to $(0, 0, 0, 0, 0, 1, 1, 0, 1, 0)$. All the vectors in the coset $x_0 + F$ are of weight ≥ 3, thus the constructed function f satisfies $PC(2)$.

Let $\mathcal{E}$ be the set of vectors $u = (u_1, \ldots, u_5, 0, \ldots, 0)$, with $u_1 \cdots u_5 \in$ {00000, 10100, 01100, 01010, 11010, 10110, 11110, 00001, 10101, 01101, 11101, 10011, 01011, 11011, 10111, 01111}. The 16 cosets $u + F^{\perp}$, with $u \in \mathcal{E}$ are split by the linear form $t \mapsto x_0 \cdot t$, into two subsets and one of them only contains vectors of weight ≥ 3. For any $u \in \mathcal{E}$, let $\varepsilon_u \in \{0, 1\}$ be such that the coset $\{t \in u + F \mid x_0 \cdot t = \varepsilon_u\}$ only contains vectors of weight ≥ 3. The values of ε_u are respectively 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1. Thus it is possible to construct a 2-resilient function:

- define the image $\pi(E) = \mathcal{E}$, and for any $t \in E'$, choose once the same element in $\mathcal{E}$ for $\pi(t)$ and $\pi(t + x_0)$;
- for any $t \in E'$, let $u = \pi(t)$. Choose randomly $h(t)$ and define $h(t + x_0) = h(t) + \varepsilon_u$.

As $\max_{u \in \mathbb{F}_2^n} |\widehat{f_\chi}(u)| = 64$, the nonlinearity of f is $\delta(f) = 2^9 - 2^5 = 480$.

For all $x \in \mathbb{F}_2^5 \setminus \{0\}$, the indicator of $\pi(E)$ satisfies, $|\widehat{\varphi_{\pi(E)}}(x)| \leq 6$, and for all $x \in \mathbb{F}_2^5$, the function G defined by relation (10) satisfies $|\widehat{G}(x)| \leq 10$, thus $\max_{z \in \mathbb{F}_2^n \setminus \{0\}} |r_f(z)| = 640$ and the distance from f to the set of linear structures is $\sigma(f) = 256 - 160 = 96$.

Remark. If the pre-image by π is always an affine subspace of constant direction V, then it is equivalent to consider the decomposition $E' \oplus (F + V)$ and taking π one-to-one over E'.

6. Conclusion

We have studied a family of Boolean functions defined on a direct sum $E \oplus F$ of the whole space F_2^n, similarly to the Maiorana-McFarland construction. The cryptographic properties of the obtained functions depend on the parameters of the vector subspace F, seen as a linear code. In a particular construction, the vector space E is split as a union of affine subspaces. The two cases of affine subspaces of dimension 0 and 1 have been studied, enhancing the cryptographic properties of the constructed Boolean functions.

It remains to study other decompositions of vector space E as union of affine subspaces of greater dimension. Another research direction is to consider E as a union of quasi-disjoint vector spaces, that is to say vector spaces that intersect only on the zero vector.

Other important cryptographic parameters, such as algebraic immunity, have also to be considered and studied.

References

[1] P. Camion, C. Carlet, P. Charpin, N. Sendrier. *On Correlation Immune Functions. CRYPTO'91*, LNCS VOL. 576, pp. 86–100, 1992.

[2] C. Carlet. *Partially-bent functions*; *Designs, Codes and Cryptography* VOL. 3, pp. 135–145, 1993.

[3] J.F. Dillon. *Elementary Hadamard Difference Sets*; PhD. Thesis. University of Maryland, 1974.

[4] M. Meier, O. Staffelbach. *Nonlinearity Criteria for Cryptographic Functions*; *EUROCRYPT' 89*, LNCS VOL. 473, pp. 549–562, 1990.

[5] P. Sarkar, S. Maitra. *Nonlinearity Bounds and Constructions of Resilient Boolean Functions. CRYPTO'2000*, LNCS VOL. 1880, pp. 515–532, 2000.

[6] C.E. Shannon.; *Communication theory of Secrecy System*; Bell Sys. Tech journal VOL 28, pp. 656–715, 1949.

[7] Xiao Guo-Zhen, J.L. Massey. *A Spectral Characterization of Correlation-Immune Combining Functions. IEEE Trans. on Inf. Theory*, VOL. IT34, no 3, pp. 569–571, 1988.

GENERALIZED CONSTRUCTIONS OF HIGHLY NON-LINEAR MULTI-OUTPUT BOOLEAN FUNCTIONS

Zu-Ling Chang[1], Fang-Wei Fu[2] and Qiao-Yan Wen[1]

Abstract. This paper presents some generalized constructions of multi-output Boolean functions with high nonlinearity. Especially, we consider the constructions of highly nonlinear n-input m-output Boolean functions when $n < m$, multi-output bent functions and highly nonlinear multi-output balanced Boolean functions. When $n < m$, we produce the sufficient and necessary condition for existing multi-output Boolean function whose nonlinearity is nonzero and give one generalized method to construct such functions using the knowledge of Reed-Muller codes. For multi-output bent functions, we produce generalized methods to construct multi-output $\mathcal{M}$ and $\mathcal{PS}$ class bent functions. Finally we construct highly nonlinear balanced functions using the results of multi-output bent functions.

Keywords. Boolean functions, (n, m)-functions, nonlinearity, Reed-Muller code, bent functions, balanced functions.

1. **Preliminaries**

We consider Boolean functions from F_2^n to F_2(or simply functions on F_2^n), where F_2 is the finite field whose elements are 0 and 1, and the addition operator on F_2 is denoted by $\oplus$. A Boolean function $f(x_1, \ldots, x_n)$ can be described as the output column of

[1] School of Sciences, Beijing University of Posts and Telecommunications, Beijing, 100876, P. R. China, email: zlchang@eyou.com; wxq@bupt.edu.cn
[2] Temasek Laboratories, National Univ. of Singapore, Singapore 119260, Republic of Singapore, email: tslfufw@nus.edu.sg

its truth table, i.e., a binary string of length 2^n having the form

$$[f(0,0,\ldots,0), f(1,0,\ldots,0), f(0,1,\ldots,0),\ldots, f(1,1,\ldots,1)].$$

Also $f(x_1,\ldots,x_n)$ can be expressed as one n-variables polynomial

$$\begin{aligned} f(x_1,\ldots,x_n) &= a_0 \oplus \left(\bigoplus_{i=1}^{n} a_i x_i\right) \oplus \left(\bigoplus_{1\le i\ne j\le n} a_{ij}x_ix_j\right) \\ &\oplus \cdots \oplus a_{12\cdots n}x_1x_2\cdots x_n, \end{aligned}$$

where the coefficients $a_0, a_{ij},\ldots,a_{12\cdots n} \in F_2$. This representation of $f(x)$ is called the *algebraic normal form*(ANF) of $f(x)$. In the algebraic normal form of $f(x)$, every $x_{i_1}\ldots x_{i_s}$ is called one *monomial.* The *degree* of one monomial is the number of different x_i in it and the degree of one Boolean function $f(x)$ is defined as the maximum degree of monomials in its algebraic normal form.

An *affine* function $f(x_1,\ldots,x_n)$ is a function that takes the form of $f(x_1,\ldots,x_n) = a_0 \oplus a_1x_1 \oplus \cdots \oplus a_nx_n$, where $a_j \in F_2$, $j = 0,1,\ldots,n$. Furthermore f is called a *linear* function if $a_0 = 0$. The set of all affine functions on F_2^n is denoted by $A(n)$.

The *Hamming weight* of one binary string S, denoted by $wt(S)$, is the number of 1's in S, and the *Hamming distance* between two binary strings S_1, S_2 with same length, denoted by $d(S_1,S_2)$, is equal to $wt(S_1 \oplus S_2)$. One Boolean function on F_2^n is *balanced* if $wt(f) = 2^{n-1}$.

The *nonlinearity* of a Boolean function $f(x_1,\ldots,x_n)$ is

$$N_f = \min_{l\in A(n)} d(f,l).$$

We call one Boolean function a *bent function* if its nonlinearity is $2^{n-1} - 2^{\frac{n}{2}-1}$, i.e., the maximum nonlinearity of Boolean functions. At this time n is even.

An n-input m-output function, or an (n,m)-function $F(x)$ is a function from F_2^n to F_2^m, and it can be expressed as

$$F(x) = (f_1(x), f_2(x),\ldots, f_m(x)),$$

where every $f_i(x)$, the *component function* of $F(x)$, is a Boolean function on F_2^n having one output. The nonlinearity of $F(x)$ is

$$N_F = \min N_g,$$

where N_g is the nonlinearity of an arbitrary nonzero linear combination of component functions of $F(x)$, i.e., $g(x) = a_1 f(x) \oplus \cdots \oplus a_n f(x)$, where $a_1, \ldots, a_n \in F_2$, are not all zero.

For one (n, m)-function $F(x) = (f_1(x), f_2(x), \ldots, f_m(x))$, if for any output $(a_1, a_2, \ldots, a_m) \in F_2^m$, the number of input of $F(x)$ satisfying $F(x) = (a_1, a_2, \ldots, a_m)$ is 2^{n-m}, i.e.,

$$|\{F(x) = (a_1, a_2, \ldots, a_m) | x \in F_2^n\}| = 2^{n-m},$$

then we say this function is *balanced*. Furthermore, an (n, m)-function is balanced if and only if any nonzero linear combination of its component functions is a single-output balanced function. Obviously, if an (n, m)-function is balanced, then $n \geq m$. An (n, n)-function is called a *permutation* if it is balanced.

One (n, m)-function is one (n, m)*-bent function* if any nonzero linear combination of these component functions is one single output bent function. So the nonlinearity of (n, m)-bent function is $2^{n-1} - 2^{\frac{n}{2}-1}$, which is the maximum nonlinearity of all functions with n input.

(n, m)-function is a main component of one secure secret key cryptosystems such as stream ciphers and block ciphers. To resist cryptanalysis for symmetric-key block ciphers, including the linear cryptanalysis [12], we want (n, m)-functions to have high nonlinearity . Some useful results about multi-output Boolean functions have been given in [16, 17, 23], and some constructions for (n, m)-function with high nonlinearity are presented. But there are still many questions about this issue. For example, when $n < m$, some (n, m)-functions with nonzero nonlinearity have never been exhibited, the problem about constructing such multi-output Boolean functions with high nonlinearity is still unsolved. How to construct highly nonlinear multi-output bent functions and balanced functions are both interesting.

In this paper we will mainly solve such questions and find generalized methods to construct multi-output functions with high nonlinearity. When $n < m$, we produce the sufficient and necessary condition for existing multi-output Boolean function whose nonlinearity is nonzero and construct such functions using the knowledge of Reed-Muller codes. We also consider the problem about constructing multi-output bent functions, and produce generalized methods to construct multi-output $\mathcal{M}$ and $\mathcal{PS}$ class bent functions. Finally we construct highly nonlinear balanced functions using the results of multi-output bent functions.

The remaining part of this paper is organized as follows. In section 2 we mainly consider the constructions of highly nonlinear (n, m)-functions. we produce the sufficient and necessary condition for existing multi-output Boolean function whose nonlinearity is nonzero and construct such functions using the knowledge of Reed-Muller codes. In Section 3 we consider the generalized constructions multi-output bent functions. We talk about the constructions of $\mathcal{M}$ and $\mathcal{PS}$ class bent functions. In section 4 we produce some constructions of highly nonlinear balanced multi-output functions. We mainly use (n, m)-bent functions to construct such functions. Finally we end this paper with some conclusions in section 5.

2. Constructions of highly nonlinear multi-output functions when $n < m$

In this section we will mainly consider the construction of nonlinear multi-output functions when $n < m$ using some results of Reed-Muller code. Let $0 \leq r \leq n$. The *r-th Reed-Muller code* $RM(r, n)$ is the set of all binary strings of length 2^n associated with the algebraic normal form $f(x_1, \ldots, x_n)$ of degree at most r. Obviously, $A(n)$, the set of all affine functions, is $RM(1, n)$. More properties of Reed-Muller code can be found in [11], and in this paper we will use the following theorem.

Theorem 2.1. *The Reed-Muller code $RM(r, n)$ has minimum distance 2^{n-r}, and then has parameters $[2^n, 1 + \binom{n}{1} + \ldots + \binom{n}{r}, 2^{n-r}]$.*

Let $f(x)$ be a Boolean function with degree $r > 1$, then it is in $RM(r, n)$. Obviously $A(n) = RM(1, n) \subseteq RM(r, n)$, so for any affine function $a(x) \in A(x)$, the degree of $f(x) \oplus a(x)$ is still r and the distance between $f(x)$ and $a(x)$ must be not less than 2^{n-r} because of the minimum distance of $RM(r, n)$ being 2^{n-r}. From this fact we deduce that if the degree of $f(x)$ is $r > 1$, then the nonlinearity of $f(x)$ is at least 2^{n-r}.

The following lemma is well known [21] and therefore stated without proof.

Lemma 2.2. *Let $g(x), x \in F_2^n$ and $h(y), y \in F_2^m$ be two Boolean functions. Then the nonlinearity of $f(x, y) = g(x) \oplus h(y)$ is*

$$N_f = 2^m N_g + 2^n N_h - 2N_g N_h.$$

Not all the (n, m)-functions have nonzero nonlinearity. For $n = 1$, 2 and $m \geq 2$, it is not possible to get any nonlinearity. So at first we prove the following theorem about the existence of (n, m)-function with nonzero nonlinearity.

Theorem 2.3. *For two integers n, m, there exists an (n, m)-function whose nonlinearity is not zero if and only if $2^n - n - 1 \geq m$.*

Proof. Firstly we prove the sufficiency. The number of monomial $x_{i_1} x_{i_2} \ldots x_{i_s}$ whose degree is at least two is $2^n - n - 1$. Because $2^n - n - 1 \geq m$, we can build one (n, m)-function $F(x) = (f_1(x), \ldots, f_m(x))$ by arbitrarily choosing m different monomials as its component functions from these $2^n - n - 1$ monomials. For this function $F(x)$, the degree of nonzero linear combinations of its component functions is at most n and at least 2, the truth tables of these according functions are all in $RM(n, n)$. From the definition of nonlinearity and theorem 2.1, we deduce that the nonlinearity of $F(x)$ is at least 1.

Conversely, in order to make the nonlinearity of one (n, m)-function $F(x) = (f_1(x), \ldots, f_m(x))$ nonzero, from the definition of nonlinearity, these $2^m - 1$ nonzero linear combinations of component functions of $F(x)$ must be nonlinear functions and in each function the monomials whose degree is at least two must not be all the same with other functions, so we need $2^m - 1$ different nonzero linear combinations of monomials whose degree is at least two. For an n, the number of monomials whose degree is at least two is $2^n - n - 1$, and the number of nonzero linear combinations of these monomials is $2^{2^n - n - 1} - 1$. If $2^n - n - 1 < m$, then $2^{2^n - n - 1} - 1 < 2^m - 1$. At this time there are at least two nonzero linear combinations of component functions of $F(x)$ whose monomials with degree at least two must be identical, then at least one nonzero linear combination of component functions of $F(x)$ is an affine function, and the nonlinearity of $F(x)$ is zero. So if the nonlinearity of $F(x)$ is not zero, then we must have $2^n - n - 1 \geq m$. □

From theorem 2.1 and theorem 2.3, we deduce that the construction for (n, m)-functions with high nonlinearity is as follows: we take the different monomials whose degree is at least two and at most r as the component functions of $F(x)$. If the number of such monomials is not less than m, then we can always construct one function satisfying the $2^m - 1$ nonzero linear combinations of component functions of $F(x)$ and in each function the monomials whose degree is at least two and at most r are not all the same with other functions, and the degree of such functions is at most r. From the definition of nonlinearity and the properties of Reed-Muller code, the nonlinearity of $F(x)$ is at least 2^{n-r}.

From this construction, the larger n, the larger nonlinearity of function is constructed by this method. Especially, the number of monomials whose degree is at least two and at most $n-1$ is $2^n - n - 2$. If $2^n - n - 2 \geq m$, we can construct one (n, m)-function whose nonlinearity is at least 2, and if $2^n - 2n - 2$, i.e., the number of monomials whose degree is at least two and at most $n-2$, is larger than or equal to m, we can construct one (n, m)-function whose nonlinearity is at least 4. When $\binom{n}{2} \geq m$, we can get the following corollary.

Corollary 2.4. *For two integers n, m, if $\binom{n}{2} \geq m$, then there exists an (n, m)-function whose nonlinearity is at least 2^{n-2}.*

Proof. If $\binom{n}{2} \geq m$, then there exist at least m different monomials with degree 2. Let such monomials be the component functions of one (n, m)-function $F(x)$. Then the nonzero linear combinations of component functions are all with degree 2, so we have the nonlinearity of $F(x)$ is 2^{n-2} according to theorem 2.1. □

For example, when $n = 4$, $m = 6$, the monomials whose degree is two are

$$x_1x_2,\ x_1x_3,\ x_1x_4,\ x_2x_3,\ x_2x_4,\ x_3x_4.$$

We construct one $(4, 6)$-function

$$F(x) = (f_1(x), f_2(x), f_3(x), f_4(x), f_5(x), f_6(x))$$

satisfying

$$f_1(x) = x_1x_2, \quad f_2(x) = x_1x_3, \quad f_3(x) = x_1x_4,$$

$$f_4(x) = x_2x_3, \quad f_5(x) = x_2x_4, \quad f_6(x) = x_3x_4.$$

The nonlinearity of $F(x)$ is 4, which is the maximum nonlinearity of $(4,6)$-function.

Corollary 2.5. *For two integers n, m, if $\binom{n}{2} \geq 2m$, then there exists an (n,m)-function whose nonlinearity is at least $2^{n-1}-2^{n-3}$.*

Proof. If $\binom{n}{2} \geq 2m$, then there exist at least $2m$ different monomials with degree 2. We construct $F(x) = (f_1(x), f_2(x), \ldots, f_m(x))$ as follows: for $1 \leq i \leq m$, $f_i(x) = x_{i_1}x_{i_2} \oplus x_{i_3}x_{i_4}$, and $i_1, i_2, i_3, i_4 \in \{1, 2, \ldots, n\}$ are four different integers. Moreover, each monomial with degree 2 is within at most one component function, and the nonzero linear combinations of these component functions can't be written as the following form:

$$(x_{i_1} \oplus \cdots \oplus x_{i_s})(x_{j_1} \oplus \cdots \oplus x_{j_t}),$$

where $i_1, \ldots, i_s, j_1, \ldots, j_t$ are different integers. This condition can be satisfied if for arbitrary two component functions

$$f_i(x) = x_{i_1}x_{i_2} \oplus x_{i_3}x_{i_4}, f_j(x) = x_{j_1}x_{j_2} \oplus x_{j_3}x_{j_4}$$

the set $\{i_1, i_2, i_3, i_4\}$ is the not same as $\{j_1, j_2, j_3, j_4\}$.

According to [2], all Boolean functions with degree 2 are partially-bent functions. So all the nonzero linear combinations of these component functions have the form:

$$g(x) = x_1x_2 \oplus x_3x_4 \oplus h(x_5, \ldots, x_n)$$

after an affine nonsingular transformation. Obviously, $b(x) = x_1x_2 \oplus x_3x_4$ is bent function on F_2^4, so $N_b = 2^{4-1} - 2^{\frac{4}{2}-1} = 6$. According to lemma 2.2, the nonlinearity of $g(x)$ is

$$\begin{aligned} N_g &= 2^{n-4}N_b + 2^4N_h - 2N_bN_h \\ &= 2^{n-4}6 + (2^4 - 2 \times 6)N_h \\ &= 2^{n-1} - 2^{n-3} + 4N_h \\ &\geq 2^{n-1} - 2^{n-3}. \end{aligned}$$

So the nonlinearity of $F(x)$ at least $2^{n-1} - 2^{n-3}$. □

For example, when $n = 6$, $m = 7$, we can construct one $(6, 7)$-function whose component functions are:

$$\begin{array}{ll} f_1(x) = x_1x_2 \oplus x_3x_6, & f_2(x) = x_1x_3 \oplus x_2x_5, \\ f_3(x) = x_1x_4 \oplus x_2x_6, & f_4(x) = x_1x_5 \oplus x_3x_4, \\ f_5(x) = x_1x_6 \oplus x_3x_5, & f_6(x) = x_2x_3 \oplus x_4x_5, \\ f_7(x) = x_2x_4 \oplus x_5x_6. & \end{array}$$

Then the nonlinearity of $F(x) = (f_1(x), f_2(x), \ldots, f_7(x))$ is

$$2^{6-1} - 2^{6-3} = 24.$$

Using the same idea, we can construct more (n, m)-functions with high nonlinearity. In fact, the maximum nonlinearity of (n, m)-function for $3 \le n \le 8$, $1 \le m \le 8$ have been given in [23], but the authors have not considered the general results. The results in theorem 2.3 and its following corollaries can be viewed as the general results about the nonlinearity of (n, m)-functions when $n < m$. Furthermore, in [19] the authors mainly discussed how to construct resilient functions with very high nonlinearity using linear codes, where highly nonlinear (n, m)-functions with $n < m$ were also used. So the (n, m)-functions constructed in this paper can be used to improve the results in [19].

3. Constructions of Multi-output Bent Functions

For the construction of (n, m)-function with high nonlinearity when $n \ge m$, there are lots of results. In [16] Nyberg gave two examples of highly nonlinear permutations. We suppose that F_2^n is identified to the Galois field F_{2^n} [10].

Proposition 3.1. *Let $F(x) = x^{2^k+1}$ be a power polynomial on F_{2^n} with n odd and $(k, n) = 1$. Then $F(x)$ and its inverse have nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$.*

Proposition 3.2. *The permutation on F_{2^n}*

$$F(x) = \begin{cases} x^{-1} & x \neq 0, x \in F_{2^n} \\ 0 & x = 0 \end{cases}$$

has nonlinearity

$$N_F \ge 2^{n-1} - 2^{\frac{n}{2}}.$$

For constructing highly nonlinear (n,m)-functions when $n \geq m$, we can firstly construct one permutation on F_{2^n} using proposition 3.1 or 3.2 when n is odd or even respectively. After deleting some component functions we can get one (n,m)-function with the same nonlinearity as the corresponding permutations, which is at least $2^{n-1} - 2^{\lfloor \frac{n}{2} \rfloor}$.

Furthermore, when $n \geq 2m$ and n is even, we can construct (n,m)-functions with the highest nonlinearity, i.e., (n,m)-bent functions. Here is the following theorem in [17].

Theorem 3.3. *There exists an (n,m)-bent function if and only if $n \geq 2m$ and n is even.*

Bent functions were firstly defined in [20]. There are many good properties about bent functions, and they have many applications not only in cryptography, but also in algebraic coding theory, sequences and design theory etc. [1,14,16,18]. In the remaining part of this paper we will mainly study the constructions of (n,m)-bent functions.

3.1. Construction of $\mathcal{M}$ Class (n,m)-Bent Functions

One important class of bent functions: Maiorana-McFarland class [7,13], i.e., $\mathcal{M}$ class, is the set of all the Boolean functions on $F_2^n = \{(x,y), x, y \in F_2^p\}$, of the form:

$$f(x,y) = x \cdot \pi(y) \oplus g(y)$$

where π is any permutation on F_2^p and g any Boolean function on F_2^p and $n = 2p$. Let $F : F_2^n \to F_2^m$, $m \leq p$, be one (n,m)-function and denote the m output component functions of F by $f_1, f_2, \ldots, f_m$. Assume that every f_i is a function in $\mathcal{M}$ class,

$$f_i(x,y) = x \cdot \pi_i(y) \oplus g_i(y),$$

then $F = (f_1, f_2, \ldots, f_m)$ is m-output bent function if every nonzero linear combination of these permutations $\pi_i, i = 1, 2, \ldots, m$, is again a permutation on F_2^p. We call such (n,m)-bent functions a $\mathcal{M}$ *class* (n,m)*-bent functions* and call such permutations one class of *orthogonal permutations*. For constructing such functions, what we need to do is find families of permutations with the required property.

Now we will find such permutations. Let

$$p(x) = x^p \oplus a_{p-1}x^{p-1} \oplus \cdots \oplus a_1 x \oplus a_0 \in F_2[x]$$

be one *irreducible polynomial* with degree p and $S = (s_{ij})_{p\times p}, s_{ij} \in F_2$ be the companion matrix of $p(x)$ [10, 15], i.e.,

$$S = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & a_0 \\ 1 & 0 & 0 & \dots & 0 & a_1 \\ 0 & 1 & 0 & \dots & 0 & a_2 \\ & & & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 & a_{p-1} \end{bmatrix}.$$

According to matrix representation of the elements of a finite field [10,15], for any matrix $A = (a_{ij})_{p\times p}, a_{ij} \in F_2$ which is similar with S, then any nonzero linear combination of

$$A^0 = I, A, A^2, \dots, A^{p-1}$$

is still a full rank matrix, so let

$$\pi_i(x) = xA^{i-1}, \quad i = 1, 2, \dots, p,$$

then any nonzero linear combination of these permutations will still be one linear permutation, and such permutation satisfies the required property. $\{\pi_1(x), \dots, \pi_p(x)\}$ is a basis of orthogonal permutations.

Lemma 3.4. *Suppose that matrix $A_{p\times p}$ is similar with the companion matrix of one irreducible polynomial with degree p, then any nonzero linear combination of*

$$\pi_i(x) = xA^{i-1}, \quad i = 1, 2, \dots, p,$$

is a linear permutation.

Furthermore, if $\pi_1, \pi_2, \dots, \pi_p$ is one basis of one class of orthogonal permutations, then for any permutation H on F_2^n, $\pi_1 \circ H, \pi_2 \circ H, \dots, \pi_p \circ H$ is still a basis of one class of orthogonal permutations because for any not all zero $c_1, c_2, \dots, c_p \in F_2$

$$\begin{aligned} &c_1\pi_1 \circ H(x) \oplus c_2\pi_2 \circ H(x) \oplus \cdots \oplus c_p\pi_p \circ H(x) \\ &= (c_1\pi_1 \oplus c_2\pi_2 \oplus \cdots \oplus c_p\pi_p) \circ H(x) \end{aligned}$$

is still a permutation. Also for any linear permutation $L(x)$, $L \circ \pi_1, L \circ \pi_2, \ldots, L \circ \pi_p$ is still a basis of one class of orthogonal permutations because for any not all zero $a_1, a_2, \ldots, a_p \in F_2$

$$\begin{array}{rl} & c_1 L \circ \pi_1(x) \oplus c_2 L \circ \pi_2(x) \oplus \cdots \oplus c_p L \circ \pi_p(x) \\ = & L \circ (c_1\pi_1 \oplus c_2\pi_2 \oplus \cdots \oplus c_p\pi_p)(x) \end{array}$$

is still a permutation, so we get the following theorem.

Theorem 3.5. *Let $\pi_1, \pi_2, \ldots, \pi_p$ be one basis of one class orthogonal permutations. Let H be any permutation on F_2^p and L be any linear permutation on F_2^p. Then*

$$L \circ \pi_1 \circ H, L \circ \pi_2 \circ H, \ldots, L \circ \pi_p \circ H$$

is still one basis of one class of orthogonal permutations.

The above results also can be easily extended to general finite fields F_{q^n}. Also constructing orthogonal permutations is related to the problem of complete mappings and orthogonal Latin squares. More information about orthogonal Latin squares can be found in [6].

Here we talk about the algebraic degree of such bent functions. For the component function

$$f_i(x, y) = x \cdot \pi_i(y) \oplus g_i(y),$$

we suppose that $g_i(y) = 0$, then $\deg(f_i(x, y)) = \deg(\pi_i(y)) + 1$. If $\pi_i(y) = yA^{i-1}$, then it is a linear permutation, and $\deg(f_i) = 2$. To improve the degree of bent functions, we must use permutation $H(y)$ in theorem 3.5 with high degree. Here we provide one simple method for constructing permutations on F_2^p with high degree as follows: Let $H'(y_1, \ldots, y_{p-1}) = (h_1, \ldots, h_{p-1})$ be an arbitrary permutation on F_2^{p-1}. We construct one (p, p)-function $H(y_1, \ldots, y_{p-1}, y_p)$ using H' by adding one more component function

$$h_p = g(y_1, \ldots, y_{p-1}) \oplus y_p,$$

where $g(y_1, \ldots, y_{p-1})$ is an arbitrary Boolean function on F_2^{p-1} with degree $p-1$. Then the new function $H(y) = (h_1, \ldots, h_{p-1}, h_p)$ will be permutation on F_2^p with degree $p-1$. So the bent function

$$x \cdot (\pi_i \circ H)(y) \oplus g_i(y)$$

will have degree p, which is the maximum degree of bent functions. Especially, we can directly use permutation defined in proposition 3.2, which degree is also $p-1$.

In [22] the authors also talked about constructing (n,p)-bent functions. For a binary vector $(y_1,\dots,y_p)\in F_2^p$,define

$$dec(y_1,\dots,y_p)\overset{\triangle}{=}2^{p-1}y_1+2^{p-2}y_2+\cdots+y_p.$$

For an element $\beta\in F_{2^p}$, let $[\beta]$ denote a vector representation of β. Let α be one primitive element of F_{2^p}. Define one (n,p)-function $F(x,y)=(f_1,\dots,f_p)$ such that

$$f_i(x,y)=x\cdot\varphi_i(y)\oplus g_i(y)$$

where

$$\varphi_i(y)\overset{\triangle}{=}\begin{cases}0 & \text{if } \ y=(0,\dots,0),\\ \alpha^{dec(y)+i-1} & \text{otherwise}\end{cases}$$

and g_i is any Boolean function. Such (n,p)-function are $\mathcal{M}$ class bent functions with degree p. In fact, if $y\neq(0,\dots,0)$, then permutation

$$\varphi_i(y)=\alpha^{i-1}\alpha^{dec(y)}=\pi_i\circ\alpha^{dec(y)},$$

$\alpha^{dec(y)}$ can be viewed as the permutation $H(y)$ in theorem 3.5. We deduce that the constructing method in [22] is just one special case of theorem 3.5.

3.2. Construction of $\mathcal{PS}$ Class (n,m)-Bent Functions

Another important class of bent functions is partial spread class [7], i.e., $\mathcal{PS}$ class bent functions. The *indicator function* of one subspace E of F_2^n is defined as

$$I_E(x)=\begin{cases}1 & x\in E,\\ 0 & x\notin E.\end{cases}$$

The class $\mathcal{PS}$ bent functions is the set of all the sums in F_2 of the indicators of 2^{p-1} or $2^{p-1}+1$ "disjoint" p-dimensional subspaces of F_2^n("disjoint" means that the only common element of any two subspaces is $\mathbf{0}$, then the direct sum of any two such subspaces is F_2^n). Dillon denotes by $\mathcal{PS}^-$ (resp. $\mathcal{PS}^+$) the class of those functions for which the number of p-dimensional subspaces is 2^{p-1} (resp. $2^{p-1}+1$).

Let $F : F_2^n \to F_2^m$, $m \leq p$, be one (n, m)-function and denote the m output component functions of F by $f_1, f_2, \ldots, f_m$. If every nonzero linear combination of these component functions is $\mathcal{PS}$ bent function, then we call such (n, m)-bent functions $\mathcal{PS}$ *class* (n, m)*-bent functions*.

$\mathcal{PS}$ bent functions is one interesting class of bent functions having good algebraic structure and is useful for studying the general structure of bent functions [3]. The dual of any $\mathcal{PS}$ bent function is exactly the sum in F_2 of the indicator functions of the corresponding dual subspaces. The elements of $\mathcal{PS}^-$ have degree p exactly, but not those of $\mathcal{PS}^+$ which contain for instance all the quadratic bent functions if p is even. It is an open problem to characterize the algebraic normal forms of the elements of class $\mathcal{PS}$.

The key to construct $\mathcal{PS}$ bent functions is to divide F_2^n into $2^p + 1$ disjoint p-dimensional subspaces. In [4,5] the authors considered the division of F_2^n and construction of $\mathcal{PS}$ bent functions. Suppose we have divided F_2^n into $2^p + 1$ disjoint p-dimensional subspaces, say, $E_0, E_1, \ldots, E_{2^p}$. For $0 \leq i \leq 2^p$, we have

$$E_i = \{\ \overline{x}A_i \mid \overline{x} \in F_2^p\ \},$$

where A_i is one $p \times n$ matrix on F_2 satisfying $rank(A_i) = p$. We can call A_i as the generating matrix of E_i. Because these subspaces are disjoint, then for arbitrary $i \neq j$, $\begin{pmatrix} A_i \\ A_j \end{pmatrix}$ is an invertible $n \times n$ matrix. For any invertible matrix $M_{n\times n}$,

$$\begin{pmatrix} A_i \\ A_j \end{pmatrix} M = \begin{pmatrix} A_iM \\ A_jM \end{pmatrix}$$

is still an invertible $n \times n$ matrix, A_iM and A_jM are still $p \times n$ matrices with *rank* p. So the subspaces

$$E_iM = \{\ \overline{x}A_iM \mid \overline{x} \in F_2^p\ \}, \quad 0 \leq i \leq 2^p$$

is still one division of F_2^n, and we can get the following lemma.

Lemma 3.6. *Under one linear transformation, a division of F_2^n is still a division of F_2^n. Furthermore, a $\mathcal{PS}$ bent function is still a $\mathcal{PS}$ bent function under one linear transformation.*

From this lemma we can choose one special matrix

$$M = \begin{pmatrix} A_0 \\ A_1 \end{pmatrix}^{-1},$$

then

$$A_0M = (I_p, \mathbf{0}), \quad A_1M = (\mathbf{0}, I_p),$$

where I_p is $p \times p$ identity matrix and $\mathbf{0}$ is $p \times p$ zero matrix. We denote the new subspaces and matrices E'_i and A'_i respectively, where $A'_0 = (I_p, \mathbf{0})$,$A'_1 = (\mathbf{0}, I_p)$. For $2 \le i \le 2^p$, $A'_i = (S_{i1}, S_{i2})$ must satisfies that

$$D = \begin{pmatrix} A'_0 \\ A'_i \end{pmatrix} = \begin{pmatrix} I_p & \mathbf{0} \\ S_{i1} & S_{i2} \end{pmatrix}$$

is an invertible matrix, so

$$|D| = |I_p| \times |S_{i2}| = |S_{i2}| \bmod 2$$

is nonzero. So S_{i2} is an invertible matrix, and similarly S_{i1} is also invertible matrix according to A'_1.

From the above result, we can suppose that for $2 \le i \le 2^p$, A'_i, has the form (I_p, B_i), where B_i is an $p \times p$ invertible matrix, and because E'_i and E'_j is disjoint for any $i \neq j$, then

$$\begin{vmatrix} I_p & B_i \\ I_p & B_j \end{vmatrix} = \begin{vmatrix} I_p & B_i \\ \mathbf{0} & B_i + B_j \end{vmatrix} = |B_i + B_j|,$$

so $B_i + B_j$ is also invertible matrix for $i \neq j$. We call such two matrices as *orthogonal matrices.* Under one linear transformation $M = \begin{pmatrix} I_p & \mathbf{0} \\ \mathbf{0} & B_2^{-1} \end{pmatrix}$, then $(I_p, B_2)M = (I_p, I_p)$. According the above results, we get the following theorem [5].

Theorem 3.7. *Let $E_0, E_1 \ldots, E_{2^p}$ be any division of F_2^n, and $A_0, A_1 \ldots, A_{2^p}$ be the generating matrices respectively. After one linear transformation, these matrices can have the following properties:*

1: *$A_0 = (I_p, \mathbf{0})$, $A_1 = (\mathbf{0}, I_p)$, $A_2 = (I_p, I_p)$;*

2: *for $3 \le i \le 2^p$, $A_i = (I_p, B_i)$, where B_i is an $p \times p$ invertible matrix;*

3: *$B_2 = I_p, B_3, \ldots, B_{2^p}$ is one class of orthogonal matrices, i.e., $B_i + B_j$ is still one invertible matrix for $i \neq j$.*

According to this theorem, we choose the matrix A defined in lemma 3.4 and let

$$B_i = A^{i-2},\ i = 2, 3, \ldots, p+1,$$

then the nonzero linear combinations of such matrices will be $2^p - 1$ orthogonal matrices, then we can get one division of F_2^n. Such a division is still one division after one linear transformation.

Suppose that $E_0, E_1 \ldots, E_{2^p}$ is one division of F_2^n, we use them to construct $\mathcal{PS}$ class (n,p)-bent functions. For one vector $x = (x_1, \ldots, x_p) \in F_2^p$, we define a one-to-one mapping $\varphi(x)$

$$\varphi(x) = x_1 2^{p-1} + x_2 2^{p-2} + \cdots + x_p.$$

Obviously $\varphi(x) \in \{0, 1, \ldots, 2^p - 1\}$. For arbitrary permutation function $H(x) = (h_1(x), \ldots, h_p(x))$ on F_2^p, we define subset S_i, $i = 1, \ldots, p$, of $\{0, 1, \ldots, 2^p - 1\}$ as follows:

$$S_i = \{\ \varphi(\alpha) \mid h_i(\alpha) = 1,\ \alpha \in F_2^p\ \}.$$

Obviously the size of S_i is $|S_i| = 2^{p-1}$. For such subsets, we define one (n,p)-function $F(x) = (f_1(x), \ldots, f_p(x))$ such that

$$f_i(x) = \sum_{j \in S_i} I_{E_j}(x) \oplus c_i I_{E_{2^p}}(x) \mod 2, \quad c_i \in F_2, \quad i = 1, \ldots, p.$$

The nonzero linear combination of such functions $b_1 f_1(x) \oplus \cdots \oplus b_p f_p(x)$ has the following form

$$\sum_{j \in S'} I_{E_j}(x) \oplus c_i' I_{E_{2^p}}(x) \mod 2, \quad c_i' \in F_2,$$

where

$$S' = \{\ \varphi(\alpha) \mid (b_1 h_1 \oplus \cdots \oplus b_p h_p)(\alpha) = 1,\ \alpha \in F_2^p\ \}$$

satisfying $|S'| = 2^{p-1}$ according to the property of permutation, so it is the sum (modulo 2) of the indicators of 2^{p-1} or $2^{p-1} + 1$ disjoint p-dimensional subspaces of F_2^n. Such function is $\mathcal{PS}^-$ or $\mathcal{PS}^+$ bent function and $F(x) = (f_1(x), \ldots, f_p(x))$ is $\mathcal{PS}$ class (n,p)-bent function.

In the above construction, there are many ways to choose that one-to-one mapping $\varphi(x)$. Generally, Let $\varphi(x)$ be one-to-one mapping from F_2^p to $\{0, 1, \ldots, 2^p\}\backslash\{s\}$, where $s \in \{0, 1, \ldots, 2^p\}$. For example, let

$$\varphi(x) = x_1 2^{p-1} + x_2 2^{p-2} + \cdots + x_p + 1,$$

then it is one-to-one mapping from F_2^p to $\{1, 2, \ldots, 2^p\}$. We can produce the general form of $\mathcal{PS}$ class (n, m)-bent functions according to such one-to-one mappings.

Theorem 3.8. *Let $E_0, E_1, E_2, \ldots, E_{2^p}$ be one division of F_2^n. Let $\varphi(x)$ be one-to-one mapping from F_2^p to $\{0, 1, \ldots, 2^p\}\backslash\{s\}$, $s \in \{0, 1, \ldots, 2^p\}$ and $H(x) = (h_1(x), \ldots, h_m(x))$ be an arbitrary balanced (p, m)-function. Define*

$$S_i = \{\varphi(\alpha)|h_i(\alpha) = 1, \alpha \in F_2^p\} \subseteq \{0, 1, \ldots, 2^p\}\backslash\{s\},\ i = 1, \ldots, m.$$

At this time $F(x) = (f_1(x), \ldots, f_m(x))$, where

$$f_i(x) = \sum_{j \in S_i} I_{E_j}(x) \oplus c_i I_{E_s}(x) \mod 2, \quad c_i \in F_2, \quad i = 1, \ldots, m,$$

is $\mathcal{PS}$ class (n, m)-bent function.

In [24] the authors produced the definition of *hyper-bent functions*. One function $f(x)$ from F_{2^n} to F_2 is hyper bent if and only if for any positive integer c satisfying $(c, 2^n - 1) = 1$ and any $\lambda \in F_{2^n}$,

$$\sum_{x \in F_{2^n}} (-1)^{f(x)+Tr(\lambda x^c)} = \pm 2^{\frac{n}{2}}.$$

Obviously, $f(x)$ is a hyper-bent function if and only if for any c such that $(c, 2^n - 1) = 1$, $f(x^c)$ is still bent function. It is not hard to prove that bent functions in one subclass of $\mathcal{PS}$ bent functions: $\mathcal{PS}_{ap}$ [7], are hyper-bent functions. So the method of constructing $\mathcal{PS}$ Class (n, m)-bent functions can be used to construct multi-output hyper-bent functions.

4. Construction of Highly Nonlinear Balanced (n, m)-Functions

In this section we discuss the problem of constructing balanced (n, m)-functions with high nonlinearity using (n, m)-bent functions.

Dobbertin, in [9] produces one method to construct single-output Boolean balanced functions with high nonlinearity. The author shows that, if a bent function is constant on a $\frac{n}{2}$, i.e., p-dimensional subspace of F_2^n, then it is possible to deduce a highly nonlinear balanced Boolean function. Furthermore, the bent function being chosen must be normal because such p-dimensional subspace exists only if the bent function is normal(more information about normal bent functions can can be found in [8]). Let $f(x)$ on F_2^n be a normal bent function. If $wt(f) = 2^{n-1} - 2^{\frac{n}{2}-1}$, then we choose one p-dimensional subspace E such that $f(x) = 0$ when $x \in E$, and if $wt(f) = 2^{n-1} + 2^{\frac{n}{2}-1}$ we choose one p-dimensional subspace E such that $f(x) = 1$ when $x \in E$. Let g be any balanced function on E. Then the Boolean function $f' = f + g$ whose value at any $x \in E$ is $g(x)$ and whose value at any $x \in F_2^n \backslash E$ is $f(x)$ is a balanced function. Denote by $N_n(f')$ the nonlinearity of f' and by $N_{\frac{n}{2}}(g)$ the nonlinearity of g, we have:

$$N_n(f') \geq 2^{n-1} - 2^{\frac{n}{2}} + N_{\frac{n}{2}}(g).$$

Using the same idea, we can construct highly nonlinear balanced (n, m)-functions. Let $F(x) = (f_1, \ldots, f_m)$ be (n, m)-bent function, whose component functions are all constant on a $\frac{n}{2}$-dimensional subspace E. Let $H = (h_1, \ldots, h_m)$ be one balanced $(\frac{n}{2}, m)$-function on E with nonlinearity N_H, which can be provided by proposition 3.1 or 3.2. For new (n, m)-function $F' = F + H$, any nonzero linear combination of its components is nonzero linear combination of components of F plus nonzero linear combination of components of H, which is balanced and with nonlinearity at least $2^{n-1} - 2^{\frac{n}{2}} + N_H$, so $F' = F + H$ is a balanced (n, m)-function satisfying

$$N_F \geq 2^{n-1} - 2^{\frac{n}{2}} + N_H.$$

According to the above method and the construction of (n, m)-bent functions in section 3($\mathcal{M}$ bent functions and $\mathcal{PS}$ bent functions are all normal), we get the following corollaries whose proofs are omitted.

Corollary 4.1. (For $\mathcal{M}$ class) *Let $F(x, y) = (f_1(x, y), ., f_m(x, y))$ be one $\mathcal{M}$ class (n, m)-bent function on F_2^n. For $i = 1, \dots, m$,*

$$f_i(x, y) = x \cdot \pi_i(y) \oplus g_i(y)$$

and $\pi_i(\mathbf{0}) = 0$. Let $H(x) = (h_1(x), \dots, h_m(x))$ be one balanced $(\frac{n}{2}, m)$-function with nonlinearity N_H. Define one new (n, m)-function $F'(x, y) = (f'_1(x, y), \dots, f'_m(x, y))$ satisfying that for $i = 1, \dots, m$,

$$f'_i(x, y) = \begin{cases} h_i(x) & y = \mathbf{0}, \\ f_i(x, y) & y \neq \mathbf{0}. \end{cases}$$

Then $F'(x, y)$ is one balanced (n, m)-function and its nonlinearity

$$N_{F'} \geq 2^{n-1} - 2^{\frac{n}{2}} + N_H.$$

Corollary 4.2. (For $\mathcal{PS}$ class) *Let $p = \frac{n}{2}$. Let $E_0, E_1, E_2, \dots, E_{2^p}$ be one division of F_2^n and $H = (h_1, \dots, h_m)$ be one balanced (p, m)-function on some E_s with nonlinearity N_H, where $s \in \{0, 1, .., 2^p\}$. Let φ be one-to-one mapping from F_2^p to $\{0, 1, \dots, 2^p\} \backslash \{s\}$ and $G = (g_1, \dots, g_m)$ be an arbitrary balanced (p, m)-function. Define subset S_i, $i = 1, \dots, m$, as follows:*

$$S_i = \{ \varphi(\alpha) \mid g_i(\alpha) = 1, \ \alpha \in F_2^p \}.$$

Then the new (n, m)-function $F'(x) = (f'_1(x), \dots, f'_m(x))$ satisfying that for $i = 1, \dots, m$,

$$f'_i(x) = \begin{cases} h_i(x) & x \in E_s, \\ \sum_{j \in S_i} I_{E_j}(x) & \textit{otherwise}. \end{cases}$$

is balanced and its nonlinearity is

$$N_{F'} \geq 2^{n-1} - 2^{\frac{n}{2}} + N_H.$$

Using the above results, we can construct balanced multi-output function with fairly high nonlinearity. For example, when $n = 8$, we can construct one $\mathcal{M}$ or $\mathcal{PS}$ class $(8, 4)$-bent function and one $(4, 4)$-permutation with nonlinearity 4 according to proposition 3.2, then using corollary 4.1 or 4.2 we can construct one balanced $(8, 4)$-function with nonlinearity

$$2^7 - 2^4 + 4 = 116,$$

which is the best known nonlinearity for balanced functions with 8 input variables [9,21].

5. Conclusions

In this paper we mainly present some generalized constructions of (n, m)-functions with high nonlinearity. We give sufficient and necessary conditions for existing (n, m)-function whose nonlinearity is nonzero and general construction of (n, m)-functions when $n < m$. We also provide generalized constructions of multi-output $\mathcal{M}$ and $\mathcal{PS}$ class bent functions and balanced highly nonlinear multi-output functions. Especially, the constructions of bent functions can be used to construct generalized bent functions on finite fields. These results are useful in designing secure secret key cryptosystems as well as random number generators, they also can be used to construct other functions with good cryptographical properties.

References

[1] E. F. Assmus, J. D. Key, *Designs and their Codes*, Cambridge University Press, 1992.

[2] C. Carlet, “Partially-bent functions”, *Designs, Codes and Cryptography*, vol. 3, 1993, pp. 135-145.

[3] C. Carlet, “Generalized partial spreads”, *IEEE Transactions on Information Theory*, vol. 41, no. 5, 1995, pp. 1482-1487.

[4] Z. L. Chang, L. S. Chen, and F. W. Fu, “One method for constructing bent functions of class $\mathcal{PS}$”, *ACTA Eletronica Sinica*, vol. 32, no. 10, 2004, pp. 1649-1653.(in chinese)

[5] Z. L. Chang, F. W. Fu, and Q. Y. Wen, “On Division of F_2^n and construction of $\mathcal{PS}$ bent functions”, preprint.

[6] J. Denes, A. D. Keedwell, *Latin squares and their applications*. The English Universities Press Ltd, London, 1974.

[7] J. F. Dillon, *Elementary hadamard difference sets*, Ph. D. Dissertation, University of Maryland, 1974.

[8] J. F. Dillon, H. Dobbertin, "New cyclic difference sets with Singer parameters", *Finite Fields their Applic.*, 2004, pp. 342-389.

[9] H. Dobbertin, "Construction of bent functions and balanced Boolean functions with high nonlinearity", *Proceedings of the 1994 Leuven Workshop on Cryptographic Algorithms*(Lecture Notes in Computer Science, vol. 1008). Berlin, Germay: Springer-Verlag, 1995, pp. 61-74.

[10] R. Lidl, H. Niederreiter, *Finite Fields.* Encyclopedia of Mathematics and Its Applications, vol. 20, Addison-Wesley Publishing Company, 1983.

[11] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error Correcting Codes*, Amsterdam: North Holland, 1977.

[12] M. Matsui, "Linear cryptanalysis method for DES cipher", in *Advances in Cryptology-EUROCRYPT'93* (Lecture Notes in Computer Science, vol. 765). Berlin, Germay: Springer-Verlag, 1994, pp. 386-397.

[13] R. L. McFarland, "A family of noncyclic difference sets", *Journal of Combinatorics Theory*, Series A, vol. 15, 1973, pp. 1-10.

[14] W. Meier, O. Staffelbach, "Nonlinearity criteria for Cryptographic functions", in *Advances in Cryptology-EUROCRYPT'89*(Lecture Notes in Computer Science, vol. 434). Berlin, Germay: Springer-Verlag, 1992, pp. 549-562.

[15] Nathan Jacobson, *Lecture in Abstract Algebra II. Linear Algebra*, GTM 31, Springer-Verlag, 1953.

[16] K. Nyberg, "Differentially uniform mappings for cryptography", In *Advances in Cryptology — EUROCRYPT'93 (Lecture Notes in Computer Science)*, Berlin: Springer-Verlag, 1994, vol. 765, pp. 55-64.

[17] K. Nyberg, "Constructions of bent functions and difference sets", In *Advances in Cryptology — EUROCRYPT'90 (Lecture Notes in Computer Science).* Berlin: Springer-Verlag, 1991, vol. 473, pp. 155-160.

[18] J. D. Olsen, R. A. Scholtz, L. R. Welch, "Bent function sequence", *IEEE Transactions on Information Theory*, vol. 28, no. 6, 1982, pp. 858-864.

[19] E. Pasalic, S. Maitra, "Linear codes in generalized construction of resilient functions with very high nonlinearity", *IEEE Transactions on Information Theory*, vol. 48, no. 8, 2002, pp. 2182-2191.

[20] O. S. Rothaus, "On bent functions", *Journal of Combinatorics Theory*, Series A, vol. 20, 1976, pp. 300-305.

[21] P. Sarkar, S. Maitra, "Nonlinearity bounds and constructions of resilient Boolean functions ", *Proceedings of CRYPTO'2000* (Lecture Notes in Computer Science), Berlin: Springer-Verlag, 2000, vol. 1880, pp. 515-532.

[22] T. Satoh, K. Kurosawa, "On cryptographically secure vectorial Booolean functions ", In *Advances in Cryptology— ASIACRYPT'99 (Lecture Notes in Computer Science)*, Berlin: Springer-Verlag, 1999, vol. 1716, pp. 20-28.

[23] T. Wadayama, T. Hada, K. Wakasugi, and M. Kasahara, "Upper and lower bounds on maximum nonlinearity of n-input m-output Boolean function", *Designs, Codes and Cryptography*, vol. 23, 2001, pp. 23-33.

[24] A. M. Youssef, G. Gong, "Hyper-bent functions", in *Advances in Cryptology-EUROCRYPT'2001*(Lecture Notes in Computer Science, vol. 2045). Berlin, Germay: Springer-Verlag, 2001, pp. 406-419.

This work is supported by the National Natural Science Foundation of China(60373059), the National Research Foundation for the Doctoral Program of Higher Education of China(20040013007) and the research foundation of the State Key Laboratory of Information Security.

EXPONENTIAL SUMS AND BOOLEAN FUNCTIONS

Julien Bringer[1] and Valérie Gillot, Philippe Langevin[2]

Abstract. We study the nonlinearity of Boolean functions constructed by means of a subgroup of the multiplicative group of a finite field. The functions that we consider are constant over the non trivial cosets of a subgroup of small index. Classical properties of Gauss sums lead us to propose a new conjecture of the Patterson-Wiedemann type. One of the major steps of this approach consists in finding good estimations of exponential sums restricted over subgroup.

1. **Nonlinearity**

All along the paper, L denotes a finite extension of degree m of $\mathbf{F}_2$ the field of order two. The canonical additive character of L is denoted by μ. It is defined by means of the abolute trace of L over $\mathbf{F}_2$ by $\mu(x) = (-1)^{\mathrm{Tr}_L(x)}$. The *Fourier coefficient* of a complex mapping f is defined, at $a \in L$, by

$$\widehat{f}(a) = \sum_{x\in L} f(x)\mu(ax). \tag{1}$$

We denote by $R(f) := \sup_{a\in L} |\widehat{f}(a)|$ the *spectral amplitude* of f. One of the most exciting challenge at the intersection of the coding theory and cryptography consists in finding the minimal spectral amplitude that can achieve a binary function i.e. a mapping from L into ±1. For a such function, the Parseval relation says that $R(f)$ is greater than or equal to $\sqrt{2^m}$. This fact splits the problem

[1] SAGEM Défense Sécurité SA. Avenue du Gros Chêne, 95610 Eragny-sur-Oise, France. email: `julien.bringer@sagem.com`
[2] GRIM, USTV. Bat. U, B.P. 20132. 83957 La Garde, France. email: `{gillot,langevin}@univ-tln.fr`

in two cases according to the parity of m. In the case when m is even, there exists *bent functions* of spectral amplitude $\sqrt{2^m}$ and that is the best that we can do. The main questions are : how to construct bent functions, how to classify or merely how to count them. In the case when m is odd, the exact value of $R_m = \inf_f R(f)$ is not known, and the famous conjecture of Patterson-Wiedemann [6] claims the asymptotic behavior:

$$R_m \sim \sqrt{2^m}. \tag{2}$$

Now, let G be the subgroup of $L^\times$ of index v. We ask similar questions. What is the maximal value, say $R^v(f)$, of the character sums

$$\tilde{f}(a) = \sum_{x \in G} f(x)\mu(ax)?$$

The minimal value, say R^v_m of the $R^v(f)$'s when f ranges the set of binary functions is called the *spectral radius of index* v, in this paper we study theses numbers. The main goal of the present contribution is to exhibit examples of groups with small index so that R^v_m is rather small. For one thing that could seem artificial but recent works of Bringer, summarized in the next section, show links with the Patterson-Wiedemann conjecture. In section (5), we recall the basic notion over exponential sums that we apply to construct our examples.

2. **Bringer construction**

Let G be a subgroup of index v of $L^\times$ and let Ω be the quotient group $L^\times/G$. Let s be a balanced mapping defined over Ω such that $s(\omega) = \pm 1$ for all $\omega \neq 1$, $s(1) = 0$, and $\sum_{\omega \in \Omega} s(\omega) = 0$. We consider the binary function

$$h(x) = f(x)g(x) + \sum_{1 \neq \omega \in \Omega} s(\omega) g(x/\omega) \tag{3}$$

where f is a binary function, and where g is the indicating function of G i.e. $g(x) = \begin{cases} 1, & x \in G; \\ 0, & x \notin G. \end{cases}$. In this paper, we will say that the binary function h is a configuration of index v by the sequence s and the section f, briefly a (v, s, f)-configuration. The function h

is constant over all cosets of G except over G itself. As in [4], we write the Fourier coefficient of g at a by means of Gauss sums

$$\hat{g}(a) = \frac{1}{v} \sum_{\chi \perp G} \tau_L(\chi)\bar{\chi}(a). \tag{4}$$

See [5], for generality on Gauss sums. Hence

$$\hat{h}(a) = \frac{1}{v} \sum_{\chi \perp G} \tau_L(\chi)s(\chi)\bar{\chi}(a) + \tilde{f}(a). \tag{5}$$

where $s(\chi) = \sum_{\omega\in\Omega} s(\omega)\bar{\chi}(\omega)$. Note this last sum is nothing but the multiplicative Fourier coefficient of s considered as a mapping from the group G into $\{-1, 0, +1\}$. For $\chi \neq 1$, let us set $\tau_L(\chi) = \upsilon(\chi)\sqrt{q}$, note that $|\upsilon(\chi)| = 1$. Since s is balanced, we have

$$\hat{h}(a) = \frac{\sqrt{q}}{v} \sum_{1\neq\chi\perp G} \upsilon(\chi)s(\chi)\bar{\chi}(a) + \tilde{f}(a). \tag{6}$$

The last expression allows us to guess sufficient conditions in order to construct a configuration with a small spectral amplitude. For example, if the $\upsilon(\chi)$'s are closed to 1, for all the non trivial χ, then thanks to orthogonality relations, the previous equation becomes $\hat{h}(a) \sim s(\omega)\sqrt{q} + \tilde{f}(a)$, where $a \in \omega \in \Omega$. And so, if the second term is negligeable compared to $\sqrt{q}$, then h would have a spectral amplitude near $\sqrt{q}$. This kind of construction would be helpfull to confirm the Patterson and Wiedemann conjecture. The hypothesis of the example can be achieve in some special case (e.g. for some values of m or for m growing to infinity). A main problem is how small the second term can be.

This is a more general problem than the conjecture of Patterson and Wiedemann, but it is interesting to notice that, if we want to find functions with high non-linearity over L in such a way, we do not have to be very tight over G.

These are the reasons why, as we said in the introduction, we focus our interest on the last point and we try to understand the behaviour of R^v_m. First, note that the Parseval relation, as in the all space case, gives us a lower bound :

$$\sum_{a\in L} \tilde{f}(a)^2 = 2^m \frac{2^m - 1}{v} \Longrightarrow R^v(f) \geq \sqrt{\frac{2^m - 1}{v}}. \tag{7}$$

Again, the question is how far to this lower bound are we ? By analogy with the all-space case, and due to numerical results, we guess that the Patterson-Wiedemann conjecture would become :

Conjecture 2.1. Let v be an odd integer. For a large integer m such that $v \mid (2^m - 1)$:

$$R_m^v \sim \sqrt{\frac{2^m}{v}}$$

3. Quadratic residue construction

In this section, we present a nice configuration involving quadratic residue that gives a higly nonlinear Boolean function of 15 variables constant on the group of index 7 of $\mathbf{F}^{\times}{}_{2^{15}}$.

Let $v > 3$ be a prime congruent to 3 modulo 4 such that 2 generates the group of quadratic residues modulo v. In the terminology of [3], the pair $(v, 2)$ satisfies the *quadratic residue conditions.* Let χ be a multiplicative character of order v. There exist integers t, A and B such that :

$$\tau_L(\chi) = 2^t(A + B\sqrt{-v}), \quad 2 \not| AB;$$

where t is deeply connected to both Stickelberger theorem and the class number of the quadratic field $\mathbf{Q}(\sqrt{-l})$. For all $0 \leq j < v$,

$$\tau_L(\chi^j) = 2^t\big(A + \left(\frac{j}{v}\right)B\sqrt{-v}\big)$$

Let γ be primitive root of L. We assume that $\chi(\gamma)$ is equal to ζ_v the principal root of order v. The elements γ^0, γ^1,..., γ^{v-1} forms a system of representatives of Ω. We define the *quadratic residue spread* by

$$h(x) = \sum_{j=1}^{v-1} \left(\frac{j}{v}\right) g(\gamma^{-j}x).$$

It is a balanced function, $\hat{h}(0) = 0$ and the other Fourier coefficients are given by means of the Legendre symbole

$$\hat{h}(\gamma^k) = 2^t \times \big(\left(\frac{k}{v}\right)A - B + vB\delta_0(k)\big) \tag{8}$$

where $\delta_0(k) = 1$ or 0 according to whether $k = 0$ or not. Indeed, from Gauss we know $\sum_{j=0}^{v-1} \left(\frac{j}{v}\right)\zeta_v^{ks} = \left(\frac{s}{v}\right)\sqrt{-v}$. In particular, $s(\chi^j) = \sum_{i=0}^{v-1} \left(\frac{i}{v}\right)\bar{\chi}(\gamma^{ij}) = -\left(\frac{j}{v}\right)$. The remainder is a straightforward calculation:

$$\begin{aligned}
v\hat{h}(\gamma^k) &= \sum_{j=1}^{v-1} \tau_L(\chi^j) s(\bar{\chi}^j)\chi^j(\gamma^k) = -\sum_{j=1}^{v-1} \tau_L(\chi^j)[\left(\frac{j}{v}\right)\sqrt{-v}]\zeta^{kj} \\
&= -2^t \sum_{j=1}^{v-1} [A\left(\frac{j}{v}\right)\sqrt{-v} - Bv]\zeta^{kj} \\
&= -2^t A\sqrt{-v} \sum_{j=1}^{v-1} \left(\frac{j}{v}\right)\zeta^{kj} + 2^t Bv \sum_{j=1}^{v-1} \zeta^{kj} \\
&= 2^t Av\left(\frac{k}{v}\right) + 2^t Bv \sum_{j=1}^{v-1} \zeta^{kj}.
\end{aligned}$$

Let χ be a multiplicative character of order 7 in $\mathbf{F}_{2^{15}}$. We can realize χ as the lift of a non trivial multiplicative character χ' of $\mathbf{F}_8$, so that

$$\tau_{\mathbf{F}_{2^{15}}}(\chi) = (\tau_{\mathbf{F}_8}(\chi'))^5 = (-1 + \sqrt{-7})^5 = -16(11 + \sqrt{-7})$$

i.e. $A = 11$ and $B = 1$, whence the Fourier transform of the quadratic spread takes the values -160, -96 and 192.

By an exhaustive computer search among the monomial x^s, we have found that the binary function

$$h(x^s) = \mu(x^{755})g(x) + \sum_{j=1}^{v-1} \left(\frac{j}{v}\right) g(\gamma^{-j}x).$$

has spectral amplitude 232 when $s = 755$. The spectrum of the function is detailed in TABLE 1. We believe it is possible to obtain such good nonlinearity for all the instances $m = 3r$ for which the Gauss sums lie within a narrow angular sector. It is the case for $m = 15$. According to the table TABLE 2 below, the best situation for that point of view seems $r = 13$ i.e. for dimension 39.

value	-216	-152	-88	-24	40	104	168	232
multiplicity	7550	6494	1208	3020	755	151	6795	6795

TABLE 1. Spectrum of the quadratic residue spread $h(x) = \mu(x^{755})g(x) + \sum_{j=1}^{v-1} \left(\frac{j}{v}\right) g(x/\gamma^j)$.

r	13	26	39	52	65	78	91	96	83	70	57	31	44	18	5
Δ	1	3	5	7	9	11	11	15	17	19	21	23	13	25	27

TABLE 2. arguments of the Gauss sums for the group of index 7 in an extension of degree r of $\mathbf{F}_8$ which lies in a sector of Δ degree.

4. Asymptotic Bound

Asymptotically, it is known [7] that almost all boolean functions have high non linearities, and so that they have low spectral amplitudes. For binary functions over a subgroup G of $L^\times$, we show here that this phenomenon is always true.

First, let us recall known bounds on sums of binomial coefficients.

Lemma 4.1. *Let N be any positive integer and $0 < \lambda < 1/2$. Then*

$$\frac{2^{NH_2(\lambda)}}{\sqrt{8N\lambda(1-\lambda)}} \leq \sum_{0 \leq i \leq \lambda N} \binom{N}{i} \leq 2^{NH_2(\lambda)} < 2^N e^{-2N(1/2-\lambda)^2}$$

where $H_2(x) = -x\log_2(x) - (1-x)\log_2(1-x)$ is the entropy function.

This lemma implies the following result :

Theorem 4.2. *Let $m > 0$ be an integer, G a subgroup of $L^\times$ and N, v the order and the index of G. Let c be any strictly positive real number such that $N > 2c^2m$. Then, the density of the set $\{f : G \to \{\pm 1\}, R^v(f) \leq c\sqrt{2Nm}\}$ is greater than $1 - 2^{m(1-c^2\log_2(e))}$.*

If $c^2\log_2(e) > 1$, then this density tends to 1 when m tends to infinity. For every $m \geq 3$ and G such that $N > 2m$, a majority of functions f defined over G are such that $R^v(f) \leq \sqrt{2Nm}$.

Proof. Let $l : L \to \mathbf{F}_2$ be a linear function and l_G its restriction over G, then the number of functions $f : G \to \{\pm 1\}$, such that the distance between f and $\mu(l_G)$ over G is lower than $N/2 - c\sqrt{m}\sqrt{N/2}$, is :

$$A = \sum_{0 \leq i \leq N/2 - c\sqrt{m}\sqrt{N/2}} \binom{N}{i}.$$

Thanks to lemma 4.1, we deduce that : $A \leq 2^N e^{-2N(1/2-\lambda)^2}$, where $0 < \lambda = 1/2 - c\sqrt{m}/\sqrt{2N} < 1/2$. So, $A \leq 2^{N - mc^2 \log_2(e)}$.

Hence, the number of functions f at a distance over G lower than $N/2 - c\sqrt{m}\sqrt{N/2}$ from a linear function is at most $2^m A = 2^{m+N-mc^2 \log_2(e)}$. As $\tilde{f}(a) = N - 2d(f, x \mapsto \mu(ax))$, we obtain that the density of the set defined previously in the theorem, among all the binary functions defined over G, is greater than $1 - 2^{m(1-c^2 \log_2(e))}$.

Moreover, if $c^2 \log_2(e) > 1$ and if we have a sequence $(G_m)_m$, where for all m, G_m is a subgroup of order $N_m > 2c^2 m$ of $\mathbf{F}_{2^m}^\times$, then the density, of the functions defined over G_m such that $R^{v_m}(f) \leq c\sqrt{2N_m m}$, tends toward 1 when m grows to the infinity.

For the last result, notice that we have $2^{m(1-c^2 \log_2(e))} < \frac{1}{2}$ if $m \geq 3$ and $c = 1$. □

Hence, if $m \geq 3$ and $N > 2m$, then

$$\sqrt{\frac{2^m - 1}{v}} \leq R_m^v \leq \sqrt{2m}\sqrt{\frac{2^m - 1}{v}},$$

and a majority of functions are between these two bounds. Notice that in particular, if $N = o(2^m/m)$, then the majority of binary functions f defined over G are such that $R^v(f) = o(\sqrt{2^m})$. Which is sufficient, added to the others points seen in section (2), to construct boolean functions with high non linearities.

5. Exponential Sums

We consider a polynomial $f(X) \in L[X]$ and we write $\tilde{f}(a)$ the Fourier coefficient of the binary function $x \mapsto \mu\big(f(x)\big)$:

$$\tilde{f}(a) = \sum_{x \in G} \mu\big(f(x) + ax\big) = \frac{1}{v} \sum_{x \in L^\times} \mu\big(f(x^v) + ax^v\big). \qquad (9)$$

In particular, if the degree of $f(X)$ is an odd integer $s > 1$ the famous Hasse-Weil bound gives the estimation

$$R^v(f) \leq \frac{1}{v}(sv-1)\sqrt{2^m} + \frac{1}{v} \lesssim s\sqrt{2^m}. \tag{10}$$

This in comparison of (7) seems bad. However, when the index of G is fixed and m increases then (10) is the best that one can say. Whence, for a given polynomial, there is infinitely many extensions such that the Parseval bound (7) is far from the reality.

The goal of this section is to estimate the spectral amplitude of index v of monomials $f(x) = \gamma x^s$ for certain $\gamma \in L$ and integer s. If m is not prime $(m = lt)$, the strategy consists in evaluatinf the exponential sum over $K = \mathbf{F}_q$ instead of L, with $[L : K] = l$ and $q = 2^t$, like in [2]. So, we search instances of (m, l, t, v, s) where v is the index of a group G and s an exponent such that $R^v(\gamma x^s)$ is small for a good choice of $\gamma \in L$. In practice, it is difficult to obtain smooth hypersurfaces from any γx^s. So, we determine the forms of s and vs to apply the results of [2]. Let $w_q(e)$ be the sum of the digits of the q-ary expansion of an integer e. Assume that $w_q(s) \neq w_q(sv)$, denote $w = \max\{w_q(s), w_q(sv)\}$ and let $d \in \{v, sv\}$ the integer such that $w = w_q(d)$.

If $d < q$ is odd or if the q-ary expansion of d is $d = 1 + kq^j$ for any even integer k and $j < (m/l)$, then Theorem 2.1 in [2] gives the following estimation

$$R^v(f) \leq \frac{1}{v}(w-1)^l\sqrt{2^m} + \frac{1}{v} \tag{11}$$

With a computer, we can find a lot of numerical instances (m, l, t, v, s) satisfying $(w-1)^l < (sv-1)$. Unfortunately, we did not find any which satisfy the inequality $(w-1)^l < v$. However, we obtain the following proposition for the groups with index 3.

Proposition 5.1. *Set $m = 2t$, with odd t. Consider $f(x) = \gamma x^s$, with $\mathrm{Tr}_{L/K}(\gamma) \neq 0$. The instance $\big(2t, 2, t, 3, (q+1)/3\big)$ satisfies*

$$R^3_m(f) \leq \frac{4}{3}\sqrt{2^m} + \frac{1}{3} \tag{12}$$

Proof. Set $m = 2t$, $v = 3$, $vs = q + 1$. If $f(x) = \gamma x^s$, we have to estimate

$$\tilde{f}(a) = \frac{1}{v} \sum_{x \in L^\times} \mu\big(\gamma x^{sv} + ax^v\big) = \frac{1}{v} \sum_{x \in L^\times} \mu\big(\gamma x^{q+1} + ax^3\big)$$

If $a \neq 0$, $\max\{w_q(3), w_q(q+1)\} = 3$, the estimation (11) gives (12). If $a = 0$, we have to calculate

$$\tilde{f}(0) = \frac{1}{v} \sum_{x \in L^\times} \mu\big(\gamma x^{q+1}\big)$$

Let μ_K be the additive character of K and let be $x \in L^\times$,

$$\mu(\gamma x^{q+1}) = \mu_K(\mathrm{Tr}_{L/K}(\gamma x^{q+1})) = \mu_K(x^{q+1}\mathrm{Tr}_{L/K}(\gamma)).$$

The map from $L^\times$ to $K^\times$ defined by $x \mapsto x^{q+1}$ is onto, so we have

$$\tilde{f}(0) = \frac{q+1}{v} \sum_{y \in K^\times} \mu_K(y\mathrm{Tr}_{L/K}(\gamma)) = -\frac{q+1}{v}$$

Thus, the inequality (12) rises from $|\tilde{f}(0)| = \frac{q+1}{3} \leq \frac{4}{3}q + \frac{1}{3}$.

□

References

[1] Bringer J., Nonlinearity of some Patterson-Wiedemann type functions. *Yacc-04 Conference*, Porquerolles, France (2004).

[2] Gillot V., Bounds for Exponential Sums over Finite Fields *Finite Fields and Their Applications*,vol. 1, pp. 421–436 (1995).

[3] Langevin P., A New Class of Two Weight Codes. *Finite Fields Conference Fq3*, Glasgow, Scotland pp. 181–187 (1996).

[4] Langevin P., Zanotti J.-P., A Note on the Counter-Example of Patterson-Wiedemann. *Finite Fields Conference Fq6*, Oaxaca, Mexico pp. 214–219 (2001)

[5] Lidl R., Niederreiter H., Finite Fields. Encyclopedia of Mathematics and its Applications, vol. 20 (1983).

[6] Patterson N. J., Wiedemann D. H., The covering radius of the (1, 15) Reed-Muller code is at least 16276 *IEEE Transactions on Information Theory*, vol. 29, pp. 354–356 (1983).

[7] Rodier F., Sur la non-linéarité des fonctions booléennes, *Acta Arithmetica*, vol. 115, pp. 1–22 (2004).

RANDOMLY GENERATED BENT BOOLEAN FUNCTIONS [*]

Anna Grocholewska-Czurylo[1]

Abstract. Arguably, one of the most challenging issues in the field of cryptography, is the design of a basic building block of a stream or block cipher - a cryptographically sound Boolean function. A function that at the same time fulfills to the maximum a number of, often contradicting, cryptographic criteria. This article presents a new method for obtaining highly nonlinear balanced functions by means of random bent function generation. The technique described herein easily yields functions with so far best known nonlinearity for a certain number of arguments, and gives nonlinearities higher than other known methods in other cases (however lower than best known examples).

1. **Introduction**

Over the recent years, a variety of criteria has been identified that a single Boolean function should maximally fulfill in order to be considered as a cryptographically sound basic building block of a strong cipher (be it block or stream cipher). These are balancedness, nonlinearity, autocorrelation, correlation immunity, algebraic degree etc. Some of these criteria are contradictory (like balancedness and highest nonlinearity) and tradeoffs have to be made. These tradeoffs have been the subject of much research, e.g. [2, 16, 18, 24, 27–29]. The more criteria that have to be taken

[*] *This scientific paper has been financed as a research project from Polish State Committee For Scientific Research funds in the years 2004-2006.*

[1] Institute of Control and Information Engineering, Poznan University of Technology, pl. Marii Sklodowskiej-Curie 5, Poznan, Poland email: czurylo@sk-kari.put.poznan.pl

into account, the more difficult the problem is. For some of the properties, it is unclear how tight the best theoretical bounds are. For example, the most interesting for us in this paper, is the upper bound on achievable nonlinearity, which is a subject of conjecture [8].

This paper deals with two of the above mentioned criteria, namely balancedness and nonlinearity. These two criteria are absolutely essential in the design of a cipher. The algorithm presented aims at randomly generating a balanced Boolean function with very high nonlinearity. In fact, to the best of the author's knowledge, achieved nonlinearity is higher than any of the previously published methods.

The paper is organized as follows. Section 2 provides some basic definitions and notations that are used throughout the remainder of the article. In Section 3 a random bent function generator is described, which is used as a foundation for obtaining highly nonlinear balanced functions. In Section 4 some results are presented on comparing S-boxes built from different types of bent functions. Section 5 deals with balancing the bent functions. Experimental results and comparisons to other researchs are given in Section 6. Then conclusions follow in Section 7.

2. Preliminaries

We use square brackets to denote vectors like $[a_1, \ldots, a_n]$ and round brackets to denote functions like $f(x_1, \ldots, x_n)$.

2.1. Boolean function

Let $GF(2) = \langle \sum, \oplus, \bullet \rangle$ be two-element Galois field, where $\sum = \{0, 1\}$, $\oplus$ and $\bullet$ denotes the sum and multiplication mod 2, respectively. A function $f : \sum^n \mapsto \sum$ is an n-argument Boolean function. Let $z = x_1 \cdot 2^{n-1} + x_2 \cdot 2^{n-2} + \ldots + x_n \cdot 2^0$ be the decimal representation of arguments $(x_1, x_2, \ldots, x_n)$ of the function f. Let us denote $f(x_1, x_2, \ldots, x_n)$ as y_z. Then $[y_0, y_1, \ldots, y_{2^n-1}]$ is called a truth table of the function f.

2.2. Linear and nonlinear Boolean functions

An n-argument Boolean function f is linear if it can be represented in the following form: $f(x_1, x_2, \ldots, x_n) = a_1x_1 \oplus a_2x_2 \oplus \ldots \oplus a_nx_n$. Let L_n be a set of all n-argument linear Boolean functions. Let $M_n = \{g : \sum^n \mapsto \sum \mid g(x_1, x_2, \ldots, x_n) = 1 \oplus f(x_1, x_2, \ldots, x_n)$ and $f \in L_n\}$. A set $A_n = L_n \cup M_n$ is called a set of n-argument affine Boolean functions. A Boolean function $f : \sum^n \mapsto \sum$ that is not affine is called a nonlinear Boolean function.

2.3. **Balance**

Let $N_0[y_0, y_1, \ldots, y_{2^n-1}]$ be a number of zeros (0's) in the truth table $[y_0, y_1, \ldots, y_{2^n-1}]$ of function f, and $N_1[y_0, y_1, \ldots, y_{2^n-1}]$ be a number of ones (1's). A Boolean function is balanced if

$$N_0[y_0, y_1, \ldots, y_{2^n-1}] = N_1[y_0, y_1, \ldots, y_{2^n-1}]$$

2.4. **Algebraic Normal Form**

A Boolean function can also be represented as a maximum of 2^n coefficients of the Algebraic Normal Form. These coefficients provide a formula for the evaluation of the function for any given input $x = [x_1, x_2, \ldots, x_n]$:

$$f(x) = a_0 \oplus \sum_{i=1}^{n} a_i x_i \oplus \sum_{1 \le i < j \le n} a_{ij} x_i x_j \oplus \ldots \oplus a_{12\ldots n} x_1 x_2 \ldots x_n$$

where $\sum$, $\oplus$ denote modulo 2 summation.

The order of nonlinearity of a Boolean function $f(x)$ is a maximum number of variables in a product term with non-zero coefficient a_J, where J is a subset of $\{1, 2, 3, \ldots, n\}$. In the case where J is an empty set the coefficient is denoted as a_0 and is called a zero order coefficient. Coefficients of order 1 are $a_1, a_2, \ldots, a_n$, coefficients of order 2 are $a_{12}, a_{13}, \ldots, a_{(n-1)n}$, coefficient of order n is $a_{12\ldots n}$. The number of all ANF coefficients equals 2^n.

Let us denote the number of all (zero and non-zero) coefficients of order i of function f as $\sigma_i(f)$. For n-argument function f there are as many coefficients of a given order as there are i-element combinations in n-element set, i.e. $\sigma_i(f) = \binom{n}{i}$.

2.5. **Hamming distance**

Hamming weight of a binary vector $x \in \sum^n$, denoted as $hwt(x)$, is the number of ones in that vector.

Hamming distance between two Boolean functions $f, g : \sum^n \mapsto \sum$ is denoted by $d(f, g)$ and is defined as follows:

$$d(f,g) = \sum_{x \in \sum^n} f(x) \oplus g(x)$$

The distance of a Boolean function f from a set of n-argument Boolean functions X_n is defined as follows:

$$\delta(f) = \min_{g \in X_n} d(f,g)$$

where $d(f, g)$ is the Hamming distance between functions f and g. The distance of a function f to a set of affine functions A_n is the distance of function f from the nearest function $g \in A_n$.

The distance of function f from a set of all affine functions is called the nonlinearity of function f and is denoted by N_f.

2.6. SAC and SAC(k)

A Boolean function f satisfies SAC if complementing any single input bit changes the output bit with probability of 0.5.

A Boolean function $f(x_1, \ldots, x_n)$ SAC (the strict avalanche criterion) if $f(x) \oplus f(x \oplus \alpha)$ is balanced for any $\alpha \in \sum^n$ such that $hwt(\alpha) = 1$.

$f(x)$ satisfies SAC(k) if any function obtained from $f(x)$ by keeping any k input bits constant satisfies SAC. We say that f is a SAC(k) function if $f(x)$ satisfies SAC(k).

There exists no SAC($n-1$) functions [9].

If $f(x_1, \ldots, x_n)$ satisfies SAC($n-2$) then $deg(f) = 2$.

If $f(x_1, \ldots, x_n)$ satisfies SAC(k) for $0 \leq k \leq n-3$, then $deg(f) \leq n-k-1$ [25].

2.7. Bent functions

A Boolean function $f : \sum^n \mapsto \sum$ is perfectly nonlinear if and only if $f(x) \oplus f(x \oplus \alpha)$ is balanced for any $\alpha \in \sum^n$ such that $1 \leq hwt(\alpha) \leq n$.

For a perfectly nonlinear Boolean function, any change of inputs causes the change of the output with probability of 0.5.

Meier and Staffelbach [20] proved that the set of perfectly nonlinear Boolean functions is the same as the set of Boolean bent functions defined by Rothaus [26].

Perfectly nonlinear functions (or bent functions) have the same, and the maximum possible distance to all affine functions. So their correlation to any affine function is consistently bad (minimal). Linear cryptanalysis works if it is possible to find a good linear approximation of the S-box.

Bent functions are not balanced. This property prohibits their direct application in S-box construction, however there exists numerous methods for modifying bent function in such a way so that the resulting function is balanced and still maintains the good cryptographic properties of a bent function [20]. Hamming weight of a bent function equals $2^{n-1} \pm 2^{\frac{n}{2}-1}$.

Differential analysis [19] can be seen as an extension of the ideas of attacks based on the presence of linear structures [23]. As perfect nonlinear Boolean function have maximum distance to the class of linear structures (equal to 2^{n-2}), they are a useful class of functions for constructing mappings that are resistant to differential attacks.

Bent functions exist only for even n. The nonlinear order of bent functions is bounded from above by $\frac{n}{2}$ for $n > 2$. The number of Boolean bent function for $n > 6$ remains an open problem.

3. Random generation of bent functions

There exists a number of algorithms for constructing bent Boolean functions. Such constructions have been given by Rothaus [26], Kam and Davida [11], Maiorana [13], Adams and Tavares [1], and others.

As an example let's consider the following [1, 11]:

Method 1: Let B_n denote a set of bent functions $f : \sum^n \mapsto \sum$ with n even. Given a set of bent functions B_6, bent functions in B_8 can be constructed using the following method (method 1):

Let $A, B \in B_6$. Then the function $f : \sum^8 \mapsto \sum$ defined by:

$$f(x_0 \dots x_7) = \begin{cases} a(x_0 \dots x_5), & x_6 = 0, x_7 = 0 \\ a(x_0 \dots x_5), & x_6 = 0, x_7 = 1 \\ b(x_0 \dots x_5), & x_6 = 0, x_7 = 0 \\ b(x_0 \dots x_5) \oplus 1, & x_6 = 0, x_7 = 0 \end{cases}$$

is bent. Rearrangements of the 64 blocks in the expression above also result in bent functions.

Another method for bent function construction was given by Rothaus in [26] (method 2): Let $x = (x_1, \dots, x_n)$ and let $a(x)$, $b(x)$ and $c(x)$ be bent functions such that $a(x) \oplus b(x) \oplus c(x)$ is also bent. Then a function $f(x, x_{n+1}, x_{n+2}) = a(x)b(x) \oplus b(x)c(x) \oplus c(x)a(x) \oplus [a(x) \oplus b(x)]x_{n+1} \oplus [a(x) \oplus c(x)]x_{n+2} \oplus x_{n+1}x_{n+2}$ is bent.

Most of the known bent function constructions take bent functions of n arguments as their input and generate bent functions of $n + 2$ arguments. One major drawback of these methods is the fact that they are deterministic. Only short bent functions ($n = 4$ or 6) are selected at random and the resulting function is obtained using the same, deterministic formula every time.

The use of randomly chosen Boolean functions with good cryptographic properties (if we are able to find such functions) is probably better than the use of functions with similar parameters which are obtained by an explicit construction. The main reason is that explicit constructions usually lead to functions which have very particular (algebraic or combinatorial) structures, which may induce weaknesses regarding existing or future attacks. Therefore, authors considered finding and studying randomly generated Boolean functions (at least with a few inputs and outputs) with good cryptographic properties, to be of high interest.

Drawing bent functions at random is not feasible already for small number of arguments ($n > 6$). To make such generation possible, an algorithm was designed to generate random Boolean functions in Algebraic Normal Form thus making use of some basic properties of bent functions to considerably narrow the search space. This makes the generation of bent functions feasible for $n \geq 6$.

The algorithm for the generation of bent functions in ANF domain takes as its inputs the minimum and maximum number of ANF coefficients of every order that the resulting functions are allowed to have. Since the nonlinear order of bent functions is less than or equal to $\frac{n}{2}$, clearly an ANF of a bent function can not be any ANF coefficient of order higher then $\frac{n}{2}$. This restriction is the major reason for random generation feasibility, since it considerably reduces the possible search space.

Presented in Table 1 are numbers of 12-argument bent functions of all nonlinear orders that a non-optimized, PC implementation of the algorithm finds them in a minute (3GHz machine). In Table

Order	*Number of bent functions*
2	819
3	662
4	315
5	88
6	8

TABLE 1. Number of 12-argument bent functions of all nonlinear orders generated in one minute on a 3GHz PC machine

Order	*Number of bent functions*
2	130000
3	60000
4	11000

TABLE 2. Number of 8-argument bent functions of all nonlinear orders generated in one minute on a 3GHz PC machine

2 similar data are given for 8-argument bent functions (as mentioned earlier, the nonlinear order of bent functions is bounded from above by $\frac{n}{2}$).

4. Properties of random and constructed bent functions

4.1. Nonlinearity of pairs (8x2 S-boxes)

Now some comparative results are presented. Three sets of 8-argument bent Boolean functions are analyzed: bent functions constructed using method 1 mentioned earlier, bent functions constructed using method given in [22] (Maiorana functions with permuted inputs) and randomly generated bent functions. For random, distinct i, j the nonlinearity of $f_i \oplus f_j$ was calculated. Figures 1 and 2 show the resulting nonlinearity distribution (in percentage). The random bent functions were generated with the following parameters: number of 2nd order coefficients was between 7 and 14 (statistically that yields the highest number of bent functions), number of 3rd order coefficients was fixed at 2 and number of 4th order ANF coefficients was also fixed at 2. There was no

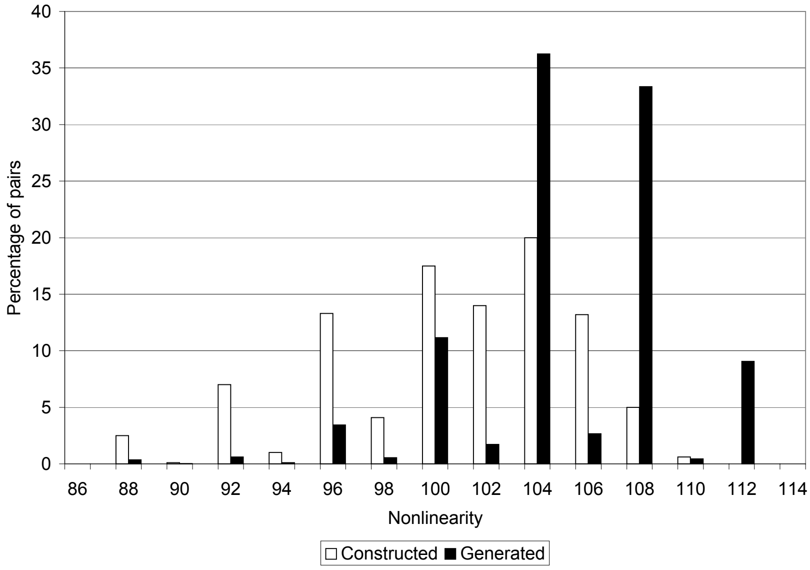


FIGURE 1. Pairs nonlinearity distribution. Constructed bent (method I) vs. Generated bent

coefficient of order 0 and 1 to prevent the occurrences of bent functions that would be just a linear transformations of one another.

As shown on those figures among randomly generated functions more pairs have higher nonlinearity than in other sets, including the set of Maiorana functions with permuted inputs whose results are presented in [22].

4.2. Comparing S-boxes

In this section we concentrate on nonlinearity of S-boxes built using randomly generated bent functions. We give comparative results of the performance of S-boxes built form bent functions constructed using a method introduced by Rothaus [26], bent functions generated with our algorithm described in this paper and random Boolean functions (not bent).

We compare the nonlinearity of 6 × 6 S-boxes testing the feasibility of its linear approximation (since the feasibility of the best linear approximation is a measure of nonlinearity).

Then we simply test (by calculation) the nonlinearity of 8 × 6 S-boxes.

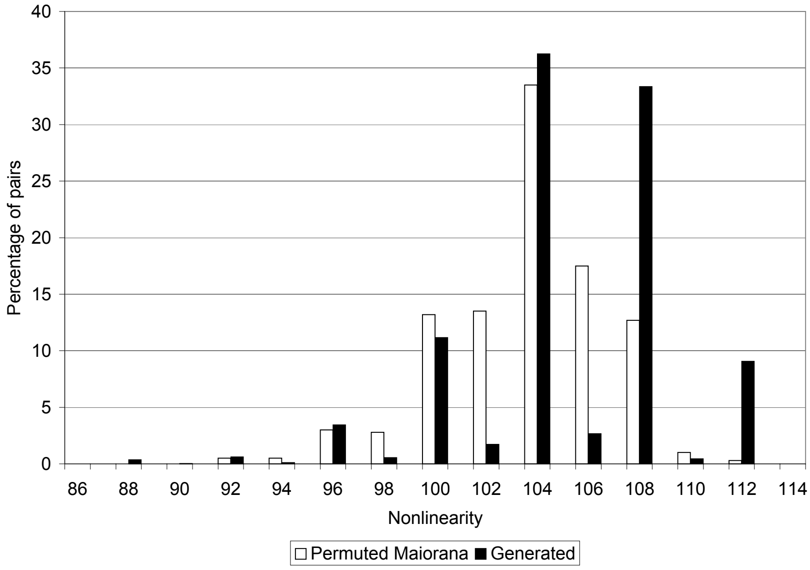


Figure 2. Pairs nonlinearity distribution. Constructed bent (Permuted Maiorana) vs. Generated bent

Finally some interesting results are presented pertaining to SAC criterion (for 8 × 2 and 8 × 4 S-boxes).

One has to note that for real-life applications bent functions would have to be modified to be balanced prior to their use in S-boxes. Such modification algorithms are covered later in the article. Also, for the sakes of clarity, we do not transform random bent functions into balanced, highly nonlinear functions (to compare S-boxes) as some differences between constructed and random bent functions could not perhaps be so clearly visualized.

4.3. **Linear approximations of S-boxes**

By linear approximation of a Boolean function $h : \sum^n \mapsto \sum^m$, written as $Y = h(X)$, we mean any equation of the form:

$$\sum_{i\in Y'} y_i = \sum_{j\in X'} x_j, \quad \text{for } Y' \subseteq \{1,2,\ldots,m\}, X' \subseteq \{1,2,\ldots,n\},$$

fulfilled with the probability of $p = N(X',Y')/2^n$, where $N(X',Y')$ denotes the number of pairs (X,Y) fulfilling the equation, and $\sum$

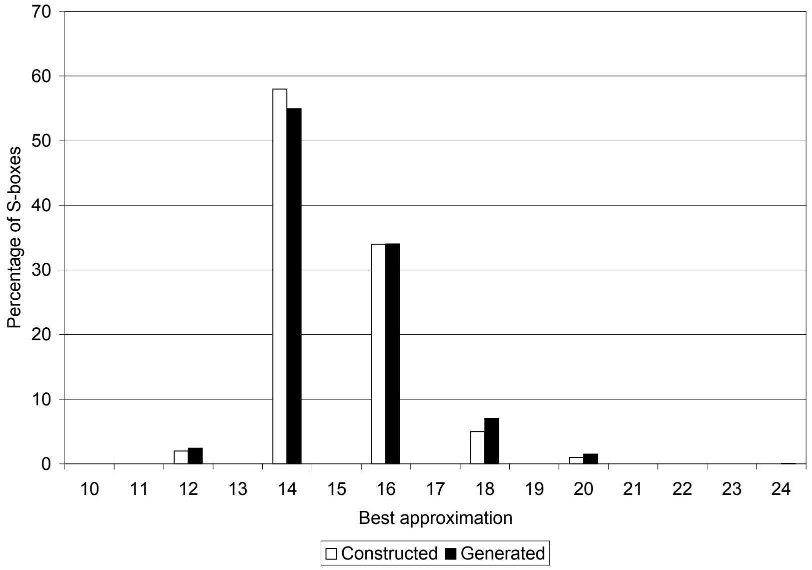


FIGURE 3. Best S-box approximation distribution. Constructed (Rothaus) vs. Generated bent

is a modulo 2 summation. The sets of indices X', Y' are called input and output masks.

The measure of linear approximation effectiveness is the value of a probability $\Delta p = |p - \frac{1}{2}|$ called differential probability. For a fixed n a measure of effectiveness can also be defined as a value of $\Delta N(X', Y') = |N(X', Y') - 2^{n-1}|$.

In our experiment we tested linear approximations of 6 × 6 S-boxes, i.e. functions $Y = h(X) : \sum^6 \mapsto \sum^6$, where sub-functions of function h were constructed bent functions and randomly generated bent functions. The distribution of the best approximations was tested, i.e. maximum value of $\Delta N(X', Y')$ among all possible sets of input and output masks (except empty output mask). For each type of functions, 10000 of random S-boxes were tested.

Differences between S-boxes built from bent functions constructed using Rothaus method (method 2) and S-boxes built from randomly generated bent functions are not very evident.

4.4. Nonlinearity

Now we will show the results of testing the S-boxes for high nonlinearity. We consider 6×8 S-boxes (each S-box is constructed of six 8-argument functions).

The nonlinearity of an S-box, that as to say a function: $F : \sum^n \mapsto \sum^m$ such that $F(x) = (f_1(x), f_2(x), \ldots, f_m(x))$ i $x \in \sum^n$ is calculated as minimal nonlinearity of all linear combinations of F's sub-functions. The nonlinearity of a S-box is then defined as follows:

$$N_F = \min\{N_{f_J} | f_J = \sum_{i \in J} f_i, J \subseteq (1, 2, \ldots, m)\}$$

To calculate nonlinearity of a single S-box $2m$ linear combinations have to be constructed and their distance to affine functions calculated. The lowest of all calculated nonlinearities (distances to affine functions) is the nonlinearity of the S-box.

Using Rothaus construction the maximum achieved S-box nonlinearity was 100 (for about 2% of all S-boxes). For S-boxes built using randomly generated bent function maximum nonlinearity was 112 (for about 5% of all S-boxes!). It is worth noting that it is the highest known nonlinearity for 8×6 S-box.

This means that using randomly generated bent functions may lead to constructing S-boxes of better cryptographic qualities in lesser time.

However, one has to note the fact that in case of randomly generated bent functions there are also S-boxes of relatively poor nonlinearity (like 80). So building S-boxes from these functions requires (more then in other cases) to carefully check the resulting S-boxes for possible low nonlinearity.

4.5. Strict avalanche criterion for S-boxes

The generalization of SAC(k) to vector output Boolean functions (S-Boxes) has been proposed by Kurosawa and Satoh in [12] and as a step toward the security of block ciphers against attacks which keep some input bits constant.

We say that $F(x_1, \ldots, x_n) = (f_1, \ldots, f_m)$ is an $(n, m)-$SAC(k) function if all nonzero linear combinations of $f_1, \ldots, f_m$ satisfy SAC(k).

For the purpose of the research described in this paper various sample S-boxes were tested. These S-boxes were not necessarily

k	*Percentage of S-boxes*
0	26.80
1	33.00
2	15.60
3	2.40
4	0.20

Table 3. (8,2)-SAC(k) fulfilling S-boxes

k	*Percentage of S-boxes*
0	37.80
1	19.60
2	3.90
3	0.30
4	0.00

Table 4. (6,2)-SAC(k) fulfilling S-boxes

chosen to fit any particular application but to clearly show the differences between constructed bent functions and randomly generated bent functions. Such an S-box in its simplest form would be just a pair of two Boolean functions (S-box size 8×2). 10000 S-boxes have been tested during each experiment mentioned below.

And so, for S-boxes of size 8×4 built from constructed bent functions (Rothaus method) there has not been a single S-box found that would satisfy even the original SAC ((8,4)-SAC(0)), while among the same size S-boxes built from randomly generated bent functions there have been ca. 71.5% of non-SAC fulfilling S-boxes, 28.2% of (8,4)-SAC(0) fulfilling S-boxes and the remaining 0.3% of (8,4)-SAC(1) S-boxes.

For 8×2 S-boxes (pairs of bent functions) there also hasn't been a single S-box found that would satisfy SAC for constructed bent functions. For randomly generated bent functions the proportions of SAC fulfilling S-boxes were shown in Table 3.

Similarly, for S-boxes of size 6×4, there has been no SAC S-boxes built from constructed bent functions while there were about 3% of (6,4)-SAC(0) S-boxes built from randomly generated bent functions.

For S-boxes of size 6 × 2 (pairs of 6-argument bent functions) there have been ca. 3% of (6,2)-SAC(0) S-boxes built from constructed bent functions, and results for randomly generated bent functions are given in Table 4.

As one can easily see from the examples above, randomly generated bent functions possess some interesting cryptographic qualities, quite distinct from those of constructed bent functions. While generation times for randomly generated bent functions are up to 40 times shorter than the time required for constructing a bent function (using a Rothaus method, 8-argument bent functions) this opens new possibilities for designing fast algorithms for strong S-box constructions.

5. Balancing bent functions

As bent functions achieve maximum possible nonlinearity they are often used as a foundation for constructing highly nonlinear balanced functions that could be used directly in for example S-boxes. In recent years some methods have been proposed that transform bent functions to balanced Boolean functions with minimal loss in nonlinearity. Examples of such methods are given in [14, 15].

For the purpose of this research none of these methods has been implemented. Instead the adopted approach was to balance generated bent functions randomly, without any sort of optimization and then to test the nonlinearity of resulting balanced functions.

Thus balancing is performed as follows. A bent function is randomly generated (using the above mentioned generator). Depending on whether bent function's hamming weight is $2^{n-1} - 2^{\frac{n}{2}-1}$ or $2^{n-1} + 2^{\frac{n}{2}-1}$ missing 1's or 0's are added at random positions.

This approach suggests unbiased comparison of the nonlinearities achieved. Had any of the balancing methods been used it would be uncertain if high nonlinearity comes from properties of randomly generated bent functions, or from carefully designed balancing techniques.

n	8	9	10	11	12
LUB	118	244	494	1000	2014
BK	116	240	492	992	2010
DC	116		492		2010
BC	112	240	480	992	1984
R	112	230	472	962	1955
RHC	114	236	476	968	1961
GA	116	236	484	980	1976
DNL	114	236	480	974	1972
NLT	116	238	486	984	1992
ACT	116	238	484	982	1986
GEN	**116**	**240**	**488**	**992**	**2002**

Table 5. Conjectured upper bounds and attained values for nonlinearity of balanced functions

6. Experimental results for balanced functions

Table 5 summarizes the results obtained. The table shows nonlinearity of balanced Boolean functions achieved by the best currently known techniques along with best theoretical upper bounds and the best currently known examples. The table gives values for Boolean functions of 8 up to 12 arguments. Results for lower number of arguments are the same for every method and are in fact maximum achievable.

Abbreviations used in Table 5 are: LUB - Lowest Upper Bound, BK - Best Known [10], DC - Dobertin's Conjecture [8], BC - Bent Concatenation, R - Random, RHC - Random + Hill-Climb [7], GA - Genetic Algorithms [21], DNL - Direct Non-Linearity [7], NLT - Non-Linearity Targeted [7], ACT - Auto-Correlation Targeted [7], and finally GEN - Balanced Randomly Generated functions (results presented in this paper)

As it can be clearly seen from Table 5, a random generation method presented in this paper gives the same results of nonlinearity of balanced functions only for $n = 8$, and is better than any of the other methods for all higher numbers of arguments, and most profoundly so in case of 11 and 12 arguments. It is also worth noting, that for $n = 9$ and $n = 11$ random generation methods yield results equal to the best known examples of highly nonlinear balanced Boolean functions.

Nonlinearity	*Number of functions*
1986	3
1988	41
1990	543
1992	3979
1994	22458
1996	76942
1998	87004
2000	9474
2002	6

Table 6. Nonlinearities of balanced 12-argument functions obtained by random balancing generated bent functions

Other interesting results are presented in Table 6. 200000 12-argument bent functions were randomly generated and then randomly balanced (as described in Section 5). Table 6 shows the nonlinearity of the resulting balanced functions.

The vast majority (>97%) of balanced Boolean functions in Table 6 has nonlinearity between 1994 and 2000, which is very high (see Table 5).

7. Conclusions

The main relevant cryptographic properties for block ciphers are the nonlinearity of the S-box, its propagation characteristics and its resistance to differential attacks. One of the underlying problem is then to construct S-boxes with high nonlinearity, whose Boolean components are some highly nonlinear functions which are randomly chosen.

From the results presented in this paper it seems that random generated bent functions (and obtained balanced functions) offer an interesting alternative to construction methods. Not only nonlinear characteristics of these functions are equal or better than those of constructed functions but also generated functions have a very compact (small) Algebraic Normal Form which can be used for efficient storage and fast cryptographic routines.

A new search algorithm has been presented, which proved to be very efficient in finding highly nonlinear balanced Boolean functions.

However, before practical applications more research is needed. In particular, the resiliency of balanced functions has to be investigated.

References

[1] C. M. Adams, S. E. Tavares. *Generating and Counting Binary Bent Sequences.* In *IEEE Transactions on Information Theory*, IT-36:1170–1173, 1990.

[2] C. Carlet. *On the coset weight divisibility and nonlinearity of resilient and correlation immune functions.* In *SETA 2001*, 2001

[3] J. A. Clark, J. L. Jacob. *Two stage optimisation in the design of Boolean functions.* In E. Dawson, A. Clark, and C. Boyd, editors, *5th Australasian Conference on Information Security and Privacy, ACISP 2000*, pages 242–254, LNCS 1841, Springer, 2000.

[4] J. A. Clark, J. L. Jacob, S. Stepney. *The Design of S-Boxes by Simulated Annealing.* In *CEC 2004: International Conference on Evolutionary Computation, Portland, USA, June 2004*, 2004.

[5] J. A. Clark, J. L. Jacob, S. Stepney. *Secret agents leave big footprints: how to plant a cryptographic trapdoor, and why you might not get away with it.* In *Genetic and Evolutionary Computation Conference: GECCO 2003*, pages 2022–2033, LNCS 2724, Springer, 2003.

[6] J. A. Clark, J. L. Jacob, S. Stepney. *Functions satisfying multiple criteria.* In *Progress in Cryptology: INDOCRYPT 2002*, pages 246–259, LNCS 2551, Springer, 2002.

[7] J. A. Clark, J. L. Jacob, S. Stepney. *for cost functions.* In *CEC 2004: International Conference on Evolutionary Computation, Portland OR, USA, June 2004*, pages 1517–1524, IEEE 2004.

[8] H. Dobbertin. *Construction of bent functions and balanced functions with high nonlinearity.* In *Fast Software Encryption, 1994 Leuven Workshop*, pages 61–74, LNCS 1008, Springer, 1994.

[9] R. Forré. *The strict avalanche criterion: spectral properties of Boolean functions with high nonlinearity.* In *Advances in Cryptology: CRYPTO 1988*, Springer-Verlag, 1990.

[10] X. D. Hou. *On the norm and covering radius of first-order Reed-Muller codes.* In *IEEE Transactions on Information Theory*, 43(3):1025–1027, May 1997.

[11] J. B. Kam, G. Davida. *Structured Design of Substitution-Permutation Encryption Networks.* In *IEEE Transactions on Computers*, C-28:747–753, 1979.

[12] K. Kurosawa, T. Satoh. *Generalization of higher order SAC to vector output Boolean functions.* In *IEICE Transactions*, vol. E90, No. 1, January 1998.

[13] J. A. Maiorana. *A Class of Bent Functions.* In *R41 Technical Paper*, 1971.

[14] S. Maity, T. Johansson. *Construction of Cryptographically Important Boolean Functions.* In *INDOCRYPT 2002*, 234–245.

[15] S. Maity, S. Maitra. *Distance between Bent and 1-Resilient Boolean Functions.* In *FSE 2004*, 143–160.

[16] S. Maitra. *Highly nonlinear balanced Boolean functions with very good autocorrelation property* In *Technical Report 2000/047*, Indian Statistical Institute, Calcutta, 2000.

[17] S. Maitra. *Autocorrelation properties of correlation immune Boolean functions.* In *INDOCRYPT 2001*, pages 242–253, Springer, 2001.

[18] S. Maitra, E. Pasalic. *Further constructions of resilient Boolean functions with very high nonlinearity.* In *SETA 2001*, 2001.

[19] M. Matsui. *Linear cryptanalysis method for DES cipher.* In *Abstracts of EUROCRYPT 1993*, 1993.

[20] W. Meier, O. Staffelbach. . *Nonlinearity criteria for cryptographic functions.* In J. J. Quisquater, J. Vandewalle, editors, *Advances in Cryptology: EUROCRYPT 1989*, pages 549–562, LNCS 434, Springer, 1989.

[21] W. Millan, A. Clark, E. Dawson. *Heuristic design of cryptographically strong balanced Boolean functions.* In *Advances in Cryptology: EUROCRYPT 1998*, pages 489–499, LNCS 1403, Springer, 1998.

[22] S. Mister, C. Adams. *Practical S-Box Design.* In *Workshop on Selected Areas in Cryptography: SAC 1996*, Workshop Record, Queens University, pages 61-76, 1996.

[23] K. Nyberg. *Perfect nonlinear S-boxes.* In *Advances of Cryptology: EUROCRYPT 1991*, LNCS, 547:378–386, 1991.

[24] E. Pasalic, S. Maitra, T. Johansson, P. Sarkar. *New constructions of resilient and correlation immune Boolean functions achieving upper bound on nonlinearity.* In *Workshop on Coding Theory, Electronic Notes in Discrete Mathematics.*, Elsevier, 2001.

[25] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, J. Vandewalle. *Propagation characteristics of Boolean functions.* In *Advances in Cryptology: EUROCRYPT 1990*, LNCS, 473:161–173, 1991.

[26] O. S. Rothaus. *On bent functions.* In *Journal of Combinatorial Theory: Series A*, 20:300–305, 1976.

[27] J. J. Son, J. I. Lim, S. Chee, S. H. Sung. *Global Avalanche Characteristics and nonlinearity of balanced Boolean functions.* In *Information Processing Letters*, 65(3):139–144, 1998.

[28] S. H. Sung, S. Chee, C. Park. *Global Avalanche Characteristics and propagation criterion of balanced Boolean functions.* In *Information Processing Letters*, 69(1):21–24, 1999.

[29] Y. Tarannikov. *On resilient Boolean functions with maximal possible nonlinearity.* In *Technical Report 2000/005*, Mech. and Math. Department, Moscow State University, 2000.

[30] X. M. Zhang, Y. Zheng. *GAC - the criterion for Global Avalanche Characteristics of cryptographic functions.* In *Journal of Universal Computer Science*, 1(5):316–333, 1995.

BFCA'05 - Proceedings

About the book

Held March, 2005, in Rouen, BFCA'05 was the first workshop about Boolean Functions and their applications (notably in cryptography). During two days, many different international scientists met each other and talked about their work. This book contains the acts of the proceedings of the conference.

À propos de cet ouvrage

En mars 2005 s'est tenu à Rouen, BFCA'05, le premier atelier sur le thème des Fonctions Booléennes et de leurs applications (en particulier cryptographiques). Pendant deux jours, de nombreux chercheurs internationaux s'y sont rencontrés et y ont parlé de leurs travaux. Cet ouvrage est composé des articles associés aux différentes conférences qui s'y sont tenues.

About the editors :

Jean-Francis Michon is Computer Science Professor and Director of the LIFAR (Computer Science Laboratory) at University of Rouen, France.
Pierre Valarcher is Computer Science Assistant Professor and Member of the LIFAR (Computer Science Laboratory) at University of Rouen, France.
Jean-Baptiste Yunès is Computer Science Assistant Professor and Member of the LIAFA (Computer Science Laboratory) at University of Paris 7 - Denis Diderot, France.

With the collaboration of :
Gwénolé Ars, Julien Bringer, Claude Carlet, Vincent Carlier, Hervé Chabanne, Zu-Ling Chang, Ali Doganaksoy, Emmanuelle Dottax, Gérard Duchamp, Jean-Charles Faugère, Fang-Wei Fu, Philippe Gaborit, Valérie Gillot, Philippe Guillot, Anna Grocholewska-Czurylo, Hatem Hadj Kacem, Martin Hell, Philippe Langevin, Eric Laugerotte, Yuzhen Liu, Subhamoy Maitra, Alexander Maximov, Qingshu Meng, Sihem Mesnager, François Rodier, Elif Yildrim Saygi, Zülfükar Saygi, Meltem Sönmez Turan, Qiao-Yan Wen, Min Yang, Huanguo Zhang.

AVEC LE CONCOURS DU CONSEIL GÉNÉRAL DE LA SEINE-MARITIME

Seine-Maritime
Le Département

UNIVERSITE DE ROUEN