HAL
archives-ouvertes.fr

Boolean functions with restricted input and their robustness; application to the FLIP cipher

Claude Carlet, Pierrick Méaux, Yann Rotella

▶ **To cite this version:**

Claude Carlet, Pierrick Méaux, Yann Rotella. Boolean functions with restricted input and their robustness; application to the FLIP cipher. IACR Transactions on Symmetric Cryptology, Ruhr Universität Bochum, 2017, 2017 (3), pp.192–227. <10.13154/tosc.v2017.i3.192-227>. <hal-01633506>

HAL Id: hal-01633506
https://hal.inria.fr/hal-01633506

Submitted on 13 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Boolean functions with restricted input and their robustness; application to the FLIP cipher

Claude Carlet[1], Pierrick Méaux[2] and Yann Rotella[3]

[1] LAGA, Department of Mathematics, University of Paris 8, and Paris 13 and CNRS, Saint–Denis cedex 02, France.
claude.carlet@univ-paris8.fr
[2] Inria, CNRS, ENS and PSL Research University, Paris, France.
pierrick.meaux@ens.fr
[3] Inria, University Pierre & Marie Curie Paris 6, Paris, France.
yann.rotella@inria.fr

Abstract. We study the main cryptographic features of Boolean functions (balancedness, nonlinearity, algebraic immunity) when, for a given number n of variables, the input to these functions is restricted to some subset E of $\mathbb{F}_2^n$. We study in particular the case when E equals the set of vectors of fixed Hamming weight, which plays a role in the FLIP stream cipher and we study the robustness of the Boolean function in this cipher.

Keywords: FLIP · Boolean function · balance · nonlinearity · algebraic immunity · constrained input

1 Introduction

In a cryptographic framework, Boolean functions are classically studied with an input ranging over the whole vector space $\mathbb{F}_2^n$ of binary vectors of some length n. This is the case when Boolean functions are used as the (main) nonlinear components of a stream cipher, in the so-called combiner and filter models of pseudo-random generators. However, it can happen that the function be in fact restricted to a subset (say E) of $\mathbb{F}_2^n$. A recent example of such situation is given by the cipher FLIP (see [MJSC16]).

1.1 FLIP: filtering a constant Hamming weight register

The cipher FLIP is an encryption scheme that appeared recently. It is specifically to be combined with an homomorphic encryption scheme to improve the efficiency of somewhat homomorphic encryption frameworks. As for Kreyvium [CCF+16] and LowMC [ARS+15] the goal of the cipher is to present a decryption algorithm whose homomorphic evaluation is as insignificant as possible in term of homomorphic error growth. This homomorphic-friendly design requires to drastically reduce the multiplicative depth of the decryption circuit and in the case of FLIP, it led to use a non generic construction: the filter permutator. This symmetric primitive consists in updating a key register only by wire-cross permutations and then in filtering it with a Boolean function wich inputs the whole register to generate the keystream. At each clock cycle the wire-cross permutation used to shuffle the secret key is given by the output of a PRNG. The PRNG seed acting as an IV, at each clock cycle the input to the filtering function is only a reordering of the secret key bits.

This specificity produces an unusual situation for stream ciphers: the Hamming weight of the key register is invariant, equal to the Hamming weight of the secret key. For the

Licensed under Creative Commons License CC-BY 4.0.
IACR Transactions on Symmetric Cryptology ISSN 2519-173X, Vol. 2017, No. 3, pp. 192–227
DOI:10.13154/tosc.v2017.i3.192-227

instances of FLIP, the Hamming weight of the secret key is set to $\frac{n}{2}$, with n the size of the secret key, way larger than usual key sizes (> 500 and > 1000 bits for security parameters 80 and 128). These sizes prevent an exhaustive search but it still restricts the input of the filtering function to the half Hamming weight vectors of $\mathbb{F}_2^n$. It raises a very natural question, which was not addressed in [MJSC16] but which is mandatory when evaluating the security: Does the filtering function maintain good behavior on this restricted input with respect to classical attacks? The Boolean criteria commonly used to study the robustness of a filtering function are always considered on the whole space $\mathbb{F}_2^n$; hence they do not apply on restricted inputs. Therefore in this work we study Boolean functions on restricted inputs, focusing on criteria adapted to restricted sets.

1.2 Boolean criteria on restricted sets

Let us begin with a preliminary remark: for the FLIP family of stream ciphers, the divide-and-conquer technique introduced by Siegenthaler [Sie84, Sie85] does not seem to apply. Siegenthaler's attack applies on a combination of several generators filtered by a Boolean function, when there is a correlation between the output of the function and some of its input coordinates, which allows to make an exhaustive search reduced to the outputs of the corresponding generators, without needing to consider the outputs of the other generators. To withstand the attack, the function needs in such framework to have large resilience order. However, in the FLIP family of stream ciphers, a permutation is applied at each clockcycle to only one register. It seems then very difficult to find a bias between the output to a function and a fixed set of input variables. More generally, it seems very difficult to apply Siegenthaler's attack on ciphers in which a filter function applies on restricted input, because the principle of the correlation attack, as explained above, is to make an exhaustive search on some part of the initial state without having any restriction on the rest of the state; the restriction (like fixing the Hamming weight) imposes a dependence between the two parts. Consequently we do not study the resilience of "restricted Boolean functions". But all the other classical features of Boolean functions (namely balancedness, algebraic immunity and nonlinearity) continue to play a direct role with respect to attacks in such new framework. However, their behavior changes because of the restriction on the input.

1.2.1 Balancedness

A first commonly accepted requirement on cryptographic Boolean functions is to be balanced -or at least almost balanced- since otherwise, if there is a fairly big bias in the output distribution of the function, then the attacker could detect the resulting statistical bias between the plaintext and the ciphertext, allowing to distinguish when two texts of the same length have high probability to be a plaintext and the corresponding ciphertext. We shall then be focussed on those functions which are balanced on the input set E. But since E may change in the process (this is not the case in FLIP but it could be in a variant), we are interested in Boolean functions whose restrictions to all sets E in some family $\mathcal{E}$ are balanced. Even if E does not change, we may wish to have a Boolean function which is balanced on a family of sets E, so that it can be used in a variety of situations. Given some family $\mathcal{E}$ of subsets of $\mathbb{F}_2^n$, we shall say that a Boolean function f is perfectly balanced over $\mathcal{E}$ if its restriction to any set $E \in \mathcal{E}$ of even size is balanced. We shall be in particular interested in the case of $\mathcal{E} = \{E_{n,1}, \ldots, E_{n,n-1}\}$, where $E_{n,k} = \{x \in \mathbb{F}_2^n; \mathsf{w_H}(x) = k\}$, $\mathsf{w_H}$ denoting the Hamming weight. We shall then call such functions *weightwise perfectly balanced*.

Notation 1. *We denote by* $\mathsf{w_H}(f)_k$ *the Hamming weight of the evaluation vector of the*

function f on all the entries of fixed Hamming weight k:

$$\mathsf{w_H}(f)_k = |\{x \in \mathbb{F}_2^n, \mathsf{w_H}(x) = k, f(x) = 1\}|,$$

where $\mathsf{w_H}$ *denotes the Hamming weight. We accordingly denote* $\overline{\mathsf{w_H}(f)_i} = |\{x, \mathsf{w_H}(x) = i, f(x) = 0\}| = \binom{n}{i} - \mathsf{w_H}(f)_i$. *We denote by* $E_{n,k}$ *the set of such entries:* $E_{n,k} = \{x \in \mathbb{F}_2^n; \mathsf{w_H}(x) = k\}$.

Definition 1. Let f be a Boolean function defined over $\mathbb{F}_2^n$. It will be called *weightwise perfectly balanced* (*WPB*) if, for every $k \in \{1, \ldots, n-1\}$, the restriction of f to $E_{n,k}$, is balanced, that is, $\forall k \in [1, n-1], \mathsf{w_H}(f)_k = \frac{\binom{n}{k}}{2}$.
To make the function balanced on its whole domain $\mathbb{F}_2^n$, we shall additionally impose that $f(0, \ldots, 0) \neq f(1, \ldots, 1)$ and more precisely that

$$f(0, \ldots, 0) = 0; \quad f(1, \ldots, 1) = 1.$$

This last constraint does not reduce the generality (when $f(0, \ldots, 0) \neq f(1, \ldots, 1)$), up to the addition of constant 1 to f, and it makes some constructions clearer. Note that weightwise perfectly balanced Boolean functions exist only if, for every $k \in [1, n-1]$, $\binom{n}{k}$ is even and this property is satisfied if and only if n is a power of 2. Note that $\mathsf{w_H}(f)_k = \frac{\binom{2^\ell}{k}}{2}$ is then even for $k \in [1, \ldots, 2^{\ell-1} - 1] \cup [2^{\ell-1} + 1, \ldots, 2^\ell - 1]$ and odd for $k = 2^{\ell-1} = n/2$. To be able to address the case where n is not a power of 2, we introduce:

Definition 2. Let f be a Boolean function defined over $\mathbb{F}_2^n$. It will be called *weightwise almost perfectly balanced functions* (*WAPB*) if, for every $k \in [1, n-1]$, $\mathsf{w_H}(f)_k = \frac{\binom{n}{k}}{2}$ when $\binom{n}{k}$ is even and $\mathsf{w_H}(f)_k = \frac{\binom{n}{k} \pm 1}{2}$ when $\binom{n}{k}$ is odd.

1.2.2 Nonlinearity

A second parameter, which plays an important role for quantifying the contribution of the function to the resistance against attacks by affine approximations, like the fast correlation attack [MS88b], is the minimum of the Hamming distance $\mathsf{d}_H(f, h) = |\{x \in \mathbb{F}_2^n; f(x) \neq h(x)\}|$ between $f(x)$ and affine functions $h(x) = a \cdot x + \varepsilon$, $a \in \mathbb{F}_2^n, \varepsilon \in \mathbb{F}_2$ (where "$\cdot$" is some inner product in $\mathbb{F}_2^n$; any choice of an inner product will give the same definition). This parameter is called the *nonlinearity* of the function, and we shall denote it by $\mathsf{NL}(f)$ when there is no restriction on the input to f, and $\mathsf{NL}_E(f)$ when the input to f is taken from a set E.

Let E be any subset of $\mathbb{F}_2^n$ and f any Boolean function defined over E (i.e. any function from E to $\mathbb{F}_2$). Let $\ell(x) = a \cdot x + \varepsilon$ be any affine function. Denoting by $f_a(x)$ the sum (in $\mathbb{F}_2$) of $f(x)$ and $a \cdot x$, we have: $\sum_{x \in E}(-1)^{f(x)+a\cdot x} = \sum_{x \in E}(1 - 2f_a(x))$, and the Hamming distance between f and $a \cdot x$ on inputs ranging over E equals $\sum_{x \in E} f_a(x) = \frac{|E|}{2} - \frac{1}{2}\sum_{x \in E}(-1)^{f(x)+a\cdot x}$ (sums performed in $\mathbb{Z}$). Hence, the Hamming distance between f and ℓ over E equals:

$$\frac{|E|}{2} - \frac{(-1)^\epsilon}{2} \sum_{x \in E} (-1)^{f(x)+a\cdot x}.$$

Definition 3. Let E be any subset of $\mathbb{F}_2^n$ and f any Boolean function defined over E. We call *nonlinearity of f over E* and denote by $\mathsf{NL}_E(f)$ the minimum Hamming distance between f and the restrictions to E of affine functions over $\mathbb{F}_2^n$.

1.2.3 Algebraic immunity

A third parameter plays a role for quantifying the contribution of the function to the resistance against algebraic attacks, giving the degree of the algebraic system obtained

by the Courtois-Meier method [CM03] (which needs to be solved for recovering the initialization of the register). It is called the *algebraic immunity* of the function; we shall denote it by $\mathsf{AI}(f)$ when there is no restriction on the input to f, and $\mathsf{AI}_E(f)$ when the input to f is taken from a set E.

Let E be any subset of $\mathbb{F}_2^n$ and f any Boolean function defined over E. The principle of the algebraic attack is to use the existence of Boolean functions g and h over $\mathbb{F}_2^n$, such that h and gf coincide over E, while g is not identically null on E. In the case of the standard attack, both functions g and h must have low algebraic degree, and in the case of the fast algebraic attack, g must have low algebraic degree and h must have an algebraic degree reasonably low (say, not much larger than $n/2$). The algebraic immunity of f is then defined as $AI(f) = \min\{\max(deg(g), deg(fg)); g \neq 0\}$ and equals $\min\{deg(g); fg = 0 \text{ or } (f+1)g = 0; g \neq 0\}$ because if $fg = h$ then $f(g+h) = 0$. It enables to define the algebraic immunity over a restricted set:

Definition 4. We call *algebraic immunity* of a function f over a set E the number:

$$\min\{\deg(g); g \text{ annihilator of } f \text{ or of } f+1 \text{ over } E \text{ and } g \text{ not identically null over } E\}.$$

1.3 Previous works

Studying the robustness of Boolean functions from these criteria has been largely applied for the security analysis of stream ciphers, and the corresponding attacks are considered as the standard attacks to consider for any stream cipher. In that sense, many works consider the three Boolean criteria presented above for the particular case of a design it introduces or of the cryptanalysis it develops. More specifically, some papers have already considered Boolean functions whose inputs are restricted.

The bias of a stream cipher output in presence of Hamming weight leakage is considered in [JD06]. Precisely, it is shown that knowing the Hamming weight of a register when the updating function is an LFSR in a particular representation enables to distinguish the keystream from a random binary stream, and the authors also describe a correlation attack in this setting. This result can be deduced from an application of a subpart of our results: they use the balancedness flaw on a function on the sets $\{x \mid \mathsf{wH}(x) = k\}$ (the fact that this function is not weightwise perfectly balanced), and combine it with other equations to mount a correlation attack on these LFSR.

Concerning the algebraic immunity criteria, the work [CFGR12] realises a theoretical study of the algebraic phase of the so-called algebraic side channel attacks on block ciphers. The authors modify the notion of algebraic immunity to include the information (on Hamming weight or Hamming distance) obtained by exploiting the leakage and are able to obtain enough equations of degree one to solve the algebraic system with Gröbner methods. In the present paper, our modification of the definition of algebraic immunity is also related to Hamming weight but is of a different nature, being related to the fact that the input is restricted. Another major difference is that we focus on functions with one bit of output and not S-boxes functions.

A study has been made by Yuval Filmus et al. on the restrictions of Boolean functions to sets of inputs of fixed Hamming weight (that he calls "slices") [Fil16a, Fil16b, FKMW16, FM16]; this study is asymptotical and does not really fit with our cryptographic framework; the results from these papers have no overlap with ours.

Finally, the nonlinearity of Boolean functions under non-uniform input distribution has been recently studied in [GGPS17], but the chosen distribution is binomial and there is no overlap with our work in this case as well.

1.4 Our contributions

We realise the first study of balancedness, nonlinearity and algebraic immunity on Boolean functions with restricted inputs, centred on the fixed Hamming weight input. In this case we study the degradation of the parameters for functions optimal in the whole space, commonly used for basic cryptographic constructions. More suprisingly we determine bent functions which are linear for every restricted Hamming weight and hence have null $\mathsf{NL}_{E_{n,k}}$ for every k and relatively to algebraic immunity we prove counter-intuitive results for direct sums (used in the design of the FLIP Boolean function).

Then for each criterion, we compare its behavior in this constrained framework to the properties well known in $\mathbb{F}_2^n$, we consider the functions with highest criterion on $E_{n,k} = \{x \mid \mathsf{wH}(x) = k\}$ that we can construct. More precisely, for the balancedness criterion, we prove necessary conditions on the Algebraic Normal Form of f to be weightwise perfectly balanced and we give a primary construction (i.e. we exhibit a class) of such functions and a secondary construction for designing them. Since weightwise perfectly balanced functions can exist only in numbers of variables which are powers of 2, we also give a construction of weightwise almost perfectly balanced function for all n and present a relation between balancedness on fixed weight inputs and a transform similar to the Walsh transform involving symmetric functions. For the nonlinearity criterion, we give for every subset E of $\mathbb{F}_2^n$ an upper bound for those functions restricted to E and show that, contrarily to the case of $\mathbb{F}_2^n$, this bound (related to bent functions) cannot be reached for most $E_{n,k}$. We use an error correcting code perspective to construct functions with non null $\mathsf{NL}_{E_{n,k}}$ for all $k \in [1, n]$. For the algebraic immunity criterion, we generalise the upper bound for all set $E$of $\mathbb{F}_2^n$ and give precise results in the constant Hamming weight case, showing how the general algebraic immunity can decrease on $E_{n,k}$.

We give a cryptanalysis aspect of this study analyzing the 4 instances of the cipher FLIP. For these functions we prove bounds for the three main criteria, also considering possible attack improvements with guess and determine attacks. We provide a new security analysis of this cipher, based on filtered function with fixed Hamming weight input.

1.5 Paper organisation

In Section 2 we show how much the restriction to inputs of fixed hamming weight can influence the cryptographic criteria. Then Section 3 concerns the behavior of the criteria of balancedness, nonlinearity and algebraic immunity on restricted inputs, and finally Section 4 presents the security analysis of FLIP with fixed Hamming weight input.

2 Fixed Hamming weight inputs and criteria degradations

In this section we show how the restriction to inputs of fixed Hamming weight can affect the cryptographic criteria. This restriction makes that some functions which are known as having optimally good cryptographic property when they are defined over the whole space totally lose this property when their input becomes restricted. It is trivially the case of so-called symmetric functions (whose output depends only on the Hamming weight of the input) when the input weight becomes restricted, like the majority function (which has optimal algebraic immunity over $\mathbb{F}_2^n$). But there are other examples. Thereafter we exhibit some functions highly degraded by a weightwise restriction.

2.1 Balancedness degradation in fixed input weight framework

First we consider the behavior of balanced functions, or highly resilient functions, compared to weightwise perfectly balanced functions as defined in Section 1.2.1. Weightwise perfectly

balancedness implies balancedness over all $\mathbb{F}_2^n$ whereas the inverse is false, as illustrated by the next Remark on highly resilient functions.

Remark 1. For all $n \geq 2$, there exists an $(n-1)$-resilient function (i.e. a balanced Boolean function which remains balanced when at most $n-1$ of its variables are arbitrarily fixed) which is unbalanced for all weight $k \in [1, n-1]$.
Indeed, the first elementary symmetric Boolean function $\sigma_1 = \sum_{i=1}^{n} x_i = \mathsf{wH}(x) \pmod 2$ is $(n-1)$-resilient and is constant on all fixed weight input, its weightwise restrictions are as much unbalanced as possible.

2.2 Nonlinearity degradation in fixed input weight framework

Fixing the input Hamming weight may deteriorate in an extreme way the nonlinearity of a Boolean function.

Proposition 1. *For every n, there exist n-variable bent functions f such that, for every $k = 0, \ldots, n$, $\mathsf{NL}_{E_{n,k}}(f) = 0$.*

Proof: This is for instance the case of the function $f(x) = \binom{\mathsf{wH}(x)}{2} = \sum_{1 \leq i < j \leq n} x_i x_j$. This function is, up to the addition of an affine function, the only bent symmetric function (see e.g. [Car10]). Since it is symmetric, fixing the Hamming weight of its input makes it constant and therefore with null nonlinearity. □

More generally, it would be interesting to characterize those bent functions whose restrictions to $E_{n,k}$ have null nonlinearity (i.e. are affine), for every k. This task seems very difficult but we are able to achieve it in the particular case of quadratic functions. We begin with an observation:

Remark 2. A Boolean function satisfies $\mathsf{NL}_{E_{n,k}}(f) = 0$ for every k, i.e. has all its restrictions to $E_{n,k}$ affine, if and only if there exist symmetric Boolean functions $\varphi_0, \varphi_1, \ldots, \varphi_n$ such that $f(x) = \varphi_0(x) + \sum_{i=1}^{n} \varphi_i(x)\, x_i$. Any symmetric Boolean functions $\varphi(x)$ can be written in the form $\ell \circ \Sigma(x)$ where ℓ is affine and Σ is the vectorial (n,n)-function whose ith coordinate function is the elementary symmetric function $\sum_{1 \leq j_1 < \cdots < j_i \leq n} \prod_{l=1}^{i} x_{j_l}$. We deduce that f satisfies $\mathsf{NL}_{E_{n,k}}(f) = 0$ for every k if and only if it has the form $f(x) = \ell_0 \circ \Sigma(x) + \sum_{i=1}^{n} \ell_i \circ \Sigma(x) x_i$, where the ℓ_i's are affine. In other words (after gathering all the terms in this expression which involve each elementary symmetric function σ_i):

$$f(x) = \ell'_0(x) + \sum_{i=1}^{n} \sigma_i(x) \ell'_i(x),$$

where the ℓ'_i's are all affine.

Then we have:

Proposition 2. *For every even $n \geq 4$, the quadratic bent functions satisfying $\mathsf{NL}_{E_{n,k}}(f) = 0$ for every k are those functions of the form $f(x) = \sigma_1(x)\ell(x) + \sigma_2(x)$ where $\ell(1, \ldots, 1) = 0$.*

Proof: According to Remark 2, a quadratic function satisfies $\mathsf{NL}_{E_{n,k}}(f) = 0$ for every k if and only if, up to the addition of an affine function, it has the form:

$$f(x) = \sigma_1(x)\ell(x) + \epsilon\sigma_2(x)$$

where ℓ is linear and $\epsilon \in \mathbb{F}_2$. The symplectic form associated $(x, y) \to f(x+y) + f(x) + f(y) + f(0)$ (see e.g. [Car10]) equals:

$$\sigma_1(y)\ell(x) + \sigma_1(x)\ell(y) + \epsilon \sum_{1 \leq j \neq i \leq n} x_j y_i.$$

Denoting $\ell(x) = \sum_{i=1}^{n} l_i x_i$, the kernel

$$E = \{x \in F_2^n; \forall y \in F_2^n, f(x+y) + f(x) + f(y) + f(0) = 0\}$$

of this symplectic form is the vector space of equations:

$$(L_i) : \ell(x) + l_i \sum_{j=1}^{n} x_j + \epsilon \sum_{j \neq i} x_j = 0,$$

where i ranges from 1 to n. The sum $L_i + L_{i'}$ of two of these equations equals

$$(L_i + L_{i'}) : (l_i + l_{i'}) \sum_{j=1}^{n} x_j + \epsilon(x_i + x_{i'}) = 0.$$

If $l_i = l_{i'}$ we obtain: $\forall x \in E$, $x_i = x_{i'}$ if $\epsilon = 1$ and no condition on $x \in E$ otherwise. If $l_i \neq l_{i'}$, we obtain: $\forall x \in E$, $\sum_{j=1}^{n} x_j = x_i + x_{i'}$ if $\epsilon = 1$ and $\forall x \in E$, $\sum_{j=1}^{n} x_j = 0$ otherwise. Hence, denoting

$$I = \{i = 1, \ldots, n; l_i = 0\},$$

we have that, if $\epsilon = 1$, then all the coordinates of indices $i \in I$ of an element of E are equal to some bit η and all those such that $i \in I^c$ are equal to $\eta + \sum_{j=1}^{n} x_j$, and if $\epsilon = 0$, there is no condition on $x \in E$ if $I = \emptyset$ or $I = \{1, \ldots, n\}$ and if $I \neq \emptyset, \{1, \ldots, n\}$, the condition is $\sum_{j=1}^{n} x_j = 0$. We then have two cases:

- if $x \in E$ is such that $\sum_{j=1}^{n} x_j = 0$ then:
  - if $\epsilon = 1$, then either all x_i's are null, in which case (L_i) is satisfied, or all are equal to 1, in which case (L_i) becomes (since n is even) $\ell(1, \ldots, 1) = 1$; hence, if this latter equality is true (i.e. if I has odd cardinality), $E \neq \{0\}$;
  - if $\epsilon = 0$ then all equations L_i are equal to $\ell(x) = 0$; then $E \neq \{0\}$ unless the hyperplane $\ker \ell$ has a trivial intersection with the hyperplane of equation $\sum_{j=1}^{n} x_j = 0$, which is possible only if $n = 2$; the case $\epsilon = 0$ is then compatible with f bent only for $n = 2$; we shall not consider it anymore.
- if $x \in E$ is such that $\sum_{j=1}^{n} x_j = 1$ then if $\epsilon = 1$, all x_i's such that $i \in I$ are equal to η and those x_i's such that $i \in I^c$ are equal to $\eta + 1$, which implies $\eta|I| + (\eta + 1)|I^c| = |I^c| = 1 \pmod 2$; hence I has odd cardinality and we have seen that $E \neq \{0\}$ in such case.

The only case where f is bent, *i.e.* where $E = \{0\}$, for $n \geq 4$, is then $\begin{cases} \epsilon = 1 \\ \ell(1, \ldots, 1) = 0 \end{cases}$. □

Two n-variable Boolean functions f and g are called EA-equivalent if there exist an affine automorphism L over $\mathbb{F}_2^n$ and an affine n-variable function ℓ such that $f = g \circ L + \ell$. All the functions above are EA-equivalent to each others, since all quadratic bent functions are EA-equivalent to each others, but EA-equivalence is not preserving the Hamming weight, so the nonlinearity degradation with weightwise consideration cannot be seen equivalence class by equivalence class.

2.3 Algebraic immunity degradation in fixed input weight framework

The majority function is well-known for its optimal algebraic immunity and, as all symmetric functions, is constant on all inputs of the same Hamming weight. Therefore it is a trivial example where the algebraic immunity collapses in our context. To go further we investigate the algebraic immunity of direct sums of functions.

The so-called *direct sum* is a well-known secondary construction of Boolean functions which on the entire space $\mathbb{F}_2^n$, enables to guarantee some algebraic immunity of a function based one two fonctions on a smaller number of variables, we prove here that it behaves differently when the inputs are restricted to a fixed Hamming weight.

Definition 5 (Direct Sum)**.** Let f be a Boolean function of n variables and g a Boolean function of m variables, f and g depending in distinct variables, the direct sum h of f and g is defined by:

$$h(x,y) = f(x) + g(y), \quad \text{where } x \in \mathbb{F}_2^n \text{ and } y \in \mathbb{F}_2^m.$$

Theorem 1. *[Link between* AI_k *and* AI *in direct sum] Let F be the direct sum of f and g with n and m variables respectively. Let k be such that $n \leq k \leq m$. Then the following relation holds:*

$$\mathsf{AI}_k(F) \geq \mathsf{AI}(f) - \deg(g).$$

Proof: Let $h(x,y)$ be a non-null annihilator of F over $E_{n+m,k}$. Let $(a,b) \in \mathbb{F}_2^{n+m}$ have Hamming weight k and be such that $h(a,b) = 1$. Since (a,b) has Hamming weight k, we may, up to changing the order of the coordinates of b (and without loss of generality), assume that for every $j = 1, \ldots, n$, we have $b_j = a_j + 1$ and for every $j = n+1, \ldots k$, we have $b_j = 1$ (so that for every $j = k+1, \ldots m$, we have $b_j = 0$). We define the following affine function over $\mathbb{F}_2^n$:

$$L(x) = (x_1 + 1, x_2 + 1, \ldots, x_n + 1, 1, \ldots, 1, 0, \ldots, 0),$$

where the length of the part "$1, \ldots, 1$" equals $k - n$. We have $L(a) = b$. The n-variable function $h(x, L(x))$ is then non-zero and is an annihilator of $f(x) + g(L(x))$ over $\mathbb{F}_2^n$. If $g(b) = 0$, then function $h(x, L(x))\,(g(L(x)) + 1)$ is a non-zero annihilator of f and has algebraic degree at most $\deg(h) + \deg(g)$; then we have $\deg(h) + \deg(g) \geq AI(f)$. If $g(b) = 1$, then by applying the same reasoning to $f + 1$ instead of f and $g + 1$ instead of g, we have $\deg(h) + \deg(g) \geq AI(f)$. If $h(x,y)$ is a non-null annihilator of $F + 1$ over $E_{n+m,k}$, we have the same conclusion by replacing f by $f + 1$ or g by $g + 1$. This completes the proof. □

This bound proves in particular that, if $k \geq n$, then adding $m \geq k$ virtual variables to a function (taking $g = 0$) does not lower the algebraic immunity with inputs of Hamming weight k with respect to the (global) original algebraic immunity. This was already true (with no condition on n, k, m) when dealing with functions with no restriction on the input and it was completely straightforward to prove it, while here it was less obvious. Note that the bound of Theorem 1 is tight when $\deg(g) = 0$: take for f a function whose algebraic immunity equals its algebraic degree; we have then that $\mathsf{AI}_k(F)$ equals $\mathsf{AI}(f) = \deg(f)$, since it cannot be larger than the algebraic degree of f over $E_{n,k}$ (formally proved in the Corollary 5) which is at most equal to $\deg(f)$; the three parameters $\mathsf{AI}_k(F)$, the algebraic degree of f over $E_{n,k}$ and $\deg(f)$ are then equal.

Nevertheless, the bound of Theorem 1 also suggests that making the direct sum with a non-constant Boolean function g may lower the algebraic immunity over inputs of Hamming weight k with respect to the (global) original algebraic immunity. This may seem rather counter-intuitive, but it is true. Let us give an example: take $f(x_1, x_2, x_3) = x_1 + x_2x_3$; $g(x_4, \ldots, x_{10}) = \sum_{i=4}^{10} x_i$ and $k = 5$, then $\mathsf{AI}(f) = \deg(f) = 2$, $\mathsf{AI}(f) - \deg(g) = 1$, and x_2 being an annhilator of $f(x_1, x_2, x_3) + g(x_4, \ldots, x_{10})$ over inputs of Hamming weight 5, because $x_2(f(x_1, x_2, x_3) + g(x_4, \ldots, x_{10})) = x_2(1 + \sum_{i=1}^{10} x_i)$ vanishes when the input has weight 5, we have $\mathsf{AI}_5(f(x_1, x_2, x_3) + g(x_4, \ldots, x_{10})) = 1$; the bound is then tight here. In fact, making the direct sum with a non-constant Boolean function g may decrease drastically the algebraic immunity over inputs of Hamming weight k: take n odd, $f(x) = 1 + maj(x)$ where maj is the majority function over n variables (which has optimal algebraic immunity

$\frac{n+1}{2}$) and $g(y) = maj(y)$ over n variables as well. Then $F(x,y) = f(x) + g(y)$ is null at fixed input weight n, because if $\mathsf{w_H}(x) + \mathsf{w_H}(y) = n$ then either $\mathsf{w_H}(x) \leq \frac{n-1}{2}$ and $\mathsf{w_H}(y) \geq \frac{n+1}{2}$ or $\mathsf{w_H}(x) \geq \frac{n+1}{2}$ and $\mathsf{w_H}(y) \leq \frac{n-1}{2}$. We fall then down to a null algebraic immunity with input weight n (however, the bound is not tight here because the algebraic degree of maj is in general strictly larger than its algebraic immunity).

3 General study of restricted inputs criteria, and constructions

3.1 Balancedness

In this part we study the criterion of balancedness with weightwise consideration; we first determine the necessary conditions for a function to be weightwise perfectly balanced, then we construct such functions and finally we describe a new transform adapted to weightwise balancedness.

3.1.1 Relation with ANF

Recall that any Boolean function over $\mathbb{F}_2^n$ has a unique algebraic normal form (ANF) $f(x) = \sum_{I \subseteq \{1,\ldots,n\}} a_I \prod_{i \in I} x_i$, where $a_I \in \mathbb{F}_2$. Any term $\prod_{i \in I} x_i$ in such ANF is called a *monomial* and its degree equals $|I|$. The algebraic degree of f equals the global degree $\max_{I;a_I=1} |I|$ of its ANF. The function f is affine if and only if its algebraic degree is at most 1.
In the following, we give more insights on necessary conditions on the ANF of *WPB* or *WAPB* (recall definitions in Section 1.2.1).

Remark 3. For every even n and $\varepsilon = 0$ or 1, function $\ell(x_1, x_2, \ldots, x_n) = \varepsilon + x_1 + x_2 + \cdots + x_{\frac{n}{2}}$ is balanced on all words of fixed odd Hamming weight, since for such word, either $\mathsf{w_H}(x_1, \ldots, x_{\frac{n}{2}})$ is odd and $\ell(x_1, x_2, \ldots, x_n) = \varepsilon + 1$, or $\mathsf{w_H}(x_{\frac{n}{2}+1}, \ldots, x_n)$ is odd and $\ell(x_1, x_2, \ldots, x_n) = \varepsilon$, and the words of the former kind are the shifted by $\frac{n}{2}$ positions of the words of the latter kind and are then no more and no less numerous. Conversely, any affine function balanced on words of Hamming weight 1 has the form $\varepsilon + x_{i_1} + x_{i_2} + \cdots + x_{i_{\frac{n}{2}}}$, where $\varepsilon = 0$ or 1. Any weightwise perfectly balanced Boolean function has then the following form :

$$f(x_1, x_2, \ldots, x_n) = \varepsilon + x_{i_1} + x_{i_2} + \cdots + x_{i_{\frac{n}{2}}} + g(x_1, x_2, \ldots, x_n),$$

where g is non null and is the sum of monomials of degrees at least 2, since all monomials of degree at least 2 vanish at inputs of Hamming weight 1.

More precisely, we can derive necessary conditions of the ANF of f for weightwise perfectly balancedness:

Proposition 3. *If f is a weightwise (almost) perfect Boolean function of n variables then the ANF of f contains $\lceil n/2 \rfloor$ monomials of degree 1 and at least $\lfloor n/4 \rfloor$ monomials of degree 2, where $\lceil n/2 \rfloor$ equals $n/2$ if n is even and $(n \pm 1)/2$ if n is odd.*

Proof: In the particular case where f is linear, $\mathsf{w_H}(f)_k$ is exactly the number of entries of weight k for which an odd number of the monomials of f are set to 1. Therefore denoting by d the number of (degree 1) monomials in the ANF of f, we have: $\mathsf{w_H}(f)_k = \sum_{i \text{ odd}} \binom{d}{i}\binom{n-d}{k-i}$. For any function f, as $\mathsf{w_H}(f)_k$ is only determined by the monomials of f of degree at most k, let us partition f into ℓ_f, q_f and f', respectively made of the monomials of degree 1, 2 and strictly larger than 2 in the ANF of f. For $k = 1$, we have:

$$\mathsf{w_H}(f)_1 = \mathsf{w_H}(\ell_f)_1 = \binom{|\ell_f|}{1},$$

where $|\ell_f|$ is the number of monomials of ℓ_f.
Therefore, if f is (almost) balanced for fixed weight 1, then $|\ell_f| = \frac{n}{2}$ for n even and $|\ell_f| = \frac{n\pm1}{2}$ for n odd. We have:

$$\mathsf{w_H}(f)_2 = \mathsf{w_H}(\ell_f + q_f)_2 = \mathsf{w_H}(\ell_f)_2 + \mathsf{w_H}(q_f)_2 - 2\mathsf{w_H}(\ell_f \cdot q_f)_2,$$

Therefore, if f is (almost) balanced for fixed weights 1 and 2, then, for n even, $\mathsf{w_H}(\ell_f)_2 = \binom{\frac{n}{2}}{1}\binom{\frac{n}{2}}{1} = \frac{n^2}{4}$ and $\mathsf{w_H}(\ell_f)_2 - \frac{\binom{n}{2}}{2} = \frac{n}{4}$ so $\mathsf{w_H}(q_f)_2 \geq \lfloor n/4 \rfloor$, and for n odd, $\mathsf{w_H}(\ell_f)_2 = \binom{\frac{n+1}{2}}{1}\binom{\frac{n-1}{2}}{1} = \frac{n^2-1}{4}$ and $\mathsf{w_H}(\ell_f)_2 - \frac{\binom{n}{2}}{2} = \frac{n-1}{4}$ so $\mathsf{w_H}(q_f)_2 \geq \lfloor n/4 \rfloor$.

□

Proposition 4. *If f is a weightwise perfectly balanced Boolean function of n variables, then the ANF of f contains at least one monomial of degree $n/2$.*

Proof: Let m_d be a monomial of degree d, we focus on the parity of $\mathsf{w_H}(m_d)_k$; for all $1 \leq k \leq n-1$ and $1 \leq d \leq k$:

$$\mathsf{w_H}(m_d)_k = \binom{n-d}{k-d}$$

More particularly when $k = d$, $\mathsf{w_H}(m_k)_k = \binom{n-k}{0} = 1$. We have seen that f being perfectly balanced implies that $n = 2^\ell$ and therefore $\mathsf{w_H}(f)_k = \frac{\binom{2^\ell}{k}}{2}$ is even for $k \in [1, \ldots, 2^{\ell-1} - 1] \cup [2^{\ell-1} + 1, \ldots, 2^\ell - 1]$ and odd for $k = 2^{\ell-1} = n/2$. This enables to determine the parity of the number of monomials of each degree of f smaller than or equal to $2^{\ell-1} = n/2$. Concretely, f has an even number of monomials of degree d for $1 \leq d \leq n/2 - 1$ (by induction at weight $k = d$ this number has to be even due to $\mathsf{w_H}(m_k)_k = 1$) and an odd number of monomials of degree $n/2$, finishing the proof.

□

3.1.2 Constructions

The direct sum construction (see Definition 5) can be a starting point to build weightwise perfectly balanced function. This secondary construction does not build a weightwise perfectly balanced function from two weightwise perfectly balanced functions as we can see from the next Lemma and Corollary.

Lemma 1. *Let f be a Boolean function with $n = 2^\ell$ variables ($\ell \in \mathbb{N}^*$) such that there exist two Boolean functions g_1 and g_2 in $\frac{n}{2}$ variables such that $f(x_1, \ldots, x_n) = g_1(x_1, \ldots, x_{\frac{n}{2}}) + g_2(x_{\frac{n}{2}+1}, \ldots, x_n)$, and such that $g_1(0\ldots0)+g_1(1\ldots1)+g_2(0\ldots0)+g_2(1\ldots1) \equiv 0 \mod 2$, then f cannot be weightwise perfectly balanced.*

Proof: Let f be a Boolean function such that f is a direct sum of two Boolean functions g_1 and g_2 of n_1 and n_2 variables. As f is a direct sum of g_1 and g_2, we can link the value of $\mathsf{w_H}(f)_k$ to $\mathsf{w_H}(g_1)_i$ and $\mathsf{w_H}(g_2)_i$ with $i \leq k$ for every $k \in [1, n-1]$. First we do a partition of the entries of f of Hamming weight k depending on the Hamming weight of the entries of g_1 and g_2, this gives a partition of $k+1$ sets where g_1 is evaluated on $E_{n_1,i}$ and g_2 is evaluated on $E_{n_2,k-i}$. Then $f(x_1, \ldots, x_n) = 1$ is equivalent to $g_1(x_1, \ldots, x_{n_1}) \neq g_2(x_{n_1+1}, \ldots, x_n)$, so we can link $\mathsf{w_H}(f)_k$ to the number of entries where g_1 gives 1 and g_2 gives 0 plus the number of entries where g_1 gives 0 and g_2 gives 1. Finally we obtain:

$$\mathsf{w_H}(f)_k = \sum_{i=0}^{k} \mathsf{w_H}(g_1)_i \left(\binom{n_2}{k-i} - \mathsf{w_H}(g_2)_{k-i} \right) + \mathsf{w_H}(g_2)_{k-i} \left(\binom{n_1}{i} - \mathsf{w_H}(g_1)_i \right)$$

Now we suppose that f is weightwise perfectly balanced and we use that $n_1 = n_2 = \frac{n}{2}$; in particular, $\mathsf{w_H}(f)_{\frac{n}{2}} = \frac{1}{2}\binom{n}{\frac{n}{2}} \equiv 1 \mod 2$ and developing:

$$\mathsf{w_H}(f)_{\frac{n}{2}} = \sum_{i=0}^{\frac{n}{2}} \binom{\frac{n}{2}}{\frac{n}{2}-i} \mathsf{w_H}(g_1)_i + \binom{\frac{n}{2}}{i} \mathsf{w_H}(g_2)_{\frac{n}{2}-i} - 2\mathsf{w_H}(g_1)_i \mathsf{w_H}(g_2)_{\frac{n}{2}-i}$$

Moreover, we know that, as $\frac{n}{2}$ is also a power of 2, then for each $i \in [1, \frac{n}{2}-1]$, $\binom{\frac{n}{2}}{i}$ is even. To conclude, if f is weightwise perfectly balanced, then we have the following relation:

$$1 \equiv \mathsf{w_H}(g_1)_0 + \mathsf{w_H}(g_1)_{\frac{n}{2}} + \mathsf{w_H}(g_2)_0 + \mathsf{w_H}(g_2)_{\frac{n}{2}} \mod 2$$

Then we need that $g_1(0\ldots0) + g_1(1\ldots1) + g_2(0\ldots0) + g_2(1\ldots1) \equiv 1 \mod 2$ □

The corollary below is a direct consequence:

Corollary 1. *If $g_1(x_1, \ldots, x_{\frac{n}{2}})$ and $g_2(x_{\frac{n}{2}+1}, \ldots, x_n)$ are two weightwise perfectly balanced functions, then the Boolean function defined by the direct sum of g_1 and g_2 cannot be weightwise perfectly balanced.*

Hence, the direct sum, when applied to perfectly balanced functions, does not lead to a weightwise perfectly balanced function; nevertheless we can derive such construction from weightwise perfectly balanced functions by applying the direct sum after modifying one of the functions: if f and g are two n-variable weightwise perfectly balanced functions, then $h(x,y) = f(x) + \prod_{i=1}^{n} x_i + g(y)$ is a $2n$-variable weightwise perfectly balanced function. In fact, this result is a particular case of a more general construction, inspired by the so-called indirect sum, which builds a Boolean function from four Boolean functions as follows: $h(x,y) = f(x) + g(y) + (f(x) + f'(x))(g(y) + g'(y))$, and which allowed to construct bent and correlation immune functions:

Theorem 2. *Let f, f' and g be three weightwise perfectly balanced n-variable functions and let g' be any n-variable Boolean function, then $h(x,y) = f(x) + \prod_{i=1}^{n} x_i + g(y) + (f(x) + f'(x))g'(y)$, where $x, y \in \mathbb{F}_2^n$, is a weightwise perfectly balanced $2n$-variable function.*

Proof:

- If $k = 0$, then $\mathsf{w_H}(x,y) = k$ is equivalent to $x = y = (0, \ldots, 0)$ and we have $h(x,y) = f(0,\ldots,0) + g(0,\ldots,0) = 0$.
- If $k \in \{1, \ldots, n-1\}$, then, the set $\{(x,y) \in \mathbb{F}_2^{2n}; \mathsf{w_H}(x,y) = k\}$ equals the disjoint union of the following sets:
  - $\{(0,\ldots,0)\} \times \{y \in \mathbb{F}_2^n; \mathsf{w_H}(y) = k\}$, on which $h(x,y)$ equals $f(0,\ldots,0) + g(y)$ (since $f(0,\ldots,0) + f'(0,\ldots,0) = 0$) and is then balanced;
  - $\{x \in \mathbb{F}_2^n; \mathsf{w_H}(x) = i\} \times \{y\}$, where $1 \le i \le k$ and $\mathsf{w_H}(y) = k - i$, on each of which $h(x,y)$ equals $f(x) + g(y)$ if $g'(y) = 0$ and $f'(x) + g(y)$ if $g'(y) = 1$; in both cases, it is balanced;
- If $k = n$, then the set $\{(x,y) \in \mathbb{F}_2^{2n}; \mathsf{w_H}(x,y) = k\}$ equals the disjoint union of the following sets:
  - $\{((0,\ldots,0),(1,\ldots,1))\} \cup \{((1,\ldots,1),(0,\ldots,0))\}$, on which $h(x,y)$ equals respectively $f(0,\ldots,0) + g(1,\ldots,1) = 1$ (since $f(0,\ldots,0) + f'(0,\ldots,0) = 0$) and $f(1,\ldots,1) + g(0,\ldots,0) + 1 = 0$ (since $f(1,\ldots,1) + f'(1,\ldots,1) = 0$) and is then globally balanced;
  - $\{x \in \mathbb{F}_2^n; \mathsf{w_H}(x) = i\} \times \{y\}$, where $1 \le i \le n-1$ and $\mathsf{w_H}(y) = n - i$, on each of which $h(x,y)$ equals $f(x) + g(y)$ if $g'(y) = 0$ and $f'(x) + g(y)$ if $g'(y) = 1$; in both cases, it is balanced;

- If $k \in \{n+1, \ldots, 2n-1\}$, then the set $\{(x,y) \in \mathbb{F}_2^{2n}; \mathsf{w_H}(x,y) = k\}$ equals the disjoint union of the following sets:
  - $\{(1,\ldots,1)\} \times \{y \in \mathbb{F}_2^n; \mathsf{w_H}(y) = k-n\}$, on which $h(x,y)$ equals $f(1,\ldots,1) + g(y) + 1$ and is then balanced;
  - $\{x \in \mathbb{F}_2^n; \mathsf{w_H}(x) = i\} \times \{y\}$, where $k-n+1 \leq i \leq n-1$ and $\mathsf{w_H}(y) = k-i$, on each of which $h(x,y)$ equals $f(x)+g(y)$ if $g'(y) = 0$ and $f'(x)+g(y)$ if $g'(y) = 1$; in both cases, it is balanced;
- If $k = 2n$, then $\mathsf{w_H}(x,y) = k$ is equivalent to $x = y = (1,\ldots,1)$ and we have $h(x,y) = 1+1+1 = 1$.

□

Note that for $f = f'$ or $g' = 0$, we obtain the construction related to the direct sum mentioned above. Noting that $f(x_1,x_2) = x_1$ is perfectly balanced, we can recursively build perfectly balanced Boolean functions of 2^ℓ variables, for all ℓ in $\mathbb{N}^*$. For instance, applying the construction with $f = f'$, we get:

$$f(x_1, x_2, \ldots, x_{2^\ell}) = \sum_{a=1}^{\ell} \sum_{i=1}^{2^{\ell-a}} \prod_{j=0}^{2^{a-1}-1} x_{i+j2^{\ell-a+1}}$$

And since g' can be freely chosen and f' can be a version of f in which the coordinates of x are permuted, we have a large number of weightwise perfectly balanced functions by applying Theorem 2.

We can extend the previous example to get weightwise almost perfectly balanced Boolean function on n variables for all n.

Proposition 5. *The function f_n in $n \geq 2$ variables, recursively defined by $f_2(x_1,x_2) = x_1$ and for $n \geq 3$: $f_n(x_1,\ldots,x_n) =$*

$$\begin{cases} f_{n-1}(x_1,\ldots,x_{n-1}) & \text{if } n \text{ odd} \\ f_{n-1}(x_1,\ldots,x_{n-1}) + x_{n-2} + \prod_{i=1}^{2^{d-1}} x_{n-i} & \text{if } n = 2^d; d > 1 \\ f_{n-1}(x_1,\ldots,x_{n-1}) + x_{n-2} + \prod_{i=1}^{2^d} x_{n-i} & \text{if } n = p \cdot 2^d; p > 1 \text{ odd}\,; d \geq 1 \end{cases}$$

is a weightwise almost perfectly balanced Boolean function of degree 2^{d-1}, where $2^d \leq n < 2^{d+1}$, and with $n-1$ monomials in its ANF if n is even and $n-2$ monomials if n is odd. Note that this function can be written as a direct sum for all $n \geq 2$.

Proof: The degree and number of monomials of f_n are easily checked by induction on n for $n \geq 2$. We prove the weightwise almost perfect balance property by induction on n as well:

- for $n = 2$, $f_2 = x_1$ is *WPB*.
  We now assume that $n \geq 3$ and that, for every $2 \leq i \leq n-1$, f_i is *WAPB*. We prove under this induction hypothesis that f_n is *WAPB*.
- for n odd:
  - if $k = 0$, then $\mathsf{w_H}(f_n)_0 = \mathsf{w_H}(f_{n-1})_0 = 0$;
  - if $k \in [1, n-1]$, then $\mathsf{w_H}(f_n)_k = \mathsf{w_H}(f_{n-1})_k + \mathsf{w_H}(f_{n-1})_{k-1}$. As $n-1$ is even, at least one of the coefficients $\binom{n-1}{k}, \binom{n-1}{k-1}$ is even (as $n-1$ is even and k or $k-1$ is odd therefore one of those written in binary has a digit equal to 1 where the corresponding one of n is 0 which characterize the even parity of this binomial coefficient), therefore $\mathsf{w_H}(f_{n-1})_k + \mathsf{w_H}(f_{n-1})_{k-1} = \frac{\binom{n-1}{k}+\binom{n-1}{k-1}}{2} = \frac{\binom{n}{k}}{2}$ if both are even and $\mathsf{w_H}(f_{n-1})_k + \mathsf{w_H}(f_{n-1})_{k-1} = \frac{\binom{n-1}{k}+\binom{n-1}{k-1}\pm 1}{2} = \frac{\binom{n}{k}\pm 1}{2}$ otherwise;

– if $k = n$, then $\mathsf{w_H}(f_n)_n = \mathsf{w_H}(f_{n-1})_{n-1} = 1$

Hence, f_n is *WAPB*.

- for $n = 2^d; d > 1$, we can view f_n as the following direct sum:

$$f_n(x_1, \ldots, x_n) =$$

$$f_{2^{d-1}}(x_1, \cdots, x_{2^{d-1}-1}, x_n) + f_{2^{d-1}}(x_{2^{d-1}}, \cdots, x_{n-1}) + \prod_{i=1}^{2^{d-1}} x_{n-i}.$$

As $f_{2^{d-1}}$ is *WPB* by hypothesis, we can apply Theorem 2 with $g' = 0$, giving that f_n is *WPB*.

- $n = p \cdot 2^d; 1 < p$ odd ; we decompose f_n in a direct sum and use techniques of Theorem 2's proof:

$$f_n(x_1, \ldots, x_n) =$$

$$f(x_1, \cdots, x_{n-2^d-1}, x_n) + g(x_{n-2^d}, \cdots, x_{n-1}) + \prod_{i=1}^{2^d} x_{n-i}$$

reordering the variables we get $f = f_{n-2^d}$ and $g = f_{2^d}$, with f_{2^d} *WPB* and f_{n-2^d} *WAPB* by hypothesis. f_n being a direct sum of f and $g + \prod_{i=1}^{2^d} x_{n-i}$ we get:

– if $k = 0$: $\mathsf{w_H}(f_n)_0 = \mathsf{w_H}(f)_0\overline{\mathsf{w_H}(g)_0} + \overline{\mathsf{w_H}(f)_0}\mathsf{w_H}(g)_0 = 0$

– if $k \in [1, 2^d - 1]$:

$$\mathsf{w_H}(f_n)_k = \sum_{i=0}^{k} \mathsf{w_H}(g)_i\overline{\mathsf{w_H}(f)_{k-i}} + \overline{\mathsf{w_H}(g)_i}\mathsf{w_H}(f)_{k-i} \tag{1}$$

$$= \mathsf{w_H}(f)_k + \sum_{i=1}^{k} \mathsf{w_H}(g)_i(\overline{\mathsf{w_H}(f)_{k-i}} + \mathsf{w_H}(f)_{k-i}) \tag{2}$$

$$= \mathsf{w_H}(f)_k + \frac{1}{2}\sum_{i=1}^{k} \binom{2^d}{i}\binom{n-2^d}{k-i} \tag{3}$$

$$= \mathsf{w_H}(f)_k + \frac{1}{2}\left(\binom{n}{k} - \binom{n-2^d}{k}\right) \tag{4}$$

Equation 2 comes from g being WPB of 2^d variables, therefore $\overline{\mathsf{w_H}(g)_i} = \mathsf{w_H}(g)_i$ for $i \in [1, 2^d - 1$. Equation 3 is obtained using that $\mathsf{w_H}(f)_{k-i} + \overline{\mathsf{w_H}(f)_{k-i}} = \binom{n-2^d}{k-i}$ by definition and $\mathsf{w_H}(g)_i = \frac{1}{2}\binom{2^d}{i}$ because f is a *WPB* function. Equation 4 is obtained Vandermonde convolution: $\sum_{i=0}^{k}\binom{n}{i}\binom{m}{k-i} = \binom{n+m}{k}$.

Therefore $\mathsf{w_H}(f_n)_k = \frac{1}{2}\binom{n}{k}$ if $\binom{n-2^d}{k}$ is even and $\mathsf{w_H}(f_n)_k = \frac{1}{2}\left(\binom{n}{k} \pm 1\right)$ otherwise.

– if $k \in [2^d, n - 1]$:

$$\begin{aligned}
\mathsf{w_H}(f_n)_k &= \sum_{i=1}^{2^d-1} \mathsf{w_H}(g)_i\overline{\mathsf{w_H}(f)_{k-i}} + \overline{\mathsf{w_H}(g)_i}\mathsf{w_H}(f)_{k-i} \\
&+ \mathsf{w_H}(g)_0\overline{\mathsf{w_H}(f)_k} + \overline{\mathsf{w_H}(g)_0}\mathsf{w_H}(f)_k \\
&+ \overline{\mathsf{w_H}(g)_{2^d}\mathsf{w_H}(f)_{k-2^d}} + \mathsf{w_H}(g)_{2^d}\mathsf{w_H}(f)_{k-2^d} \\
&= \sum_{i=1}^{2^d-1} \frac{1}{2}\binom{2^d}{i}\binom{n-2^d}{k-i} + \mathsf{w_H}(f)_k + \mathsf{w_H}(f)_{k-2^d} \\
&= \frac{1}{2}\left(\binom{n}{k} - \binom{n-2^d}{k} - \binom{n-2^d}{k-2^d}\right) + \mathsf{w_H}(f)_k + \mathsf{w_H}(f)_{k-2^d}
\end{aligned}$$

As $n - 2^d \equiv 0[2^{d+1}]$ at least one of $\binom{n-2^d}{k}, \binom{n-2^d}{k-2^d}$ is even therefore $\mathsf{w_H}(f_n)_k = \frac{1}{2}\binom{n}{k}$ if both are even and $\mathsf{w_H}(f_n)_k = \frac{1}{2}\left(\binom{n}{k} \pm 1\right)$ otherwise.

– if $k = n$: $\mathsf{w_H}(f_n)_n = \mathsf{w_H}(f)_{n-2^d}\mathsf{w_H}(g)_{2^d} + \overline{\mathsf{w_H}(f)_{n-2^d}\mathsf{w_H}(g)_{2^d}} = 1$ Giving that f_n is *WAPB*.

To conclude for $n \geq 2$, f_n is weightwise (almost) perfectly balanced.

□

3.1.3 A Walsh-like transform involving symmetric functions and handling balance with fixed input weight.

For $i \in \{1, \ldots, n\}$, let us recall that σ_i denotes the ith elementary symmetric Boolean function:

$$\sigma_i(x) = \sum_{1 \leq j_1 < \cdots < j_i \leq n} \prod_{l=1}^{i} x_{j_l}$$

(sum performed in $\mathbb{F}_2$) and Σ the vectorial (n, n)-function whose ith coordinate function is σ_i.

Lemma 2. *For $k \in \{1, \ldots, n\}$, we have $\mathsf{w_H}(x) = k$ if and only if, for every $i = 1, \ldots, n$, we have $\sigma_i(x) = \binom{k}{i} \pmod 2$.*

Indeed, the σ_i's generate by linear combinations all those symmetric Boolean functions which are null at input 0, and we know that two vectors x, y have the same nonzero Hamming weight if and only if every symmetric Boolean function null at input 0 takes the same value at inputs x and y (indeed, the indicator of the set of those vectors of some nonzero Hamming weight k is a symmetric function null at input 0). We have then:

$$\begin{aligned}
\binom{n}{k} - 2\ \mathsf{w_H}(f_{|\mathsf{w_H}(x)=k}) &= \sum_{x \in \mathbb{F}_2^n;\, \mathsf{w_H}(x)=k} (-1)^{f(x)} \\
&= \sum_{\substack{x \in \mathbb{F}_2^n;\, \forall i=1,\ldots,n, \\ \sigma_i(x)=\binom{k}{i} \pmod 2}} (-1)^{f(x)} \\
&= 2^{-n} \sum_{v \in \mathbb{F}_2^n} (-1)^{\sum_{i=1}^n v_i\binom{k}{i}} \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+v\cdot\Sigma(x)}.
\end{aligned}$$

Indeed, for every $x \in \mathbb{F}_2^n$, we have:

$$\begin{aligned}\sum_{v\in\mathbb{F}_2^n}(-1)^{\sum_{i=1}^n v_i\binom{k}{i}+v\cdot\Sigma(x)} &= \sum_{v\in\mathbb{F}_2^n}(-1)^{\sum_{i=1}^n v_i[\binom{k}{i}+\sigma_i(x)]}\\ &= \begin{cases} 2^n \text{ if } \sigma_i(x)=\binom{k}{i} \pmod 2, \forall i=1,\ldots,n,\\ 0 \text{ otherwise.}\end{cases}\end{aligned}$$

With the same notation, we have $\mathsf{w_H}(x)=\mathsf{w_H}(y)$ if and only if $\Sigma(x)=\Sigma(y)$, and we have then:

$$\begin{aligned}\frac{1}{n+1}\sum_{k=0}^n\Big[\sum_{\mathsf{w_H}(x)=k}(-1)^{f(x)}\Big]^2 &= \frac{1}{n+1}\sum_{k=0}^n\Big[\sum_{\mathsf{w_H}(x)=k}(-1)^{f(x)}\Big]\Big[\sum_{\mathsf{w_H}(y)=k}(-1)^{f(y)}\Big]\\ &= \frac{1}{n+1}\sum_{\mathsf{w_H}(x)=\mathsf{w_H}(y)}(-1)^{f(x)+f(y)}\\ &= \frac{2^{-n}}{n+1}\sum_{x,y,v\in\mathbb{F}_2^n}(-1)^{f(x)+f(y)+v\cdot(\Sigma(x)+\Sigma(y))}\\ &= \frac{2^{-n}}{n+1}\sum_{v\in\mathbb{F}_2^n}\Big(\sum_{x\in\mathbb{F}_2^n}(-1)^{f(x)+v\cdot\Sigma(x)}\Big)^2.\end{aligned}$$

Hence, the quadratic mean of the sequence: $k\to\sum_{\mathsf{w_H}(x)=k}(-1)^{f(x)}$ equals $\frac{1}{\sqrt{n+1}}$ times the quadratic mean of the sequence:

$$g\to\sum_{x\in\mathbb{F}_2^n}(-1)^{f(x)+g(x)} \tag{5}$$

where g ranges over the set of all symmetric Boolean functions null at 0 input.

Expression (5) corresponds to a transformation similar to the Walsh transform where the linear functions $a\cdot x$ are replaced by the symmetric functions null at input 0.

3.2 Nonlinearity

In this part we study the criterion of nonlinearity on restricted inputs; first we study the bound on the maximal nonlinearity reachable by a function on a restricted set, then we investigate the behavior of this bound for the fixed Hamming weight case. We give an error correcting code perspective on these investigations, enabling to construct functions with a guaranteed amount of nonlinearity when the Hamming weight is fixed and finally we show how direct sums can provide some nonlinearity in this setting.

3.2.1 Nonlinearity upper bound for all restricted sets

From the definition of nonlinearity over a set of Section 1.2.2 we deduce:

Proposition 6. *For every n-variable Boolean function f over $\mathbb{F}_2^n$ and every subset E of $\mathbb{F}_2^n$, we have:*

$$\mathsf{NL}_E(f)=\frac{|E|}{2}-\frac{1}{2}\max_{a\in\mathbb{F}_2^n}|\sum_{x\in E}(-1)^{f(x)+a\cdot x}|.$$

Note that, for every $b\in\mathbb{F}_2^n$, denoting $f'(x)=f(x)+b\cdot x$, we have $\mathsf{NL}_E(f')=\mathsf{NL}_E(f)$. This obvious observation will be useful below.

We have:

$$\sum_{a\in\mathbb{F}_2^n}\left(\sum_{x\in E}(-1)^{f(x)+a\cdot x}\right)^2 = \sum_{x,y\in E}(-1)^{f(x)+f(y)}\sum_{a\in\mathbb{F}_2^n}(-1)^{a\cdot(x+y)} \quad (6)$$

$$= 2^n\,|E|. \quad (7)$$

Equation 6 is obtained by changing the order of the two summations and applying the classical equality $(\sum_{i\in I} a_i)^2 = \sum_{i,j\in I} a_i a_j$ expressing the square of a summation. The second sum being not null only when $x+y=0$, we get Equation 7. As the maximum of a sequence of numbers is always bounded below by the arithmetic mean, we deduce:

Proposition 7. *For every subset E of $\mathbb{F}_2^n$ and every Boolean function f defined over E, we have:*

$$\mathsf{NL}_E(f) \leq \frac{|E|}{2} - \frac{\sqrt{|E|}}{2}.$$

This bound when applied with $E=\mathbb{F}_2^n$ is called the covering radius bound and the functions achieving it with equality are called *bent* and are characterized by the balancedness of their derivatives $D_a f(x) = f(x)+f(x+a)$, for $a\neq 0$.

We show that this bound can be improved for some E and in particular when E is the set of vectors of fixed Hamming weight:

Proposition 8. *Let E be any subset of $\mathbb{F}_2^n$, f a Boolean function over E, and F a vectorspace where there exists v in $\mathbb{F}_2^n$ such that $v\cdot(x+y)=1$ for all $(x,y)\in E^2$ such that $0\neq x+y\in F^\perp$. Then we have:*

$$\mathsf{NL}_E(f) \leq \frac{|E|}{2} - \frac{1}{2}\sqrt{|E|+\lambda},$$

where

$$\lambda = \left|\sum_{\substack{(x,y)\in E^2\\ 0\neq x+y\in F^\perp}}(-1)^{f(x)+f(y)}\right|.$$

Proof: Let F be any vector subspace of $\mathbb{F}_2^n$. Then we have:

$$\begin{aligned}\sum_{a\in F}\left(\sum_{x\in E}(-1)^{f(x)+a\cdot x}\right)^2 &= \sum_{(x,y)\in E^2}(-1)^{f(x)+f(y)}\sum_{a\in F}(-1)^{a\cdot(x+y)}\\ &= |F|\sum_{\substack{(x,y)\in E^2\\ x+y\in F^\perp}}(-1)^{f(x)+f(y)}\\ &= |F|\left(|E| + \sum_{\substack{(x,y)\in E^2\\ 0\neq x+y\in F^\perp}}(-1)^{f(x)+f(y)}\right),\end{aligned}$$

which implies:

$$\max_{a\in\mathbb{F}_2^n}\left|\sum_{x\in E}(-1)^{f(x)+a\cdot x}\right| \geq \sqrt{|E| + \sum_{\substack{(x,y)\in E^2\\ 0\neq x+y\in F^\perp}}(-1)^{f(x)+f(y)}},$$

and

$$\mathsf{NL}_E(f) \leq \frac{|E|}{2} - \frac{1}{2}\sqrt{|E| + \sum_{\substack{(x,y)\in E^2\\ 0\neq x+y\in F^\perp}}(-1)^{f(x)+f(y)}}.$$

Let us assume that there exists v in $\mathbb{F}_2^n$ such that, for all $(x,y) \in E^2$ such that $0 \neq x+y \in F^\perp$, we have $v \cdot (x+y) = 1$. Suppose that:

$$\sum_{\substack{(x,y)\in E^2 \\ 0\neq x+y\in F^\perp}} (-1)^{f(x)+f(y)} = \lambda \neq 0.$$

Then λ may be without loss of generality assumed to be positive. Indeed, if λ is negative, then let v be as above, and let $f'(x) = f(x) + v \cdot x$; we have:

$$\sum_{\substack{(x,y)\in E^2 \\ 0\neq x+y\in F^\perp}} (-1)^{f'(x)+f'(y)} = \sum_{\substack{(x,y)\in E^2 \\ 0\neq x+y\in F^\perp}} (-1)^{f(x)+f(y)+v\cdot(x+y)} = -\lambda > 0.$$

Therefore we deduce the bound of the Proposition. □

Moreover, we can also take a family of vectorspaces $\mathcal{F}$, and the proposition above can then lead to the corollary below.

Corollary 2. *Let E be any subset of $\mathbb{F}_2^n$, f a Boolean function over E, and $\mathcal{F}$ a family of vectorspaces F for each of which there exists v in $\mathbb{F}_2^n$ such that $v \cdot (x+y) = 1$ for all $(x,y) \in E^2$ such that $0 \neq x+y \in F^\perp$. Then we have:*

$$\mathsf{NL}_E(f) \leq \frac{|E|}{2} - \frac{1}{2}\sqrt{|E| + \lambda},$$

where

$$\lambda = \max_{F\in\mathcal{F}} \left| \sum_{\substack{(x,y)\in E^2 \\ 0\neq x+y\in F^\perp}} (-1)^{f(x)+f(y)} \right|.$$

In particular, taking for $\mathcal{F}$ the family of all linear hyperplanes of $\mathbb{F}_2^n$ (for which such v always exists since $F^\perp$ has dimension 1), we have:

Corollary 3. *Let E be a subset of $\mathbb{F}_2^n$ and f a Boolean function over E. Then:*

$$\mathsf{NL}_E(f) \leq \frac{|E|}{2} - \frac{1}{2}\sqrt{|E| + \lambda},$$

where

$$\lambda = \max_{a\in\mathbb{F}_2^n; a\neq 0} \left| \sum_{\substack{(x,y)\in E^2 \\ x+y=a}} (-1)^{f(x)+f(y)} \right|.$$

Remark 4. Note that this result applied for $E = \mathbb{F}_2^n$ proves again that the derivatives of bent functions are all balanced.

3.2.2 Nonlinearity upper bound for fixed Hamming weight input

Let us now consider the case of $E = E_{n,k}$ for $k = 0, \ldots, n$, where $E_{n,k}$ is the set of vectors of Hamming weight k in $\mathbb{F}_2^n$. We have:

$$\mathsf{NL}_{E_{n,k}}(f) \leq \frac{\binom{n}{k}}{2} - \frac{1}{2}\sqrt{\binom{n}{k}}.$$

Note that this bound could be tight only if $\binom{n}{k}$ is a square, but we shall see that even in that case, it is not. Of course, we have

$$\mathsf{NL}_{E_{n,k}}(f) \leq \left\lfloor \frac{\binom{n}{k}}{2} - \frac{1}{2}\sqrt{\binom{n}{k}} \right\rfloor,$$

and it seems difficult to determine for which values of n this latter bound is tight.
Let us denote by i the Hamming weight of a. If i is odd then $\{(x,y) \in E_{n,k}^2;\ x+y=a\}$ is empty and if i is even, then $|\{(x,y) \in E_{n,k}^2;\ x+y=a\}|$ equals the number of possible choices (for building the support of x) of $\frac{i}{2}$ indices in the support of a and of $k-\frac{i}{2}$ indices outside the support of a. Then

$$|\{(x,y) \in E_{n,k}^2;\ x+y=a\}| = \binom{i}{\frac{i}{2}}\binom{n-i}{k-\frac{i}{2}}.$$

Clearly, since the sum

$$\sum_{\substack{(x,y)\in E^2\\ x+y=a}} (-1)^{f(x)+f(y)}$$

is invariant when swapping x and y, if $\binom{i}{\frac{i}{2}}\binom{n-i}{k-\frac{i}{2}}$ is not divisible by 4, then λ equals twice the sum of an odd number of integers equal to ± 1, where λ is defined as in corollary 2; it is then strictly positive. For instance for $k=2$ and $i=4$,

$$\binom{i}{\frac{i}{2}}\binom{n-i}{k-\frac{i}{2}} = 6\binom{n-4}{0} = 6,$$

and for $n \geq 4$, the sum

$$\sum_{\substack{(x,y)\in E^2\\ x+y=a}} (-1)^{f(x)+f(y)}$$

cannot be null. We deduce:

Corollary 4. *For all n and $k \in \{1,\ldots,n-1\}$, Bound (3.2.2) is never tight, except maybe for two particular pairs (n,k): $(50,3)$ and $(50,47)$.*

Indeed, the bound can only be tight when $E_{n,k}$ is a square. Erdős showed the following theorem.

Theorem 3. *[AZ09] The equation $\binom{n}{k} = m^\ell$ has no integer solution with $\ell \geq 2$ and $4 \leq k \leq n-4$.*

The only solution for $k=3$ is $n=50$, therefore we only consider the cases $k \in 0,1,2,n-2,n-1,n$.

- $k=0$ (or $k=n$): Proposition 7 gives $\mathsf{NL}_{E_{n,0}}(f) \leq 0$ which is tight because for all n and for all Boolean function f, f_k when $k=0$ (or $k=n$) is constant.

- $k=1$ (or $k=n-1$): $|E_{n,1}|$ is a square if and only if n is a square; using Proposition 6, every function restricted to its entries of Hamming weight 1 (or $n-1$) is linear therefore $\mathsf{NL}_{E_{n,1}}(f)=0$ whereas the bound tells $\mathsf{NL}_{E_{n,1}}(f) \leq \frac{n-\sqrt{n}}{2}$.

- $k=2$ (or $k=n-2$):

  Using corollary 3, for $i=2$, $\binom{i}{\frac{i}{2}}\binom{n-i}{2-\frac{i}{2}} = 2(n-2)$ and if n is odd, the sum $\sum_{\substack{(x,y)\in E^2\\ x+y=a}} (-1)^{f(x)+f(y)}$ cannot be null, and for $i=4$, $\binom{i}{\frac{i}{2}}\binom{n-i}{2-\frac{i}{2}} = 6\binom{n-4}{0} = 6$ for $n \geq 4$ and the sum $\sum_{\substack{(x,y)\in E^2\\ x+y=a}} (-1)^{f(x)+f(y)}$ cannot be null. Therefore for all n and for all Boolean function f $\mathsf{NL}_{E_{n,2}}(f) < \frac{|E_{n,2}|-\sqrt{|E_{n,2}|}}{2}$.

3.2.3 Error correcting codes perspective

Reed Muller codes $RM(r,n)$ are binary codes of length 2^n whose codewords are the evaluations of all Boolean functions of algebraic degrees at most r in n variables on their 2^n entries. Fixing the Hamming weight of the entries gives particular punctured Reed Muller codes whose characteristics are directly linked to Boolean functions with fixed weight entries. As Reed Muller codes have been intensively studied in other contexts we do not describe fundamental new results in this part, we rather use another perspective to give interesting constructions and help to link our problematic to a quite well known topic.

Definition 6. For all $n \in \mathbb{N}^*; r,k \in [0,n]$ we denote by $RM(r,n)_k$ the punctured Reed Muller code of length $\binom{n}{k}$ obtained by puncturing $RM(r,n)$ on all entries of Hamming weight different from k.

Remark 5. $RM(1,n)_k$ corresponds to the evaluation of all affine functions in n variables on entries of Hamming weight k; therefore, for every Boolean function f, $\mathsf{NL}_{E_{n,k}}(f)$ is the distance between f's truth table restricted to Hamming weight k entries and $RM(1,n)_k$. The maximal value of $\mathsf{NL}_{E_{n,k}}(f)$ when f ranges over the set of all Boolean functions equals the covering radius of $RM(1,n)_k$.

In the next remark we exhibit the parameters of the code $RM(1,n)_k$; this provides a lower bound on the maximal value of $\mathsf{NL}_{E_{n,k}}$.

Remark 6. $RM(1,n)_k$ is a linear code with parameters $[\binom{n}{k}, n, d]$ where

$$d = \frac{\binom{n}{k} - \max_{(0<\ell\leq n/2)} \left|\sum_{i\in\mathbb{Z}} (-1)^i \binom{\ell}{i}\binom{n-\ell}{k-i}\right|}{2}.$$

Let $l(x) = \sum_{i\in I} x_i$ be any linear Boolean function whose restriction to the entries of Hamming weight k is non constant, and let $|I| = \ell$. We have $\ell \in \{1,\ldots,n-1\}$. The number of entries x of Hamming weight k such that $|supp(x) \cap I| = i$ equals $\binom{\ell}{i}\binom{n-\ell}{k-i}$. We deduce that the minimum distance of $RM(1,n)_k$ equals:

$$\min_{(0<\ell<n)} \left(\min \left(\sum_{i \text{ odd}} \binom{\ell}{i}\binom{n-\ell}{k-i}, \sum_{i \text{ even}} \binom{\ell}{i}\binom{n-\ell}{k-i} \right) \right) =$$

$$\frac{\min_{(0<\ell<n)} \left(\min \left(\sum_{i\in\mathbb{Z}} \binom{\ell}{i}\binom{n-\ell}{k-i} - \sum_{i\in\mathbb{Z}} (-1)^i \binom{\ell}{i}\binom{n-\ell}{k-i}, \sum_{i\in\mathbb{Z}} \binom{\ell}{i}\binom{n-\ell}{k-i} + \sum_{i\in\mathbb{Z}} (-1)^i \binom{\ell}{i}\binom{n-\ell}{k-i} \right) \right)}{2} =$$

$$\frac{\min_{(0<\ell<n)} \left(\sum_{i\in\mathbb{Z}} \binom{\ell}{i}\binom{n-\ell}{k-i} - \left|\sum_{i\in\mathbb{Z}} (-1)^i \binom{\ell}{i}\binom{n-\ell}{k-i}\right| \right)}{2}.$$

In other words, writing $P[X^k]$ for the coefficient of X^k in a polynomial $P(X)$, the minimum distance of $RM(1,n)_k$ equals:

$$\frac{\min_{(0<\ell<n)} \left((1+X)^\ell (1+X)^{n-\ell}[X^k] - \left|(1-X)^\ell (1+X)^{n-\ell}[X^k]\right| \right)}{2} =$$

$$\frac{\min_{(0<\ell<n)} \left(\binom{n}{k} - \left|(1-X)^\ell (1+X)^{n-\ell}[X^k]\right| \right)}{2} =$$

$$\frac{\binom{n}{k} - \max_{(0<\ell<n)} \left|\sum_{i\in\mathbb{Z}} (-1)^i \binom{\ell}{i}\binom{n-\ell}{k-i}\right|}{2}.$$

Note that $\left|\sum_{i\in\mathbb{Z}} (-1)^i \binom{\ell}{i}\binom{n-\ell}{k-i}\right|$ is invariant when changing ℓ into $n-\ell$ (by changing i into $k-i$); we can then replace $\max_{(0<\ell<n)}$ by $\max_{(0<\ell\leq n/2)}$.

Dumer and Kapralova studied this punctured Reed and Muller code of order 1 in 2013 and more recently they also studied the general case of the punctured Reed and Muller codes of order r. See the results in the two following papers [DK17, DK13].
Note that the maximal value of $\mathsf{NL}_{E_{n,k}}(f)$ *when* f *ranges over the set of all Boolean functions (i.e. the covering radius of* $RM(1,n)_k$*) is bounded from below by* $\frac{d}{2}$. It is then nonzero except for particular values of k and enables to directly build functions reaching this minimal bound for all k from this error correcting code perspective.

3.2.4 Direct sum and $\mathsf{NL}_{E_{n,k}}$

Let N be any positive integer and $k \in \{1, \ldots, N\}$. We recall that the nonlinearity $\mathsf{NL}_{E_{N,k}}(F)$ of an N-variable function F over a set $E_{N,k}$ equals the minimum Hamming distance between the restriction of F to $E_{N,k}$ and the restrictions of affine functions to $E_{N,k}$.

Lemma 3 (Direct sum and $\mathsf{NL}_{E_{N,k}}$)**.** *Let* F *be the direct sum of* f *and* g*, we have:*

$$\mathsf{NL}_{E_{N,k}}(F) \geq \sum_{i=0}^{k} \binom{n}{i} \mathsf{NL}_{E_{m,k-i}}(g) + \sum_{i=0}^{k} \mathsf{NL}_{E_{n,i}}(f) \left(\binom{m}{k-i} - 2\mathsf{NL}_{E_{m,k-i}}(g) \right)$$

Proof: We have:

$$\begin{aligned}
\mathsf{NL}_{E_{N,k}}(F) &= \frac{\binom{N}{k}}{2} - \frac{1}{2} \max_{(a,b)\in\mathbb{F}_2^n\times\mathbb{F}_2^m} \left| \sum_{(x,y)\in E_{N,k}} (-1)^{F(x,y)+a\cdot x+b\cdot y} \right| \\
&\geq \frac{\binom{N}{k}}{2} - \frac{1}{2} \max_{a\in\mathbb{F}_2^n, b\in\mathbb{F}_2^m} \sum_{i=0}^{k} \left| \sum_{x\in E_{n,i}, y\in E_{m,k-i}} (-1)^{f(x)+g(y)+a\cdot x+b\cdot y} \right| \\
&= \frac{\binom{N}{k}}{2} - \frac{1}{2} \max_{a\in\mathbb{F}_2^n, b\in\mathbb{F}_2^m} \sum_{i=0}^{k} \left| \sum_{x\in E_{n,i}} (-1)^{f(x)+a\cdot x} \right| \left| \sum_{y\in E_{m,k-i}} (-1)^{g(y)+b\cdot y} \right| \\
&\geq \frac{\binom{N}{k}}{2} - \frac{1}{2} \sum_{i=0}^{k} \left(\max_{a\in\mathbb{F}_2^n} \left| \sum_{x\in E_{n,i}} (-1)^{f(x)+a\cdot x} \right| \right) \left(\max_{b\in\mathbb{F}_2^m} \left| \sum_{y\in E_{m,k-i}} (-1)^{g(y)+b\cdot y} \right| \right) \\
&= \frac{\binom{N}{k}}{2} - \frac{1}{2} \sum_{i=0}^{k} \left(\binom{n}{i} - 2\mathsf{NL}_{E_{n,i}}(f) \right) \left(\binom{m}{k-i} - 2\mathsf{NL}_{E_{m,k-i}}(g) \right) \\
&= \sum_{i=0}^{k} \binom{n}{i} \mathsf{NL}_{E_{m,k-i}}(g) + \sum_{i=0}^{k} \binom{m}{k-i} \mathsf{NL}_{E_{n,i}}(f) - 2\sum_{i=0}^{k} \mathsf{NL}_{E_{n,i}}(f) \mathsf{NL}_{E_{m,k-i}}(g) \\
&= \sum_{i=0}^{k} \binom{n}{i} \mathsf{NL}_{E_{m,k-i}}(g) + \sum_{i=0}^{k} \mathsf{NL}_{E_{n,i}}(f) \left(\binom{m}{k-i} - 2\mathsf{NL}_{E_{m,k-i}}(g) \right).
\end{aligned}$$

□

Although this inequality does not provide a tight bound, it enables to guarantee some nonlinearity on fixed Hamming weight input of a function from two simpler functions with high nonlinearity in this context.

3.3 Algebraic Immunity

In this part we study the criterion of algebraic immunity on restricted inputs; first we study the bound on the maximal algebraic immunity reachable by a function on a restricted set,

then we investigate the behavior of this bound for the fixed Hamming weight case and give more detailed results in relation with this particular case. Finally we show how direct sums can provide some algebraic immunity in this setting.

3.3.1 Algebraic immunity upper bound for all restricted sets

In the case of $E = \mathbb{F}_2^n$, Courtois and Meier [CM03] have shown that, for every non-negative integers d and e such that $d+e \geq n$, there exists a nonzero Boolean function g of algebraic degree at most e and a Boolean function h of algebraic degree at most d such that $h = gf$. For $e = d = \lceil n/2 \rceil$, this proved that the so-called algebraic immunity of f (see Definition 4 in Section 1.2.3) is at most $\lceil n/2 \rceil$. We revisit these results for functions defined over a subset of $\mathbb{F}_2^n$.

Proposition 9. *Let E be any non-empty subset of $\mathbb{F}_2^n$ and f any Boolean function defined over E. Let d and e be two non-negative integers. Let $M_{d,E}$ be the $(\sum_{i=0}^{d} \binom{n}{i}) \times |E|$ matrix whose term at row indexed by $u \in \mathbb{F}_2^n$, $\mathsf{w_H}(u) \leq d$, and at column indexed by $x \in E$ equals $x^u := \prod_{i=1}^{n} x_i^{u_i}$. Assume that the ranks of matrices $M_{d,E}$ and $M_{e,E}$ are such that*

$$\mathsf{rank}(M_{d,E}) + \mathsf{rank}(M_{e,E}) > |E|,$$

then there exists a Boolean function g of algebraic degree at most e over $\mathbb{F}_2^n$, whose restriction to E is not identically null, and a Boolean function h of algebraic degree at most d on $\mathbb{F}_2^n$, such that functions gf and h coincide on E.

Proof: Let $\mathcal{F}_d$ (resp. $\mathcal{F}_e$) be a maximum size free family of restrictions to E of monomials x^u of algebraic degree $\mathsf{w_H}(u)$ at most d (resp. at most e). By definition of the rank of a matrix, the size of $\mathcal{F}_d$ equals $\mathsf{rank}(M_{d,E})$ and that of $\mathcal{F}_e$ equals $\mathsf{rank}(M_{e,E})$. Let us consider now the family, that we shall denote by $\mathcal{F}_e f$, whose elements (with possible repetitions) are the products of the elements of $\mathcal{F}_e$ by the function f. By hypothesis, $|\mathcal{F}_d| + |\mathcal{F}_e f|$ is strictly larger than the dimension of the $\mathbb{F}_2$-vectorspace of all Boolean functions over E, that is, $|E|$ (indeed, the number of Boolean functions over E equals $2^{|E|}$). There exists then a linear combination of the elements of $\mathcal{F}_d$ and of those of $\mathcal{F}_e f$, which equals the zero function and whose coefficients are not all null. Gathering the part of this linear combination dealing with the elements of $\mathcal{F}_d$ and those dealing with $\mathcal{F}_e f$, we obtain respectively functions h and g such that h and gf coincide over E, and the restrictions of g and h to E are not both null (since both families $\mathcal{F}_e$ and $\mathcal{F}_d$ are free), that is, the restriction of g to E is nonzero. □

Taking $e = 0$ in Proposition 9, we have $\mathsf{rank}(M_{e,E}) = 1$, since constant function 1 does not vanish over E, and:

Corollary 5. *Let E be any non-empty subset of $\mathbb{F}_2^n$ and f any Boolean function defined over E. Let n and d be such that $\mathsf{rank}(M_{d,E}) = |E|$, then there exists a Boolean function over $\mathbb{F}_2^n$ of algebraic degree at most d which coincides with f on E.*

In other words, the algebraic degree of any Boolean function over E is bounded above by the least value of d such that $\mathsf{rank}(M_{d,E}) = |E|$. Indeed, in Proposition 9, we have $g = 1$ and $gf = h$ on E where h has algebraic degree at most d.

Taking $d = 0$, we have $\mathsf{rank}(M_{d,E}) = 1$ and, calling *annihilator of f on E* any Boolean function g over E whose product with f vanishes:

Corollary 6. *Let E be any non-empty subset of $\mathbb{F}_2^n$ and f any non-constant Boolean function defined over E. Let n and e be such that $\mathsf{rank}(M_{e,E}) = |E|$, then there exists a nonzero annihilator of f of algebraic degree at most e over E.*

Indeed, in Proposition 9, we have h constant and since $gf = 1$ on E is impossible, we have then $h = 0$.
Note that this shows that the algebraic immunity of a function (see Definition 4 in Section 1.2.3), which for a random Boolean function over $\mathbb{F}_2^n$ lies not far from $n/2$ as shown by F. Didier in [Did06], can tumble down when the input is restricted to a set E. Notice also that Corollary 6 can be viewed as a corollary of Corollary 5, since we can take $f + 1$ for annihilator.

This being observed, we have in fact a stronger result when taking $e = d$; we have:

Corollary 7. *Let E be any non-empty subset of $\mathbb{F}_2^n$ and f any Boolean function defined over E. Let n and e be such that $\mathsf{rank}(M_{e,E}) > \frac{|E|}{2}$, then there exists g of algebraic degree at most e, whose product with f or $f + 1$ is null on E, and whose restriction to E is nonzero.*

Indeed, using a classical idea of Meier et al. [MPC04], either the functions g and h of Proposition 9 coincide on E, and we have then $gf + h = g(f + 1) = 0$ on E, where g has algebraic degree at most e and nonzero restriction to E, or they do not and we have, after multiplication of equality $h = gf$ by f, that $(g + h)f = 0$, where $g + h$ has algebraic degree at most e and nonzero restriction to E.

The situation is then similar to that described by Meier et al. and this explains our definition of restricted algebraic immunity (see Definition 4 in Section 1.2.3) and leads to the following property:

Corollary 8. *The algebraic immunity of any Boolean function over E is bounded above by* $\min\{e;\ \mathsf{rank}(M_{e,E}) > \frac{|E|}{2}\}$.

3.3.2 Algebraic immunity upper bound for fixed Hamming weight input

In this section, we focus on the particular case when the input is restricted to the words of Hamming weight fixed: $E_{n,k}$ for some $k \in [1, n-1]$, note that $M_{n,k,d}$ is a generator matrix of the code $RM(d, n)_k$ from definition 6. To be able to evaluate efficiently in such situation the algebraic immunity by using Proposition 9 and its corollaries, there remains to calculate the rank of the matrix $M_{n,k,d}$ for each d and k:

Theorem 4. *The rank of $M_{n,k,d}$ is equal to*

$$\binom{n}{\min(d, k, n-k)}$$

Proof: The principle of this proof is to find a recurring relation on the rank of $M_{n,k,d}$. To this aim, we use a construction which looks like the well-known $u \parallel u + v$ construction of Reed-Muller codes: every Boolean function f of algebraic degree at most d can be written in the form :

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

where g has algebraic degree at most d and h has algebraic degree at most $d - 1$. In the sequel of the proof, we shall use the notations:

$$N_k = \binom{n}{k} \quad \text{and} \quad D = \sum_{0 \le i \le d} \binom{n}{i}.$$

Let $\psi_{n,k,d}$ be the following linear application, mapping every Boolean function in n variables (defined by its ANF) of algebraic degree at most d to the restriction of its truth table to the elements in $E_{n,k}$:

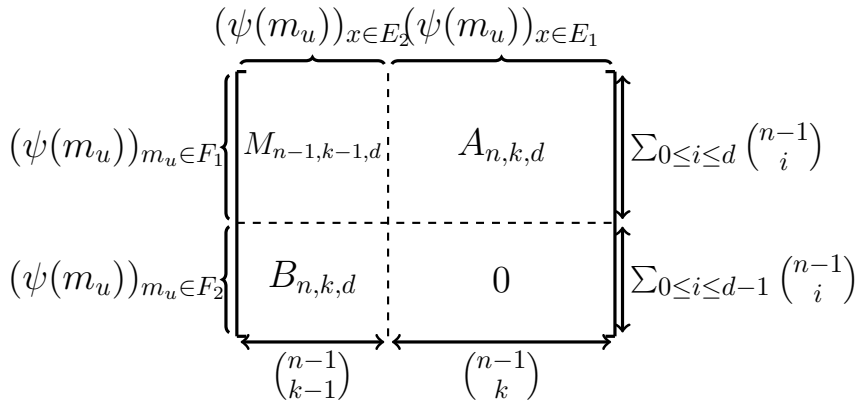

Figure 1

$$\begin{array}{rccc} \psi_{n,k,d} & : & \mathbb{F}_2^D & \longrightarrow & \mathbb{F}_2^{N_k} \\ & & (a_u)_{u\in\mathbb{F}_2^n,\mathsf{w_H}(u)\le d} & \longmapsto & \left(\sum_{u\preceq x} a_u\right)_{x\in\mathbb{F}_2^n,\mathsf{w_H}(x)=k} \end{array}$$

where $u \preceq x$ means $u_i \le x_i$ for every i. This application $\psi_{n,k,d}$ is linear and moreover, the rank of $M_{n,k,d}$ is exactly the rank of this linear application $\psi_{n,k,d}$.
Denoting by m_u the monomial x^u, the rank of $\psi_{n,k,d}$ is the rank of the family of the following vectors:

$$(\psi_{n,k,d}(m_u))_{u\in\mathbb{F}_2^n,\mathsf{w_H}(u)\le d}.$$

We split the family of vectors $u \in \mathbb{F}_2^n, \mathsf{w_H}(u) \le d$, into:

$$F_1 = \{u \in \mathbb{F}_2^n, \mathsf{w_H}(u) \le d; u_n = 0\} \quad \text{and} \quad F_2 = \{u \in \mathbb{F}_2^n, \mathsf{w_H}(u) \le d; u_n = 1\}.$$

We also split the vectors x of $\mathbb{F}_2^n$ of Hamming weight k into:

$$E_1 = \{x \in \mathbb{F}_2^n, \mathsf{w_H}(x) = k; x_n = 0\} \quad \text{and} \quad E_2 = \{x \in \mathbb{F}_2^n, \mathsf{w_H}(x) = k; x_n = 1\}.$$

Notice that for every $u \in F_2$ and every $x \in E_1$, we have $m_u(x) = 0$.

The $\mathbb{F}_2^D \times \mathbb{F}_2^{N_k}$ matrix $M_{n,k,d}$ representing the linear application $\psi_{n,k,d}$ has then the form given in figure 1.

$A_{n,k,d}$ takes its entries on the set of monomials in which x_n does not occur and of degrees at most d. The output of the linear application defined by $A_{n,k,d}$ is the truth table of Boolean functions on those inputs of Hamming weight k where the value of x_n is set to 0. Then, as x_n does not occur in the entries and is fixed to 0 in the output, $A_{n,k,d}$ defines exactly the linear application which gives the truth table on words of weight k of all Boolean functions with $n-1$ variables (x_n is fixed) and of degrees at most d.

$B_{n,k,d}$ defines the linear application which gives the truth table on words of weight $k-1$ (because x_n is fixed to 1 and not 0 anymore) of all Boolean functions with $n-1$ variables (x_n being fixed) and of degrees at most $d-1$. Hence, $A_{n,k,d}$ defines the linear application $\psi_{n-1,k,d}$ and $B_{n,k,d}$ defines $\psi_{n-1,k-1,d-1}$.

Moreover, let us prove that the rank of this matrix is equal to the rank of $A_{n,k,d}$ plus the rank of $B_{n,k,d}$ (i.e. $M_{n-1,k-1,d}$ does not play any role in the rank of the whole matrix). Indeed, if we have a vector of length $\binom{n}{k}$ which is a linear combination on the lines such that the last $\binom{n-1}{k}$ coordinates of the resulting vector are null, (i.e. we are in the kernel of $A_{n,k,d}$) then this vector is linearly dependent from the vectors defined by $B_{n,k,d}$. By

viewing this in terms of Boolean functions, we prove that if f is a Boolean function in the linear span of F_1 such that $\forall x \in E_1, f(x) = 0$ (i.e. in the kernel of $A_{n,k,d}$) then f is in the linear span of F_2; indeed: $f(x_1, \cdots, x_n) = x_n g(x_1, \cdots, x_{n-1}) + h(x_1, \cdots, x_{n-1})$. The Boolean function f is of degree less than d, then h is of degree less than d and g is of degree less than $d-1$. But for all $x \in E_1$, we have $f(x) = 0$, then that means that $h(x_1, \cdots, x_{n-1}) = 0$, then f is in the linear span of F_2. Then we deduce the following recurring relation:

$$\dim(\Im(\psi_{n,k,d})) = \dim(\Im(\psi_{n-1,k-1,d-1})) + \dim(\Im(\psi_{n-1,k,d}))$$

Moreover, if $d \geq k$ then $\dim(\Im(\psi_{n,k,d})) = \binom{n}{k}$. In fact, the monomials of degree exactly k correspond to the canonical basis of the Boolean functions defined over $E_{n,k}$ (representing within their truth table). For $d \geq n-k$, we can choose the Boolean functions defined by $f(x) = (1+x_{i_1})(1+x_{i_2})\cdots(1+x_{i_{n-k}})$ which are of degree less than d and form also the canonical basis of the Boolean functions defined over $E_{n,n-k}$.
Then, as we found a recurring relation between $\dim(\Im(\psi_{n,k,d}))$, $\dim(\Im(\psi_{n-1,k-1,d-1}))$ and $\dim(\Im(\psi_{n-1,k,d}))$, at some point there will be $k = d$ or $n-k = d$. Then the initialization step ($d = k$ or $d = n-k$ or $d = 0$) of the recurring relation is true.

Then we deduce by induction that

$$\dim(\Im(\psi_{n,k,d})) = \binom{n}{\min(d, k, n-k)}$$

□

From Corollary 7 and Theorem 4, we deduce:

Corollary 9. *Let k be any positive integer such that $k \leq n/2$. The algebraic immunity of the restriction of F to $E_{n,k}$ is bounded above by*

$$\min\left\{e;\ 2\binom{n}{e} > \binom{n}{k}\right\}.$$

Remark 7. For $r > 0$, we have: $2\binom{n}{k-r} = \binom{n}{k}\frac{2k(k-1)\ldots(k-r+1)}{(n-k+r)\ldots(n-k+1)}$ and if $\frac{k-r+1}{n-k+1} > 2^{-1/r}$, that is if $k > \frac{2^{-1/r}(n+1)+r-1}{1+2^{-1/r}} = \frac{n+1+(r-1)2^{1/r}}{2^{1/r}+1}$, then we have *a fortiori* $\frac{k-r+2}{n-k+2} > 2^{-1/r}, \ldots, \frac{k}{n-k+r} > 2^{-1/r}$, and we have then $2\binom{n}{k-r} > \binom{n}{k}$. For $k = n/2$, the condition $k > \frac{n+1+(r-1)2^{1/r}}{2^{1/r}+1}$ becomes $n(2^{1/r}+1) > 2(n+1+(r-1)2^{1/r})$, that is, $n > \frac{2+2(r-1)2^{1/r}}{2^{1/r}-1}$. Hence, the best possible algebraic immunity of a function with constrained input Hamming weight is lower than for unconstrained functions.

With theorem 4, we have the dimension of the image of $\psi_{n,k,d}$ and then of its kernel. Without direct application to the others sections we can exhibit more properties on the basis of this kernel.

Proposition 10. *Let k, r and n be such that $k \leq \frac{n}{2}$ and let $0 \leq i_1 < i_2 < \cdots < i_r \leq n$. Then any Boolean function defined as:*

$$x_{i_1}x_{i_2}\cdots x_{i_r}\left(\sum_{j \neq i_1, i_2, \cdots, i_r} x_j\right) \text{ if } k - r \equiv 0 \mod 2,$$

$$x_{i_1}x_{i_2}\cdots x_{i_r}\left(1 + \sum_{j \neq i_1, i_2, \cdots, i_r} x_j\right) \text{ if } k - r \equiv 1 \mod 2$$

is null on the set $E_{n,k}$ of all binary vectors of size n with Hamming weight equal to k. More generally, for every $j < k$ and s, and any u of Hamming weight equal to j, the function defined as:

$$x^u \times \left(\sum_{\{i_1,\dots,i_{s-j}\} \cap supp(u) = \emptyset} \prod_{l=1}^{s-j} x_{i_l} \right) \text{ if } \binom{k-j}{s-j} \equiv 0 \mod 2$$

$$x^u \times \left(1 + \sum_{\{i_1,\dots,i_{s-j}\} \cap supp(u) = \emptyset} \prod_{l=1}^{s-j} x_{i_l} \right) \text{ if } \binom{k-j}{s-j} \equiv 1 \mod 2$$

is null on $E_{n,k}$.

Proof: Without loss of generality, we take $x^u = x_1 x_2 \cdots x_j$ We note f such a function. Let x of Hamming weight k, then if $x^u = 0$ then $f(x) = 0$, if $x^u = 1$, then $f(x) = \sum_{\{i_1,\dots,i_{s-j}\} \cap supp(u) = \emptyset} \prod_{l=1}^{s-j} x_{i_l} + \binom{k-j}{s-j} \mod 2$, but as $x^u = 1$, $x_1 = x_2 = \cdots = x_j = 1$ the Hamming weight of the vector $(x_{j+1}, x_{j+2}, \cdots, x_n)$ is then fixed to $k-j$. The Boolean function $\sum_{\{i_1,\dots,i_{s-j}\} \cap supp(u) = \emptyset} \prod_{l=1}^{s-j} x_{i_l}$ is a elementary symmetric Boolean function on $n-j$ variables of degree $s-j$, then it is constant when the Hamming weight of the entry is fixed (which is the case when $x^u = 1$ here) and its value is $\binom{k-j}{s-j} \mod 2$. So $f(x) = 0$ if x is of Hamming weight k. □

The sum involved in this definition is an elementary symmetric Boolean function but defined on a smaller set of variables.

Corollary 10. *If $d \geq k$, then a basis of $\Im(M_{n,k,d}) = \mathbb{F}_2^{\binom{n}{k}}$ is the set of all the monomials of degree k.*

Corollary 11. *If $d \geq n-k$, then a basis of $\Im(M_{n,k,d}) = \mathbb{F}_2^{\binom{n}{k}}$ is the set of all the Boolean functions of the form $(1+x_{i_1})(1+x_{i_2})\cdots(1+x_{i_{n-k}})$ with $i_1 < i_2 < \cdots < i_{n-k}$*

Proof: See the end of the proof of theorem 4 □

3.3.3 Direct sum and AI_k

One of the main purposes of this paper is to discuss (see the Section 4) the robustness of the filter function in FLIP [MJSC16]. This function being built as a direct sum, we need then to study the algebraic immunity of direct sums. Complementary to the results of Section 2.3 linking classic algebraic immunity and algebraic immunity on fixed Hamming weight input, we investigate here the behavior of AI_k for a direct sum construction. As for the nonlinearity case, it enables to build functions with a guaranteed algebraic immunity from functions with a lesser number of variables.

For every Boolean function say F for example, we will denote by F_k the Boolean function restricted on its input of Hamming weight k.

Lemma 4 (Direct sum and AI_k)**.** *Let F be the direct sum of f and g with n and m variables respectively. Then for all $k \leq \min(n,m)$, $\mathsf{AI}_k(f)$ follows the bound :*

$$\mathsf{AI}_k(F) \geq \min_{0 \leq \ell \leq k} (\max[\mathsf{AI}_\ell(f), \mathsf{AI}_{k-\ell}(g)]).$$

Proof: Suppose that we have $A(x)$ a non-zero annihilator of F_k. Then we will show that A is a non-zero annihilator of f_ℓ or $1+f_\ell$ and $g_{k-\ell}$ or $1+g_{k-\ell}$ for some ℓ.
For all $X \in E_{n+m,k}$, $A(X)F(X)=0$. Moreover, A is non-null on $E_{n+m,k}$, so, there exists $\widetilde{X} \in E_{n+m,k}$ such that $A(\widetilde{X})=1$. We write $\widetilde{X}$ as $(\widetilde{x},\widetilde{y})$ where $\widetilde{x} \in \mathbb{F}_2^n$ and we define ℓ as $\mathsf{wH}(\widetilde{x})$, then the weight of $\widetilde{y} \in \mathbb{F}_2^m$ is $k-\ell$. Finally for this ℓ, we fix for $X \in E_{n+m,k}$ the x part to the value $\widetilde{x}$ and we consider all possible values for $y \in \mathbb{F}_2^m$ of Hamming weight $k-\ell$. By doing so, it appears that A_k is an annihilator of $g_{k-\ell}$ or $1+g_{k-\ell}$, and is non null by construction. We can also fix y to $\widetilde{y}$ and consider all possible values for x of Hamming weight ℓ and find out that A_k is also a non-zero annihilator of f_ℓ or $1+f_\ell$. Therefore $deg(A) \geq deg(A_k) \geq \max[\mathsf{AI}_\ell(f), \mathsf{AI}_{k-\ell}(g)]$. Recalling that $\ell = \mathsf{wH}(\widetilde{x})$ and then $0 \leq \ell \leq k$ finishes the proof. □

3.4 Open Questions

Considering Boolean functions on restricted input with a cryptographic point of view is quite new. Our theoretical study focusing on fixed Hamming weight input tries to address most of the natural questions in this context. In this part we enlight some other questions of variable interest and presumed difficulty which are not answered yet.

WPB function with minimal number of monomials. We proved in Section 3.1.1 that a WPB function has an ANF containing at least $\frac{3n}{4}+1$ monomials (for $n>4$) and we exhibited a construction with $n-1$ monomials. It remains then to determine whether this number of monomials in the ANF is the smallest number of monomials to obtain a WPB Boolean function. Determining the minimal number of monomials necessary to fulfill a Boolean criterion could lead to low cost functions usable in the FHE or MPC context.

Tightness of $\mathsf{NL}_{E_{N,k}}$ bound. In Section 3.2.2 we proved the upper bound:

$$\mathsf{NL}_{E_{n,k}}(f) \leq \frac{\binom{n}{k}}{2} - \frac{1}{2}\sqrt{\binom{n}{k}}.$$

As this bound is unreachable for almost all values of n and k, it would be nice to investigate if the floor of this value is a tight upper bound. Moreover this quantity being the covering radius of a punctured Reed-Muller code, various approaches could help to precise its value or give intuition on weightwise bent functions.

Exact behavior of algebraic immunity. We proved in Section 3.3.2 that the algebraic immunity e when the input is restricted to the Hamming weight k is bounded with the relation:

$$2\binom{n}{e} > \binom{n}{k}.$$

It would be nice to determine the smallest integer e satisfying this relation, and the asymptotic behavior of e relatively to the standard algebraic immunity upper bound $\lceil n/2 \rceil$ for meaningfull values of k (around $n/2$). A common function considered in cryptography for its optimal AI is the majority function, which is constant when the input weight is fixed, therefore optimal functions for restricted weight algebraic immunity could lead to very different constructions.

Tightness of AI_k bound. Back to Section 3.3.2, we linked the algebraic immunity upper bound to the rank of the generator matrix of a punctured Reed-Müller code $RM(d,n)_k$. This matrix can also be used to compute the exact AI_k of a given function, by partitioning the columns depending on the value of f on the column entry and determining when the rank of one of these two matrices is strictly inferior to the rank of the global one. For matrices with rank r at least twice the number of columns, proving the existence of a

partition of the columns in two rank r matrices will prove the tightness of the AI_k upper bound e.

4 Case study on FLIP

The stream cipher FLIP [MJSC16] has a non standard design (*i.e.* the filter permutator) where the updating process of the internal state consists in permuting its coordinates. Therefore, the Hamming weight of the internal state is constant during all the encryption. In the four proposed instances, the Hamming weight of this register is forced to $\frac{n}{2}$ where n is the size of the register (n is larger than the security parameter λ, enough to ensure that $\binom{n}{n/2} \geq 2^\lambda$).

For classical filtered pseudo-random generators (for example filtered Linear Feedback Shift Registers), when the next-state function reaches all elements in $\mathbb{F}_2^n$ or $\mathbb{F}_2^n \setminus \{0\}$, the three main criteria (for functions defined over $\mathbb{F}_2^n$) are relevant. Indeed, as the input is the whole space, designers can ensure that there are no extra relations on the filtering function inputs. However, if all inputs of the filtering function are not all reachable by the next-state function as in this case, then the security analysis does not rely on the classical criteria defined for Boolean functions over all $\mathbb{F}_2^n$, because the internal state itself does not reach all possible values. Then, the security analysis must be done in the adequate model: the stream cipher function is only evaluated on entries from $E_{n,\frac{n}{2}}$, and the security should be studied relatively to the robustness of Boolean functions over $E_{n,\frac{n}{2}}$.

The purpose of this section is first to stand on the fixed input weight criteria of the proposed filtering functions, and then to analyze the security of this stream cipher adapting standard cryptanalysis over $\mathbb{F}_2^n$ to fixed weight entries. In an article published at CRYPTO 2016 [DLR16], Duval et al. gave a guess and determine attack on preliminary instances of FLIP, leading the authors of FLIP to add more triangular functions in the filtering function. Therefore we also study the case of guess and determine attacks, combined with algebraic like attacks and with correlation like attacks, when the Hamming weight is fixed.

4.1 Fixed Hamming weight input cryptanalysis on FLIP filtering function

4.1.1 FLIP instances

We recall the 4 filtering functions proposed in FLIP in table 1, each one defined by 4 parameters n_1, n_2, nb and h where the filtering function is the direct sum of a linear function of n_1 variables, a quadratic function of n_2 variables (obtained by direct sum) and nb triangular functions of degree h. A triangular function of degree h is a direct sum of one monomial of each degree from 1 to h. Notice that in the ANF of f, each variable is used once, and f can therefore be expressed as a direct sum in various ways, which is determinant to bound its parameters on fixed Hamming weight inputs.

Table 1: FLIP filtering function instances, N is the total number of variables, n_1 is the number of variables over the linear part, n_2 is the number of variables over the quadratic part, nb is the number of triangular functions, h is the degree of the triangular functions and λ is the security parameter.

Name	N	n_1	n_2	nb	h	λ
FLIP-530	530	42	128	8	9	80
FLIP-662	662	46	136	4	15	80
FLIP-1394	1394	82	224	8	16	128
FLIP-1704	1704	86	238	5	23	128

4.1.2 Balancedness of FLIP and distinguishing attack

Distinguisher: For any stream cipher, it is important to guarantee that the keystream has good statistical properties (*i. e.* looks like a random sequence), to avoid the possibility for an attacker to distinguish the keystream from a random sequence. That is a reason why Boolean functions used in cryptography should be balanced and therefore, in a model where the Hamming weight is known, Boolean functions should be weightwise perfectly balanced functions. Indeed, let us denote $p_k = Pr_{x \in E_{N,k}}[f(x) = 1] = \frac{1}{2} - \varepsilon_k$. Then if there exists k such that $\varepsilon_k \neq 0$, then there exists a distinguisher on the function f for this weight. The amount of data needed to detect the bias ε_k is equal to $\frac{1}{{\varepsilon_k}^2}$; If we consider all entries of f, we scale the probability of having a word of weight k, so the amount of data needed for our distinguisher to detect a bias is therefore:

$$\min_k\left(\frac{1}{{\varepsilon_k}^2} \times \frac{2^N}{\binom{N}{k}}\right)$$

As FLIP cipher always applies the filtering function on entries of constant Hamming weight k, there is no need to scale the probability; k being fixed to $\frac{N}{2}$, we care about $\varepsilon_{N/2}$ and balancedness over $E_{N,N/2}$. Balancedness for all weight is still important in this setting; a guess and determine technique consists in fixing some entries, modifying the weight. Therefore we study the balancedness criterion for all Hamming weight of the FLIP function instances.

The number of variables in the functions of the FLIP instances is never a power of 2. Moreover, the filtering function is defined with a direct sum, and has a small degree compared to the number of variables. In this sense, there is no way that the filtering function of FLIP could be weightwise perfectly balanced neither weightwise almost perfectly balanced. However, we calculate the bias for each weight on toy versions of FLIP, and it appears that for all not extreme k (k close to 0 or N) the Boolean function is not balanced. Nevertheless the calculated bias are totally not exploitable to distinguish the output from another random sequence, regarding that we cannot have more than 2^{80} or 2^{128} bits of keystream.

Using the direct sum structure of the FLIP functions into the proof of Lemma 1, we can exactly compute the values $|\{x \mid f(x) = 1, \mathsf{wH}(x) = k\}|$ for all k for the four filtering functions and the bias from a random sequence. In table 2 we summarize our computation results, providing for which weights k the bias is inferior to $2^{-\frac{\lambda}{2}}$. As the impact of guess and determine attack on the balancedness on restricted inputs is not known, for the 4 instances we study this criterion for three particular guesses. The first guess consists in canceling (forcing to be 0) $\lambda/2$ linear variables, the second one on $\lambda/2$ variables of the quadratic part and the third one on $\lambda/2$ variables of the highest degree monomials. These three strategies represent the best deterioration that an attacker can obtain on one part of a FLIP filtering function. We assume that if none of these strategy reveal a sufficient bias for a concrete cryptanalysis then no hybrid approach will lead to an efficient attack.

Interpreting the results of table 2, as $k = \frac{N}{2}$ is in the ranges of the v_0 column for the 4 instances of f, we conclude that we can not apply our distinguisher based on the balancedness criterion as it will require more than 2^λ keystream bits. Even considering a combination with guess and determine attack, the biases on the simpler functions obtained are insufficient to mount a concrete attack.

4.1.3 $\mathsf{NL}_{E_{N,k}}$ of FLIP and fast correlation attack

Fast correlation attack: In our model we target only the function f generating the keystream, in this sense correlation attacks cannot work because there is no way we can do a divide-and-conquer technique as in Siegenthaler's attack [Sie84] in this case only fast correlation attack [MS88a] could have smaller complexity than an exhaustive search in our

Table 2: Balancedness (with constant weight inputs) bias on FLIP filtering function instances, and modified instances. N is the number of variables of the instance, ℓ is the number of variables guessed to be 0. v_i stands for the range of weights with bias $< 2^{-\ell}$ for the various strategies of guesses i and N_i the number of variables of the resulting function, with v_0 the attack without guesses of the N variable function.

Instance	N	v_0	ℓ	N_1	v_1	N_2	v_2	N_3	v_3
FLIP-530	530	$[78, 482]$	40	490	$[134, 446]$	450	$[70, 409]$	250	$[30, 190]$
FLIP-662	662	$[102, 621]$	40	622	$[178, 585]$	582	$[97, 547]$	242	$[29, 185]$
FLIP-1394	1394	$[207, 1325]$	64	1330	$[348, 1266]$	1266	$[203, 1205]$	594	$[69, 514]$
FLIP-1704	1704	$[257, 1643]$	64	1640	$[429, 1582]$	1576	$[254, 1519]$	610	$[70, 533]$

Table 3: Lower bound on $\mathsf{NL}_{E_{N,k}}$ of FLIP instances, N refers to the number of variables, v_0 the range of weight k for which $\frac{\mathsf{NL}_{E_{N,k}}(f)}{\binom{N}{k}} \geq 0.499$. ℓ refers to the number of canceled monomials of the quadratic part by the guess strategy, N_2 and v_2 are the corresponding number of variables and range of weights.

Instance	N	v_0	ℓ	N_2	v_2
FLIP-530	530	$[107, 464]$	40	450	$[142, 383]$
FLIP-662	662	$[136, 556]$	40	582	$[189, 453]$
FLIP-1394	1394	$[221, 1239]$	64	1266	$[296, 1094]$
FLIP-1704	1704	$[266, 1492]$	64	1576	$[363, 1321]$

model. The attacker first computes the linear approximations l_k of f_k where $\mathsf{NL}_{E_{N,k}}(f)$ is small. Approximating the keystream equations by their linear approximation, she builds a linear system and relies it to a decoding problem. When no particular code structure is used, this system can be seen by the attacker as an instance of the Learning Parity with Noise problem, where the noise parameter is $\varepsilon_k = \frac{\mathsf{NL}_{E_{N,k}}}{\binom{N}{k}}$. Standard algorithms can be used to solve this instance, as BKW [BKW03] or LF [LF06] algorithms giving an attack with data complexity of $\mathcal{O}(2^h \varepsilon_k^{-2(x+1)})$ where the parameters h and x depend on the algorithm used and the number of variables used in ℓ_f.

The high number of variables of FLIP instances makes impossible to compute exactly the nonlinearity on constant weight inputs. Nevertheless using the lower bound of the $\mathsf{NL}_{E_{N,k}}$ of a direct sum (see Section 3.2.4) we can give a lower bound of the nonlinearity on constant weight inputs for the 4 instances. To do so we recursively compute the $\mathsf{NL}_{E_{N,k}}$ for a direct sum with lower bounds for $\mathsf{NL}_{E_{n,i}}(g)$ and exact value of $\mathsf{NL}_{E_{m,k-i}}(h)$. In table 3 we summarize the results, the exact $\mathsf{NL}_{E_{n,k}}$ values are computed on functions with $n \leq 22$ for the quadratic functions (Dickson functions), $n \leq 15$ for the triangular functions and $n \leq 17$ for sums of two monomials functions. Note that the $\mathsf{NL}_{E_{n,k}}$ of a function in n variables consisting in only one degree n monomial is null for all k. The results are obtained by first combining the quadratic functions, then the triangular part and finally the linear (and higher than 9 degree) part.

The results in table 3 show that for $k = \frac{N}{2}$, $\mathsf{NL}_{E_{N,k}}(f)$ is higher enough to ensure that considering the system of equations described in the attack scenario 2, the LPN system will be unsolvable with data complexity inferior to 2^λ as the error is considered as coming from a Bernouilli distribution with mean $0.499 \leq p \leq 0.5$. Even with a guess and determine attack (with simulated results form the right part of the table), The $\mathsf{NL}_{E_{N,k}}$ of the various functions and the number of variables are too big to lead to a concrete cryptanalysis with an attack scenario based on $\mathsf{NL}_{E_{N,k}}$.

4.1.4 AI_k of FLIP and algebraic attack

Algebraic attack: Assuming that an attacker does not want to distinguish the keystream but instead to recover some internal state of the cipher, it could improve the so-called algebraic attacks. Algebraic attacks (and fast algebraic attacks [Cou03]) were first introduced by Courtois and Meier in [CM03] and applied to the stream cipher Toyocrypt. Their main idea is to build an over-defined system of equations with the initial state of the stream cipher as unknown, and to solve this system with Gaussian elimination. More precisely, by using a nonzero function g such that both g and $h = gf$ have low algebraic degree, an adversary is able to obtain T equations with monomials of degree at most $AI(f)$. Function g can be taken equal to an annihilator of f or of $f \oplus 1$, *i.e.* such that $gf = 0$ or $g(f \oplus 1) = 0$. After a linearization step, the adversary obtains a system of T equations in $D = \sum_{i=0}^{\mathsf{AI}(f)} \binom{N}{i}$ variables. Therefore, the time complexity of the algebraic attack is $\mathcal{O}(D^\omega)$, that is, $\mathcal{O}(n^{\omega AI(f)})$ where ω is the exponent in the complexity of solving a linear system ($\omega = 3$ for the naive approach, we consider $\omega = \log 7$ for a realistic attack).

The fast version consists in finding a function g with low degree and a function h with degree slightly higher than $\mathsf{AI}(f)$ which are solutions of the equation $h = gF$, providing an easier algebraic system to solve. In our context of fixed Hamming weight, the data complexity will drop to $D' = \sum_{i=0}^{\mathsf{AI}_k(f)} \binom{N}{i}$, the number of independent equation needed to mount a solvable algebraic system.

For the FLIP family of stream cipher, the filtering function is defined by a direct sum but we cannot conclude on the exact algebraic immunity of FLIP instances (regarding the restricted input) with the corresponding bound (given in Section 3.3.3). However, we have shown with Theorem 1 (in Section 2.3) that defining a Boolean function with a direct sum can lower the algebraic immunity regarding the degree when the input has a fixed Hamming weight. Therefore Theorem 1 can be useful to find a lower bound on the algebraic immunity of FLIP. To apply it, we define each filtering function of the four instances of FLIP using the following form:

$$F(x, y) = f(x) \oplus g(y)$$

where f has n variables and a high algebraic immunity and g has m variables with the smallest degree as possible. The inputs in FLIP are the words of Hamming weight $\frac{N}{2}$ where $N = n + m$. Then to apply Theorem 1, n and m have to satisfy the condition $n \leq \frac{N}{2} \leq m$. To garantee that g has the smallest degree possible (to get the bound of Theorem 1 as meaningful as possible) we set the function g to be the first sum of all monomials of degree lesser or equal to k which has more than $\frac{N}{2}$ variables. Then we need to know the algebraic immunity of the other function f; due to the particular shape of f, we need more results on the algebraic immunity of functions with few monomials. We begin with a property linking the algebraic immunity of two functions.

Proposition 11. *Let $f(x_1, x_2, x_3, \cdots, x_n)$ be a Boolean function in n variables such that there exists two variables (x_1 and x_2 without loss of generality) satisfying:*

$$\forall x \in \mathbb{F}_2^{n-2} \; f(0, 0, x) = f(0, 1, x) = f(1, 0, x)$$

Let $F(X, x_3, \cdots, x_n)$ be the Boolean function in $n - 1$ variables defined by :

$$\forall x \in \mathbb{F}_2^{n-2} \; F(1, x) = f(1, 1, x) \text{ and } F(0, x) = f(0, 0, x)$$

If $\mathsf{AI}(f) \leq d$ then $\mathsf{AI}(F) \leq d$.

Proof: Formally we prove that if there exists a non null function g in n variables of degree $\leq d$ such that $fg = 0$ (respectively $(f + 1)g = 0$) over all $\mathbb{F}_2^n$ then there exists

a non null function G in $n-1$ variables of degree $\leq d$ such that $FG = 0$ (respectively $(F+1)G = 0$) over all $\mathbb{F}_2^{n-1}$.

First we decompose g in an unique way:

$g = g(x_1, x_2, \cdot, x_n) = g_1 x_1 + g_2 x_2 + g_{mix} x_1 x_2 + g_{none}$

where:

- $g_{mix} x_1 x_2$ contains all monomials with both x_1 and x_2,
- $g_1 x_1$ contains all monomials with x_1 and without x_2,
- $g_2 x_2$ contains all monomials with x_2 and without x_1,
- g_{none} contains all monomials without x_1 and without x_2.

As g is non null there exists at least one $\bar{x} \in \mathbb{F}_2^{n-2}$ such that for at least one of the four entries $(0,0,\bar{x}), (0,1,\bar{x}), (1,0,\bar{x})$ or $(1,1,\bar{x})$ the function g is not null. Therefore we realise a disjunction of cases, for the four possible values of (x_1, x_2) we build a different function G. In each case G is an annihilator of F (respectively $F+1$) in $n-1$ variables of degree $\leq d$ based on the fact that the function in $n-2$ variables defined as $\forall x \in \mathbb{F}_2^{n-2}\ g(x_1, x_2, x)$ is non null and with degree $\leq d$.

- Case $g(1,1,x)$ is not the null function:

  As $\forall x \in \mathbb{F}_2^{n-2}$, $g(1,1,x) = g_1 + g_2 + g_{mix} + g_{none}$, we can define G as

  $$G(X,x) = (g_1 + g_2 + g_{mix})X + g_{none},$$

  therefore, $\forall x \in \mathbb{F}_2^{n-2}$:

  if $X = 0$ then $F(X,x)G(X,x) = f(0,0,x)g_{none} = f(0,0,x)g(0,0,x) = 0$,

  else $X = 1$ then $FG = f(1,1,x)(g_1 + g_2 + g_{mix} + g_{none}) = f(1,1,x)g(1,1,x) = 0$.

  In both cases G is an annihilator of F (respectively of $1+F$), non null because $g(1,1,x)$ is non null and of degree $\leq d$ because the degree of g_1 and g_2 are upper bounded by $d-1$, the degree of g_{mix} is upper bounded by $d-2$ and the degree of g_{none} is upper bounded by d.

- Case $g(1,0,x)$ is not the null function:

  we define $G(X,x) = g_1 + g_{none} + (g_2 + g_{mix})X$, then $\forall x \in \mathbb{F}_2^{n-2}$:

  if $X = 0$ then $F(X,x)G(X,x) = f(0,0,x)(g_1 + g_{none}) = f(1,0,x)g(1,0,x) = 0$,

  else $X = 1$ then

  $$F(X,x)G(X,x) = f(1,1,x)(g_1 + g_2 + g_{mix} + g_{none}) = f(1,1,x)g(1,1,x) = 0.$$

  We can conclude similarly than the previous item.

- Case $g(0,1,x)$ is not the null function:

  we define $G(X,x)$ as $g_2 + g_{none} + (g_1 + g_{mix})X$; we only switch the impact of x_1 and x_2 from the previous item.

- Case $g(0,0,x)$ is not the null function:

  we define $G(X,x) = (g_1 + g_2 + g_{mix}) * X + g_{none}$ as for the first item and as $g(0,0,x)$ is not null in this case we can conclude the same way.

We conclude that for all f with this property relatively to two variables there exists a function G with the described property, and therefore $\mathsf{AI}(f) \leq d$ then $\mathsf{AI}(F) \leq d$.

□

The contraposition of this Proposition gives that for such functions $\mathsf{AI}(F) > d$ implies $\mathsf{AI}(f) > d$, therefore, from the lower bound on the algebraic immunity of a function F we can derive a lower bound on the algebraic immunity of a function f in more variables.

We use this Proposition on the specific functions we obtain by partitioning in two the instances of FLIP; we want to determine the algebraic immunity of a function being the direct sum of high degree monomials.

Proposition 12. *Let $f(x_1, \cdots, x_n)$ be a Boolean function in n variables satisfying the two following properties:*

1. *f is a direct sum of d monomials,*
2. *$\forall i \in [k, d]$, f has a monomial of degree i; where k is the smallest degree over all monomials of f.*

Then $\mathsf{AI}(f) = d$.

Proof: First, property 1 implies that $\mathsf{AI}(f) \leq d$; each one of the d monomials can be annihilated by the degree one function $1 - x_i$ with x_i one variable of the monomial; therefore the product, a function of degree d annihilates f.

We then prove that $\mathsf{AI}(f) \geq d$ using Proposition 11.

The first property guarantees that each variable is used only once, therefore for two variables in the same monomial the contribution to f is the same. Indeed, without loss of generality calling x_1 and x_2 these two variables, parts of a monomial of degree ℓ, and reordering the variables we can write f as:

$$f(x_1, x_2, \cdots, x_\ell, x_{\ell+1}, \cdots, x_n) = x_1 x_2 x_3 \cdots x_\ell + h(x_{\ell+1}, \cdots, x_n),$$

where h is a direct sum of $d - 1$ monomials, in $n - \ell$ variables.

Denoting $x_3, \cdots, x_\ell, x_{\ell+1}, \cdots, x_n$ by x we can then verify that $f(0, 0, x) = f(1, 0, x) = f(0, 1, x)$. It implies that for each couple of variables in the same monomial we can apply Proposition 11. Therefore, we can reduce the number of variable of f, one by one, contracting a product of two variables in a new one at each step, keeping the property that f and the newly obtained function F share the same AI.

The second property guarantees that f can lead to a triangular function of degree d, as all monomials of degree $\geq k$ are already in f, there are $d - k$ other monomials, all of degree $\geq k$. These monomials are recursively taken and contracted to be the monomials of degree $k - 1$ to 1 of the triangular, by applying several time Proposition 11.

As the triangular function of degree d has an algebraic immunity of d we get $\mathsf{AI}(f) = d$

□

The partitioning of the instances in a direct sum of high degree monomials and a low degree function, makes the first part to be a direct sum of degree d with more than d monomials of high degree. The algebraic immunity of a direct sum of two functions being at least as high as the highest one we can apply Proposition 12 on the d monomials of highest degree of this function and we conclude that the algebraic immunity of the direct sum is d, its degree.

To sum up; for the 4 instances we can write the function as the direct sum of f and g minimizing the degree of g. We can determine exactly the algebraic immunity of f and apply the lower bound of Theorem 1 for the AI_k of the 4 instances.

With the lower bounds on the four instances of FLIP, we then can calculate a lower bound on the complexity of an algebraic attack on FLIP which is $\left(\sum_{i=1}^{\mathsf{AI}} \binom{n}{i}\right)^{\log 7}$. The results and the bounds giving us the complexity are shown in table 4.

Table 4: Lower bounds on AI_k of FLIP instances and complexity of the algebraic attack, N refers to the number of variables, $\mathsf{AI}(f)$ is equal to the degree of f, n is the number of variables in f, and the lower bound is given with theorem 1. f is the Boolean function defined by the direct sum of all the monomials of degree strictly greater than $\deg(g)$ and g takes all monomials of degree less or equal to $\deg(g)$, D refers to the number of variables after linearization and $D^{\log(7)}$ to the corresponding attack complexity.

Instance	N	$\mathsf{AI}(f)$	$\deg(g)$	n	*Bound*	D	$D^{\log(7)}$
FLIP-530	530	9	5	240	4	$2^{31.6}$	$2^{88.7}$
FLIP-662	662	15	9	300	6	$2^{46.7}$	$2^{131.1}$
FLIP-1394	1394	16	10	648	6	$2^{53.2}$	$2^{150.2}$
FLIP-1704	1704	23	15	780	8	$2^{70.6}$	$2^{198.2}$

It is important to notice that the lower bounds here could be not tight and the algebraic immunity (on constant Hamming weight inputs) of FLIP instances could be as great as $\mathsf{AI}(F)$. Then it remains to prove the exact $\mathsf{AI}_{\frac{n}{2}}$ of FLIP instances, which we could not determine by computation due to the high number of variables of these functions.

Moreover, it remains not clear whether or not a guess and determine attack targeting the algebraic immunity could apply. Considering all possible guesses leads to functions where the lower bound decreases enough to contemplate an attack, but different aspects impeach us to exhibit an attack. Firstly, the guessing technique conducts to cancel or diminish the degree of some monomials; when a monomial is canceled some variables unguessed inside could be ones, and therefore the considered function is evaluated on an input where we cannot know exactly the Hamming weight. Secondly, the probability of obtaining a targeted weight on a targeted simpler function enough times (*i.e.* disposing of enough keystream bits with a coherent set of permutations) fastly decreases. Finally, the time complexity of the attack exhibited with the lower bound (Table 4) is out of reach when no guesses are made; computational trials make us believe that the additional complexity cost of fixing ℓ variables of the filtering function (additional cost of 2^ℓ) compensate enough the potential decrease of AI_k of the weaker function obtained.

An interesting point is that the high number of triangular functions used in FLIP were used to prevent the guess and determine attack combined with fast algebraic attack. Nevertheless, it appears here that if the number of triangular functions is smaller, then the lower bound on the algebraic immunity is increased. Supposing that we cannot compute the exact AI_k of FLIP filtering functions, there is then for now a compromise to find on the number of triangular functions: if there are few triangular functions, then the guess and determine attack consisting in canceling monomials with high degree is more efficient, but if there are too many triangular functions, then the lower bound on the algebraic immunity of FLIP is decreasing. Consequently, this attack cannot serve to determine the optimal number and size of triangular functions.

4.2 Conclusions and cautionary note

As a preliminary conclusion, the lower bounds exhibited for balancedness, nonlinearity and algebraic immunity do not reveal any concrete attack on the four instances of the FLIP family of stream ciphers. In this section the security analysis focusing on the Boolean function is made in the right model (*i.e.* we have taken into account that the entries of the filtering function only reach $E_{n,\frac{n}{2}}$ and not $\mathbb{F}_2^n$).

On one side, we get only lower bounds on the two criteria of nonlinearity and algebraic immunity, leading to lower bounds on the attack complexity rather than practical bounds. On the other side, if the bounds we proved for AI_k are tight, we could not infirm that a deviation of a guess and determine attack on FLIP could be efficient in that way.

Nevertheless, our security analysis of FLIP is in the right model and the lower bounds (which can be not tight) do not exhibit any attack regarding the security claims of the authors of FLIP.

We remark that the two potential tweaks proposed by the authors (use whitening or add a linear layer) avoid a constant Hamming weight cryptanalysis; in both cases, f's input is then $\mathbb{F}_2^n$, but the whitening still enables to know its parity.

References

[ARS+15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT (1)*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454. Springer, 2015.

[AZ09] Martin Aigner and Gnter M. Ziegler. *Proofs from THE BOOK*. Springer Publishing Company, Incorporated, 4th edition, 2009.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[Car10] Claude Carlet. *Boolean Functions for Cryptography and Error Correcting Codes,*, pages 257–397. Y. Crama and P. Hammer eds, Cambridge University Press, 2010.

[CCF+16] Anne Canteaut, Sergiu Carpov, Caroline Fontaine, Tancrède Lepoint, María Naya-Plasencia, Pascal Paillier, and Renaud Sirdey. Stream Ciphers: A Practical Solution for Efficient Homomorphic-Ciphertext Compression. In Thomas Peyrin, editor, *FSE*, volume 9783 of *Lecture Notes in Computer Science*, pages 313–333. Springer, 2016.

[CFGR12] Claude Carlet, Jean-Charles Faugère, Christopher Goyet, and Guénaël Renault. Analysis of the algebraic side channel attack. *J. Cryptographic Engineering*, 2(1):45–62, 2012.

[CM03] Nicolas Courtois and Willi Meier. Algebraic Attacks on Stream Ciphers with Linear Feedback. In *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, pages 345–359, 2003.

[Cou03] Nicolas Courtois. Fast Algebraic Attacks on Stream Ciphers with Linear Feedback. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, pages 176–194, 2003.

[Did06] F. Didier. A New Upper Bound on the Block Error Probability After Decoding Over the Erasure Channel. *IEEE Transactions on Information Theory*, 52(10):4496–4503, Oct 2006.

[DK13] Ilya Dumer and Olga Kapralova. Spherically punctured biorthogonal codes. *IEEE Trans. Information Theory*, 59(9):6010–6017, 2013.

[DK17] Ilya Dumer and Olga Kapralova. Spherically punctured reed-muller codes. *IEEE Trans. Information Theory*, 63(5):2773–2780, 2017.

[DLR16] Sébastien Duval, Virginie Lallemand, and Yann Rotella. Cryptanalysis of the FLIP Family of Stream Ciphers. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 457–475. Springer, 2016.

[Fil16a] Yuval Filmus. Friedgut-kalai-naor theorem for slices of the boolean cube. *Chicago J. Theor. Comput. Sci.*, 2016, 2016.

[Fil16b] Yuval Filmus. An orthogonal basis for functions over a slice of the boolean hypercube. *Electr. J. Comb.*, 23(1):P1.23, 2016.

[FKMW16] Yuval Filmus, Guy Kindler, Elchanan Mossel, and Karl Wimmer. Invariance principle on the slice. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 15:1–15:10, 2016.

[FM16] Yuval Filmus and Elchanan Mossel. Harmonicity and invariance on slices of the boolean cube. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 16:1–16:13, 2016.

[GGPS17] Sugata Gangopadhyay, Aditi Kar Gangopadhyay, Spyridon Pollatos, and Pantelimon Stanica. Cryptographic boolean functions with biased inputs. *Cryptography and Communications*, 9(2):301–314, 2017.

[JD06] Antoine Joux and Pascal Delaunay. Galois LFSR, Embedded Devices and Side Channel Weaknesses. In Rana Barua and Tanja Lange, editors, *Progress in Cryptology - INDOCRYPT 2006, 7th International Conference on Cryptology in India, Kolkata, India, December 11-13, 2006, Proceedings*, volume 4329 of *Lecture Notes in Computer Science*, pages 436–451. Springer, 2006.

[LF06] Éric Levieil and Pierre-Alain Fouque. An Improved LPN Algorithm. In Roberto De Prisco and Moti Yung, editors, *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 2006.

[MJSC16] Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet. Towards Stream Ciphers for Efficient FHE with Low-Noise Ciphertexts. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT (1)*, volume 9665 of *Lecture Notes in Computer Science*, pages 311–343. Springer, 2016.

[MPC04] Willi Meier, Enes Pasalic, and Claude Carlet. Algebraic Attacks and Decomposition of Boolean Functions. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 474–491. Springer, 2004.

[MS88a] W. Meier and O. Staffelbach. Fast Correlation Attacks on Certain Stream Ciphers. In *EUROCRYPT*, pages 301–314, 1988.

[MS88b] Willi Meier and Othmar Staffelbach. Fast Correlation Attacks on Stream Ciphers (Extended Abstract). In Christoph G. Günther, editor, *EUROCRYPT*, volume 330 of *Lecture Notes in Computer Science*, pages 301–314. Springer, 1988.

[Sie84] Thomas Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Trans. Information Theory*, 30(5):776–780, 1984.

[Sie85] Thomas Siegenthaler. Decrypting a Class of Stream Ciphers Using Ciphertext Only. *IEEE Trans. Computers*, 34(1):81–85, 1985.