A Class of Weightwise Almost Perfectly Balanced Boolean Functions with High Weightwise Nonlinearity*

Deepak Kumar Dalai[1] and Krishna Mallick[2]

[1]School of Mathematical Sciences,
[2]School of Computer Sciences,
National Institute of Science Education and Research,
An OCC of Homi Bhabha National Institute,
Bhubaneswar, Odisha 752050, India
Email: {deepak, krishna.mallick}@niser.ac.in

Abstract

A Boolean function with good cryptographic properties over a set of vectors with constant Hamming weight is significant for stream ciphers like FLIP [MJSC16]. This paper presents a construction weightwise almost perfectly balanced (WAPB) Boolean functions by perturbing the support vectors of a highly nonlinear function in the construction presented in [DM]. As a result, the nonlinearity and weightwise nonlinearities of the modified functions improve substantially.

Keywords: Boolean function, FLIP cipher, Weightwise perfectly balanced (WPB), Weightwise almost perfectly balanced (WAPB), Nonlinearity

1 Introduction

An n-variable Boolean function f is a mapping from the n-dimensional vector space $\mathbb{F}_2^n$ to $\mathbb{F}_2$, where $\mathbb{F}_2$ is a finite field with two elements $\{0, 1\}$. Depending upon the underlying algebraic structure, the '+' symbol is used for the addition operation in both $\mathbb{F}_2$ and $\mathbb{R}$. Boolean functions are crucial in the construction of nonlinear components in symmetric ciphers. In stream ciphers, Boolean functions are employed as filter functions to generate a pseudo-random sequence. All the cryptographic criteria have been established to analyze and construct Boolean functions for use in ciphers that can withstand various attacks. The cryptographic criteria for a filter function are generally defined over the entire vector space

*The extended abstract is presented in the 8th International Workshop on Boolean Functions and their Applications (BFA) 2023.

$\mathbb{F}_2^n$. The study of the Boolean functions over a restricted domain became interesting after the appearance of the FLIP cipher in 2016 [MJSC16]. The main idea of proposing the FLIP cipher is to combine a symmetric cipher with homomorphic encryption to improve the efficiency of computations through cloud-based services. The new stream cipher design involves using a bit permutation generator to permute the inputs before entering them into the filter function for every updating process. Therefore, the entire setup is known as a filter permutator. As a result, the Hamming weight (i.e., the number of non-zero coordinates) of the inputs to the filter function remains the same as the Hamming weight of the secret key. This restriction of the inputs significantly changes the viewpoint toward the security analysis of it. Cryptanalysis of the initial version of FLIP and some modifications in the filter function are presented in [DLR16]. However, the motivation to construct Boolean functions in the FLIP frame of reference arises. An initial cryptographic study of Boolean function in a restricted domain is introduced by Carlet et al. in [CMR17]. The mathematical introduction of the required parameters (i.e., balancedness, nonlinearity, and algebraic immunity) of a Boolean function in a restricted domain of $\mathbb{F}_2^n$ is presented in the paper. The set of vectors with constant Hamming weight in $\mathbb{F}_2^n$ is also called a "slice" in the hypercube $\{0,1\}^n$ in [FI19]. In this paper, Filmus and Ihringer studied the Boolean functions with constant degrees over the slice of the hypercube $\{0,1\}^n$.

Boolean functions used as filter functions in stream cipher are distributed uniformly over $\mathbb{F}_2^n$ and are called balanced Boolean functions. In the FLIP cipher, the Hamming weight of the key register is $\frac{n}{2}$, for n is even. Therefore, the keystream generated by FLIP should look like a random sequence over the set of vectors of Hamming weight $\frac{n}{2}$ for n is even, or we can say the filter function in FLIP should be balanced over the set $E_{n,\frac{n}{2}} = \{x \in \mathbb{F}_2^n | \mathtt{wt}(x) = \frac{n}{2}\}$. The Boolean functions balanced over the subsets of $\mathbb{F}_2^n$ containing vectors with constant Hamming weight are said to be weightwise perfectly balanced (WPB). In order for WPB Boolean functions to exist, n must be of the form 2^l for $l \in \mathbb{N}$. In [CMR17], the author extended the concept of WPB Boolean function for any $n \in \mathbb{N}$ and named these functions as weightwise almost perfectly balanced (WAPB) Boolean functions by allowing these functions to be almost balanced over the restricted domain, depending on the cardinality. The WPB and WAPB functions with good cryptographic criteria over the restricted domains are essential in the FLIP frame of reference. Several cryptographic criteria of a Boolean function over a restricted domain are studied in [CMR17]. An upper bound on the nonlinearity and algebraic immunity of a Boolean function on restricted inputs is presented in [CMR17]. The nonlinearity bound is further improved in [MZD19]. Following are the constructions of WPB/WAPB Boolean functions in literature:

1. The first weightwise perfectly balanced (WPB) Boolean function construction was introduced in [CMR17] in 2017 using the indirect sum of four Boolean functions. The construction is based on the direct sum of two WPB Boolean functions of 2^n- variable by modifying one of the WPB Boolean functions. A generalized result is presented by using four Boolean functions. The author also presented a recursive construction for weightwise almost perfectly balanced (WAPB) Boolean functions in the same paper. Upper bounds on the weightwise nonlinearities and weightwise algebraic immunity are presented in this paper.

2. Tang and Liu [TL19] proposed a construction of a class WAPB Boolean functions for an even number of variables, which satisfy optimal algebraic immunity. The authors also discussed that the WAPB function in their construction also has good weightwise algebraic immunity over some subsets of $\mathbb{F}_2^n$.

3. Liu and Mesnager [LM19] presented a class of WPB Boolean functions that are 2-rotation symmetric. These functions have the best weightwise nonlinearities and nonlinearity compared to the available constructions till now.

4. In 2020, Jingjing Li and Sihong Su in [LS20] constructed a class of WAPB Boolean functions of 2^{q+2} variables for $q \geq 1$. Then, they constructed a WPB Boolean function of 2^{q+2} variables by modifying the support of the WAPB Boolean function.

5. Several constructions of WPB and WAPB Boolean functions are presented in [MS21] by modifying the support of linear and quadratic functions.

6. In [ZS22], Linya Zhu and Sihong Su presented a method of constructing a WAPB Boolean function for an arbitrary number of variables using the direct sum of known WPB Boolean functions.

7. In [GM22a, GM22b, GM23b, GM23a], Agnese Gini and Pierrick Méaux have proved several results on WPB/WAPB Boolean functions. The authors have discussed the Hamming weight of linear functions restricted to the set of vectors with constant Hamming weights that can be expressed by Krawtchouk polynomials. Furthermore, an upper and lower bound on the nonlinearity of $f \in \mathcal{W}_n$, where $\mathcal{W}_n$ is the set of all WPB Boolean functions, have been studied in [GM23a].

There are many other constructions for WPB/WAPB Boolean functions presented in [Su21, ZS22, GS22, DM, ZLC, ZJZQ23]. Recently, such functions with high nonlinearity have been searched using genetic programming (GP) and genetic algorithm (GA) in [MPJ]. However, these functions may not be suitable for cryptographic implementation due to their structureless representation. Therefore, analyzing the cryptographic properties like nonlinearity, algebraic immunity, and efficiency of a WPB/WAPB Boolean function is vital from the perspective of ciphers like FLIP. Indeed, the upper bounds for nonlinearity and weightwise nonlinearity are not tight for such Boolean functions. Moreover, it is also significant to figure out the algebraic structure of these WPB/WAPB Boolean functions such that these functions with better trade-offs on cryptographic properties can be constructed from the known Boolean functions in lower dimensions.

1.1 Our Contribution

There are few works on the construction of WAPB Boolean functions available in the literature, whereas many works on the construction of WPB are found. In this paper, we have presented constructions of WAPB Boolean functions. At first, we improve the nonlinearity of the WAPB Boolean function proposed in [DM] by perturbing the support of a highly nonlinear function. The nonlinearity is improved by perturbing bits using bent functions.

1.2 Organisation

The research objectives and our contributions are already outlined. The remaining part of the paper is organized as follows:

i. In Section 2, we precisely define all the required definitions and notations of Boolean functions and their cryptographic properties. Furthermore, the definitions of WAPB and WPB Boolean functions and some previous construction of the WPB/ WAPB Boolean function is also discussed.

ii. Section 3 presents a class of WAPB Boolean function, which is a modification of the construction proposed in [DM]. We try to improve the nonlinearity and weightwise nonlinearity of the WAPB Boolean function using the support of highly nonlinear Boolean function. Finally, we show our experimental results by comparing the nonlinearities of the proposed functions with the known functions. Appendix A presents the algorithm used to compute the function.

iv. The paper is concluded with future scope in Section 4.

2 Preliminaries

Let $\mathbb{F}_2 = \{0, 1\}$ be a finite field with addition '+' and multiplication '.'. The multiplication $x.y$ is written as xy. We denote $[i, j] = \{i, i + 1, \ldots, j\}$ for two integers i, sj with $i \leq j$. An n-variable Boolean function is a mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. $\mathcal{B}_n$ is denoted as the set of all n-variable Boolean functions. For any $v = (v_1, v_2, \ldots, v_n) \in \mathbb{F}_2^n$, the Hamming weight of v is defined as $\mathtt{wt}(v) = |\{i \in [1, n] : v_i = 1\}|$. The support of a Boolean function $f \in \mathcal{B}_n$ is $\mathtt{sup}(f) = \{v \in \mathbb{F}_2^n : f(v) = 1\}$ and Hamming weight of f is $\mathtt{wt}(f) = |\mathtt{sup}(f)|$. Denote $E_{n,k} = \{v \in \mathbb{F}_2^n : \mathtt{wt}(v) = k\}$ for every $k \in [0, n]$. The support and Hamming weight of f restricted to $E_{n,k}$ are denoted as $\mathtt{sup}_k(f) = \{v \in E_{n,k} : f(v) = 1\}$ and $\mathtt{wt}_k(f) = |\mathtt{sup}_k(f)|$, respectively. The Hamming distance between two functions $f, g \in \mathcal{B}_n$ is given as $\mathtt{d}(f, g) = |\{v \in \mathbb{F}_2^n : f(v) \neq g(v)\}| = \mathtt{wt}(f + g)$ and the Hamming distance between two functions $f, g \in \mathcal{B}_n$ restricted to $E_{n,k}$ is given as $\mathtt{d}_k(f, g) = |\{v \in E_{n,k} : f(v) \neq g(v)\}| = \mathtt{wt}_k(f + g)$.

The truth table representation of a Boolean function $f \in \mathcal{B}_n$ is a 2^n-dimensional vector representation, i.e., $f = (f(0, 0, \ldots, 0), f(0, 0, \ldots, 1), \ldots, f(1, 1, \ldots, 1))$. The algebraic normal form (ANF) representation is defined as $f(x) = \sum_{u \in \mathbb{F}_2^n} a_u x^u$, where $a_u \in \mathbb{F}_2$ and $x^u = x_1^{u_1} x_2^{u_2} \cdots x_n^{u_n}$ for $x = (x_1, x_2, \ldots, x_n)$ and $u = (u_1, u_2, \ldots, u_n)$. The algebraic degree of a Boolean function $f \in \mathcal{B}_n$ is defined as $\deg(f) = \max\{\mathtt{wt}(u) : u \in \mathbb{F}_2^n, a_u \neq 0\}$. Any $f \in \mathcal{B}_n$ with $\deg(f) \leq 1$ is said to be an affine Boolean function, and the set of all affine Boolean functions in $\mathcal{B}_n$ is denoted by $\mathcal{A}_n$.

A Boolean function $f \in \mathcal{B}_n$ is balanced, if $\mathtt{wt}(f) = 2^{n-1}$. The Boolean function that is used for cryptographic algorithms necessarily be balanced to generate a random-looking sequence when the input goes through all the elements of $\mathbb{F}_2^n$. The nonlinearity of $f \in \mathcal{B}_n$ denoted as $\mathtt{nl}(f)$, is the minimum Hamming distance of f to any affine function. That is, $\mathtt{nl}(f) = \min_{g \in \mathcal{A}_n} \mathtt{d}(f, g)$. Similarly, all these cryptographic criteria are also defined for the n-variable Boolean function when the inputs are restricted to $E_{n,k}$.

Definition 2.1. *A Boolean function $f \in \mathcal{B}_n$ is said to be weightwise almost perfectly balanced (WAPB) if for all $k \in [0, n]$,*

$$\mathtt{wt}_k(f) = \begin{cases} \frac{\binom{n}{k}}{2} & \text{if } \binom{n}{k} \text{ is even,} \\ \frac{\binom{n}{k} \pm 1}{2} & \text{if } \binom{n}{k} \text{ is odd.} \end{cases}$$

For $k \in [0, n]$, $\delta_k^f \in \{-1, 0, 1\}$ is defined as $\delta_k^f = 2\mathtt{wt}_k(f) - \binom{n}{k}$. That is, $\mathtt{wt}_k(f) = \frac{1}{2}\left[\binom{n}{k} + \delta_k^f\right]$.

Hence, for any WAPB $f \in \mathcal{B}_n$,

$$\delta_k^f = \begin{cases} 0 & \text{if } \binom{n}{k} \text{ is even,} \\ -1 & \text{if } \mathtt{wt}_k(f) < \frac{\binom{n}{k}}{2}, \\ 1 & \text{if } \mathtt{wt}_k(f) > \frac{\binom{n}{k}}{2}. \end{cases}$$

For $x = (x_1, x_2, \ldots, x_n), y = (y_1, y_2, \ldots, y_n) \in \mathbb{F}_2^n$, we say y covers x (i.e., $x \preceq y$), if $x_i \leq y_i, \forall i \in [1, n]$ i.e., $y_i = 1$ if $x_i = 1, \forall i \in [1, n]$.

Proposition 2.2 (Lucas' Theorem)**.** *Let the binary representation of n and k be $(n_1, n_2, \ldots, n_l)$ and $(k_1, k_2, \ldots, k_l)$ respectively, where $n_i, k_i \in \{0, 1\}$ for $i \in [1, l]$, then*

$$\binom{n}{k} = \begin{cases} 1 (mod\, 2) & \text{if } k \preceq n \\ 0 (mod\, 2) & \text{if } k \not\preceq n. \end{cases}$$

Hence, it is straightforward from Proposition 2.2 that $\binom{n}{k}$ is even for all $k \in [1, n-1]$ iff $n = 2^m$ for a nonnegative integer m. Then we have the following special class of WAPB Boolean functions when $n = 2^m$ for a nonnegative integer m.

Definition 2.3. *A Boolean function $f \in \mathcal{B}_n$ is said to be weightwise perfectly balanced (WPB) if the restriction of f to $E_{n,k}$, is balanced for all $k \in [1, n-1]$, i.e., $\binom{n}{k}$ is even and $\mathtt{wt}_k(f) = \frac{\binom{n}{k}}{2}$ for all $k \in [1, n-1]$.*

Therefore, if $f \in \mathcal{B}_n$ is an WPB Boolean function, then $n = 2^m$ for a nonnegative integer m and $\delta_k^f = 0$ for all $k \in [1, n-1]$. A WPB Boolean function $f \in \mathcal{B}_n$ is balanced, if $f(0, 0, \ldots, 0) \neq f(1, 1, \ldots, 1)$. Hence, there are $2\prod_{k=1}^{n-1} \binom{\binom{n}{k}}{\binom{n}{k}/2}$ balanced WPB Boolean functions.

Definition 2.4. *Let $f \in \mathcal{B}_n$ be a Boolean function, then its Walsh transform W_f at $a \in \mathbb{F}_2^n$ is defined as:*

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

Let $f \in \mathcal{B}_n, E \subseteq \mathbb{F}_2^n$, then its Walsh transform restricted to E at $a \in \mathbb{F}_2^n$ is defined as

$$W_{f,E}(a) = \sum_{x \in E} (-1)^{f(x)+a.x}.$$

If $E = E_{n,k}$, we denote $W_{f,E}(a)$ as $W_{f,k}(a)$.

Therefore, $W_{f,k}(0) = \sum_{x \in E_{n,k}} (-1)^{f(x)} = \binom{n}{k} - 2\mathtt{wt}_k(f) = -\delta_k^f$.

Such kind of Boolean functions with good cryptographic criteria over $E_{n,k}$ are significant for the stream ciphers like FLIP. The nonlinearity and algebraic immunity of a Boolean function over a restricted domain is introduced in [CMR17].

Definition 2.5 (Weightwise nonlinearity)**.** *The nonlinearity of $f \in \mathcal{B}_n$ over $E_{n,k}$, denoted as $\mathtt{nl}_k(f)$, is the Hamming distance of f to the set of all affine functions $\mathcal{A}_n$ when evaluated over $E_{n,k}$. That is, $\mathtt{nl}_k(f) = \min_{g \in \mathcal{A}_n} d_k(f,g) = \min_{g \in \mathcal{A}_n} \mathtt{wt}_k(f+g)$.*

The following identity and upper bound on the nonlinearity of a Boolean function over $E_{n,k}$ can be derived. The upper bound is further improved by Mesnager et al. in [MZD19].

Lemma 2.6. *[CMR17] If $f \in B_n$ then*

$$\mathtt{nl}_k(f) = \frac{|E_{n,k}|}{2} - \frac{1}{2} \max_{a \in \mathbb{F}_2^n} \left| \sum_{x \in E_{n,k}} (-1)^{f(x)+a.x} \right| \text{ and}$$

$$\mathtt{nl}_k(f) \leq \frac{1}{2}[|E_{n,k}| - \sqrt{|E_{n,k}|}]$$

for $k \in [0,n]$ (where $|E_{n,k}| = \binom{n}{k}$).

Definition 2.7. *Given $f \in \mathcal{B}_n$, a nonzero $g \in \mathcal{B}_n$ is called an annihilator of f if $f.g = 0$, i.e., $f(x)g(x) = 0$ for all $x \in \mathbb{F}_2^n$. For $E \subseteq \mathbb{F}_2^n$, a function $g \in \mathcal{B}_n$ is called an annihilator of f over E if $g(x) \neq 0$ for some $x \in E$ and $f(x)g(x) = 0$ for all $x \in E$. The set of all annihilators of $f \in \mathcal{B}_n$ is denoted by $An(f)$ and the set of all annihilators of f over E is denoted by $An_E(f)$. The algebraic immunity of $f \in \mathcal{B}_n$ is defined as*

$$\mathtt{AI}(f) = \min\{\deg(g) : g \in An(f) \cup An(1+f)\}.$$

For $E \subseteq \mathbb{F}_2^n$, the algebraic immunity of f over E is defined by

$$\mathtt{AI}_E(f) = \min\{\deg(g) : g \in An_E(f) \cup An_E(1+f)\}.$$

For $E = E_{n,k}$, we denote $An_E(f)$ and $\mathtt{AI}_E(f)$ as $An_k(f)$ and $\mathtt{AI}_k(f)$ respectively.

Note 2.8. *For $f \in \mathcal{B}_n$ and $E \subseteq \mathbb{F}_2^n$, if $g \in An_E(f)$ then $g \neq 0$ over E. This implies that an annihilator of f is not necessarily an annihilator of f on E. That is, $An(f) \not\subseteq An_E(f)$ and hence $\mathtt{AI}_E(f) \not\leq \mathtt{AI}(f)$ for any $f \in \mathcal{B}_n$ and $E \subseteq \mathbb{F}_2^n$.*

The following propositions present some of the constructions of WPB and WAPB Boolean functions which are basis of our construction presented in Section 3 . Let $\triangle$ be the symbol represents the symmetric diffrence between two sets.

Proposition 2.9. *[MS21] For a positive integer $n = 2^m$, let $f_n \in \mathcal{B}_n$ with support*

$$\begin{aligned} \mathtt{sup}(f_n) &= \triangle_{i=1}^{m} \{(x,y,x,y,\ldots,x,y) \in \mathbb{F}_2^n : x,y \in \mathbb{F}_2^{2^{m-i}}, \mathtt{wt}(x) \text{ is odd}\}. \\ &= \begin{cases} \{(1,y) : y \in \mathbb{F}_2\} & \text{if } n = 2, \\ \{(x,y) : x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}\} \triangle \{(x,x) : x \in \mathtt{sup}(f_{\frac{n}{2}})\} & \text{if } n > 2. \end{cases} \end{aligned}$$

Then, f_n is a WPB Boolean function.

Corollary 2.10. *[MS21] The ANF of the Boolean function f_n proposed in Proposition 2.9 is*

$$f_n(x_1, x_2, \ldots, x_n) = \begin{cases} x_1 & \text{if } n = 2, \\ \sum_{i=1}^{\frac{n}{2}} x_i + f_{\frac{n}{2}}(x_1, x_2, \ldots, x_{\frac{n}{2}}) \prod_{i=1}^{\frac{n}{2}} (x_i + x_{\frac{n}{2}+i} + 1) & \text{if } n > 2. \end{cases}$$

Proposition 2.11. *[DM] For $n \geq 2$, let $f_n \in \mathcal{B}_n$ with support*

$$\mathtt{sup}(f_n) = \begin{cases} \{(x,1) \in \mathbb{F}_2^2 : x \in \mathbb{F}_2\} = \{(0,1),(1,1)\} & \text{if } n = 2, \\ \{(x,0) \in \mathbb{F}_2^n : x \in \mathtt{sup}(f_{n-1})\} \cup \{(x,1) \in \mathbb{F}_2^n : x \notin \mathtt{sup}(f_{n-1})\} & \text{if } n > 2 \text{ and odd}, \\ \{(x,y) \in \mathbb{F}_2^n : x, y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}\} \triangle \{(z,z) \in \mathbb{F}_2^n : z \in \mathtt{sup}(f_{\frac{n}{2}})\}, & \text{if } n > 2 \text{ and even}. \end{cases}$$

Then, f_n is a WAPB Boolean function.

The construction proposed in Proposition 2.11 is a generalization of the construction proposed in Proposition 2.9 to get WAPB Boolean functions on n variables. The construction proposed in Proposition 2.11 is important for our study as we will provide a construction that improves its nonlinearity by using highly nonlinear Boolean function.

Theorem 2.12. *[DM] Let $f_n \in \mathcal{B}_n$ ($n > 2$), defined as in Proposition 2.11. Then $\mathtt{nl}(f_n) = 2\mathtt{nl}(f_{n-1})$ if n is odd and $\mathtt{nl}(f_n) \leq \mathtt{wt}(f_{\frac{n}{2}})$ if n is even.*

For n even, the nonlinearity of f_n (in Proposition 2.11) is very low as $X_1 = \{(x,y) \in \mathbb{F}_2^n : x, y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}\}$ is the support of a linear function $\sum_{i=1}^{\frac{n}{2}} x_i$ and the cardinality of $X_2 = \{(z,z) \in \mathbb{F}_2^n : z \in \mathtt{sup}(f_{\frac{n}{2}})\}$ is $\mathtt{wt}(f_{\frac{n}{2}})$. Further, for n even and k odd, $\mathtt{sup}_k(f_n) = \mathtt{sup}(f_n) \cap E_{n,k} = \{(x,y) \in \mathbb{F}_2^n : x, y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}\} \cap E_{n,k} = \mathtt{sup}_k(\sum_{i=1}^{\frac{n}{2}} x_i)$ and hence $\mathtt{nl}_k(f_n) = 0$. Therefore, in our technique (in Section 3), we attempt to permute the coordinates of the vectors of weight k in X_1 to improve the nonlinearity by avoiding the linear patterns and preserving the weightwise balancedness.

3 A class of WAPB Boolean functions with good nonlinearity

In this section, we will present a class of WAPB Boolean functions by modifying $\mathtt{sup}(f_n)$ presented in Proposition 2.11. We observed (see the last paragraph of Section 2) that the nonlinearity becomes weak because the $\mathtt{sup}(f_n)$ when n is even is close to a linear function. In our technique, we attempt to increase the nonlinearity by permuting the coordinates of some support vectors in $\mathtt{sup}(f_n)$ when n is even.
Therefore, it is assumed that $n > 2$ and is **even** in this section. Hence, when n is even, as Proposition 2.11, $\mathtt{sup}(f_n) = \{(x,y) \in \mathbb{F}_2^n : x, y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}\} \triangle \{(z,z) \in \mathbb{F}_2^n : z \in$

$\mathtt{sup}(f_{\frac{n}{2}})\}$. Then

$$\mathtt{sup}_k(f_n) = \begin{cases} \{(x,y) \in \mathbb{F}_2^n : x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}, \mathtt{wt}(x)+\mathtt{wt}(y)=k\} \\ \qquad \triangle\{(z,z) \in \mathbb{F}_2^n : z \in \mathtt{sup}_{\frac{k}{2}}(f_{\frac{n}{2}})\} & \text{if } k \text{ is even} \\ \{(x,y) \in \mathbb{F}_2^n : x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}, \mathtt{wt}(x)+\mathtt{wt}(y)=k\} & \text{if } k \text{ is odd} \end{cases}$$

Now we will consider both cases of k (i.e., odd and even) and will propose to permute the coordinates of some vectors in $\mathtt{sup}_k(f_n)$.

3.1 When k is odd

In this case, $\mathtt{sup}_k(f_n) = \{(x,y) \in \mathbb{F}_2^n : x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}, \mathtt{wt}(x)+\mathtt{wt}(y)=k\} = \mathtt{sup}_k(l(x,y))$ where $l(x,y) = \sum_{i=1}^{\frac{n}{2}} x_i$, as we discussed at the end of Section 2. The linear function $l(x,y) = \sum_{i=1}^{\frac{n}{2}} x_i$ is independent of y. We attempt to break the independence and linearity on the coordinates in y using the support of a nonlinear function $u \in \mathcal{B}_{\frac{n}{2}}$. That is, for every $x \in \mathbb{F}_2^{\frac{n}{2}}$ satisfying $l(x,y)$ (i.e., $\mathtt{wt}(x)$ is odd), we keep (x,y) if $y \in \mathrm{sup}(u)$ otherwise we replace (x,y) by (y,x). If u is a highly nonlinear function, then the component y is expected to be far from the linear functions and as a result, we have a high nonlinearity in f.

Here, if $\mathtt{wt}(x,y) = k$ then $\mathtt{wt}((y,x)) = k$. Further, if $(x,y) \in \mathrm{sup}_k(f_n)$ then $\mathtt{wt}(y)$ is even as $\mathtt{wt}(x)$ is odd. So, $(y,x) \notin \mathrm{sup}_k(f_n)$ if $(x,y) \in \mathrm{sup}_k(f_n)$. Therefore, the replacement of $(x,y) \in \mathrm{sup}_k(f_n)$ by (y,x) does not change the weight of the resultant function in the domain $E_{n,k}$.

Lemma 3.1. *Let $u \in \mathcal{B}_{\frac{n}{2}}$. A function $f \in \mathcal{B}_n$ such that for every $k \in [0,n]$ and odd,*

$$\begin{aligned} \mathtt{sup}_k(f^u) &= \{(x,y) \in \mathbb{F}_2^n : x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}, \mathtt{wt}(y) = k - \mathtt{wt}(x), y \in \mathtt{sup}(u)\} \\ &\cup \{(y,x) \in \mathbb{F}_2^n : x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}, \mathtt{wt}(y) = k - \mathtt{wt}(x), y \notin \mathtt{sup}(u)\}. \quad (1) \end{aligned}$$

Then $\mathtt{wt}_k(f^u) = \frac{1}{2}\binom{n}{k}$.

Proof. Let $A = \{(x,y) \in \mathbb{F}_2^n | x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}, \mathtt{wt}(x)+\mathtt{wt}(y)=k\}$. Hence,

$$|A| = \sum_{\substack{i=1 \\ i \text{ is odd}}}^{k} \binom{\frac{n}{2}}{i}\binom{\frac{n}{2}}{k-i} = \frac{1}{2}\binom{n}{k}$$

For any $u \in \mathcal{B}_{\frac{n}{2}}$, we have $A = A_1 \cup A_2$ where

$$\begin{aligned} A_1 &= \{(x,y) \in \mathbb{F}_2^n : x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}, \mathtt{wt}(x)+\mathtt{wt}(y)=k, y \in \mathtt{sup}(u)\} \text{ and} \\ A_2 &= \{(x,y) \in \mathbb{F}_2^n : x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}, \mathtt{wt}(x)+\mathtt{wt}(y)=k, y \notin \mathtt{sup}(u)\}. \end{aligned}$$

So, $A_1 \cap A_2 = \emptyset$. Further denote,

$$A_2^s = \{(y,x) \in \mathbb{F}_2^n : x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}, \mathtt{wt}(y) = k - \mathtt{wt}(x), y \notin \mathtt{sup}(u)\}.$$

Here, $\mathtt{wt}(y)$ is even in A as $\mathtt{wt}(x)$ and k are odd. So, $|A_2^s| = |A_2|$ and $A \cap A_2^s = \emptyset$. As $\mathrm{sup}_k(f^u) = (A \setminus A_2) \cup A_2^s$, $\mathtt{wt}_k(f^u) = |(A \setminus A_2) \cup A_2^s| = |A| - |A_2| + |A_2^s| = |A| = \frac{1}{2}\binom{n}{k}$. □

3.2 When k is even

In this case, $\mathtt{sup}_k(f_n) = \{(x,y) \in \mathbb{F}_2^n : x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}, \mathtt{wt}(x)+\mathtt{wt}(y) = k\} \triangle \{(z,z) \in \mathbb{F}_2^n : z \in \mathtt{sup}_{\frac{k}{2}}(f_{\frac{n}{2}})\}$. Let denote the set $L = \{(x,y) \in \mathbb{F}_2^n : x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}, \mathtt{wt}(x) + \mathtt{wt}(y) = k\}$ and $M = \{(z,z) \in \mathbb{F}_2^n : z \in \mathtt{sup}_{\frac{k}{2}}(f_{\frac{n}{2}})\}$. In this case, the replacement of $(x,y) \in \mathrm{sup}_k(f_n)$ by (y,x) is not straight forward as in Subsection 3.1. If $(x,y) \in L$ then $\mathtt{wt}(y)$ is odd as $\mathtt{wt}(x)$ is odd. As a result, (y,x) could be present in L. Therefore, replacement of $(x,y) \in \mathrm{sup}_k(f_n)$ by (y,x) can possibly duplicate an existing vector in L, which reduces the weight of the resultant function. Therefore, we attempt to swap two bits of x and y instead of swapping x and y as in the following lemma. For given $(x,y) \in \mathbb{F}_2^n$ where $x = (x_1,\ldots,x_{\frac{n}{2}}), y = (y_1,\ldots,y_{\frac{n}{2}}) \in \mathbb{F}_2^{\frac{n}{2}}$, we denote $(x^i,y^i) = (x_1,\ldots,x_{i-1},y_i,x_{i+1},\ldots,x_{\frac{n}{2}},y_1,\ldots,y_{i-1},x_i,y_{i+1},\ldots,y_{\frac{n}{2}})$. That is, (x^i,y^i) is obtained by swapping the i-th bits of x and y.

Lemma 3.2. *Let $f_n \in \mathcal{B}_n$ be the function defined in Proposition 2.11. For every $k \in [0,n]$ and even, let $W_k = \{(x,y) \in \mathtt{sup}_k(f_n) |\ \mathtt{wt}(x)$ is odd, and there is an $i \in [1,\frac{n}{2}]$ such that $x_j = y_j$ for $1 \le j \le i-1$ and $x_i = 0, y_i = 1\}$ and*
$W_k' = \{(x^i,y^i) |\ (x,y) \in W_k$ and $i \in [1,\frac{n}{2}]$ such that $x_j = y_j$ for $1 \le j \le i-1$ and $x_i = 0, y_i = 1$ i.e., the i obtained for (x,y) in $W_k\}$.
A function $g_n \in \mathcal{B}_n$ such that for $k \in [0,n]$ and even, such that $\mathtt{sup}_k(g_n) = (\mathtt{sup}_k(f_n) \setminus W_k) \cup W_k'$ for every $k \in [0,n]$ and even.
Then $\mathtt{wt}_k(g_n) = \mathtt{wt}_k(f_n)$ if k is even.

Proof. From Proposition 2.11, $\mathtt{sup}_k(f_n) = \{(x,y) \in \mathbb{F}_2^n : x,y \in \mathbb{F}_2^{\frac{n}{2}}, \mathtt{wt}(x) \text{ is odd}, \mathtt{wt}(x) + \mathtt{wt}(y) = k\} \triangle \{(z,z) \in \mathbb{F}_2^n : z \in \mathtt{sup}_{\frac{k}{2}}(f_{\frac{n}{2}})\}$ for $k \in [0,n]$ and even. Here, the weight of each vector in W_k' is k and $|W_k| = |W_k'|$. As k is even, $\mathtt{wt}(x)$ and $\mathtt{wt}(y)$ are odd for every $(x,y) \in W_k$. That implies, $\mathtt{wt}(x^i)$ and $\mathtt{wt}(y^i)$ are even for every $(x^i,y^i) \in W_k'$. Hence, $W_k \cap W_k' = \emptyset$. Further, $x^i \neq y^i$ for every $(x^i,y^i) \in W_k'$ as i-th bit in x and y are different. Hence, $\mathtt{sup}_k(f_n) \cap W_k' = \emptyset$. Hence, $\mathtt{wt}_k(g_n) = \mathtt{wt}_k(f_n) - |W_k| + |W_k'| = \mathtt{wt}_k(f_n)$. □

Like in Lemma 3.1, now we will use the support of another Boolean function (possibly, a highly nonlinear) to swap x^i and y^i in some of $(x^i,y^i) \in W_k'$ as defined in Lemma 3.2.

Lemma 3.3. *Let $v \in \mathcal{B}_{\frac{n}{2}}$. Let $g_n \in \mathcal{B}_n$ as defined in Lemma 3.2 with W_k and W_k'. A function $h_n^v \in \mathcal{B}_n$ such that for every $k \in [0,n]$ and even,*
$\mathtt{sup}_k(h_n^v) = \{(x,y) \in \mathtt{sup}_k(g_n) : (x,y) \notin W_k'\} \cup \{(x,y) : (x,y) \in W_k'$, and $y \in \mathtt{sup}(v)\} \cup \{(y,x) : (x,y) \in W_k'$ and $y \notin \mathtt{sup}(v)\}$.
Then $\mathtt{wt}_k(h_n^v) = \mathtt{wt}_k(g_n)$.

Proof. Denote $\mathtt{sup}_k(h_n^v) = H_k \cup W_k^v \cup W_k^{\overline{v}}$, where $H_k = \{(x,y) \in \mathtt{sup}_k(g_n) : (x,y) \notin W_k'\}, W_k^v = \{(x,y) : (x,y) \in W_k',$ and $y \in \mathtt{sup}(v)\}$ and $W_k^{\overline{v}} = \{(y,x) : (x,y) \in W_k'$ and $y \notin \mathtt{sup}(v)\}$.

From Lemma 3.2, we have $\mathtt{sup}_k(g_n) = (\mathtt{sup}_k(f_n) \setminus W_k) \cup W_k'$. Here, $\mathtt{wt}(x)$ and $\mathtt{wt}(y)$ are even for every $(x,y) \in W_k'$. Since, $x_i \neq y_i$ in $(x,y) \in W_k'$, $(x,y) \neq (y,x)$ for each $(x,y) \in W_k'$. For any $v \in \mathcal{B}_{\frac{n}{2}}$, W_k' can be partitioned as

$$W_k' = \{(x,y) \in W_k' | y \in \mathtt{sup}(v)\} \cup \{(x,y) \in W_k' | y \notin \mathtt{sup}(v)\}.$$

Then $|\{(x,y)|(x,y) \in W_k^{'} \text{ and } y \notin \mathtt{sup}(v)\}| = |\{(y,x)|(x,y) \in W_k^{'} \text{ and } y \notin \mathtt{sup}(v)\}| = |W_k^{\overline{v}}|$. From the definition of $W_k^{'}$, for every $(x,y) \in W_k^{'}$ there is an $i \in [1, \frac{n}{2}]$ such that $x_j = y_j$ for $1 \leq j \leq i-1$ and $y_i = 0, x_i = 1$. Hence, $W_k^{'} \cap \{(y,x)|(x,y) \in W_k^{'} \text{ and } y \notin \mathtt{sup}(v)\} = W_k^{'} \cap W_k^{\overline{v}} = \emptyset$.

Further, as $\mathtt{wt}(x), \mathtt{wt}(y)$ are odd for every $(x,y) \in H_k$ and $\mathtt{wt}(x), \mathtt{wt}(y)$ are even for every $(x,y) \in W_k^{'} \cup W_k^{\overline{v}}$, $H_k \cap W_k^{\overline{v}} = \emptyset$. Hence $\mathtt{wt}_k(h_n^v) = |H_k| + |W_k^v| + |W_k^{\overline{v}}| = (\mathtt{wt}_k(g_n) - |W_k^{'}|) + |W_k^{'}| = \mathtt{wt}_k(g_n)$. □

3.3 A class of WAPB Boolean function

Now we will apply Lemma 3.1 and Lemma 3.3 to construct a WAPB Boolean function with improved nonlinearity.

Theorem 3.4. *Let $u, v \in \mathcal{B}_{\frac{n}{2}}$. Let $f_n \in \mathcal{B}_n$ be the function defined in Proposition 2.11. Let $F_n \in \mathcal{B}_n$ with support $\mathtt{sup}_k(F_n) = \begin{cases} \mathtt{sup}_k(h_n^v) & \text{if } k \text{ is even} \\ \mathtt{sup}_k(f_n^u) & \text{if } k \text{ is odd,} \end{cases}$ where f_n^u, h_n^v are as defined in Lemma 3.1 and Lemma 3.3 respectively. Then F_n is a WAPB Boolean function.*

The following is a recursive construction of a WAPB Boolean function.

Construction 3.5. *For $n \geq 2$, let $F_n \in \mathcal{B}_n$ with support*

$$\mathtt{sup}(F_n) = \begin{cases} \{(x,1) \in \mathbb{F}_2^2 : x \in \mathbb{F}_2\} = \{(0,1),(1,1)\} & \text{if } n = 2, \\ \{(x,0) \in \mathbb{F}_2^n : x \in \mathtt{sup}(F_{n-1})\} \cup \{(x,1) \in \mathbb{F}_2^n : x \notin \mathtt{sup}(F_{n-1})\} & \text{if } n > 2 \text{ and odd}, \\ S_n \triangle \{(z,z) \in \mathbb{F}_2^n : z \in \mathtt{sup}(F_{\frac{n}{2}})\} & \text{if } n > 2 \text{ and even.} \end{cases}$$

Here $S_n = \cup_{k=0}^n \mathtt{sup}_k(F_n)$ and $\mathtt{sup}_k(F_n) = \begin{cases} \mathtt{sup}_k(h_n^v) & \text{if } n > 2 \text{ and even and } k \text{ is even} \\ \mathtt{sup}_k(f_n^u) & \text{if } n > 2 \text{ and even and } k \text{ is odd.} \end{cases}$

The algorithm of computing $F_n(x), x \in \mathbb{F}_2^n$ is presented in Appendix A. The time complexity of computing $F_n(x)$ for $x \in \mathbb{F}_2^n$ is $O(n \max\{O(u(\frac{n}{2}), v(\frac{n}{2}))\})$. If the chosen functions u and v are easily computable, then computation would be very fast. If u and v are quadratic bent function as taken in Section 3.4, the time complexity would be $O(n^2)$. Such efficient functions with good cryptographic properties can be used for implementation of ciphers for lightweight cryptography.

3.4 Experimental results on nonlinearity

In this section, we have presented experimental results on the nonlinearity ($\mathtt{nl}(F_n)$) and weightwise nonlinearity ($\mathtt{nl}_k(F_n)$) of F_n. We have chosen $u, v \in \mathcal{B}_{\frac{n}{2}}$, a highly nonlinear function

$$u(y) = v(y) = \begin{cases} y_1y_2 + \cdots + y_{\frac{n}{2}-1}y_{\frac{n}{2}} & \text{if } \frac{n}{2} \text{ is even} \\ y_1y_2 + \cdots + y_{\frac{n}{2}-2}y_{\frac{n}{2}-1} + y_{\frac{n}{2}} & \text{if } \frac{n}{2} \text{ is odd.} \end{cases}$$

This function is a bent function when $\frac{n}{2}$ is even and concatenation of two bent functions

when $\frac{n}{2}$ is odd. Further, these two functions are easy to compute which is helpful for implementation in light weight cryptography. Table 1 presents the nonlinearity and weightwise nonlinearity of the functions F_n for $n = 8, 9, \ldots, 16$, which are generated using Construction 3.5.

n	$\mathtt{nl}$	$\mathtt{nl}_2$	$\mathtt{nl}_3$	$\mathtt{nl}_4$	$\mathtt{nl}_5$	$\mathtt{nl}_6$	$\mathtt{nl}_7$	$\mathtt{nl}_8$	$\mathtt{nl}_9$	$\mathtt{nl}_{10}$	$\mathtt{nl}_{11}$	$\mathtt{nl}_{12}$	$\mathtt{nl}_{13}$	$\mathtt{nl}_{14}$	$\sum_{k=0}^{n} \mathtt{nl}_k$
8	96	4	16	20	16	4	0	0	-	-	-	-	-	-	60
9	192	6	22	45	45	22	6	0	0	-	-	-	-	-	146
10	416	9	36	69	94	73	12	9	0	0	-	-	-	-	302
11	832	11	50	113	163	173	117	34	11	0	0	-	-	-	672
12	1596	12	36	146	264	286	264	148	36	14	0	0	-	-	1206
13	3192	15	69	219	507	660	660	495	240	69	17	0	0	-	2951
14	6904	19	102	336	764	1083	1484	1079	654	299	30	18	0	0	5868
15	13808	22	147	474	1155	2013	2735	2670	1965	1154	465	75	22	0	12897
16	28152	24	64	564	1216	2547	5036	4610	5036	2919	1216	516	64	24	23836

Table 1: Listing of $\mathtt{nl}(F_n), \mathtt{nl}_k(F_n)$ and $\sum_{k=0}^{n} \mathtt{nl}_k(F_n)$ for $8 \leq n \leq 16$.

We have presented a comparison of weightwise nonlinearities of F_n with the upper bound presented in [CMR17] in Table 2. Further, no upper bound is available for the nonlinearity of WAPB Boolean functions. Therefore, we have presented a comparison of the nonlinearity of F_n with the upper bound of the nonlinearity of n variable Boolean functions [dH97].

n	*function*	$\mathtt{nl}$	$\mathtt{nl}_2$	$\mathtt{nl}_3$	$\mathtt{nl}_4$	$\mathtt{nl}_5$	$\mathtt{nl}_6$	$\mathtt{nl}_7$	$\mathtt{nl}_8$	$\mathtt{nl}_9$	$\mathtt{nl}_{10}$	$\mathtt{nl}_{11}$	$\sum_{k=0}^{n} \mathtt{nl}_k$
8	*UB*	120	11	24	30	24	11	-	-	-	-	-	100
	F_8	96	4	16	20	16	4	-	-	-	-	-	60
9	*UB*	244	15	37	57	57	37	15	-	-	-	-	218
	F_9	192	6	22	45	45	22	6	-	-	-	-	146
10	*UB*	496	19	54	97	118	97	54	19	-	-	-	498
	F_{10}	416	9	36	69	94	73	12	9	-	-	-	302
11	*UB*	1000	23	76	155	220	220	155	76	23	-	-	948
	F_{11}	832	11	50	113	163	173	117	34	11	-	-	672
12	*UB*	2016	28	102	236	381	446	381	236	102	28	-	1940
	F_{12}	1596	12	36	146	264	286	264	148	36	14	-	1206
13	*UB*	4050	34	134	344	625	837	837	625	344	134	34	3948
	F_{13}	3192	15	69	219	507	660	660	495	240	69	17	2951

Table 2: Comparison of $\mathtt{nl}_k(F_n)$ with the upper bound(UB) presented in [CMR17]

A comparison of the nonlinearities of our result with some recent constructions for $n = 8$ are presented in Table 3.

4 Conclusion

We have presented constructing a class of WAPB Boolean functions in n variables from the idea of constructions presented in [MS21, DM]. The experimental results on nonlinearity and

WPB/ WAPB functions	$\mathtt{nl}$	$\mathtt{nl}_2$	$\mathtt{nl}_3$	$\mathtt{nl}_4$	$\mathtt{nl}_5$	$\mathtt{nl}_6$
Upper Bound [CMR17]		11	24	30	24	11
[CMR17]	88	2	12	19	12	6
[LM19]	96,108	6,9	0,8,14,16, 18,20, 21,22	19,22,23,24, 25,26,27	19,20,21,22	6,9
[TL19]	88, 90	6,8	8	20, 22, 24	8	6,7
[LS20, $g_{2^{q+2}}$ Equation(9)]		2	12	19	12	2
[MS21, f_m Equation(13)]		2	0	3	0	2
[MS21, g_m Equation(22)]		2	14	19	14	2
[MSL21, f_m Equation(2)]		2	8	8	8	2
[MSL21, f_m Equation(3)]		6	8	26	8	6
[GM22b, Table 1]		5,3,2,2	10,7,12,12	16,15,18,19	12,11,12,12	5,3,2,6
[GM22b, Table 3]		5	16	20	17	5
[ZS23, g_m Equation(11)]		2	12	19	12	6
[GM23a]		6,6,7	19,14,15	21,20,18	11,11,14	3,6,6
[ZJZQ23]		6	17	23	17	6
F_8 [DM]	82	7	13	14	14	7
F_8 [Construction 3.5]	96	4	16	20	16	4

Table 3: Comparison of $\mathtt{nl}_k$ of 8-variable WPB constructions.

weightwise nonlinearities show a good improvement. For future work, we are studying the cryptographic properties of this class of WAPB functions and attempting to further improve the nonlinearities and weightwise nonlinearities by modifying this class of functions.

References

[CMR17] Claude Carlet, Pierrick Méaux, and Yann Rotella. Boolean functions with restricted input and their robustness; application to the FLIP cipher. *IACR Trans. Symmetric Cryptol.*, 2017(3):192–227, 2017.

[dH97] Xiang dong Hou. On the norm and covering radius of the first-order Reed-Muller codes. *IEEE Transactions on Information Theory*, 43(3):1025–1027, 1997.

[DLR16] Sébastien Duval, Virginie Lallemand, and Yann Rotella. Cryptanalysis of the FLIP family of stream ciphers. In *Advances in Cryptology - CRYPTO 2016*, volume 9814 of *Lecture Notes in Computer Science*, pages 457–475. Springer, 2016.

[DM] Deepak Kumar Dalai and Krishna Mallick. A class of weightwise almost perfectly balanced Boolean functions. *Advances in Mathematics of Communications*, 18(2): 480–504, 2024.

[FI19] Yuval Filmus and Ferdinand Ihringer. Boolean constant degree functions on the slice are juntas. *Discret. Math.*, 342(12), 2019.

[GM22a] Agnese Gini and Pierrick Méaux. On the weightwise nonlinearity of weightwise perfectly balanced functions. *Discrete Applied Mathematics*, 322:320–341, 2022.

[GM22b] Agnese Gini and Pierrick Méaux. Weightwise almost perfectly balanced functions: Secondary constructions for all n and better weightwise nonlinearities. In *Progress in Cryptology - INDOCRYPT 2022*, volume 13774 of *Lecture Notes in Computer Science*, pages 492–514. Springer, 2022.

[GM23a] Agnese Gini and Pierrick Méaux. On the algebraic immunity of weightwise perfectly balanced functions. *IACR Cryptol. ePrint Arch.*, page 495, 2023.

[GM23b] Agnese Gini and Pierrick Méaux. Weightwise perfectly balanced functions and nonlinearity. In *Codes, Cryptology and Information Security - C2SI 2023*, volume 13874 of *Lecture Notes in Computer Science*, pages 338–359. Springer, 2023.

[Gou72] H.W. Gould. *Combinatorial Identities: A Standardized Set of Tables Listing 500 Binomial Coefficient Summations*. Gould, 1972.

[GS22] Xiaoqi Guo and Sihong Su. Construction of weightwise almost perfectly balanced Boolean functions on an arbitrary number of variables. *Discrete Applied Mathematics*, 307:102–114, 2022.

[LM19] Jian Liu and Sihem Mesnager. Weightwise perfectly balanced functions with high weightwise nonlinearity profile. *Designs, Codes and Cryptography*, 87(8):1797–1813, 2019.

[LS20] Jingjing Li and Sihong Su. Construction of weightwise perfectly balanced Boolean functions with high weightwise nonlinearity. *Discrete Applied Mathematics*, 279:218–227, 2020.

[MJSC16] Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet. Towards stream ciphers for efficient FHE with low-noise ciphertexts. In *Advances in Cryptology - EUROCRYPT 2016*, volume 9665 of *Lecture Notes in Computer Science*, pages 311–343. Springer, 2016.

[MPJ] Luca Mariot, Stjepan Picek, Domagoj Jakobovic, Marko Djurasevic, and Alberto Leporati. Evolutionary construction of perfectly balanced boolean functions. In *IEEE Congress on Evolutionary Computation, CEC 2022, Padua, Italy, July 18-23, 2022*, pages 1–8. IEEE, 2022.

[MS21] Sihem Mesnager and Sihong Su. On constructions of weightwise perfectly balanced Boolean functions. *Cryptography and Communications*, 13(6):951–979, 2021.

[MSL21] Sihem Mesnager, Sihong Su, and Jingjing Li. On concrete constructions of weightwise perfectly balanced functions with optimal algebraic immunity and high weightwise nonlinearity. In *The 6th International Workshop on Boolean Functions and Applications*, 2021.

[MZD19] Sihem Mesnager, Zhengchun Zhou, and Cunsheng Ding. On the nonlinearity of Boolean functions with restricted input. *Cryptography and Communications*, 11(1):63–76, 2019.

[Su21] Sihong Su. The lower bound of the weightwise nonlinearity profile of a class of weightwise perfectly balanced functions. *Discrete Applied Mathematics*, 297:60–70, 2021.

[TL19] Deng Tang and Jian Liu. A family of weightwise (almost) perfectly balanced Boolean functions with optimal algebraic immunity. *Cryptography and Communications*, 11(6):1185–1197, 2019.

[ZJZQ23] Qinglan Zhao, Yu Jia, Dong Zheng, and Baodong Qin. A new construction of weightwise perfectly balanced functions with high weightwise nonlinearity. *Mathematics*, 11(5):1193, 2023.

[ZLC] Qinglan Zhao, Mengran Li, Zhixiong Chen, Baodong Qin, and Dong Zheng. A unified construction of weightwise perfectly balanced boolean functions. *Discret. Appl. Math.*, 337:190–201, 2023.

[ZS22] Linya Zhu and Sihong Su. A systematic method of constructing weightwise almost perfectly balanced Boolean functions on an arbitrary number of variables. *Discrete Applied Mathematics*, 314:181–190, 2022.

[ZS23] Rui Zhang and Sihong Su. A new construction of weightwise perfectly balanced Boolean functions. *Advances in Mathematics of Communications*, 17(4):757–770, 2023.

A Algorithm of Computing $F_n(x)$ in Construction 3.5

Algorithm 1: Output of $F_n(x)$

Input: n; $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$; $u, v \in \mathcal{B}_{\frac{n}{2}}$
Output: $F_n(x)$

if n *is odd* **then**
  $z := (x_1, x_2, \ldots, x_{n-1})$;
  if $x_n = 0$ **then return** $F_{n-1}(z)$;
  else return $1 + F_{n-1}(z)$;
end
else
  $X_{[1,\frac{n}{2}]} := (x_1, x_2, \ldots, x_{\frac{n}{2}})$; $X_{[\frac{n}{2}+1,n]} := (x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n)$;
  $k := wt(x)$; $k_1 := wt(X_{[1,\frac{n}{2}]})$; $k_2 := wt(X_{[\frac{n}{2}+1,n]})$;
  if k *is odd* **then**
    if k_1 *is odd* **then return** $u(X_{[\frac{n}{2}+1,n]})$;
    else return $1 + v(X_{[1,\frac{n}{2}]})$;
  end
  else
    if $X_{[1,\frac{n}{2}]} = X_{[\frac{n}{2}+1,n]}$ **then**
      if k_1 *is even* **then return** $F_{\frac{n}{2}}(X_{[1,\frac{n}{2}]})$;
      else return $1 + F_{\frac{n}{2}}(X_{[1,\frac{n}{2}]})$;
    end
    else
      $i := 1$;
      while $x_i = x_{\frac{n}{2}+i}$ **do** $i++$;
      if k_1 *is even* **then**
        if $x_i > x_{\frac{n}{2}+i}$ **then return** $v(X_{[\frac{n}{2}+1,n]})$;
        else return $v(X_{[1,\frac{n}{2}]})$;
      end
      else
        if $x_i > x_{\frac{n}{2}+i}$ **then return** *1*;
        else return *0*;
      end
    end
  end
end