A wide class of Boolean functions generalizing the hidden weight bit function*

Claude Carlet,
University of Bergen, Norway

E-mail: claude.carlet@gmail.com

Abstract

Designing Boolean functions with fast output to compute and meeting all the criteria necessary for allowing the stream ciphers in which they are used as nonlinear components to resist all major attacks has been an open problem since the beginning of this century, when the algebraic attacks were invented (in 2003). Functions allowing good resistance are known since 2008, but their output is too slow and complex to compute. Functions with a fast and simple to compute output are known, such as majority functions and the so-called hidden weight bit (HWB) functions, but they all have a cryptographic weakness: their too small nonlinearity. In the present paper, we introduce a generalization of the HWB function into a construction of n-variable balanced functions f from $(n-1)$-variable Boolean functions g having some property held by a large number of functions. Function f is defined by its support, equal to the image set of a vectorial function depending on g. This makes the function complex enough for allowing good cryptographic parameters, while its output is light to compute. The HWB function is what we obtain with f when the initial function g equals 1. Other well chosen functions g provide functions f having good cryptographic parameters.

We analyze the constructed functions f, we provide a fast way to compute their output, we determine their algebraic normal forms and we show that, most often, their algebraic degree is optimal. We study their Walsh transform and their nonlinearity and algebraic immunity. We observe with computer investigations that this generalization of the HWB function allows to keep its quality of being fast to compute and having good enough algebraic immunity, while significantly improving its nonlinearity. The functions already obtained in the investigations provide a quite good (and never reached before) trade-off between speed and security. Further (probably difficult) work should allow obtaining, among such generalized HWB functions whose number is huge, still better filter functions to be used in stream ciphers.

Keywords: Boolean function, code, cryptography.

*The research of the author is partly supported by the Trond Mohn Foundation.

1 Introduction

Boolean functions can be used for designing error correcting codes [14]; the important parameter in such coding theoretic case is their nonlinearity. They can also be used in stream ciphers as feedback functions in nonlinear feedback shift registers (NFSR), or as filter functions in the filter model of pseudo-random generator (see e.g. [3]) or as filter functions in the recent filter permutator [13, 12]. In all these cryptographic cases, they must be chosen with care and need to satisfy at the best possible levels some security criteria, one of which is also the nonlinearity. It is a huge problem to determine the proper criteria in the case of NFSR, while the criteria are well known in the two latter use cases, and are quantified by proper parameters: balance (and when guess and determine attacks are possible, resiliency), algebraic degree, nonlinearity, algebraic immunity and fast algebraic immunity (and, when guess and determine attacks are possible, the same parameters on the so-called descendant functions, obtained by fixing some coordinates). Some properties are specific to the filter permutator (see [7, 9]). Most functions known for having a good algebraic immunity (one of the most important criteria in cryptography) have a weak nonlinearity and cannot then be used in stream ciphers. This is for instance the case of the majority function.
The known functions that satisfy all security criteria are few, see [5, 3]. Moreover, they are too complex for allowing the stream ciphers using them as nonlinear components to be fast enough (a stream cipher is supposed to be faster than the AES in counter mode and this sets the bar high) and to be lightweight enough (a stream cipher is supposed to need lighter computational means than block ciphers). Designing such Boolean functions with fast output to compute can be viewed as "the big single-output Boolean problem" (we refer here to "the big APN problem", whose expression has been introduced by Dillon, and which designates the most important problem for vectorial functions to be used as substitution boxes in block ciphers). Solving this problem needs to find constructions providing enough complexity for potentially reaching good cryptographic parameters with the constructed functions, but also allowing an output simple and fast to compute, which is often contradictory in practice. We need also to have a rather large class of functions within which diverse kinds of properties and trade-offs can be favored. Before the invention of algebraic attacks, the large class of Maiorana-McFarland functions [2] allowed a good trade-off between simplicity and security, but these functions do not behave quite well with respect to algebraic attacks. We need then a large enough new class of functions offering such good trade-off. A Boolean function proposed by R. E. Bryant [1], mentioned by R. Knuth in Vol. 4 of "The Art of Computer Programming", and called the hidden weight bit function (in brief, HWB function), vanishes at 0 and takes at every nonzero input $x \in \mathbb{F}_2^n$ the value x_i where i is the Hamming weight of x. This function shares with the majority function a nice property: while it has good algebraic immunity (less good than the majority function, though, see below), its output is considerably faster to compute than those of the other currently known functions having good algebraic immunity. The cryptographic

properties of the HWB function have been studied in [15] (we shall recall them and see that, as the majority function, it has a very weak point: a low nonlinearity, which seems related to the simplicity of the function). In [4], we have studied an apparently rather simple general construction using that the support of any n-variable balanced Boolean function f can be obtained as the image set of an injective function F from $\mathbb{F}_2^{n-1}$ to $\mathbb{F}_2^n$. We call such F a *parameterization* of f. In the present paper, we use the parameterization technique to design a construction of n-variable balanced functions from $(n-1)$-variable Boolean functions satisfying some condition that we shall explain (and that is satisfied by many functions, roughly $2^{3\cdot 2^{n-3}}$ of them, among which are all monotone Boolean functions). The hidden weight bit function is then what we get with the construction when the chosen initial function is the simplest nonzero function: constant function 1. We observe with computer investigations that the algebraic degree and the algebraic immunity of the constructed functions can be as good as (and sometimes, for the algebraic immunity, better than) those of the HWB function in the same number of variables, and the nonlinearity is strictly better, while keeping the property of the hidden weight bit function to be fast to compute.

The paper is organized as follows. After preliminaries, we recall from [4] in Section 3 the parameterization of balanced Boolean functions. In Section 4, we deduce a generalization of the HWB function and we study the constructed functions, that we call GHWB; we determine the condition under which they are balanced. We show that despite their rather simple definition, their structure is complex. In Section 5, we provide an expression for the output of GHWB functions, which allows to compute it in a fast way. These functions combine then the merits of being fast to compute and complex enough for potentially allowing good cryptographic parameters. In Section 6, we transform this expression into an algebraic normal form, we study the algebraic degree and show that it is most often optimal. In Section 7, we study the Walsh transform, nonlinearity and resiliency of GHWB functions and we provide an upper bound on the nonlinearity, which gives some clue on conditions for a good nonlinearity. We show thanks to computer investigations that GHWB functions can reach good nonlinearity. In Section 8, we study their algebraic immunity (AI) and show by computer investigations that their AI can be good as well. We end with a conclusion in which we draw quite positive perspectives.

2 Preliminaries

In this paper, we shall denote the same way, by $+$, additions in $\mathbb{F}_2$, in $\mathbb{F}_2^n$, in $\mathbb{F}_{2^n}$, and in $\mathbb{R}$, since there will be no ambiguity. We shall denote by 0 the zero vector in any of the vector spaces over $\mathbb{F}_2$ and when needing to specify, we shall denote by 0_n the zero vector of length n. We shall also denote by 1_n the all-1 vector of length n and by $w_H(x)$ the Hamming weight of a binary vector x. We call n-variable Boolean function every function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ and we denote by $\mathcal{B}_n$ the vector space of all n-variable Boolean functions. The support

of a Boolean function f is the set $supp(f) = \{x \in \mathbb{F}_2^n;\ f(x) = 1\}$, while the support of a vector $x \in \mathbb{F}_2^n$ equals $\{i \in \{1, \ldots, n\}; x_i = 1\}$. We call co-supports the complements of the supports. The Hamming weight $w_H(f)$ of a Boolean function f (or of a vector) equals the size of its support. An n-variable Boolean function is called balanced if it has Hamming weight 2^{n-1}. The Hamming distance between two Boolean functions f, g is $d_H(f, g) = w_H(f + g)$. The functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ are called (n, m)-*functions*. Such function F being given, the n-*variable* Boolean functions $f_1, \ldots, f_m$, defined at every $x \in \mathbb{F}_2^n$ by $F(x) = (f_1(x), \ldots, f_m(x))$, are called the *coordinate functions* of F. When the numbers m and n are not specified, (n, m)-functions are called *vectorial Boolean functions* or simply *vectorial functions*. We refer to [3] for a complete state of the art.

Two vectorial functions F and G are called affine equivalent (resp. linearly equivalent) if there exist two affine (resp. linear) permutations L over $\mathbb{F}_2^m$ and L' over $\mathbb{F}_2^n$ such that $G = L \circ F \circ L'$.

Among the classical representations of Boolean functions and of vectorial functions are the *truth-table* in the case of Boolean functions and the *look-up table* (LUT) in the case of vectorial functions. Both are the table of all pairs of an element of $\mathbb{F}_2^n$ (an ordering of $\mathbb{F}_2^n$ being fixed) and of the value of the function at this input. The *algebraic normal form* (in brief the *ANF*), which contains a little more information directly usable on the cryptographic strengths of functions, is the unique n-variable multivariate polynomial representation of the form

$$F(x) = \sum_{I \subseteq \{1,\ldots,n\}} a_I \left(\prod_{i \in I} x_i \right) = \sum_{I \subseteq \{1,\ldots,n\}} a_I\, x^I, \tag{1}$$

where a_I belongs to $\mathbb{F}_2$ in the case of Boolean functions and to $\mathbb{F}_2^m$ in the case of (n, m)-functions (and where "x^I" is a notation that we shall use all along this paper). Note that we can deduce the ANF of the i-th coordinate function of F by replacing in (1) each coefficient $a_I \in \mathbb{F}_2^m$ by its i-th coordinate.
The degree of the ANF shall be denoted by $d_{alg}(f)$ (resp. $d_{alg}(F)$); it is called the *algebraic degree* of the function and equals $\max\{|I|;\ a_I \neq 0\}$, where $|I|$ denotes the size of I (with the convention that the zero function has algebraic degree 0). This makes sense thanks to the existence and uniqueness of the ANF. Note that the algebraic degree of an (n, m)-function F equals the maximal algebraic degree of its coordinate functions. It also equals the maximal algebraic degree of its *component functions*, that is, of the nonzero linear combinations over $\mathbb{F}_2$ of the coordinate functions, *i.e.* the functions of the form $v \cdot F$, where $v \in \mathbb{F}_2^m \setminus \{0\}$ and "$\cdot$" is an inner product in $\mathbb{F}_2^m$. It is an affine invariant (that is, its value does not change when we compose F, on the right or on the left, by an affine automorphism). We have:

$$\forall x \in \mathbb{F}_2^n,\ f(x) = \sum_{I \subseteq supp(x)} a_I, \tag{2}$$

which is valid for Boolean and vectorial functions, and where $supp(x)$ denotes the support of x.

The converse is also true: for all $I \subseteq \{1, \ldots, n\}$, we have:

$$\forall I \subseteq \{1, \ldots, n\},\ a_I = \sum_{x \in \mathbb{F}_2^n;\ supp(x) \subseteq I} f(x), \tag{3}$$

for f Boolean or vectorial. According to Relation (3), we have the well known property (see [11, 3]):

Proposition 1 *An n-variable Boolean function f satisfies $d_{alg}(f) = n$ if and only if $w_H(f)$ is odd. More generally, an (n, m)-function F satisfies $d_{alg}(F) = n$ if and only if $\sum_{x \in \mathbb{F}_2^n} F(x) \neq 0_m$.*

The affine (Boolean or vectorial) functions are the functions of algebraic degree at most 1.

The *Fourier-Hadamard transform* of any pseudo-Boolean function φ (i.e. any function from $\mathbb{F}_2^n$ to $\mathbb{R}$) is the $\mathbb{R}$-linear mapping which maps φ to the function $\widehat{\varphi}$ defined on $\mathbb{F}_2^n$ by

$$\widehat{\varphi}(u) = \sum_{x \in \mathbb{F}_2^n} \varphi(x)\,(-1)^{u \cdot x}, \quad u \in \mathbb{F}_2^n, \tag{4}$$

where "$\cdot$" is some chosen inner product in $\mathbb{F}_2^n$. It satisfies the so-called *inverse Fourier-Hadamard transform formula*: for all $a \in \mathbb{F}_2^n$, we have:

$$\sum_{u \in \mathbb{F}_2^n} \widehat{\varphi}(u)\,(-1)^{u \cdot a} = 2^n \varphi(a),$$

which proves that the Fourier-Hadamard transform is a bijection.

Given an n-variable Boolean function f (we shall address vectorial functions below), we have two associated transforms: the Fourier-Hadamard transform of f, where f is then viewed as a function from $\mathbb{F}_2^n$ to $\{0, 1\} \subset \mathbb{Z}$, and the *Walsh transform* of f which is the Fourier-Hadamard transform of the sign function $(-1)^f$:

$$W_f(u) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) + u \cdot x}.$$

We have:

$$W_f = 2^n\,\delta_0 - 2\widehat{f}, \tag{5}$$

where δ_0 denotes the Dirac (or Kronecker) symbol over $\mathbb{F}_2^n$, whose value is nonzero only at 0_n and whose ANF is:

$$\delta_0(x) = \prod_{i=1}^{n} (x_i + 1) = \sum_{I \subseteq \{1, \ldots n\}} x^I. \tag{6}$$

The *nonlinearity* of a Boolean function f is the minimum Hamming distance between f and affine Boolean functions. We shall denote it by $nl(f)$. We have:

$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{a \in \mathbb{F}_2^n} |W_f(a)|. \tag{7}$$

The so-called the covering radius bound states:

$$nl(f) \leq 2^{n-1} - 2^{\frac{n}{2}-1}. \tag{8}$$

A function is called bent if this inequality is an equality.

Let f be any n-variable Boolean function. An n-variable Boolean function g such that $fg = 0$ is called an *annihilator* of f.
The minimum algebraic degree of nonzero annihilators of f or $f+1$ is called the *algebraic immunity* (in brief, AI) of f and is denoted by $AI(f)$. It also equals the minimal value d such that there exist $g \neq 0$ and h, both of algebraic degree at most d, such that $fg = h$. We have $AI(f) \leq \max(d_{alg}(f), \lceil \frac{n}{2} \rceil)$. We say that f has optimal algebraic immunity if $AI(f) = \lceil \frac{n}{2} \rceil$. For n odd, the functions with optimal algebraic immunity are necessarily balanced.

We address now vectorial functions. We call *Walsh transform* of an (n, m)-*function* F, and we denote by W_F, the function which maps any ordered pair $(u, v) \in \mathbb{F}_2^n \times \mathbb{F}_2^m$ to the value at u of the Walsh transform of the Boolean function $v \cdot F$ (by abuse of notation, we denote by the same way the inner products in $\mathbb{F}_2^n$ and $\mathbb{F}_2^m$):

$$W_F(u, v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{v \cdot F(x) + u \cdot x};\ u \in \mathbb{F}_2^n, v \in \mathbb{F}_2^m.$$

In other words, the Walsh transform of F equals the Fourier-Hadamard transform of the indicator (i.e. characteristic function) of its graph $\{(x, F(x)),\ x \in \mathbb{F}_2^n\}$.

The *nonlinearity* of an (n, m)-function is the minimum nonlinearity of its component functions $v \cdot F$, $v \neq 0$:

$$nl(F) = 2^{n-1} - \frac{1}{2} \max_{\substack{v \in \mathbb{F}_2^m \setminus \{0_m\} \\ u \in \mathbb{F}_2^n}} |W_F(u, v)|. \tag{9}$$

3 The parameterization of balanced Boolean functions

In [4] is introduced the following notion:

Definition 1 *Let F be an injective $(n-1, n)$-function. We call* Boolean function parameterized by F *and denote by f_F the n-variable balanced function of support $Im(F) = \{F(z), z \in \mathbb{F}_2^{n-1}\}$. Then z is called a* parameter *of the Boolean function and F a* parameterization *of f_F.*

It is observed that changing F into an affine equivalent function $L \circ F \circ L'$, where L is an affine automorphism of $\mathbb{F}_2^n$ and L' an affine automorphism of $\mathbb{F}_2^k$, transforms f_F into an affine equivalent Boolean function, since the composition by L' does not change f_F (nor does the composition by a nonlinear permutation) and:

$$f_{L \circ F} = f_F \circ L^{-1}.$$

The cryptographic parameters of f_F are studied in [4] in terms of corresponding parameters of F. We shall recall below the main results concerning these parameters, as we study those of the new functions that we are going to introduce. Five classes of Boolean functions (three known and two new) are studied in [4]. In the present paper, we introduce and study one more class, which provides an excellent trade-off between the cryptographic security of stream ciphers using it as a filter function and their speed.

4 Generalization of the hidden weight bit function

It is easily seen that the HWB function (defined in Introduction) admits as a parameterization the $(n-1,n)$-function F that maps every $z \in \mathbb{F}_2^{n-1}$ to the vector $x = F(z)$ coinciding with z on the $w_H(z)$ first positions, having the coordinate of index $w_H(z)+1$ equal to 1, and having each coordinate of index $i > w_H(z)+1$ equal to z_{i-1}. Indeed:
- for every z, we have $w_H(F(z)) = w_H(z)+1$;
- function F is then injective, since its image set is included in the support of the HWB function, and any x in this set has exactly one pre-image, obtained from x by erasing its coordinate of index $w_H(x)$;
- the image set of F equals then the whole support of the HWB function.

4.1 Principle of the generalization

Keeping the idea of the insertion of a coordinate in the input vector $z \in \mathbb{F}_2^{n-1}$ at the position of index $w_H(z)+1$ (which always correctly belongs to $\{1,\ldots,n\}$), let g be an $(n-1)$-variable Boolean function, and for every $z \in \mathbb{F}_2^{n-1}$, let $F_g(z)$ be the vector $x = (x_1,\ldots,x_n) \in \mathbb{F}_2^n$ such that $x_i = z_i$ for every $i \leq w_H(z)$, $x_{w_H(z)+1} = g(z)$ and $x_i = z_{i-1}$ for every $i \geq w_H(z)+2$:

$$F_g(z) = \big(z_1,\ldots,z_{w_H(z)}, g(z), z_{w_H(z)+1},\ldots,z_{n-1}\big).$$

Function F_g is not always injective, but its restrictions to the support of g and to its co-support are injective. Indeed, if $x = F_g(z)$ and $z \in supp(g)$, then $w_H(x) = w_H(z)+1$ and $x_{w_H(x)} = 1$; and any $x \in \mathbb{F}_2^n$ such that $x_{w_H(x)} = 1$, has at most one pre-image z in the support of g, since any such z has Hamming weight $x_H(x)-1$, and this z is then obtained from x by erasing its coordinate of index $w_H(x)$; and if $z \notin supp(g)$, then $x_{w_H(x)+1} = 0$ and any $x \in \mathbb{F}_2^n$ such that $x_{w_H(x)+1} = 0$ has at most one pre-image in the co-support of g, since any such z has Hamming weight $x_H(x)$, and this z is then obtained from x by erasing its coordinate of index $w_H(x)+1$.

4.2 Condition on the initial function g

Let us search a necessary and sufficient condition on g under which F_g is injective over $\mathbb{F}_2^{n-1}$. According to our observation above, the condition is that the images

by F_g of the support of g and of its co-support are disjoint. Denoting by x' the vector obtained from any $x \in \mathbb{F}_2^n$ by erasing its coordinate of index $w_H(x)+1$ and by x'' the vector obtained from x by erasing its coordinate of index $w_H(x)$, the condition for the injectivity of F_g writes: there does not exist $x \in \mathbb{F}_2^n$ such that $x_{w_H(x)} = 1 = g(x'')$ and $x_{w_H(x)+1} = 0 = g(x')$. We shall denote by u the vector x deprived of its coordinates of indices $w_H(x)$ and $w_H(x)+1$. Given x such that $x_{w_H(x)} = 1 = g(x'')$ and $x_{w_H(x)+1} = 0 = g(x')$, u equals x' deprived of its coordinate of index $w_H(x')$ and x'' deprived of its coordinate of index $w_H(x'')+1$. Note that we have $w_H(u) = w_H(x) - 1$, and denoting for every $j \in \{0,1\}$ by $u^{(j)}$ the vector obtained from u by inserting a coordinate of value j at position $w_H(u)+1$, we have $u^{(0)} = x''$ and $u^{(1)} = x'$. Hence, a sufficient condition for F_g to be injective writes:

$$\nexists u \in \mathbb{F}_2^{n-2};\ g(u^{(0)}) = 1 \text{ and } g(u^{(1)}) = 0.$$

This condition is also necessary, since if any $u \in \mathbb{F}_2^{n-2}$ exists such that $g(u^{(0)}) = 1$ and $g(u^{(1)}) = 0$, then $F_g(u^{(0)})$ is obtained from $u^{(0)}$ by inserting 1 in $u^{(0)}$ at position $w_H(u)+1$ (the 0 and all the subsequent coordinates being moved on the right) and $F_g(u^{(1)})$ is obtained from $u^{(1)}$ by inserting 0 at position $w_H(u)+2$, that is, just after the 1 (all the subsequent coordinates being moved on the right), and we have then $F_g(u^{(0)}) = F_g(u^{(1)})$.

Proposition 2 *Let $n \geq 2$ and let g be any $(n-1)$-variable Boolean function. For every $z \in \mathbb{F}_2^{n-1}$, let $F_g(z)$ be the vector $(x_1, \ldots, x_n) \in \mathbb{F}_2^n$ such that $x_i = z_i$ for every $i \leq w_H(z)$, $x_{w_H(z)+1} = g(z)$ and $x_i = z_{i-1}$ for every $i \geq w_H(z)+2$. Then the $(n-1,n)$-function F_g is injective if and only if, for every $u \in \mathbb{F}_2^{n-2}$, denoting for every $j \in \{0,1\}$ by $u^{(j)}$ the vector obtained from u by inserting a coordinate of value j at position $w_H(u)+1$, we have:*

$$g(u^{(0)}) \leq g(u^{(1)}). \tag{10}$$

Notation: We denote by $\mathcal{E}$ the set of those $(n-1)$-variable Boolean functions satisfying Condition (10).

Remark. This generalization of the HWB function can be further generalized: keeping the idea of an insertion of a coordinate in the input vector z, let φ be any function from $\mathbb{F}_2^{n-1}$ to $\{1, \ldots, n\}$ and g an $(n-1)$-variable Boolean function, then we can define the function $F_{\varphi,g}(z) = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$ where $x_i = z_i$ for every $i < \varphi(z)$, $x_{\varphi(z)} = g(z)$ and $x_i = z_{i-1}$ for every $i > \varphi(z)$. It seems more difficult to study the injectivity of such general vectorial function (in the case of F_g, the relative simplicity of the relation between the Hamming weight of z and that of $F_g(z)$ has been very useful) and then the cryptographic parameters of the corresponding parameterized Boolean function. But some simple choices of $\varphi(z)$ could be tried. Of course, this latter generalization can itself be generalized, by inserting several binary values instead of only one. $\diamond$

4.3 On the number of constructed functions and examples

The transformation $g \in \mathcal{E} \mapsto f_{F_g}$ can be viewed as a secondary construction of n-variable Boolean functions from those $(n-1)$-variable Boolean functions belonging to $\mathcal{E}$. Note that different functions g in $\mathcal{E}$ essentially provide different functions f_{F_g}, since given $u \in \mathbb{F}_2^{n-2}$, the three possible choices of $(g(u^{(0)}), g(u^{(1)}))$ in $\{(0,0),(0,1),(1,1)\}$ provide different values for the ordered pair $(F_g(u^{(0)}), F_g(u^{(1)}))$. But the question of the injectivity of the mapping $g \in \mathcal{E} \mapsto f_{F_g}$ should be studied more rigorously, since the pairs $\{u^{(0)}, u^{(1)}\}$ do not provide a partition of $\mathbb{F}_2^{n-1}$, because there are pairs of vectors u, u' such that $u^{(0)} = u'^{(1)}$ (this happens when $u_{w_H(u)} = 1$, $u'_{w_H(u)} = 0$ and $u_i = u'_i$ for $i \neq w_H(u)$) and there are also vectors $z \in \mathbb{F}_2^{n-1}$ that are different from any $u^{(0)}, u^{(1)}$ (and on which g can take any value): those such that $z_{w_H(z)} = 0$ and $z_{w_H(z)+1} = 1$.

There are roughly $2^{3\cdot 2^{n-3}}$ functions in $\mathcal{E}$, since for each of the 2^{n-2} pairs $(u^{(0)}, u^{(1)})$, the value of $g(u^{(1)})$ is free if $g(u^{(0)}) = 0$ and equals 1 otherwise, because of Condition (10); about 2^{n-3} values of g are then constrained and $2^{n-1} - 2^{n-3} = 3 \cdot 2^{n-3}$ are free. Of course, this is only a rough approximation, since these constraints are not independent of each others. We leave open the problems of evaluating exactly the size of $\mathcal{E}$ and determining the number of distinct functions f_{F_g}.

Number $2^{3\cdot 2^{n-3}}$ being negligible with respect to the number of all n-variable balanced functions, whose number is $\binom{2^n}{2^{n-1}} \approx \frac{2^{2^n+\frac{1}{2}}}{\sqrt{\pi 2^n}}$, the functions f_{F_g} cover then only a small part of them. But this also happens for all known non-trivial secondary constructions having initial functions in less than n variables (see [3]), and in fact even for those having initial functions in n variables like in the construction without extension of the number of variables of [3, Proposition 85] (whose number and algebraic immunity of the provided balanced functions is not known). Compared to these constructions, the present one potentially provides a huge number of non-trivial balanced functions.

An obvious example of functions in $\mathcal{E}$ is monotone Boolean functions, but many more functions satisfying (10) with $u^{(j)}$ so defined exist. Indeed, a Boolean function g is monotone if and only if it satisfies (10) for any u when $u^{(j)}$ is obtained from u by inserting a coordinate of value j at any position, while here we insert j at only one position, for each u.

The HWB function equals f_{F_1} (i.e. corresponds to constant function $g = 1$, which is a monotone function). Hence, from a constant function g (the worst possible nonzero Boolean function from a cryptographic viewpoint), we already obtain thanks to this secondary construction a function f_{F_g} which has rather good cryptographic parameters as shown in [15] (the only weak point is the nonlinearity, see below).

The simplest example of a monotone function g different from constant function 1 is the zero function (also weak, of course). The image set of F_0 equals $\{x \in \mathbb{F}_2^n;\ x \neq 1_n, x_{w_H(x)+1} = 0\}$. Function f_{F_0} is the complement, not of the HWB function, but of the HWB function composed (on the right) with the shift

$(x_1, \ldots, x_n) \mapsto (x_2, \ldots, x_n, x_1)$. Indeed, this is true for $x \notin \{0_n, 1_n\}$, since this shift moves $x_{w_H(x)+1}$ to position $w_H(x)$, and it can be checked for $x \in \{0_n, 1_n\}$.

Remark. Despite the fact that, for every $z \in \mathbb{F}_2^{n-1}$, $g(z)$ can take only one value, and despite the injectivity of F_g, there may exist pairs (x, y) of distinct elements of $supp(f)$, which coincide at all positions except one, whose index equals $\max(w_H(x), w_H(y))$. Indeed, these two elements x and y can be the images of different $z \in \mathbb{F}_2^{n-1}$ by F_g. For instance, for $g = 0$, set $x = F_0(z)$ where z is taken in $\mathbb{F}_2^{n-1}$, and let y be the vector obtained from x by changing its coordinate of index $i := w_H(x) + 1$ (which equals 0) into 1. Then if $z_i = 0$, we have $y = F_0(z')$, where z' is obtained from z by changing its coordinate of index i into 1, since $w_H(z') = i$ and $(F_0(z'))_{i+1} = 0 = x_{i+1} = y_{i+1}$ and the coordinates of $F_0(z')$ and y coincide at all positions. Note that we have $y_{w_H(y)} = 1$ and $y_{w_H(y)+1} = 0$ but this does not contradict the condition "there does not exist $x \in \mathbb{F}_2^n$ such that $x_{w_H(x)} = 1 = g(x'')$ and $x_{w_H(x)+1} = 0 = g(x')$" since $g(y'') = 0$.
We see that the construction of f_{F_g} from g is not easy to apprehend. But this is why it can reach a high complexity, while its output shall be as easy to compute as that of g. ◇

4.4 The difficulty of characterizing generalized hidden weight bit functions

Let us first give the formal definition of these functions:

Definition 2 *For every $n \geq 2$ and every $(n-1)$-variable Boolean function g in $\mathcal{E}$, we call f_{F_g} a generalized hidden weight bit (GHWB) function.*

Let us now study these functions and see if it is possible to find necessary and sufficient conditions under which a given balanced function f is a GHWB function. We would need to characterize, by means of $supp(f)$ only, the existence of a function g such that F_g is injective and onto $supp(f)$. Since f is taken balanced, it is enough to ensure the existence of g such that each element in $supp(f)$ is in the image set of F_g (the injectivity of F_g being automatically satisfied).
Note that, given a GHWB function f_{F_g} and $x \in supp(f_{F_g})$, only two $z \in \mathbb{F}_2^{n-1}$ are possible candidates as pre-images of x by F_g: those obtained by erasing from x its coordinates of indices $w_H(x)$ and $w_H(x) + 1$, respectively. Indeed, either the unique pre-image z of x by F_g has Hamming weight $w_H(x) - 1$, and z is then obtained by erasing from x its coordinate of index $w_H(x)$, which equals then 1, or it has Hamming weight $w_H(x)$ and z is then obtained by erasing from x its coordinate of index $w_H(x) + 1$, which equals then 0. Hence, an n-variable balanced function f can be GHWB only if, for every $x \in supp(f)$, we have $x_{w_H(x)} = 1$ or $x_{w_H(x)+1} = 0$, that is:

$$x_{w_H(x)} \geq x_{w_H(x)+1}.$$

Let f be any balanced function satisfying this latter property and x be any element of $supp(f)$. Then:
(i) if $x_{w_H(x)} = x_{w_H(x)+1} = 0$, then the unique possible pre-image z of x by F_g satisfies $z_{w_H(z)} = 0$ and $w_H(z) = w_H(x)$,
(ii) if $x_{w_H(x)} = x_{w_H(x)+1} = 1$, then the unique possible pre-image z satisfies $z_{w_H(z)+1} = 1$ and $w_H(z) = w_H(x) - 1$,
(iii) if $x_{w_H(x)} = 1$ and $x_{w_H(x)+1} = 0$, then there are two possible pre-images: a first one z (denoted by x' in Subsection 4.2) satisfying $z_{w_H(z)} = 1$, having Hamming weight $w_H(z) = w_H(x)$, and obtained from x by erasing $x_{w_H(x)+1}$, and a second one z' (denoted by x'' in Subsection 4.2) satisfying $z'_{w_H(z')+1} = 0$, having Hamming weight $w_H(z') = w_H(x) - 1$, and obtained from x by erasing $x_{w_H(x)}$.
To build a function g with the desired conditions, we need a way to choose a pre-image z for each $x \in supp(f)$, in such a way that each such z corresponds to a unique x. We need then that:
1. there do not exist two elements x and $\tilde{x}$ in $supp(f)$, one being of type (i) and the other being of type (ii), and both having the same pre-image; we observe that these two elements have necessarily Hamming weights differing by 1, hence, denoting by x the one of smaller weight and by $\tilde{x}$ the one of larger weight, we have $w_H(\tilde{x}) = w_H(x) + 1$ and

$$\left\{ \begin{array}{lcl} (x_{w_H(x)}, x_{w_H(x)+1}, x_{w_H(x)+2}) & = & (0,0,1) \\ (\tilde{x}_{w_H(\tilde{x})-1}, \tilde{x}_{w_H(\tilde{x})}, \tilde{x}_{w_H(\tilde{x})+1}) & = & (0,1,1) \end{array} \right. \text{ must be avoided}$$

when x and $\tilde{x}$ coincide at the other positions,
2. there do not exist in $supp(f)$ an element x of type (iii) and two elements respectively of types (ii) and (i), whose pre-images coincide respectively with those z and z' of x. This means that there do not exist three elements $x, \tilde{x}, \tilde{x}'$ in $supp(f)$ satisfying $w_H(\tilde{x}) = w_H(x) + 1$ and $w_H(\tilde{x}') = w_H(x) - 1$ and

$$\left\{ \begin{array}{lcl} (x_{w_H(x)-1}, x_{w_H(x)}, x_{w_H(x)+1}, x_{w_H(x)+2}) & = & (0,1,0,1) \\ (\tilde{x}_{w_H(\tilde{x})-2}, \tilde{x}_{w_H(\tilde{x})-1}, \tilde{x}_{w_H(\tilde{x})}, \tilde{x}_{w_H(\tilde{x})+1}) & = & (0,1,1,1) \\ (\tilde{x}'_{w_H(\tilde{x}')}, \tilde{x}'_{w_H(\tilde{x}')+1}, \tilde{x}'_{w_H(\tilde{x}')+2}, \tilde{x}'_{w_H(\tilde{x}')+3}) & = & (0,0,0,1) \end{array} \right. \text{ must be avoided}$$

when x, $\tilde{x}$ and $\tilde{x}'$ coincide at the other positions.
But this is not sufficient. Indeed, once that z or z' has been chosen as the pre-image of x, this has an incidence on the other choices for other elements x of type (iii), because situations of the type $\left\{ \begin{array}{lcl} (x_{w_H(x)}, x_{w_H(x)+1}, x_{w_H(x)+2}) & = & (1,0,0) \\ (\tilde{x}_{w_H(\tilde{x})-1}, \tilde{x}_{w_H(\tilde{x})}, \tilde{x}_{w_H(\tilde{x})+1}) & = & (1,1,0) \end{array} \right.$ may lead to a same z for two different $x, \tilde{x}$, and when both choices of z and z' are possible for some x, it may even happen that the adopted choice has been a bad one and needs to be changed in a backtracking.
We leave open the question of having a closed necessary and sufficient condition for f to be GHWB.

Remark. The complement $f_{F_g} + 1$ of a GHWB function is not a GHWB function since its support contains elements x such that $x_{w_H(x)} < x_{w_H(x)+1}$. But

as in the case of the HWB function, $f_{F_g}+1$ may be affine equivalent (and possibly permutation equivalent) to a GHWB function, and therefore have its cryptographic parameters studied as such (recall that all cryptographic parameters of Boolean functions are affine invariant, except correlation immunity and resiliency, which are permutation invariant, see [3]). We leave open the characterizations of such GHWB functions. ◇

Remark. There is a way of revisiting slightly differently the observations above. For every element x in the support of a GHWB function f_{F_g}, there exists $\epsilon \in \{0,1\}$ such that $x_{w_H(x)+\epsilon} = 1-\epsilon$ (where $1-\epsilon$ is viewed in $\mathbb{F}_2$). Indeed, as we already saw, the unique pre-image z of x by F_g either satisfies $g(z)=0$, and then we have $w_H(x)=w_H(z)$ and $x_{w_H(x)+1}=0$ ($\epsilon=1$ satisfies then the condition), or $g(z)=1$, and then we have $w_H(x)=w_H(z)+1$ and $x_{w_H(x)}=1$ ($\epsilon=0$ satisfies then the condition).
• If only one value $\epsilon\in\{0,1\}$ satisfies $x_{w_H(x)+\epsilon}=1-\epsilon$, then denoting this value by ϵ_x, we have that $x_{w_H(x)+\epsilon_x}=1-\epsilon_x$ also equals $x_{w_H(x)+1-\epsilon_x}$ (since otherwise there would not be uniqueness of ϵ_x), and the two consecutive values $x_{w_H(x)}$ and $x_{w_H(x)+1}$ in x are then equal. Then, we are in one of the cases (i) and (ii) above, and z can be obtained from x by erasing indifferently one of these two coordinates (conversely, if $x_{w_H(x)}=x_{w_H(x)+1}$, then ϵ is unique and equals $\epsilon_x=1-x_{w_H(x)}$; we have $g(z)=x_{w_H(x)}=x_{w_H(x)+1}$).
• If both values $\epsilon\in\{0,1\}$ satisfy $x_{w_H(x)+\epsilon}=1-\epsilon$, then $x_{w_H(x)}=1$ and $x_{w_H(x)+1}=0$ and we are in case (iii). One of these two values ϵ, that we shall still denote by ϵ_x, is such that z is obtained from x by erasing its coordinate of index $w_H(x)+\epsilon_x$ (whose value is $1-\epsilon_x$), and we have $g(z)=x_{w_H(z)+1}=x_{w_H(x)+\epsilon_x}=1-\epsilon_x$ and $x=F_g(z)$. Due to the injectivity of F_g, we have, denoting by z' the vector obtained from x by erasing its coordinate of index $w_H(x)+1-\epsilon_x$ (which equals ϵ_x) that $F_g(z')\neq x$, that is, $g(z')=1-\epsilon_x$ and the image $\tilde{x}$ of z' by F_g can be obtained from x by complementing its coordinate of index $w_H(x)+1-\epsilon_x$ (whose value is ϵ_x and becomes then $1-\epsilon_x$), and $\tilde{x}$ is such that $\tilde{x}_{w_H(x)}=\tilde{x}_{w_H(x)+1}$. ◇

We see that, despite the simplicity of the definition of the GHWB functions, their nature and specificity are not that simple to apprehend.

5 An expression of GHWB functions

Let us now give the general expression of f_{F_g} by means of g (assuming that g has the desired property (10)): expressing that $f_{F_g}(x)=1$ if and only if there exists z in $\mathbb{F}_2^{n-1}$ such that $x=F_g(z)$, using that such z if it exists is unique, since F_g is injective, and distinguishing the two cases $g(z)=0$ (for which we have $w_H(x)=w_H(z)$ and therefore $z=x'$, as defined in Subsection 4.2, since this 0 is at position $w_H(z)+1=w_H(x)+1$) and $g(z)=1$ (for which we have $w_H(x)=w_H(z)+1$ and therefore $z=x''$, since this 1 is at position $w_H(z)+1=w_H(x)$), we obtain that, for every x different from 0_n (so that

$w_H(x) \in \{1, \ldots, n\}$) and from 1_n (so that $w_H(x) + 1 \in \{1, \ldots, n\}$):

$$f_{F_g}(x) = (x_{w_H(x)+1} + 1)\,(g(x') + 1) + x_{w_H(x)}\, g(x''). \tag{11}$$

Moreover, since if $g(0_{n-1}) = 0$ then $F_g(0_{n-1}) = 0_n$ and then $f_{F_g}(0_n) = 1$, and if $g(0_n) = 1$, then 0_n is not a value taken by F_g and therefore $f_{F_g}(0_n) = 0$, we have:

$$f_{F_g}(0_n) = g(0_{n-1}) + 1,$$

and we have also:

$$f_{F_g}(1_n) = g(1_{n-1}),$$

since if $g(1_{n-1}) = 1$ then $F_g(1_{n-1}) = 1_n$ and then $f_{F_g}(1_n) = 1$, and if $g(1_n) = 0$ then 1_n is not a value taken by F_g and therefore $f_{F_g}(1_n) = 0$.
Taking the double convention:

$$\forall x \in \mathbb{F}_2^n,\ x_0 = 0 \text{ and } x_{n+1} = 1, \tag{12}$$

we have that Relation (11) is valid for every $x \in \mathbb{F}_2^n$. In other terms:

Proposition 3 *For every $n \geq 2$ and every $(n-1)$-variable Boolean function g in $\mathcal{E}$, we have, with the convention (12):*

$$\begin{aligned} f_{F_g}(x) &= (x_{w_H(x)+1} + 1)\,(g(x') + 1) + x_{w_H(x)}\, g(x''). \\ &= f_{F_0}(x)\,(g(x') + 1) + f_{F_1}(x)\, g(x''). \end{aligned} \tag{13}$$

We have then a fast way to compute the output of GHWB functions. Actually, the time needed to compute the output of f_{F_g} is not significantly larger than that for computing the output of g (unless of course computing the output of g is so fast that computing the Hamming weight is significantly longer). Hence, for fast functions g, the quality of HWB function is preserved by this generalization, while the function has in fact a rather complex structure. Let us take for instance for $g(z)$ the simplest possible non-constant monotone functions (for which Condition (10) is satisfied): monomials z^I, $\emptyset \neq I \subsetneq \{1, \ldots, n-1\}$. Then:

$$f_{F_g}(x) = (x_{w_H(x)+1} + 1)\,(x'^I + 1) + x_{w_H(x)}\, x''^I$$

is fast to compute and has an expression that is rather complex since x'^I and x''^I are products of coordinates of x whose choices depend on its Hamming weight. We shall see that the ANF of f_{F_g} is still more complex.

Remark. We can see with the change of the parameterization F of the HWB function F_1 into F_g that a slight modification of F (here, in one position of $F(z)$ only), may change significantly the expression of f_F, and we shall see that it also changes the values of the cryptographic parameters of f_{F_g}.
Note that changing g for an affine equivalent function g' also satisfying Property (10) transforms f_{F_g} into a function that is in general affine inequivalent, because affine permutations do not preserve the Hamming weight. Looking now

more closely at the linear permutations that preserve the Hamming weight, that is, coordinate permutations, if $g' = g \circ \sigma$ where we denote the same way a permutation σ of $\{1, \ldots, n-1\}$ and the corresponding linear permutation $z = (z_1, \ldots, z_{n-1}) \mapsto (z_{\sigma(1)}, \ldots, z_{\sigma(n-1)})$ over $\mathbb{F}_2^{n-1}$, then $f_{F_{g'}}(x)$ is not necessarily obtained from $f_{F_g}(x)$ by composing it by a linear permutation. Indeed, let us view the support of f_{F_g} as the union $\bigcup_{i=0}^{n}\{F_g(z); z \in \mathbb{F}_2^{n-1}; w_H(z) = i\}$ and the support of $f_{F_{g'}}$ as the union $\bigcup_{i=0}^{n}\{F_{g'}(z); z \in \mathbb{F}_2^{n-1}; w_H(z) = i\}$; for each i, the set $\{F_{g'}(z); z \in \mathbb{F}_2^{n-1}; w_H(z) = i\}$ is the image of $\{F_g(z); z \in \mathbb{F}_2^{n-1}; w_H(z) = i\}$ by the permutation:

$$\sigma_i(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) = (x_{\sigma(1)}, \ldots, x_{\sigma(i-1)}, x_i, x_{\sigma(i+1)}, \ldots, x_{\sigma(n)})$$

over $\mathbb{F}_2^n$, but σ_i depends on i, as we can see. Most probably, $f_{F_{g'}}(x)$ and $f_{F_g}(x)$ are not affine equivalent, in general. Note that even changing g into $g' = g + 1$ would not provide affine equivalent functions $f_{F_{g'}}(x)$ and $f_{F_g}(x)$, but anyway, changing g into $g+1$ does not preserve Property (10), except if g is constant. $\diamond$

6 Algebraic normal form and algebraic degree

We have seen in [4] that if F is an injective (k, n)-function, given by its ANF, then, denoting by f_i the i-th coordinate function of F, we have from (6):

$$f_F(x) = \sum_{z \in \mathbb{F}_2^k} \delta_0(x + F(z)) = \sum_{z \in \mathbb{F}_2^k} \Big(\prod_{i=1}^{n} (x_i + f_i(z) + 1) \Big) \tag{14}$$

$$= \sum_{\substack{I \subseteq \{1,\ldots,n\} \\ z \in \mathbb{F}_2^k}} \left(\prod_{i \in I^c} (f_i(z) + 1) \right) x^I = \sum_{\substack{I \subseteq \{1,\ldots,n\} \\ d_{alg}(\prod_{i \in I^c}(f_i(z)+1)) = k}} x^I, \tag{15}$$

$$d_{alg}(f_F) = \max\Big\{ |I|; I \subseteq \{1, \ldots, n\}, d_{alg}\Big(\prod_{i \in I^c} (f_i(z) + 1) \Big) = k \Big\}$$

$$= \max\Big\{ |I|; I \subseteq \{1, \ldots, n\}, d_{alg}\Big(\prod_{i \in I^c} (f_i(z)) \Big) = k \Big\}.$$

Proposition 4 *[4] Let F be any (k, n)-function. The algebraic degree of the parameterized function f_F equals $n - l$ where l is the minimal number of coordinate functions of F whose product has algebraic degree k (that is, odd Hamming weight). In particular, we have $d_{alg}(f_F) = n - 1$ if and only if $d_{alg}(F) = k$, that is, $\sum_{z \in \mathbb{F}_2^k} F(z) \neq 0_n$.*

The expressions of the coordinate functions of $F_g(z)$ deduced from the definition of GHWB functions:

$$f_i(z) = \begin{cases} z_i & \text{if } i \leq w_H(z) \\ g(z) & \text{if } i = w_H(z) + 1 \\ z_{i-1} & \text{if } i \geq w_H(z) + 2 \end{cases}, \tag{16}$$

and the expression of $f_{F_g}(x)$ given in (11) are not ANFs because x, respectively z, do not intervene in them only through monomials x^I, respectively z^I, but also by means of their Hamming weights. Let us then see how we can change these expressions into the actual ANFs. We shall obtain expressions that are complex[1], but let us make this work for completeness. We need first a technical lemma.

Lemma 1 *Given a positive integer k and $i \in \{1, \ldots, k\}$, let σ_i denote the i-th elementary symmetric Boolean function over $\mathbb{F}_2^k$ (equal to the sum, modulo 2, of all the monomials of degree i in z). Then, for every $j \in \{0, \ldots, k\}$ and every $z \in \mathbb{F}_2^k$, we have $w_H(z) = j$ if and only if, for every $i \in \{1, \ldots, k\}$, $\sigma_i(z)$ equals $\binom{j}{i} \pmod 2 = \begin{cases} 1 \text{ if } i \preceq j \\ 0 \text{ otherwise} \end{cases}$, where $i \preceq j$ means that the binary expansion of i has support included in that of j.*

Proof. For every $z \in \mathbb{F}_2^k$, we have $\sigma_i(z) = \binom{w_H(z)}{i} \pmod 2$, which equals $\begin{cases} 1 \text{ if } i \preceq w_H(z) \\ 0 \text{ otherwise} \end{cases}$, according to Lucas' theorem (see e.g. [11]). For having $w_H(z) = j$, the condition $\sigma_i(z) = \binom{j}{i} \pmod 2 = \begin{cases} 1 \text{ if } i \preceq j \\ 0 \text{ otherwise} \end{cases}$ is then clearly necessary. Let us prove that it is also sufficient. As we just showed, each set $E_j = \{z \in \mathbb{F}_2^k; w_H(z) = j\}$ is included in the set $E'_j = \{z \in \mathbb{F}_2^k; \forall i \in \{1, \ldots, k\}, \sigma_i(z) = \binom{j}{i} \pmod 2\}$. The sets E_j are disjoint. The sets E'_j are disjoint too since the list of the values "1 if $i \preceq j$, 0 otherwise" where i ranges over $\{1, \ldots, k\}$ is different for different values of j. Since $\bigcup_{j=0}^k E_j = \bigcup_{j=0}^k E'_j = \mathbb{F}_2^k$, we have then $E_j = E'_j$ for every j and this completes the proof. □

We deduce the ANF of f_{F_g}, using Relation (11):

Proposition 5 *For every $n \geq 2$ and every $(n-1)$-variable Boolean function g in $\mathcal{E}$, denoting for every $x \in \mathbb{F}_2^n$ by $x^{(j)}$ the vector obtained from x by deleting its coordinate of index j, the ANF of the GHWB function equals:*

$$f_{F_g}(x) = \sum_{j=0}^{k} \left(\prod_{i \in \{1,\ldots,k\}; i \preceq j} \Big(\sigma_i(x)\Big) \prod_{i \in \{1,\ldots,k\}; i \not\preceq j} \Big(\sigma_i(x) + 1\Big) \right) h_j(x), \qquad (17)$$

where $h_j(x) = (x_{j+1} + 1)\,(g(x^{(j+1)}) + 1) + x_j\, g(x^{(j)})$, with the double convention that $x_0 = 0$ and $x_{n+1} = 1$ for every $x \in \mathbb{F}_2^n$.

Addressing now the ANF of $F_g(z) = (f_1(z), \ldots, f_n(z)) \in \mathbb{F}_2^n$ as recalled in

[1] Because of this complexity, the HWB function could not be used for instance in the FLIP and FiLIP ciphers [13, 12] (which needed to use Boolean functions with very simple ANF, since these stream ciphers were meant to work in conjunction with fully homomorphic encryption). However, f_{F_g} is well adapted to a use where it is enough to compute its output (for instance in a classical stream cipher).

Relation (16) and with the same convention as above, we have:

$$f_i(z) = \sum_{j=0}^{k} \left(\prod_{i \in \{1,\ldots,k\}; i \preceq j} \Big(\sigma_i(z)\Big) \prod_{i \in \{1,\ldots,k\}; i \not\preceq j} \Big(\sigma_i(z)+1\Big) \right) h_{i,j}(z),$$

where

$$h_{i,j}(z) = \begin{cases} z_i \text{ if } i \leq j \\ g(z) \text{ if } i = j+1 \\ z_{i-1} \text{ if } i \geq j+2. \end{cases}$$

We can see that g plays a role in each coordinate function of F_g. This confirms that the resulting function f_{F_g} is rather complex, despite the fact that it has a fast to compute output.

Let us now determine whether f_{F_g} has optimal algebraic degree $n-1$. Working on the expression (17) would be a little complex. Let us rather use Proposition 4. For every $i \in \{1, \ldots, n\}$, we have, assuming $k \geq 2$:

$$\sum_{z \in \mathbb{F}_2^k} f_i(z) =$$

$$\Big(\sum_{z \in \mathbb{F}_2^k; w_H(z) \geq i} z_i \Big) + \Big(\sum_{z \in \mathbb{F}_2^k; w_H(z) = i-1} g(z) \Big) + \Big(\sum_{z \in \mathbb{F}_2^k; w_H(z) \leq i-2} z_{i-1} \Big) =$$

$$\left(\sum_{j=i-1}^{k-1} \binom{k-1}{j} + \sum_{z \in \mathbb{F}_2^k; w_H(z) = i-1} g(z) + \sum_{j=0}^{i-3} \binom{k-1}{j} \right) \pmod 2 =$$

$$\left(\binom{k-1}{i-2} + \sum_{z \in \mathbb{F}_2^k; w_H(z) = i-1} g(z) \right) \pmod 2.$$

According to Proposition 4, most functions g are then such that $d_{alg}(f_{F_g}) = n-1$:

Proposition 6 *For every $n \geq 2$ and every $(n-1)$-variable Boolean function g in $\mathcal{E}$, we have $d_{alg}(f_{F_g}) = n-1$ unless, for every $i = 1, \ldots, n$, we have $\sum_{z \in \mathbb{F}_2^k; w_H(z) = i-1} g(z) = \binom{k-1}{i-2} \pmod 2$.*

In the case of the HWB function, we have $g = 1$ and f_{F_g} has algebraic degree $n-1$, since there exists i such that $\binom{k}{i-1} \neq \binom{k-1}{i-2} \pmod 2$ (for instance, $i = k$).

7 Walsh transform, balance, nonlinearity and resiliency

It is shown in [4] that for every injective (k,n)-function F parameterizing a Boolean function f_F, we have:

$$W_{f_F}(u) = 2^n \delta_0(u) - 2\, W_F(0_k, u), \tag{18}$$

and thus, for $k = n-1$:

$$nl(f_F) = 2^{n-1} - \max_{u\in\mathbb{F}_2^n; u\neq 0_n} |W_F(0_{n-1}, u)| \quad (19)$$

$$\geq 2\,nl(F), \quad (20)$$

but it seems difficult to determine these two nonlinearity parameters.

We know that the nonlinearity of the HWB function f_{F_1} equals $2^{n-1} - 2\binom{n-2}{\lceil \frac{n-2}{2} \rceil}$ (see [15]). When n is odd, $nl(f_{F_1})$ equals then the nonlinearity of the majority function, which is known to be the worst possible for a function with optimal algebraic immunity, according to Lobanov's bound [10], which states that the nonlinearity of such function is at least $2\sum_{i=0}^{\lceil n/2 \rceil - 2}\binom{n-1}{i} = 2^{n-1} - \binom{n-1}{(n-1)/2}$. When n is even, $nl(f_{F_1})$ is slightly worse than the nonlinearity of the majority function which is itself slightly above Lobanov's bound; anyway, $nl(f_{F_1})$ is weak, despite the fact that, as the majority function, the HWB function being fast to compute (since all the complexity of its computation lies in the computation of the Hamming weight of the input), it can then be used in stream ciphers with many more variables than classical functions. Indeed, the main parameter playing a role in the complexity of the fast correlation attack is the nonlinearity bias $\frac{1}{2} - \frac{nl(f)}{2^n}$, whose value here is too large for being compensated by the number of variables.

Function f_{F_0} has the same drawback. We have seen that it is the complement of the HWB function composed with the shift $s : (x_1, \ldots, x_n) \mapsto (x_2, \ldots, x_n, x_1)$. It has then the same nonlinearity as the HWB function.

We shall see with computer investigations that, on the contrary, some functions g in $\mathcal{E}$ can provide functions f_{F_g} with much better nonlinearity. But it seems very hard to mathematically determine the nonlinearity of general GHWB functions. According to (19) and to the fact that, for a general g (satisfying (10)), we have:

$$W_{F_g}(0_{n-1}, u) = \sum_{i=0}^{n-1} \sum_{\substack{z\in\mathbb{F}_2^{n-1} \\ w_H(z)=i}} (-1)^{u_{i+1}g(z)+u^{[i+1]}\cdot z}, \quad (21)$$

where $u^{[i+1]}$ is obtained from u by puncturing (i.e. deleting) its coordinate u_{i+1} of index $i+1$, the nonlinearity of the general function f_{F_g}, where g satisfies (10), equals:

$$nl(f_{F_g}) = 2^{n-1} - \max_{u\in\mathbb{F}_2^n; u\neq 0_n} \Big| \sum_{i=0}^{n-1} \sum_{\substack{z\in\mathbb{F}_2^{n-1} \\ w_H(z)=i}} (-1)^{u_{i+1}g(z)+u^{[i+1]}\cdot z} \Big|. \quad (22)$$

Because of this puncturing at a position depending on the Hamming weight of z, it seems difficult to go further in the determination of $nl(f_{F_g})$. As a matter of fact, the quality of the construction, which provides complex functions whose output is simple to compute, becomes a drawback when we want to guess and prove general properties, in particular when dealing with the nonlinearity. Let

us make however a few observations.

• When $u = 1_n$, the double sum in (22) equals $\sum_{z\in\mathbb{F}_2^{n-1}}(-1)^{g(z)+1_{n-1}\cdot z} = W_g(1_{n-1})$ and g is then better chosen so that $W_g(1_{n-1})$ has an absolute value not too far from the quadratic mean of W_g deduced from the Parseval relation (which writes $\sum_{v\in\mathbb{F}_2^{n-1}} W_g^2(v) = 2^{2n-2}$), that is: $2^{\frac{n-1}{2}}$. Note that when g equals 0 or 1, this is the case since $W_g(1_{n-1})$ equals then 0, and the best affine approximation of the HWB function is then clearly not by the function $1_{n-1}\cdot z = w_H(z) \pmod 2$ nor by its complement. In fact, we know from [15] that it is by the coordinate functions or their complements, that is, when u has Hamming weight 1. The case studied next includes this as a particular case.

• When all the "1" in u are consecutive, that is, when u equals the vector $\sum_{j=r}^{s} e_j$, where $1 \le r \le s \le n$, and e_j is the j-th vector (of Hamming weight 1) of the canonical basis of $\mathbb{F}_2^n$, we have:

$$\sum_{i=0}^{n-1} \sum_{\substack{z\in\mathbb{F}_2^{n-1}\\ w_H(z)=i}} (-1)^{u_{i+1}g(z)+u^{[i+1]}\cdot z} =$$

$$\sum_{i=0}^{r-2} \sum_{\substack{z\in\mathbb{F}_2^{n-1}\\ w_H(z)=i}} (-1)^{\sum_{j=r}^{s} z_{j-1}} + \sum_{i=r-1}^{s-1} \sum_{\substack{z\in\mathbb{F}_2^{n-1}\\ w_H(z)=i}} (-1)^{g(z)+\sum_{j=r}^{i} z_j+\sum_{j=i+2}^{s} z_{j-1}}$$

$$+\sum_{i=s}^{n} \sum_{\substack{z\in\mathbb{F}_2^{n-1}\\ w_H(z)=i}} (-1)^{\sum_{j=r}^{s} z_j} =$$

$$\sum_{i=0}^{r-2}\left(\sum_{0\le j\le \frac{i}{2}} \binom{s-r+1}{2j}\binom{n-s+r-2}{i-2j} - \sum_{0\le j< \frac{i}{2}} \binom{s-r+1}{2j+1}\binom{n-s+r-2}{i-2j-1}\right)$$

$$+ \sum_{i=r-1}^{s-1} \sum_{\substack{z\in\mathbb{F}_2^{n-1}\\ w_H(z)=i}} (-1)^{g(z)+\sum_{j=r}^{i} z_j+\sum_{j=i+2}^{s} z_{j-1}}$$

$$+\sum_{i=s}^{n}\left(\sum_{0\le j\le \frac{i}{2}} \binom{s-r+1}{2j}\binom{n-s+r-2}{i-2j} - \sum_{0\le j< \frac{i}{2}} \binom{s-r+1}{2j+1}\binom{n-s+r-2}{i-2j-1}\right).$$

We see the difficulty of calculating the Walsh transform of general functions f_{F_g}, even at such particularly simple input u.

– When u simply equals e_r, that is, when $s = r$, this expression becomes:

$$\sum_{i=0}^{r-2}\left(\binom{n-2}{i} - \binom{n-2}{i-1}\right) + \sum_{\substack{z\in\mathbb{F}_2^{n-2}\\ w_H(z)=r-1}} (-1)^{g(z)} + \sum_{i=r}^{n}\left(\binom{n-2}{i} - \binom{n-2}{i-1}\right) =$$

$$\binom{n-2}{r-2} + \sum_{\substack{z\in\mathbb{F}_2^{n-1}\\ w_H(z)=r-1}} (-1)^{g(z)} - \binom{n-2}{r-1}. \tag{23}$$

Remark. For the HWB function, we have $g=1$ and (23) equals then $\binom{n-2}{r-2} - \binom{n-1}{r-1} - \binom{n-2}{r-1} = -2\binom{n-2}{r-1}$. The maximum of the absolute value is $2\binom{n-2}{\lceil\frac{n-2}{2}\rceil} = 2\binom{n-2}{\lfloor\frac{n-2}{2}\rfloor}$, and this maximum is taken for $r=\lceil\frac{n}{2}\rceil$ and $r=\lfloor\frac{n}{2}\rfloor$ (which, of course, makes two values if n is odd and one if it is even), and we have $W_{F_1}\left(0, e_{\lceil\frac{n}{2}\rceil}\right) < 0$ and $W_{F_1}\left(0, e_{\lfloor\frac{n}{2}\rfloor}\right) < 0$. We have already recalled that it is proved in [15] that the maximum absolute value of the Walsh transform of the HWB function at nonzero inputs is taken when the input has weight 1. We know then that the function(s) $x_{\lceil\frac{n}{2}\rceil}+1$ and $x_{\lfloor\frac{n}{2}\rfloor}+1$ are best affine approximations of the HWB function.
Similarly, for $g=0$, (23) equals $\binom{n-2}{r-2}+\binom{n-1}{r-1}-\binom{n-2}{r-1} = 2\binom{n-2}{r-2}$. The maximum is taken for $r=\lceil\frac{n}{2}\rceil+1$ and $r=\lfloor\frac{n}{2}\rfloor+1$ and we have $W_{F_0}\left(0, e_{\lceil\frac{n}{2}\rceil+1}\right) > 0$ and $W_{F_0}\left(0, e_{\lfloor\frac{n}{2}\rfloor+1}\right) > 0$. We know then that the functions $x_{\lceil\frac{n}{2}\rceil+1}$ and $x_{\lfloor\frac{n}{2}\rfloor+1}$ are best affine approximations of f_{F_0}. This will be useful in the sequel. ◇

For general g, we see that if $\sum_{\substack{z\in\mathbb{F}_2^{n-1}\\ w_H(z)=r-1}} (-1)^{g(z)}$ has an absolute value significantly smaller than $\binom{n-1}{r-1}$ for every $r=1,\ldots,n$, that is, if g is sufficiently non-constant on each set of fixed Hamming weight near $\frac{n}{2}$ (for which $\binom{n-1}{r-1}$ is large), then the absolute value in (23) will be significantly smaller than in the case of the HWB function, since $\left|\binom{n-2}{r-2} - \binom{n-2}{r-1}\right|$ is small, and the nonlinearity will have a chance, in some cases, of being better than for the HWB function. But as we shall see when reporting computer investigations, it is actually difficult not to lose with some u of Hamming weight different from 1 what has been gained with those u of Hamming weight 1, and actually we shall see that the functions having better nonlinearity than the HWB function, which fortunately exist and in some cases improve its nonlinearity in a strong way, are a small minority.
• Although the above calculations show that it is difficult to calculate all the values of the Walsh transform of f_{F_g}, it is however possible to derive a bound on the nonlinearity of this function, which will imply a necessary condition for reaching a good nonlinearity, which counterbalances a condition that we gave above. This bound can be proved by using (23), but we shall prove it slightly differently, for providing a complementary viewpoint:

Proposition 7 *For every $n\geq 2$ and every $(n-1)$-variable Boolean function g satisfying (10), we have:*

$nl(f_{Fg}) \leq nl(f_{F_0})+$

$$2 \min_{s\in\{\lceil \frac{n}{2}\rceil, \lfloor \frac{n}{2}\rfloor\}} \Big(\left|\{z \in supp(g); w_H(z) = s\}\right|, \left|\{z \in supp(g+1); w_H(z) = s-1\}\right| \Big).$$

Proof. We have, according to Relation (19):

$$nl(f_{Fg}) = 2^{n-1} - \max_{u\in\mathbb{F}_2^n; u\neq 0_n} |W_{Fg}(0_{n-1}, u)| =$$
$$2^{n-1} - \max_{\substack{u\in\mathbb{F}_2^n; u\neq 0_n \\ \epsilon\in\mathbb{F}_2}} \Big(\sum_{z\in\mathbb{F}_2^{n-1}} (-1)^{u\cdot F_0(z) + u_{w_H(z)+1}\, g(z) + \epsilon} \Big) =$$
$$2 \min_{\substack{u\in\mathbb{F}_2^n; u\neq 0_n \\ \epsilon\in\mathbb{F}_2}} w_H\Big(z \mapsto u\cdot F_0(z) + u_{w_H(z)+1}\, g(z) + \epsilon\Big).$$

We have seen in the remark above that $w_H\Big(z \mapsto u\cdot F_0(z) + \epsilon\Big)$ reaches its minimum for $u = e_r$, $r \in \{\lceil \frac{n}{2}\rceil + 1, \lfloor \frac{n}{2}\rfloor + 1\}$ and $\epsilon = 0$; then we have, denoting by $(e_r)_{w_H(z)+1}$ the coordinate of e_r of index $w_H(z)+1$:

$$2 \min_{\substack{u\in\mathbb{F}_2^n; u\neq 0 \\ \epsilon\in\mathbb{F}_2}} w_H\Big(z \mapsto u\cdot F_0(z) + u_{w_H(z)+1}\, g(z) + \epsilon\Big) \leq$$
$$2w_H\Big(z \mapsto e_r\cdot F_0(z) + (e_r)_{w_H(z)+1}\, g(z)\Big) \leq$$
$$2w_H\Big(z \mapsto e_r\cdot F_0(z)\Big) + 2w_H\Big(z \mapsto (e_r)_{w_H(z)+1}\, g(z)\Big) =$$
$$2w_H\Big(z \mapsto e_r\cdot F_0(z)\Big) + 2\left|\big\{z \in \mathbb{F}_2^{n-1}; w_H(z)+1 = r \text{ and } g(z) = 1\big\}\right| =$$
$$2 \min_{\substack{u\in\mathbb{F}_2^n; u\neq 0_n \\ \epsilon\in\mathbb{F}_2}} w_H\Big(z \mapsto u\cdot F_0(z) + \epsilon\Big) + 2\left|\{z \in supp(g); w_H(z) = r-1\}\right|,$$

and we deduce then:

$$nl(f_{Fg}) \leq nl(f_{F_0}) + 2\left|\{z \in supp(g); w_H(z) = r-1\}\right|.$$

This completes the first part of our proof. We have also:

$$nl(f_{Fg}) = 2^{n-1} - \max_{u\in\mathbb{F}_2^n; u\neq 0} |W_{Fg}(0_{n-1}, u)| =$$
$$2^{n-1} - \max_{\substack{u\in\mathbb{F}_2^n; u\neq 0_n \\ \epsilon\in\mathbb{F}_2}} \Big(\sum_{z\in\mathbb{F}_2^{n-1}} (-1)^{u\cdot F_1(z) + u_{w_H(z)+1}\,(g(z)+1)+\epsilon} \Big) =$$
$$2 \min_{\substack{u\in\mathbb{F}_2^n; u\neq 0_n \\ \epsilon\in\mathbb{F}_2}} w_H\Big(u\cdot F_1(z) + u_{w_H(z)+1}\,(g(z)+1) + \epsilon\Big) \leq$$
$$2 \min_{\substack{u\in\mathbb{F}_2^n; u\neq 0_n \\ \epsilon\in\mathbb{F}_2}} w_H\Big(u\cdot F_1(z) + \epsilon\Big) + 2\left|\{z \in supp(g+1); w_H(z) = r-2\}\right| =$$
$$nl(f_{F_1}) + 2\left|\{z \in supp(g+1); w_H(z) = r-2\}\right|,$$

where the inequality is proved similarly as above, using again the remark above. □

Remark. We saw above that to allow reaching a large nonlinearity, g should be sufficiently non-constant on each set of fixed Hamming weight near $\frac{n}{2}$. The bound of Proposition 7 shows that for allowing to reach with f_{F_g} a nonlinearity significantly larger than the HWB function, it is better to choose functions g which take significantly often value 1 on those inputs of Hamming weight $\lceil \frac{n}{2} \rceil$ and significantly often value 0 on those inputs of Hamming weight $\lceil \frac{n}{2} \rceil - 1$. Note that calculations similar to those in the proof of Proposition 7 can be made for other values of r, even those not corresponding exactly to maxima of $|W_{F_0}(0_{n-1}, u)|$ and $|W_{F_1}(0_{n-1}, u)|$. □

Computer investigation: experiments have been made with the kind help of Stjepan Picek. Some choices of g have provided better parameters (nonlinearity and algebraic immunity) than the HWB function. Only a minority of these choices did so, but they were however rather numerous. We report here the results obtained for the nonlinearity and will report in the next section those on the algebraic immunity. We tried so far the following monotone functions g: the majority function and the more general threshold functions, the functions of supports $\{x \in \mathbb{F}_2^{n-1}; supp(u) \subseteq supp(x)\}$ for all $u \in \mathbb{F}_2^n$), the function $x_1 + x_2 + x_1x_2$ and the monomial functions of any degrees, where we saw that degree 3 gives the best results among all tries:
- monomial function $g(x) = x_3x_6x_7$ in 12 variables provides f_{F_g} in 13 variables with nonlinearity 3284 (the nonlinearity $2^{n-1} - 2\binom{n-2}{\lceil \frac{n-2}{2} \rceil}$ of the HWB function equals 3172), not close enough to the covering radius bound $2^{n-1} - \lceil 2^{\frac{n}{2}-1} \rceil = 4064$ nor to the nonlinearity 3942 of the so-called Carlet-Feng function (too slow to run in a practical stream cipher). This represents a step forward, with an increase of 3.5% with respect to the HWB function (the algebraic immunity is the same as for the HWB function, see below).
- $g(x) = x_2x_7x_8$ in 13 variables provides f_{F_g} in 14 variables with nonlinearity 6668 (the nonlinearity of the HWB function equals 6344). This is here again still not close enough to the covering radius bound 8128 nor to the nonlinearity 8028 of the Carlet-Feng function, but this represents a still larger step forward, with an increase of 5.1%, compared to the HWB function, all the more since the algebraic immunity is in this case also better, see below.
- $g(x) = x_1x_7x_8$ in 14 variables provides f_{F_g} in 15 variables with nonlinearity 13372 (the nonlinearity of the HWB function equals 12952), not close enough to the covering radius bound 16320 nor to the nonlinearity 16242 of the Carlet-Feng function; this is an increase of 3.2% (the algebraic immunity is the same as for HWB).
- $g(x) = x_1x_8x_9$ in 15 variables provides f_{F_g} in 16 variables with nonlinearity 27158 (the nonlinearity of the HWB function equals 25904), not close enough to the covering radius bound 32640 nor to the nonlinearity 32530 of the Carlet-

Feng function; this is an increase of 4.8%; f_{F_g} provides a good tradeoff between security and speed, since the algebraic immunity is in this case also better than for HWB, see below.
Improvements: only a tiny part of all functions g satisfying (10) in such numbers of variables can be visited. The time needed for visiting all 7-variable functions satisfying (10) would already need thousands of centuries on a modern computer, and each incrementation of n by 1 replaces the number of functions to be investigated by roughly its square. We decided then to start from the best functions obtained in the first phase of our investigation and to modify them so that they still satisfy Condition (10); this could be easily done by changing only the values of g taken at the inputs z such that $z_{w_H(z)} = 0$ and $z_{w_H(z)+1} = 1$, since such changes preserve (10), because such z cannot be equal to any vector of the form $u^{(0)}$ nor to any vector of the form $u^{(1)}$. Unfortunately, this only allowed to slightly improve the best obtained nonlinearities (at the price of an increase of the complexity of the functions, providing bad trade-offs, then). An evolutionary method was then applied in [6, 8] with the same strategy and gave much better results. For instance, this provided a 15-variable function with a nonlinearity of 14604 (which represents an increase of 12.7% over the HWB function) and a 16-variable function with a nonlinearity of 29128 (which represents an increase of 12.4% over the HWB function). This latter result seems quite interesting. Function g being more complex than a simple monomial, function f_{F_g} represents a different tradeoff between security and speed, which is more in the direction of security. We expect that the best possible nonlinearities of all balanced functions f_{F_g} are still significantly larger than the values we obtained so far. Further theoretical and computational work will be needed for identifying good candidates to be investigated further. ◇

7.0.1 Resiliency

We have seen above that for $g(z)$ equal to a coordinate function, the corresponding function f_{F_g} is (at least) 1-resilient. ◇

8 Algebraic immunity

Of course, as already observed in [4], an n-variable function h is an annihilator of the n-variable f_F parameterized by F if and only if the Boolean function $h \circ F$ is identically zero. In the case of the GHWB function f_{F_g}, this writes:

$$h\big(z_1, \ldots, z_{w_H(z)}, g(z), z_{w_H(z)+1}, \ldots, z_{n-1}\big) = 0, \forall z \in \mathbb{F}_2^{n-1}.$$

If n is odd, then f_{F_g} has optimal algebraic immunity $\frac{n+1}{2}$ if and only if it has no nonzero annihilator of algebraic degree at most $\frac{n-1}{2}$ (indeed, this is true for any balanced Boolean function).

We know from [15] that the HWB function does not have optimal algebraic immunity but that it has algebraic immunity at least $\lfloor n/3 \rfloor + 1$. This value is

large enough for resisting the algebraic attack, because the function being fast to compute, it can be used with more variables than classical functions.

It seems very difficult to study mathematically the algebraic immunity of GHWB functions in general (we shall give below computer investigation results showing that their AI can be quite sufficient). We shall give in Corollary 2 below a way of addressing the algebraic immunity of GHWB functions. More (hard) work may be needed for determining the algebraic immunity more accuratey. To prove Corollary 2, we need the preliminary result given in Corollary 1, which has its own interest for studying algebraic immunity in general. It shows that if we know an upper bound d on the algebraic degree of a Boolean function, then the coefficients of the terms of degree d can be addressed not only by considering the values of the function at those x having some specific coordinates equal to 0 like in Relation (3), but also by considering those values at inputs having some coordinates equal to 0 and some equal to 1, with some freedom in the choice of the positions of these 0 and 1. We first need a lemma that is a straightforward consequence of Relation (3).

Lemma 2 *Let $h(x) = \sum_{I \subseteq \{1,\ldots,n\}} a_I\, x^I$ be any n-variable Boolean function. Let $\mathcal{I}$ be a set of multi-indices in $\{1,\ldots,n\}$. Then we have:*

$$\sum_{I \in \mathcal{I}} a_I = \sum_{\substack{x \in \mathbb{F}_2^n \\ |\{I \in \mathcal{I}; supp(x) \subseteq I\}| odd}} h(x).$$

Now, let D and K be subsets of $\{1,\ldots,n\}$ such that $D \subseteq K$ and let us take $\mathcal{I} = \{I \subseteq \{1,\ldots,n\}; D \subseteq I \subseteq K\}$. Then the set $\{I \in \mathcal{I}; supp(x) \subseteq I\} = \{I \subseteq \{1,\ldots,n\}; supp(x) \cup D \subseteq I \subseteq K\}$ has an odd size if and only it is a singleton (since its size equals zero or a power of 2), that is, $supp(x) \cup D = K$, or equivalently, $K \setminus D \subseteq supp(x) \subseteq K$, and we have then:

Corollary 1 *Let $h(x) = \sum_{I \subseteq \{1,\ldots,n\}} a_I\, x^I$ be any n-variable Boolean function. Let D be any subset of $\{1,\ldots,n\}$ such that there does not exist in the ANF of h a term x^I with $D \subsetneq I$ and $a_I \neq 0$. Then for every superset K of D, we have $a_D = \sum_{\substack{x \in \mathbb{F}_2^n \\ supp(x) \cup D = K}} h(x)$.*

Note that if we know that h has algebraic degree at most d, then every D of size d satisfies the hypothesis of Corollary 1. The condition $supp(x) \cup D = K$ means that all the coordinates of x whose indices lie outside K equal 0 and all those whose indices lie inside $K \setminus D$ equal 1. In other words, we have a partition of $\{1,\ldots,n\}$ into three sets $D, K \setminus D$ and $\{1,\ldots,n\} \setminus K$, where $|D| \leq d$, and every coordinate x_i of x:

1. is free if i belongs to D,
2. is fixed to 1 if i belongs to $K \setminus D$,
3. is fixed to 0 if i belongs to $\{1,\ldots,n\} \setminus K$.

We have a complete freedom in the choice of the superset K of D. This gives a tool for trying to show that any annihilator of degree bounded above by some number m is identically zero, that is, $a_D = 0$ when $|D| \leq m$: we assume that the annihilator is not zero, we call d its algebraic degree and we prove that $a_D = 0$ for every D of size d, which leads to a contradiction.

Corollary 2 *For every $n \geq 3$, let $m \leq \frac{n-1}{2}$ and let f be an n-variable Boolean function such that both f and $f+1$ are affine equivalent to a GHWB function f_{F_g} where $g \in \mathcal{E}$ is such that, for every $d \leq m$ and every subset D of $\{1, \ldots, n\}$ of size d, there exists a superset K of D such that the four following conditions are satisfied:*

1. *$[k-d, k] \subseteq K \setminus D$ where $k = |K|$,*
2. *g equals 1 on the set $\{x^{[w_H(x)]}; x \in \mathbb{F}_2^n; K \setminus D \subseteq supp(x) \subsetneq K\}$, where $x^{[w_H(x)]}$ denotes the vector equal to x deprived of its coordinate of index $w_H(x)$,*
3. *if $k+1 \in K$, then g maps to 1 the vector $z \in \mathbb{F}_2^{n-1}$ whose support equals $(K \cap [1,k]) \cup (\{i-1; i \in K \cap [k+2, n])$,*
4. *if $k+1 \notin K$, then either g maps to 0 the vector $z \in \mathbb{F}_2^{n-1}$ whose support equals $(K \cap [1,k]) \cup \{i-1; i \in K \cap [k+2, n]\}$ or g maps to 1 the vector $z \in \mathbb{F}_2^{n-1}$ whose support equals $(K \cap [1, k-1]) \cup \{i-1; i \in K \cap [k+2, n]\}$,*

or the two following conditions are satisfied:

i) $[k-d, k+1] \subseteq K^c$ where $k = |K|$,

ii) g equals 0 on the set $\{x^{[w_H(x)]}; x \in \mathbb{F}_2^n; K \setminus D \subseteq supp(x) \subseteq K\}$,

then we have $AI(f_{F_g}) \geq m+1$.

Proof. According to the fact that affine equivalence preserves algebraic immunity, it is enough to prove that any GHWB function satisfying the hypothesis has no nonzero annihilator of algebraic degree at most m. Let h be such annihilator of algebraic degree $d \leq m$ and let D be any subset of $\{1, \ldots, n\}$ of size d.
Let K satisfy Conditions 1-4. Any vector $x \in \mathbb{F}_2^n$ such that $supp(x) \cup D = K$ has Hamming weight between $k-d$ and k, since $K \setminus D \subseteq supp(x) \subseteq K$.
• If $w_H(x) < k$ then $w_H(x)$ and $w_H(x)+1$ belong to $[k-d, k]$ and then to $K \setminus D$, thanks to Condition 1, and therefore $x_{w_H(x)} = x_{w_H(x)+1} = 1$. Then x belongs to the support of f_{F_g} because, according to Condition 2, x is the image of the vector $z = x^{[w_H(x)]} = x^{[w_H(x)+1]}$ by F_g.
• If $w_H(x) = k$, then x has support K.
- If $k+1 \in K$ then x belongs to the support of f_{F_g}, according to Condition 3.
- If $k+1 \notin K$, then according to Condition 4, either $g(x^{[k+1]}) = 0$ and $x = F_g(x^{[k+1]})$ or $g(x^{[k]}) = 1$ and $x = F_g(x^{[k]})$; in both cases, x belongs to

the support of f_{F_g}. We have then $a_D = 0$.
Let now K satisfy Conditions (i)-(ii). Condition (i) ensures that $x_{w_H(x)} = x_{w_H(x)+1} = 0$ and Condition (ii) ensures then that x belongs to the support of f_{F_g} (note that "$x_{w_H(x)} = 0$ and $x_{w_H(x)+1} = 1$" would always imply that x does not belong to the support of f_{F_g}, whatever is g, and needs then to be avoided). We have then $a_D = 0$. This completes the proof. □

Remark If $m \leq \frac{\sqrt{4n+1}-1}{2} \approx \sqrt{n}$, then there always exists K satisfying Condition 1, since if there did not exist in $\{1, \ldots, n\}$ an interval $[k-d, k]$ disjoint from D, each of the $\left\lfloor \frac{n}{d} \right\rfloor$ disjoint intervals of this form existing in $\{1, \ldots, n\}$ would have a non-empty intersection with D, and the size of D would be at least $\left\lfloor \frac{n}{d} \right\rfloor > \frac{n}{d} - 1 \geq \frac{n}{m} - 1 \geq \frac{2n}{\sqrt{4n+1}-1} - 1 \geq \frac{\sqrt{4n+1}-1}{2}$, a contradiction). The condition for having (i) satisfied is similar. Corollary 2 is then of some help for selecting GHWB functions having a not very low AI but does not ensure a good AI. Further work is then needed. ◇

We give now the results of the computer investigation mentioned above.
The function $g(x) = x_3x_6x_7$ in 12 variables provides a function f_{F_g} in 13 variables with algebraic immunity 5 while the lower bound $\lfloor n/3 \rfloor + 1$ for the HWB function gives 5 too and the upper bound $\left\lceil \frac{n}{2} \right\rceil$ gives 7.
The function $x_2x_7x_8$ in 13 variables provides a function f_{F_g} in 14 variables with algebraic immunity 6 while the lower bound $\lfloor n/3 \rfloor + 1$ for the HWB function gives 5 and the upper bound $\left\lceil \frac{n}{2} \right\rceil$ gives 7. This is a good result.
The function $x_1x_7x_8$ in 14 variables provides a function f_{F_g} in 15 variables with algebraic immunity 6 and the modified function has an algebraic immunity of 7 while the lower bound $\lfloor n/3 \rfloor + 1$ for the HWB function gives 6 and the upper bound $\left\lceil \frac{n}{2} \right\rceil$ gives 8; the latter function is good.
The function $x_1x_8x_9$ in 15 variables provides a function f_{F_g} in 16 variables with algebraic immunity 7 and the modified function has an algebraic immunity of 7 while the lower bound $\lfloor n/3 \rfloor + 1$ for the HWB function gives 6 and the upper bound $\left\lceil \frac{n}{2} \right\rceil$ gives 8. This is a good result. ◇

Conclusion.

We have deduced from a construction (introduced in a previous paper) of n-variable Boolean functions f from vectorial $(n-1, n)$-functions F, a generalization of the hidden weight bit (HWB) function. This generalization is a secondary construction that builds an n-variable Boolean function from an $(n-1)$-variable Boolean function g, leading to functions complex enough to have good cryptographic features (and also being rather difficult to apprehend), but having an output very fast to compute. This provides a class of Boolean functions within which good trade-offs can be searched (as the Maiorana-McFarland class was, in a way, before the invention of algebraic attacks). We have determined the condition on g under which the resulting function F is injective (and function f is then balanced), we have shown the difficulty of characterizing all such functions f, but provided a fast way to compute their output. We have also studied their

representation and their cryptographic parameters (except their fast algebraic immunity, which will be studied in a further paper). More work is needed for evaluating the exact number of distinct balanced functions f_{F_g} but it is clear that this number is very large. A computer investigation with some particular choices of function g has provided functions in $n \in \{3, \ldots, 16\}$ variables with much better nonlinearities than the HWB function (which is known to be insufficiently nonlinear) and having good algebraic immunity. Only a tiny part of all possible functions g satisfying the condition could be visited and the best possible nonlinearities and algebraic immunities of all balanced functions f_{F_g} is probably still significantly larger. If such highly nonlinear functions can be found having good algebraic immunity and fast algebraic immunity, for choices of functions g with an output fast enough to compute, the resulting functions f_{F_g} will be by far the best possible candidates for being used in stream ciphers. Since the corpus to be investigated is too large, theoretical work is needed in the future for identifying good candidates to be investigated, but our results already show that generalized HWB function offer a good trade-off between speed (which is favored when for instance we take for g a monomial function) and security (which is favored with the functions obtained by the evolutionary approach, which have better nonlinearity but are also more complex). More work (probably very difficult) is needed for mathematically evaluating the nonlinearity, the algebraic immunity and the fast algebraic immunity of general functions f_{F_g}.

Acknowledgement We deeply thank Stjepan Picek for his great help with experiments. We are indebted to him. We also thank Sihem Mesnager for her interesting indications.

References

[1] R. E. Bryant, On the Complexity of VLSI Implementations and Graph Representations of Boolean Functions with Application to Integer Multiplication, IEEE Transactions on Computers 40 (2) (1991), 205–213. See page 2.

[2] P. Camion, C. Carlet, P. Charpin and N. Sendrier. On correlation-immune functions, *Proceedings of CRYPTO 1991, Lecture Notes in Computer Science* 576, pp. 86-100, 1991. See page 2.

[3] C. Carlet. Boolean Functions for Cryptography and Coding Theory. Monograph in *Cambridge University Press*, 562 pages, 2021. See pages 2, 4, 5, 9, and 12.

[4] C. Carlet. Parameterizing Boolean functions by vectorial functions and studying related constructions. Preprint, *IACR Cryptology ePrint Archive* (http://eprint.iacr.org/) 2021. See pages 3, 6, 7, 14, 16, and 22.

[5] C. Carlet and K. Feng. An infinite class of balanced functions with optimum algebraic immunity, good immunity to fast algebraic attacks and good nonlinearity. *Proceedings of ASIACRYPT 2008, Lecture Notes in Computer Science* 5350, pp. 425-440, 2008. See page 2.

[6] C. Carlet, D. Jakobovic and S. Picek. Evolutionary Algorithms-assisted Construction of Cryptographic Boolean Functions. *GECCO '21: Genetic and Evolutionary Computation Conference Proceedings*, 2021. See page 22.

[7] C. Carlet, P. Méaux and Y. Rotella. Boolean functions with restricted input and their robustness; application to the FLIP cipher. *IACR Transactions on Symmetric Cryptology* 2017 (3), pp. 192-227, 2017. See page 2.

[8] C. Carlet and S. Picek. On the practical limits of a generalization of the hidden weight bit function and of another construction of highly nonlinear functions. Preprint, 2021. See page 22.

[9] J. Liu and S. Mesnager. Weightwise perfectly balanced functions with high weightwise nonlinearity profile. *Designs, Codes and Cryptography* 87 (8), pp. 1797-1813, 2019. See page 2.

[10] M. Lobanov. Tight bound between nonlinearity and algebraic immunity. *IACR Cryptology ePrint Archive* (http://eprint.iacr.org/) 2005/441, 2005. See page 17.

[11] F. J. MacWilliams and N. J. Sloane. *The theory of error-correcting codes*, Amsterdam, North Holland. 1977. See pages 5 and 15.

[12] P. Méaux, C. Carlet, A. Journault and F.-X. Standaert. Improved Filter Permutators for Efficient FHE: Better Instances and Implementations. *Proceedings of Indocrypt 2019, Lecture Notes in Computer Science* 11898, pp. 68-91, 2019. See pages 2 and 15.

[13] P. Méaux, A. Journault, F.-X. Standaert and C. Carlet. Towards Stream Ciphers for Efficient FHE with Low-Noise Ciphertexts. *Proceedings of EUROCRYPT 2016, Lecture Notes in Computer Science* 9665, pp. 311-343, 2016. See pages 2 and 15.

[14] S. Mesnager. "Linear codes from functions", Chapter 20 in "A Concise Encyclopedia 1419 Coding Theory" CRC Press/Taylor and Francis Group (Publisher), London, New York, 2021 (94 pages). See page 2.

[15] Q. Wang, C. Carlet, P. Stănică and C. H. Tan. Cryptographic Properties of the Hidden Weighted Bit Function. *Discrete Applied Mathematics* 174, pp. 1-10, 2014. See pages 3, 9, 17, 18, 19, and 22.