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

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

[1] LAGA, Department of Mathematics, University of Paris 8, and Paris 13 and CNRS, Saint–Denis cedex 02, France.
E-mail: claude.carlet@univ-paris8.fr
[2] Inria, CNRS, ENS and PSL Research University, Paris, France.
E-mail: pierrick.meaux@ens.fr
[3] Inria, University Pierre
& Marie Curie Paris 6, Paris, France.
E-mail: 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 the 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 one of the encryption schemes appeared in the very last years: specifically designed 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 with input 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 by the intermediate of a generator of permutation. 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 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: how is behaving f 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 $\mathbb{F}_2^n$; and therefore they do not apply on restricted inputs. Therefore in this work we study the 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, because of the application of a permutation at each clock cycle on the entire state. It seems very difficult to find a bias between the output to the function and a fixed set of input variables. In general, it seems very difficult to apply Siegenthaler's attack on other ciphers by regarding restricted input, because the principle of the correlation attack is to make an exhaustive search on some part of the initial state, without having any restriction on the rest of the state, while here, by fixing the weight, we impose 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.

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 at least 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. We shall be in particular interested in the case of $\mathcal{E} = \{E_{n,1}, \dots, E_{n,n-1}\}$, 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*.

A second parameter, which plays an important role for quantifying the contribution of the function to a 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.

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

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 previously presented for the particular case of a design it introduce or the cryptanalysis it develops. A study has been made by Yuval Filmus et al. on the restrictions of Boolean functions to sets of inputs of fixed Kamming 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.

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. The authors also describe a correlation attack in this setting. Therefore [JD06] compares to our work by using the balancedness flaw on a function on the sets $\{x|\mathsf{w_H}(x) = k\}$ (the fact that this function is not weightwise perfectly balanced) with other equations to attack these LFSR.

On the algebraic immunity criteria, following the algebraic side channel attacks, the work [CFGR12] realises a theoretical study of the algebraic phase of this kind of attacks on block ciphers. The authors define an adapted notion of algebraic immunity depending on the leakage model, Hamming weight or Hamming distance and show that in both cases they are able to obtain enough equations of degree one and that they can solve the algebraic system with Gröbner methods. For our work we consider a reverse approach, we also study an adapted notion of the algebraic immunity (on f with restricted input) and for the case we focus mostly on - fixed Hamming weight input - it is related to side channel approaches. Then our study directly applies to the Hamming weight model but not to the Hamming distance model, another major difference is that we focus on functions with one bit of output and not S-boxes functions.

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. For each criterion, we compare its behaviour in this constrained framework to the properties well known in $\mathbb{F}_2^n$, we consider the functions with highest criterion on $E_{n,k} = \{x|\mathsf{w_H}(x) = k\}$ that we can construct. Moreover, we study the degradation of the parameters for optimal functions in the whole space.

More precisely for the balancedness criterion, we prove necessary conditions on the Algebraic Normal Form of f to be weightwise perfectly balanced and we exhibit a class of such functions and a secondary construction for designing them. 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 all input space 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 NL_k for all $k \in [1, n]$. We also determine some bent functions which are linear for all restricted Hamming weight and hence have null NL_k for every k. We generalise the algebraic immunity upper bound for all set E and give precise results in the constant Hamming weight case. We show how the general algebraic immunity can decrease on $E_{n,k}$ and prove counter-intuitive results for direct sums (used in the design of the FLIP Boolean function).

We give a cryptanalysis aspect of this study explaining in which case *constant Hamming weight cryptanalysis* is applicable in general and how it can be. We provide a proof of concept analysis on the stream cipher Grain assuming a side channel attacker. We specially analyse 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 study the criterion of balancedness, then Section 3 concerns the nonlinearity and the algebraic immunity study is made in Section 4. Section 5 details constant Hamming weight cryptanalysis and finally Section 6 presents the security analysis of FLIP with fixed Hamming weight input.

2 Balancedness

2.1 Weightwise perfectly balanced Boolean functions

Notation 1. *We denote by* $\mathsf{w_H}(f)_k$ *the Hamming weight of the evaluation vector of 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* $k \in \{1, \dots, n-1\}$*, 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* $f(0, \dots, 0) \neq f(1, \dots, 1)$ *and more precisely that*

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

This last constraint does not reduce the generality (when $f(0, \dots, 0) \neq f(1, \dots, 1)$), 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 $k \in [1, \dots, 2^{\ell-1}-1] \cup [2^{\ell-1}+1, \dots, 2^\ell - 1]$ 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.*

Relation with ANF Recall that any Boolean function over $\mathbb{F}_2^n$ has a unique algebraic normal form (ANF) $f(x) = \sum_{I \subseteq \{1,\dots,n\}} a_I \prod_{i \in I} x_i$, 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. Function f is affine if and only if its algebraic degree is at most 1.

Remark 1. For every even n and $\varepsilon = 0$ or 1, function $\ell(x_1, x_2, \dots, x_n) = \varepsilon + x_1 + x_2 + \dots + x_{\frac{n}{2}}$ is balanced on all words of fixed odd Hamming weight, since for such word, either $\mathsf{w_H}(x_1, \dots, x_{\frac{n}{2}})$ is odd and $\ell(x_1, x_2, \dots, x_n) = \varepsilon + 1$, or $\mathsf{w_H}(x_{\frac{n}{2}+1}, \dots, x_n)$ is odd and $\ell(x_1, x_2, \dots, x_n) = \varepsilon$, 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:

Proposition 1. *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 $|\ell_f| = \frac{n \pm 1}{2}$ 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 2. *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.

□

Remark 2. The previous results show that the ANF of every weightwise perfectly balanced function of 2^ℓ variables contains monomials of degrees 2^0, 2^1 and $2^{\ell-1}$. It raises the question whether having monomials whose degrees cover all powers of 2 is a necessary condition for weightwise perfectly balancedness. It turns out that it is not necessary; for example the function of 16 variables and algebraic degree 3

$$f = \sum_{i=1}^{8} x_i + \sum_{i=1}^{4} x_i x_{i+8} + x_1 x_2 x_5 + x_1 x_4 x_{16}$$

is weightwise balanced for $k \in [1, \ldots, 4]$ and can be completed in a weightwise perfectly balanced function only by adding monomials of degree greater than 4.

Constructions A well-known secondary construction of Boolean functions is the so-called *direct sum*: given two functions $f(x)$ and $g(y)$ depending on distinct variables, the Boolean function $h(x, y) = f(x) + g(y)$, where $x, y \in \mathbb{F}_2^n$, is called the direct sum of f and g. 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 exists 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 $g_1(0 \ldots 0) + g_1(1 \ldots 1) + g_2(0 \ldots 0) + g_2(1 \ldots 1) \equiv 0 \mod 2$, 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 with $\frac{n}{2}$ 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 $k \in [1, n-1]$:

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

Now we suppose that f is weightwise perfectly balanced; 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 $g_1(0 \ldots 0) + g_1(1 \ldots 1) + g_2(0 \ldots 0) + g_2(1 \ldots 1) \equiv 1 \mod 2$ □

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 1. *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 $x = y = (0,\dots,0)$ and we have $h(x,y) = f(0,\dots,0) + g(0,\dots,0) = 0$.

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

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

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 1.

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

Proposition 3. *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 characterise 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 1 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 1'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.

□

Remark 3 (Balancedness degradation with weightwise consideration). 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 $k \in [1, n-1]$.

Proof. The first elementary symmetric Boolean function $\sigma_1 = \sum_{i=1}^{n} x_i = \mathsf{w_H}(x) \pmod 2$ is $(n-1)$-resilient and is constant on all fixed weight input, its weightwise restrictions are as much unbalanced as possible. □

2.2 A relation between balancedness when the input weight is fixed and a transform similar to the Walsh transform involving symmetric functions

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

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

(sum performed in $\mathbb{F}_2$) and Σ the vectorial (n,n)-function whose ith coordinate function is σ_i. 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 Nonlinearity of Boolean functions with restricted input

3.1 Definitions and upper bound

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 betweeen 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 $\mathbb{F}_2^n$.*

We deduce from the observations above:

Proposition 4. *For every n-variable Boolean function f over $\mathbb{F}_2^n$ and every subset E of $\mathbb{F}_2^n$, we have:*

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

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

We have:

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

The maximum of a sequence of numbers being always bounded below by the arithmetic mean, we deduce:

Proposition 5. *For every subset E of $\mathbb{F}_2^n$ and every Boolean function f defined over E, we have:*

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

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:

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:

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

and

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

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

Proposition 6. *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:*

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

where

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

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

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

where

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

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

Fixed input Hamming weight Let us now consider the case of $E = E_{n,k}$ for $k = 0, \dots, n$, where $E_{n,k}$ is the set of vectors of Hamming weight k in $\mathbb{F}_2^n$. Denoting $\mathsf{NL}_{\mathsf{w_H}(x)=k}(f)$ as $\mathsf{NL}_k(f)$, we have:

$$\mathsf{NL}_k(f) \leq \frac{\binom{n}{k}}{2} - \frac{1}{2}\sqrt{\binom{n}{k}}. \tag{6}$$

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 $\mathsf{NL}_k(f) \leq \left\lfloor \frac{\binom{n}{k}}{2} - \frac{1}{2}\sqrt{\binom{n}{k}} \right\rfloor$, 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. It equals then $\binom{i}{\frac{i}{2}}\binom{n-i}{k-\frac{i}{2}}$. Clearly, since $\sum_{\substack{(x,y)\in E^2 \\ x+y=a}} (-1)^{f(x)+f(y)}$ 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, λ equals twice the sum of an odd number of integers equal to ± 1; it is then strictly positive. For instance for $k = 2$ and $i = 4$, $\binom{i}{\frac{i}{2}}\binom{n-i}{k-\frac{i}{2}} = 6\binom{n-4}{0} = 6$ 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. We deduce:

Corollary 3. *For all n and $k \in \{1, \dots, n-1\}$, Bound (6) is never tight, except maybe for two particular pairs (n,k) : $(50,3)$ and $(50,47)$.*

Proof. As already observed, the bound can only be tight when $E_{n,k}$ is a square. Erdős showed that the binomial coefficient $\binom{n}{k}$ with $3 \leq k \leq n/2$ is a power of an integer for the single case $\binom{50}{3}$, therefore we only consider the cases $k \in 0, 1, 2, n-2, n-1, n$.

- $k = 0$ (or $k = n$): Proposition 5 gives $\mathsf{NL}_0(f) \leq 0$ 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 7, every function restricted to its entries of Hamming weight 1 (or $n - 1$) is linear therefore $\mathsf{NL}_1(f) = 0$ whereas the bound tells $\mathsf{NL}_1(f) \leq \frac{n - \sqrt{n}}{2}$.
- $k = 2$ (or $k = n - 2$):
  Using corollary 2, 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 $\mathsf{NL}_{E_{n,2}}(f) < \frac{|E_{n,2}| - \sqrt{|E_{n,2}|}}{2}$.

□

For $k=0$ or n, $\mathsf{NL}_k(f)$ is of course always null and reaches the bound as $E_{n,k}$ has size 1.

3.2 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.

Definition 4. *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, $\mathsf{NL}_k(f)$ is the distance between f's truth table restricted to Hamming weight k entries and $RM(1,n)_k$. The maximal value of $\mathsf{NL}_k(f)$ when f ranges over the set of all Boolean functions equals the covering radius of $RM(1,n)_k$.

Proposition 7. *$RM(1,n)_k$ is a linear code of parameters $[\binom{n}{k}, n, d]$ with*

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

Proof. 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 $\max_{(0<\ell\leq n/2)}$. □

We observed that this gives:

- $0\leq k<n/2$, $d=\binom{n-1}{k-1}$
- $k=n/2$, $d=\binom{n-2}{k-2}$
- $n/2<k\leq n-1$, $d=\binom{n-1}{k}$
- $k=n$, $d=1$.

Note that the maximal value of $\mathsf{NL}_k(f)$ *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 correcting code perspective.

3.3 Nonlinearity degradation with weightwise consideration

Fixing the input Hamming weight may deteriorate in an extreme way the nonlinearity of a Boolean function: for every n, there exist n-variable bent functions f such that, for every $k=0,\ldots,n$, $\mathsf{NL}_k(f)=0$. This is for instance the case of the function $f(x)=\binom{\mathsf{w_H}(x)}{2}=\sum_{1\leq i<j\leq n}x_ix_j$. 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 6. A Boolean function satisfies $\mathsf{NL}_k(f)=0$ 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 $\sum_{1\leq j_1<\cdots<j_i\leq n}\quad\prod_{l=1}^i x_{j_l}$. We deduce that f satisfies $\mathsf{NL}_k(f)=0$ 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 8. *For every even* $n\geq 4$*, the quadratic bent functions satisfying* $\mathsf{NL}_k(f)=0$ *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 6, a quadratic function satisfies $\mathsf{NL}_k(f)=0$ 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:

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

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:

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

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 $l_i \ne l_{i'}$, 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

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

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 $I = \{1, \dots, n\}$ and if $I \ne \emptyset, \{1, \dots, n\}$, 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) $\ell(1, \dots, 1) = 1$; hence, if this latter equality is true (i.e. if I has odd cardinality), $E \ne \{0\}$;
  - if $\epsilon = 0$ then all equations L_i are equal to $\ell(x) = 0$; then $E \ne \{0\}$ 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 $E \ne \{0\}$ in such case.

The only case where f is bent, that is, where $E = \{0\}$, for $n \ge 4$, is then $\begin{cases} \epsilon = 1 \\ \ell(1, \dots, 1) = 0 \end{cases}$.

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 these functions are EA-equivalent to each others, since this is the case of all quadratic bent functions, but EA-equivalence is not preserving the Hamming weight.

□

4 Algebraic immunity of Boolean functions with restricted input

4.1 General restricted input

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 $n/2$).

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 5 below) 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 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 4. *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 Propositon 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 5. *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 $\mathsf{rank}(M_{e,E}) \geq |E|$, then there exists an annihilator of f of algebraic degree at most e over E.*

Indeed, in Propositon 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 5 below), 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.

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

Corollary 6. *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 leads to the following definition and properties:

Definition 5. *We call* algebraic immunity *of a function f over a set E the number:*

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

Corollary 7. *The algebraic immunity of any Boolean function over E is bounded above by* $\min\{e;\ \mathsf{rank}(M_{e,E}) > \frac{|E|}{2}\}$.

4.2 Input restricted by Hamming weight

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 4. 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 2. *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 all 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 $E_{n,k}$:

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

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:

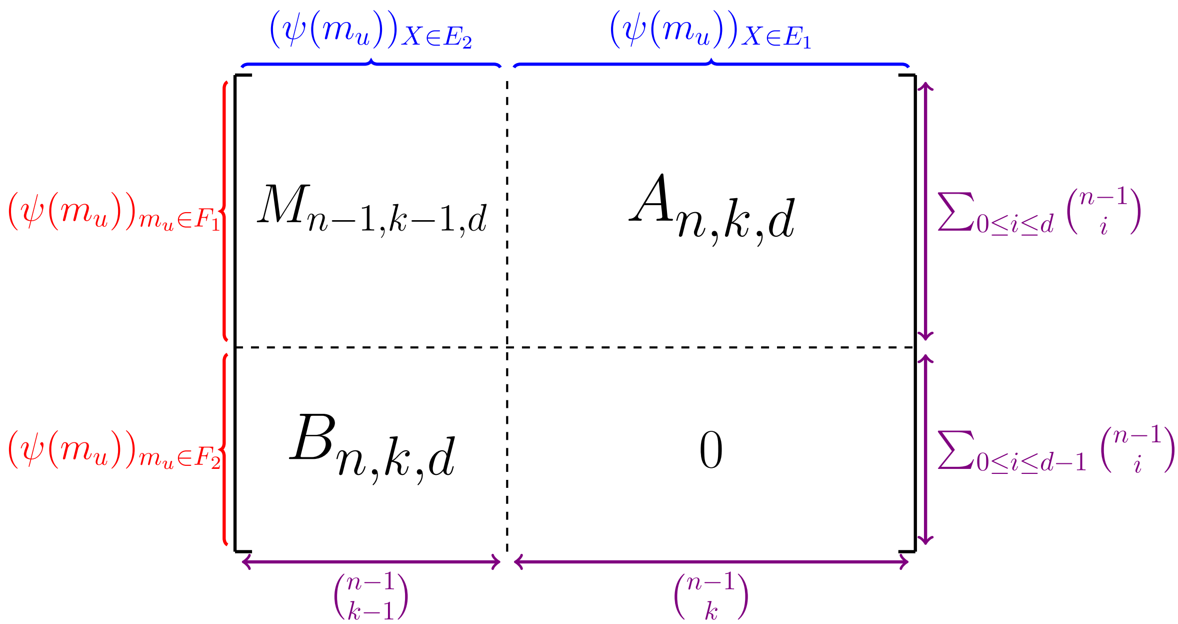

Fig. 1

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

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 initialisation 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 6 and Theorem 2, we deduce:

Corollary 8. *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: $2\binom{n}{k-r} = \binom{n}{k}\frac{2k(k-1)\dots(k-r+1)}{(n-k+r)\dots(n-k+1)}$ 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* $\frac{k-r+2}{n-k+2} > 2^{-1/r}, \dots, \frac{k}{n-k+r} > 2^{-1/r}$, 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 2, we have the dimension of the image of $\psi_{n,k,d}$ and then of its kernel. Let us exhibit a 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:

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

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

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

Proof. Without loss of generality, we take $x^u = x_1x_2\cdots x_j$ 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 $f(x) = \sum_{\{i_1,\ldots,i_{s-j}\}\cap supp(u)=\emptyset} \prod_{l=1}^{s-j} x_{i_l} + \binom{k-j}{s-j} \mod 2$, 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 $\sum_{\{i_1,\ldots,i_{s-j}\}\cap supp(u)=\emptyset} \prod_{l=1}^{s-j} x_{i_l}$ 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 9. *If $d \geq k$, then a basis of $\Im\left(M_{n,k,d}\right) = \mathbb{F}_2^{\binom{n}{k}}$ is the set of all the monomials of degree k.*

Corollary 10. *If $d \geq n-k$, then a basis of $\Im\left(M_{n,k,d}\right) = \mathbb{F}_2^{\binom{n}{k}}$ 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 2 □

4.3 Bounding the algebraic immunity of direct sums of functions

One of the main purposes of this paper is to discuss (see the Section 6) the robustness of the filter function in FLIP [MJSC16]. This function being built as a direct sum (the definition of direct sum has already been given in Section 2), we need then to study the algebraic immunity of direct sums.

Theorem 3. *[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 $n \le k \le m$. Then the following relation holds:*

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

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 input coordinates (and without loss of generality), assume that for every $j = 1, \dots, n$, we have $b_j = a_j + 1$ and for every $j = n + 1, \dots k$, we have $b_j = 1$ (so that for every $j = k + 1, \dots m$, we have $b_j = 0$). We define the following affine function over $\mathbb{F}_2^n$:

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

where the length of the part "$1, \dots, 1$" 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 $\deg(h) + \deg(g) \ge AI(f)$. If $g(b) = 1$, then by applying the same reasoning to $f + 1$ instead of f and $g + 1$ instead of g, we have $\deg(h) + \deg(g) \ge AI(f)$. 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 $k \ge n$, then adding $m \ge k$ 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 3 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 $E_{n,k}$ (as defined after Corollary 4) 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.

But the bound of Theorem 3 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 $f(x_1, x_2, x_3) = x_1 + x_2x_3$; $g(x_4, \dots, x_{10}) = \sum_{i=4}^{10} x_i$ and $k = 5$, then $\mathsf{AI}(f) = \deg(f) = 2$, $\mathsf{AI}(f) - \deg(g) = 1$, and x_2 being an annhilator of $f(x_1, x_2, x_3) + g(x_4, \dots, x_{10})$ over inputs of Hamming weight 5, because $x_2(f(x_1, x_2, x_3) + g(x_4, \dots, x_{10})) = x_2(1 + \sum_{i=1}^{10} x_i)$ vanishes when the input has weight 5, we have $\mathsf{AI}_5(f(x_1, x_2, x_3) + g(x_4, \dots, x_{10})) = 1$; 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 $\mathsf{w_H}(x) \le \frac{n-1}{2}$ and $\mathsf{w_H}(y) \ge \frac{n+1}{2}$ or $\mathsf{w_H}(x) \ge \frac{n+1}{2}$ and $\mathsf{w_H}(y) \le \frac{n-1}{2}$. 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).

5 Cryptanalysis general impact of Boolean functions with restricted input

Motivated by the fixed Hamming weight in the FLIP stream cipher instances, we develop the cryptanalysis aspect of the results presented in the first sections. We present the general aspects and proof of concepts of this approach in this section, the particular security analysis of FLIP stream cipher is developed in the next section.

Previous sections focus on the criteria of Boolean functions with restricted inputs: balancedness, nonlinearity and algebraic immunity, more particularly when the Hamming weight of the input is fixed. In this part we introduce these criteria as new mathematical tools for cryptanalysis, thereafter called the constant Hamming weight cryptanalysis. Knowing the Hamming weight of values in encryption schemes is generally a strong assumption, nevertheless a bad criterion for constant Hamming weight inputs can lead to a strong weakness too. On one side the last part of the three previous sections illustrates that some functions used for their optimality relatively to a particular criterion on all inputs are critically weak for this criterion on constant Hamming weight inputs. On the other side, the given constructions and upper bounds enable to state on the minimum robustness of functions relatively to the attacks. Therefore the following cryptanalysis can be seen as a warning for particular cases where it can be efficiently applied, and serve the designing process of an encryption scheme.

5.1 Prerequisites and principles

Prerequisite to constant Hamming weight input cryptanalysis

Three points are necessary to realise a constant Hamming weight input cryptanalysis. First the cryptanalysis focuses on pairs $\{\mathsf{w_H}(x), f(x)\}$ where the Hamming weight of the variable x is known together with the result of f applied on x. The Boolean function f is the second prerequisite, this function should be known by the adversary and its potentially bad behaviour relatively to balancedness, nonlinearity or algebraic immunity on constant weight entries is the target of the attack. The last point is therefore the weights targeted by the attack, which depend both on the pairs in her possession and the criteria of f, leading to the concept of promising set.

Definition 6. *Promising Set*

Let f be a Boolean function of n variables, for $k \in [0, n]$ we call promising set $\{f(x) | \mathsf{w_H}(x) = k\}$ and denote it S_k if its cardinality is a "big enough" portion of $\binom{n}{k}$.

The "*big enough*" cardinality defining a promising set is dictated by the data complexity of an attack based on f fixed Hamming weight input criteria. In the following part we present three attack scenarii derived from standard symmetric cryptanalysis, respectively based on balancedness, nonlinearity and algebraic immunity on fixed weight input Boolean function. We explain how the standard attack on $\mathbb{F}_2^n$ can be adapted on $E_{n,k}$, then we bound the complexity to stand on the promising sets, enabling to describe a more concrete scenario.

Attack scenarii

For these scenarii we precise our cryptanalysis on a stream cipher, with f the function generating the keystream, fed by linear retroactions. This example enables to describe various attacks as distinguishing, fast correlation attacks and algebraic attacks.

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 our model Boolean functions should be weightwise perfectly balanced functions. Indeed, let us denote $p_k = Pr[f(x) = 1 | \mathsf{w_H}(x) = k] = \frac{1}{2} - \varepsilon_k$. 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 $\frac{1}{{\varepsilon_k}^2}$; defining the promising sets for this attack. 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:

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

It is important to notice that the complexity of the distinguisher can be improved by regrouping different k where the bias ε_k has the same sign. For this improvement the attacker regroups some weights such that the number of ones in the truth table of the Boolean function restricted to this subset of input is much larger than the number of zeros, or the number of zeros much larger than the number of ones. It is then possible to obtain a better bias by combining sets that were not promising by themselves.

Attack scenario 1 (balancedness):

1. Compute $f(x)$ on all weight k entries
2. Determine p_k and ε_k
3. If $|S_k| > \varepsilon_k^2$ the attacker can distinguish the keystream from a random sequence with high probability.
   Improvement:
4. Compute the other p_k and ε_k
5. Join the S_k with same sign of bias giving the set S_K associated to the set of weight K
6. If $|S_K| > \left(0.5 - \frac{\sum_{k \in K} \binom{n}{k} p_k}{\sum_{k \in K} \binom{n}{k}}\right)^2$ the attacker can distinguish the keystream from a random sequence with high probability.

Remark 8. Notice that for this scenario we do not need to assume any property on the structure of the stream cipher itself. More precisely, for LFSR, NLFSR or other building blocks, the attack applies with the same success rate: since $\mathsf{w_H}(x)$ and $f(x)$ are known, the degree of x's coordinates (x as a Boolean vector) in the unknowns is not relevant for this bias. The bias of restricted balancedness has already been studied in a particular case; LFSR with its Galois representation. Joux and Delaunay have shown at INDOCRYPT 2006 that having some information on the weight of the state could lead to guess some output bits, using the number of XOR gates used in the implementation of these stream ciphers [JD06].

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 model. The attacker first computes the linear approximations l_k of f_k where $\mathsf{NL}_k(f)$ 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 $\varepsilon_k = \frac{\mathsf{NL}_k}{\binom{n}{k}}$. 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, and define the promising sets for this attack.

As proved in section 3 the non-linear characteristic of a Boolean function can rudely decrease on a particular restricted set. Moreover as the approximation of a function by portions is at least as good as approximating the all function, the nonlinearity on weight k inputs leads to better linear approximations, even for bent functions. The attack feasibility also relies on the size of $E_{n,k}$; a great approximation for an extreme weight is less expected to give enough equations to be able to solve the LPN problem. On the other side, a better approximation l_k on f_k for a medium weight than l on the entire function f can lead to a solvable system. An improvement of the attack is to consider various promising sets where the bias is less than a constant ε, to combine the corresponding equations, and to solve the LPN problem with error parameter ε.

Attack scenario 2 (non-linearity):

1. Compute the generator matrix M of $RM(1,n)_k$
2. Find the combination of rows of M minimizing the Hamming distance with f
3. Change all equations by $c(x)$ with c the linear function corresponding to the row combination
4. Create the LPN system with parameter $\varepsilon_k = \frac{\mathsf{NL}_k(f)}{\binom{n}{k}}$
5. if sufficiently many independent equations are available and the noise parameter is low enough, solve the system using BKW or LF algorithm.
   Improvement:
6. Compute $\mathsf{NL}_k(f)$ for all weight
7. Keep all the promising sets S_k with $nl_k(f) < \varepsilon$
8. Create the algebraic system with the corresponding equations and solve the corresponding LPN problem.

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 $AI(f)$. 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 linearisation 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, $\mathcal{O}(n^{\omega AI(f)})$.

The fast version consists in finding a function g with low degree and a function h with degree slightly higher than $\mathsf{AI}(f)$ solutions of the equation $h = gF$, providing an easier algebraic system to solve. In our context 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, this data complexity defines the promising sets.

We proved in section 4, that the algebraic immunity of Boolean functions with restricted input of fixed Hamming weight can be much smaller than the original one. For small k or k near to n (the number of variables), we showed that the algebraic immunity cannot be larger than k or $n - k$. Moreover, if k is near to $n/2$, then we also proved that the algebraic immunity cannot be $\lceil n/2 \rceil$ but is in fact much smaller. Those remarks are for any Boolean functions, but for particular Boolean functions well designed in general there is no guarantee that the algebraic immunity has a nice behaviour when restricted to a fixed Hamming weight. In our model, the time complexity of the algebraic attack is $\mathcal{O}(n^{\omega AI_k(f)})$, and it can be improved using various promising sets. The attacker can combine equations from various sets together with weight equations as the attack strategy used on blockciphers [CFGR12]. Using only low degree equations from the most promising sets the attacker can try to build an easier to solve algebraic system.

Attack scenario 3 (algebraic immunity):

1. Set r to 1
2. Compute the generator matrices M of $RM(r,n)_k$
3. Separate the columns of M: in M_0 if the corresponding input x is such that $f(x) = 0$, in M_1 otherwise
4. If $rank(M_0) = rank(M_1) = rank(M)$, increase r by 1 and go to step 2
5. Else, $AI_k(f) = r$, find the linear relation in M_0 or M_1 rows giving the functions g and h of degree equal or less than r such that $fg = h$ on $E_{n,k}$.
6. If sufficiently many independent equations T are available and g has a small enough degree, solve the algebraic system as for standard (fast) algebraic attack.

   Improvement:
7. Compute $AI_k(f)$ for all weight
8. Keep all the promising sets S_k with low $AI_k(f)$
9. Create the algebraic system with the corresponding equations and the weight equations, solve this system.

Discussions Once the prerequisites are fulfilled an attacker can mount a constant weight input cryptanalysis, trying various attack scenarii. One interest of this strategy is the easiness of translating the standard symmetric attacks on all entries to constant weight sets: we showed that we can build a distinguisher or use algebraic or fast correlation attacks in this setting; we describe and apply it to the stream cipher Grain in Section 5.3 as proof of concept. Moreover all the criteria of a Boolean function are decreasing when the input weight is fixed then the described strategy leads to a direct improvement on a standard symmetric cryptanalysis. The valour of this cryptanalysis comes when this improvement is strictly better than using the equations extracted from $\mathsf{w_H}(x) = k$ with standard attacks. It is the case when a criterion falls down, giving critical information on unknowns bit rather than not exploitable equations.

The critical point of constant Hamming weight input cryptanalysis is to find Hamming weights for which the targeted function f has a bad behaviour for one criterion, which we can qualify as a structural flaw. As shown in the previous sections, these flaws do not seem to be rare in commonly used functions. The typical functions used in stream ciphers were not designed at first to have good criteria respected to fixed Hamming weight. This is the reason why constant Hamming weight input cryptanalysis may lead to better attacks.

Another interesting point of this cryptanalysis comes from the fact that the three attack scenarii crucially depend on the promising sets. For all approaches the attack will be easier to mount if the corresponding Boolean criteria are bad for medium weights, where the sets are bigger relatively to 2^n. Nevertheless a promising set with an extreme weight will result in effective attack as the Boolean criteria are generally very low on these sets. An important focus is how the attacker acquires the promising sets; it can be a consequence of the particular design of the cipher, or the result of a guess and determine technique for example. In the following subsection we explain how constant Hamming weight input cryptanalysis can be completed with side channel attacks.

5.2 Constant Hamming weight cryptanalysis and side channel attacks

In this part we consider side channel attacks as a case of application of constant Hamming weight cryptanalysis; it illustrates a proof of concept rather than a real-world attack. As getting informations on Hamming weights corresponds to a leakage model, we describe a setting where a side channel attack enables to build the sets needed for constant Hamming weight cryptanalysis. After recalling the Hamming weight leakage model, we detail a side channel attack based on constant Hamming weight cryptanalysis before concluding on the feasibility and interest of this strategy.

Leakage model A common assumption in side channel attacks [BCO04] is to consider that when processing the algorithm, the device leaks a noisy observation of the Hamming weight of the same manipulated values [MR04]. More formally it gives leakage equations on manipulated internal values:

Definition 7. *Leakage Function*
Let $x \in \mathbb{F}_2^n$ be a manipulated value, we define as leakage function:

$$\ell(x) = \mathsf{w_H}(x) + e, \text{ where } e \xleftarrow{\$} \mathcal{N}(0, \sigma).$$

For simplicity we will consider the strongest version of this model relatively to the attacker, where $\sigma = 0$ and therefore each leakage equation obtained by this attacker is true with probability 1.

Attack description

Recall that there are three prerequisites to realise a Hamming weight input cryptanalysis, supposing that the particular function f of the cipher targeted is known, the goal of the side channel phase is to collect the sets $\{\mathsf{w_H}(x), f(x)\}$ and identify the promising sets. The value of $f(x)$ (with the value of x not known) can be obtained using plaintext-ciphertext pairs, as described with a stream cipher. The information on the Hamming weight of x is given by the equation functions $\ell(x)$, enables to collect the sets $\{\mathsf{w_H}(x), f(x)\}$ and find the promising sets relatively to f fixed Hamming weight criteria.

The setting we focus on consists in a stream cipher containing one ore more registers which are updated with linear and/or non-linear functions until the production of a keystream, which is combined with the plaintext. Let f be the function producing the keystream and the targeted function of the cryptanalysis, with inputs being key or IV bits. If the attacker gets plaintext-ciphertext pairs, she can xor it to recover values $f(x)$ and then the side channel attack goal is to obtain leakage functions $\ell(x)$ for these values. We consider in this setting that the value x is loaded on a register before computation of f in the device which enables to get a leakage function of x or equivalently that we can do a measure on f input wires. During the encryption process the leakage equations are collected at each clock time, requiring the measurement to be well synchronised with the stream cipher. This consists in a DPA targeting x the input of the last function used in the production of a keystream bit. If the DPA enables to build a promising set, the constant Hamming weight cryptanalysis described previously applies.

Conclusions In this description we presented a strong side channel attacker fulfilling the prerequisites of constant Hamming weight input cryptanalysis. The model with very high number of collected leakage functions, perfect synchronisation with the cipher and perfect leakage is not realistic but it may be interesting to see the novelties of this approach. One of the first question with this model is whether constant Hamming weight cryptanalysis is the best strategy to use; the leakage equations are already revealing a lot of information ($\log n$), and extremal weight information on key registers can be sufficient to make an exhaustive search on the key bits. It can be the case, but the interest of the described cryptanalysis is to exhibit usable equations in the values. If a non-linear retroaction is applied in the cipher the leakage informations concern relations on high degree equations in the unknowns bits. On the other side the algebraic properties of f on its restricted inputs enables to directly simplify some equation or target more efficiently the sensitive information. When the attack can be considered with only one promising set, corresponding to only one fixed Hamming weight, the effort of solving one small system of equations rather than one relative to all equations is different. and this gap is increasing when some equation are noisy.

From these considerations we argue that constant Hamming weight cryptanalysis could be used for particular ciphers in a more reasonable side channel setting; for simplicity we assumed in this section a leakage with no noise; for a (standard) noisy leakage the previous attacks can be adapted, applied with ranges of Hamming weight where f criterion has the same behaviour, or using error correcting codes techniques. For example we can expect the balancedness bias to be of the same sign for ranges of consecutive weights (as shown for Grain example in Section 5.3), or the best linear approximation or low degree annihilator of f being the same for consecutive weights. In these cases the improved attacks, combining different weights, are more suitable and can be adapted relatively to the noise parameter σ. For the SCA point of view we can also consider other leakage models where constant Hamming weight input cryptanalysis could apply, nevertheless it does not seem to apply in the Hamming distance model: the study of Boolean functions with fixed input weight does not appear relevant for this other model, it could lead to a very different work.

An interest of the presented cryptanalysis with SCA is that a particular attack can be considered relatively to a particular Hamming weight, therefore a new kind of divide and conquer method can be used for side channel attacks on stream ciphers, based on a partition following the Hamming weight. This remark shows that the side channel phase can be very adaptive in this context, the attack complexity will depend on the leakage results: more measures giving a certain Hamming weight will enable to try an attack for this particular k and increase the probability of success. Therefore the attacker strategy evolves with the DPA measures and can be optimized depending more on the samples than the algorithm itself, leading to an effective adaptive attack. On the other side; the particular flaws of the studied Boolean function can lead the first phase to target only some range of Hamming weight inputs.

5.3 Proof of concept on a stream cipher: Grain

In 2008 at the eSTREAM competition, Grain [HJMM08] was proposed as a stream cipher. This cipher has been cryptanylised a lot since. In Grain 128, the output sequence is generated by a combination of 17 bits taking in the NLFSR and the LFSR components. This Boolean function that we denote h' is the target of the Hamming weight cryptanalysis. We consider that the attacker can access to the Hamming weight of x for some samples, the plaintext-ciphertext pairs enable to compute keystream bits $h'(x)$ where the 17 bit components of x depends on the 256 bits of key and IV.

We compute the Boolean criteria of the function h' relatively to constant Hamming weight cryptanalysis, comparing it to the upper bounds from previous sections. Then we explain which attack scenario can be mounted (assuming the knowledge of samples $\mathsf{w_H}(x)$) and we finally discuss on the feasibility of this approach.

With the table 5.3 of criteria of the function h' we can state on the best attack we can try, in this case it is a distinguishing attack based on the balancedness. The function h' is not WAPB, for all possible k the probability p_k is not 0.5 and quite far from this value for all weights superior than 8. In Table 6.1 we join to the probability p_k the value $C_k = (\lceil \frac{1}{2} - p_k)^{-2}\rceil$ standing for the amount of sample needed to distinguish the output from a random sequence *i.e.* the cardinality of the corresponding set to be a promising set. Recalling

k	N	ω_k	p_k	NL_k	bound NL_k	ε_k	AI_k	$boundAI_k$
0	1	0	0.000000	0	0	0.000000	0	0
1	17	8	0.470588	0	6	0.000000	1	1
2	136	76	0.558824	4	62	0.029412	1	2
3	680	341	0.501471	61	326	0.089706	2	3
4	2380	1130	0.474790	418	1166	0.175630	3	4
5	6188	3151	0.509211	1703	3054	0.275210	3	5
6	12376	6376	0.515191	4608	6132	0.372334	3	6
7	19448	9281	0.477221	8049	9654	0.413873	3	6
8	24310	12302	0.506047	10578	12077	0.435130	3	6
9	24310	12843	0.528301	10591	12077	0.435664	3	6
10	19448	8396	0.431715	8294	9654	0.426471	3	6
11	12376	7487	0.604961	4889	6132	0.395039	3	6
12	6188	2246	0.362961	2068	3054	0.334195	3	5
13	2380	1597	0.671008	569	1166	0.239076	3	4
14	680	192	0.282353	94	326	0.138235	2	3
15	136	107	0.786765	7	62	0.051471	1	2
16	17	2	0.117647	0	6	0.000000	1	1
17	1	1	1.000000	0	0	0.000000	0	0

Table 1: Constant weight input criteria of Grain stream cipher function $h'(x) = x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8x_9 + x_{10}x_{11} + x_{12}x_{13} + x_{14}x_{15} + x_8x_{12}x_{16}$. The first columns are parameters for the three criteria, k denotes the Hamming weight and $N = \binom{n}{k}$ the binomial coefficient. For the weightwise balancedness $\omega_k = |f_k(x) = 1|$ is the number of weight k input with output 1 by h', $p_k = \omega_k/N$ stands for the probability $Pr[f(x) = 1|\mathsf{w_H}(x) = k]$, which is to compare with 0.5 for a random output distribution. 4

that h' input space without restriction is $\mathbb{F}_2^n$ we also use the notation $\delta_k = \lceil C_k * \frac{2^n}{N} \rceil$ to link the amount of data needed with weight k to the all input space.

k	N	ω_k	p_k	C_k	δ_k
0	1	0	0	4	524288
1	17	8	0.470588235294118	1156	8912896
2	136	76	0.558823529411765	289	278528
3	680	341	0.501470588235294	462401	89129153
4	2380	1130	0.474789915966387	1574	86684
5	6188	3151	0.509211376858436	11786	249647
6	12376	6376	0.515190691661280	4334	45901
7	19448	9281	0.477221308103661	1928	12994
8	24310	12302	0.506046894282188	27349	147458
9	24310	12843	0.528301110654052	1249	6735
10	19448	8396	0.431715343480049	215	1450
11	12376	7487	0.604961215255333	91	964
12	6188	2246	0.362960568842922	54	1144
13	2380	1597	0.671008403361344	35	1928
14	680	192	0.282352941176471	22	4241
15	136	107	0.786764705882353	13	12529
16	17	2	0.117647058823529	7	53979
17	1	1	1.00000000000000	4	524289

Table 2: Balancedness and distinguisher parameters for constant input cryptanalysis on Grain filtering function. k denotes the Hamming weight considered, N the binomial coefficient $\binom{n}{k}$, ω_k and p_k are criteria of h' balancedness on $E_{17,k}$, C_k and δ_k are parameters for the data complexity of the distinguishing attack.

Reading the table 5.3 we know which Hamming weight k can be used to a practical attack, having a look on the C_k column. The weights for which the quantity of data needed is greater than the potential number of samples cannot be used as for the weights $0 \leq k \leq 3$ where $C_k > N$. Nevertheless for some weights the quantity C_k is very small compared to N, as for $k = 11$ or $k = 12$ where an attacker obtaining a set as small as 1% of N gets a promising set. The column δ_k shows that with for example 2000 samples an attacker can expect to distinguish the keystream from uniform as with high probability one of the sets for $k \in [10, 11, 12, 13]$ will be a promising set. Noting that the small values of C_k are for the weights $k > 8$ and that the corresponding biases $\varepsilon_k = \frac{1}{2} - p_k$ are positive for odd values and negatives for even values, we can optimize the attack strategy by grouping weights. The parameters for this strategy are summarized in table 5.3, it leads to better biases and consequently less data complexity for the potential attack.

Based on table 5.3 we focus on finding a distinguisher rather than recover the state with an algebraic or correlation like attack for various reasons. First if we consider the different AI_k, for all medium k we get

K	N	ω_K	p_K	C_K	δ_K
$9, 11, 13, 15, 17$	39203	22035	0.562074331046093	260	870
$10, 12, 14, 16$	26333	10836	0.411498879732655	128	638

Table 3: Balancedness and distinguisher parameters for constant input cryptanalysis on Grain filtering function, grouping weights. K denotes the Hamming weights considered, N the sum of binomial coefficient $\sum_{k\in K}\binom{n}{k}$, ω_K, p_K ,C_K and δ_K are the obtain with the corresponding parameters for the various $k \in K$ considered.

$AI_k = 3$, as h' is a Boolean function of degree 3, for these weight we are not getting an important advantage over standard algebraic attack over $\mathbb{F}_2^n$. For the criterion of nonlinearity, there are no medium weight with critical NL_k leading to very practical attack. The most important argument in disfavour of these attacks is the update function of this stream cipher, containing a NLFSR. Due to the NLFSR the inputs bits of h' can be expressed as non linear equation of the key and IV bits, where the degree of the equations increase quickly to the number of unknown bits. Therefore the attack complexities relatively to the algebraic immunity and nonlinearity criteria will no more depend on n but on n' linearised variables, increasing drastically the attack complexities. Another strategy to circumvent the impact of the NLFSR is to study the behaviour of it's feedback relation g' on constant weight inputs. However our study on this function does not reveal exploitable flaws enabling to keep linear the equations of h' inputs.

Moreover, it is important to notice that we assume here that we have access to the weight of the input of the function h' but not the weight of the two registers, which is here a strong hypothesis. The complexity of the algorithms to compute AI_k and NL_k are not suitable for 128 or 256 variables if the attacker consider the entry of the function h with the all state. Nevertheless it is possible to compute the balancedness bias of h on fixed Hamming weight on the all 256 bit state. These computations give that for all weight k ($k \in [0, 256]$), the bias $\mid 0.5 - p_k \mid$ is never less that 2^{-33}, which is extremely low for medium weights. The data complexity parameter C_k which scales the number of samples needed at weight k by the proportion over all possible 2^n samples fall down to 2^{60} for various weights. The sign of $(0.5 - p_k)$ is the same for relatively large set of consecutive weights, there are only 12 sign changes and the longest set of consecutive weights with the same sign is of cardinality 93.

In conclusion the constant Hamming weight input cryptanalysis on Grain could lead to a distinguisher (and potentially bit prediction) with less than 1000 samples with exact Hamming weight samples of h' inputs or 2^{60} samples for h inputs. The feasibility of a concrete attack resides now on the complexity of getting theses samples. As the keystream is produced after an initialization step, it seems difficult to know the weight of x only choosing the IV and making guesses on the key bits. On the opposite a side channel attack appears to be a good candidate for obtaining this information on x, but this corresponds to a strong model of attacker. For the attack on h', it consists in assuming that the attacker obtains leakages on a register where x is loaded from the LFSR-NFSR component before h' computation (or wires where these inputs transit). For the attack on h we assume that the attacker gets leakage from the entire 256 bits state. The attacker is able to identify when the keystream function is computed during the keystream generation process (synchronization effort) and to obtain leakage equations without noise. The last assumption is not realistic, but some relaxations could still be compatible with the constant Hamming weight input study on Grain. The results on h' show that if the parity of the weight is determined it can apply, nevertheless it seems to be even more inadequate with the Hamming leakage model. On the other side, the results on h show that an approximation of the weight can be enough, as the sign of the bias is constant on big ranges of consecutive Hamming weights. In this context the leakage equations on the entire state of Grain obtained by the side channel attacker can be used when the leakage measure induce high confidence in the Hamming weight of x to create the sets enabling to distinguish the output of h from a random sequence.

6 Impact of Boolean functions with restricted input 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 4 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 $\binom{n}{n/2} \geq 2^\lambda$). As the Hamming weight of the input of the filtering function f is constant and known, a constant Hamming weight input cryptanalysis is directly applicable. Therefore we claim that the security analysis of FLIP([MJSC16],section 3.4) based on the criteria of Boolean functions over all $\mathbb{F}_2^n$ inputs is not done in the most adequate model. As the stream cipher function is only evaluated on entries from $E_{n,\frac{n}{2}}$, it's security should be studied in the constant Hamming weight model.

The purpose of this section is then to stand on the fixed input weight criteria of the proposed filtering functions and the potential flaws revealed by Hamming weight input cryptanalysis before concluding on the security of this stream cipher in the Hamming weight model. In an article published at CRYPTO 2016 [DLR16], Duval et al. gave an attack on preliminary instances of FLIP, leading the authors of FLIP to add more triangular functions in the filtering function. We then also study the case of Guess and Determine like attacks combined with algebraic like attacks and correlation like attacks when the Hamming weight is fixed.

Definition 8 (Direct Sum). *Let f be the direct sum of two Boolean functions g anf h:*

$$\forall x \in \mathbb{F}_2^r, \forall y \in \mathbb{F}_2^s, f(x,y) = g(x) + h(y),$$

where $n = r + s$.

Direct sum and balancedness We first show that knowing the weights of g an h on all constant weight inputs enables to determine exactly the weight of f on all E_k with $0 \leq k \leq r + s$.

Proposition 11. *Recall that we denote by $\mathsf{w_H}(f)_k$ the Hamming weight of the evaluation vector of function f on all the entries of fixed Hamming weight k: $\mathsf{w_H}(f)_k = |\{x \in \mathbb{F}_2^n, \mathsf{w_H}(x) = k, f(x) = 1\}|$ .Let f be the direct sum of h and g, then we have the following relation :*

$$\mathsf{w_H}(f)_k = \sum_{i=0}^{k} \mathsf{w_H}(g)_i \left(\binom{s}{k-i} - \mathsf{w_H}(h)_{k-i} \right) + \left(\binom{r}{i} - \mathsf{w_H}(g)_i \right) \mathsf{w_H}(h)_{k-i}$$

Direct sum and NL_k Let n be any positive integer and $k \in \{1, \ldots, n\}$. We recall that the nonlinearity $NL_{E_{n,k}}(f)$ 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 $E_{n,k}$. For readability we simply denote $NL_{E_{n,k}}(f)$ by $NL_k(f)$.

Lemma 2 (Direct sum and NL_k).

Let f be the direct sum of g and h, we have:

$$NL_k(f) \geq \sum_{i=0}^{k} \binom{r}{i} NL_{k-i}(h) + \sum_{i=0}^{k} NL_i(g) \left(\binom{s}{k-i} - 2NL_{k-i}(h) \right)$$

Proof. We have:

$$\begin{aligned}
NL_k(f) &= \frac{\binom{n}{k}}{2} - \frac{1}{2} \max_{(a,b)\in\mathbb{F}_2^r\times\mathbb{F}_2^s} \Big| \sum_{(x,y)\in E_{n,k}} (-1)^{f(x,y)+a\cdot x+b\cdot y} \Big| \\
&\geq \frac{\binom{n}{k}}{2} - \frac{1}{2} \max_{a\in\mathbb{F}_2^r, b\in\mathbb{F}_2^s} \sum_{i=0}^{k} \left| \sum_{x\in E_{r,i}, y\in E_{s,k-i}} (-1)^{g(x)+h(y)+a\cdot x+b\cdot y} \right| \\
&= \frac{\binom{n}{k}}{2} - \frac{1}{2} \max_{a\in\mathbb{F}_2^r, b\in\mathbb{F}_2^s} \sum_{i=0}^{k} \left| \sum_{x\in E_{r,i}} (-1)^{g(x)+a\cdot x} \right| \left| \sum_{y\in E_{s,k-i}} (-1)^{h(y)+b\cdot y} \right| \\
&\geq \frac{\binom{n}{k}}{2} - \frac{1}{2} \sum_{i=0}^{k} \left(\max_{a\in\mathbb{F}_2^r} \left| \sum_{x\in E_{r,i}} (-1)^{g(x)+a\cdot x} \right| \right) \left(\max_{b\in\mathbb{F}_2^s} \left| \sum_{y\in E_{s,k-i}} (-1)^{h(y)+b\cdot y} \right| \right) \\
&= \frac{\binom{n}{k}}{2} - \frac{1}{2} \sum_{i=0}^{k} \left(\binom{r}{i} - 2NL_i(g) \right) \left(\binom{s}{k-i} - 2NL_{k-i}(h) \right) \\
&= \sum_{i=0}^{k} \binom{r}{i} NL_{k-i}(h) + \sum_{i=0}^{k} \binom{s}{k-i} NL_i(g) - 2\sum_{i=0}^{k} NL_i(g) NL_{k-i}(h) \\
&= \sum_{i=0}^{k} \binom{r}{i} NL_{k-i}(h) + \sum_{i=0}^{k} NL_i(g) \left(\binom{s}{k-i} - 2NL_{k-i}(h) \right).
\end{aligned}$$

□

Direct sum and AI_k 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 3 (Direct sum and AI_k). *Let f be the direct sum of g and h with r and s variables respectively. Then for all $k \leq \min(r,s)$, $\mathsf{AI}_k(f)$ follows the bound :*

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

.

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 g_ℓ or $1+g_\ell$ and $h_{k-\ell}$ or $1+h_{k-\ell}$ for some ℓ
$\forall X \in E_k$, $A(X)f(X) = 0$. Moreover, A is non-null on E_k, so, there exists $\widetilde{X} \in E_k$ 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 $\mathsf{w_H}(\widetilde{x})$, then the weight of $\widetilde{y} \in \mathbb{F}_2^m$ is $k-\ell$. Finally for this ℓ, we fix for $X \in E_k$ the x part to the value $\widetilde{x}$ and we consider all possible values for $y \in \mathbb{F}_2^s$ of Hamming weight $k-\ell$. By doing so, it appears that A_k is an annihilator of $h_{k-\ell}$ or $1+h_{k-\ell}$, 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 g_ℓ or $1+g_\ell$. Therefore $deg(A) \geq deg(A_k) \geq \max[\mathsf{AI}_\ell(g), \mathsf{AI}_{k-\ell}(h)]$. Recalling that $\ell = \mathsf{w_H}(\widetilde{x})$ and then $0 \leq \ell \leq k$ finishes the proof. □

6.1 Fixed Hamming weight input cryptanalysis on FLIP filtering function

FLIP instances We recall the 4 filtering functions proposed in FLIP in table 6.1, 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 criteria on fixed Hamming weight inputs.

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

Table 4: 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.

Balancedness of FLIP 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 proposition 11, we can exactly compute the values $|\{f(x) = 1|\mathsf{w_H}(x) = k\}|$ for all k for the four filtering functions and the bias from a random sequence. In table 6.1 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 cancelling (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 6.1, 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. Even considering a combination with guess and determine attack, the biases on the simpler functions obtained are insufficient to mount a concrete attack.

NL_k of FLIP 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 NL_k of a direct sum

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 5: 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

we can give a lower bound of the nonlinearity on constant weight inputs for the 4 instances. To do so we recursively compute the NL_k for a direct sum with lower bounds for $\mathsf{NL}_i(g)$ and exact value of $\mathsf{NL}_{k-i}(h)$. In table 6.1 we summarize the results, the exact NL_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 NL_k of a function in d variables consisting in only one monomial is null for all k with $0 < k < d$. The results are obtained by first combining the quadratic functions, then the triangular part and finally the linear (and higher than 9 degree) part.

Instance	N	v_0	ℓ	N_1	v_1
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	$[363, 1321]$

Table 6: Lower bound on NL_k of FLIP instances, N refers to the number of variables, v_0 the range of weight k for which $\frac{\mathsf{NL}_k(f)}{\binom{n}{k}} \geq 0.499$. ℓ refers to the number of cancelled monomials of the quadratic part by the guess strategy, N_1 and v_1 are the corresponding number of variables and range of weights.

The results in table 6.1 show that for $k = \frac{n}{2}$, $NL_k(f)$ 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 NL_k of the various functions and the number of variables are too big to lead to a concrete cryptanalysis with our attack scenario based on NL_k.

AI_k of FLIP 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. However, we have shown with theorem 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. However, theorem 3 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:

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

where f has r variables and a high algebraic immunity and g has s variables with the smallest degree as possible. The inputs in FLIP are the words of Hamming weight $\frac{n}{2}$ where $n = r+s$. Then to apply theorem 3, r and s have to satisfy the condition $r \leq \frac{n}{2} \leq s$. Then for each instances of FLIP, we take f the Boolean function taking all monomials with the highest degree. In each f chosen there is more monomials in direct sum than the degree of the f, so the algebraic immunity (over $\mathbb{F}_2^r$) of f is its degree. 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 6.1.

Instance	N	$\mathsf{AI}(f))$	$\deg(g)$	r	$Bound$	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	$2^{70.6}$	$2^{198.2}$

Table 7: 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, r is the number of variables in f, and the lower bound is given with theorem 3. 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.

It is important to notice that the lower bounds here could be not tight and the algebraic immunity of FLIP instances could be as great as $\mathsf{AI}(F)$. Then it remains to prove the exact algebraic immunity of FLIP instances. Moreover, it remains not clear what could happen by considering all possible guesses in a Guess and Determine attack to the algebraic immunity regarding the large number of possible guesses that an attacker could make. Moreover, the high number of triangular functions used in FLIP where 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 is few triangular function, then the Guess and Determine attack consisting in cancelling monomials with high degree is more efficient, but if there is too much triangular functions, then the lower bound on the algebraic immunity of FLIP is decreasing.

6.2 Conclusions and cautionary note

Finally 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. The security analysis is made in the right model, however, we get only lower bounds on the different criteria. We do not know which is the exact algebraic immunity of FLIP for the four instances. If the bounds we proved are tight, then maybe a Guess and Determine attack on FLIP could be efficient in that way. Nevertheless, we still do not have the exact value of nonlinearity and algebraic immunity of FLIP, but our first 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 input is then $\mathbb{F}_2^n$, but the whitening still enables to know the parity of f input.

6.3 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 2.1 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 NL_k bound. In Section 3 we proved the upper bound:

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

As this bound is unreacheable 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 4 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 $n/2$). The most commun 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 4, we linked the algebraic immunity upper bound to the rank of the generator matrix of a punctured Reed-Müller code $RM(d,n)_k$. As shown in Section 5 in the secund attack scenario, this matrix can be used to compute the exact AI_k of a given function, by partitionning the columns depending on the value of f on the column entry. 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.

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.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In Marc Joye and Jean-Jacques Quisquater, editors, *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[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.

[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.

[HJMM08] Martin Hell, Thomas Johansson, Alexander Maximov, and Willi Meier. The grain family of stream ciphers. In Matthew J. B. Robshaw and Olivier Billet, editors, *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 179–190. Springer, 2008.

[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.

[MR04] Silvio Micali and Leonid Reyzin. Physically Observable Cryptography (Extended Abstract). In Moni Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 278–296. 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.