Plateaued Rotation Symmetric Boolean Functions on Odd Number of Variables

Alexander Maximov[1], Martin Hell[1], Subhamoy Maitra[2]

[1] Department of Information Technology, Lund University
P.O. Box 118, 221 00 Lund, Sweden
{movax, martin}@it.lth.se
[2] Applied Statistics Unit, Indian Statistical Institute,
203, B T Road, Kolkata 700 108, INDIA
subho@isical.ac.in

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

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

1 Introduction

While designing cryptographically significant Boolean functions, many requirements have to be fulfilled, such as balancedness, nonlinearity, algebraic degree, correlation immunity, resistance from algebraic attacks etc. Some of them may contradict each other, e.g., bent functions, which have highest possible nonlinearity, can not be balanced. Getting the best possible trade-off among these parameters has always been a challenging task as evident from literature (see [10,11,13] and the references in these papers). The class of Rotation symmetric Boolean functions (RSBFs) is a class of functions that are invariant under circular translation of indices. It has been shown that many functions in this class are rich in terms of cryptographic properties [2, 5, 12, 13]. Further the RSBF class is much smaller ($\approx 2^{\frac{2^n}{n}}$) compared to the space of n-variable Boolean functions (2^{2^n}) and hence search techniques work much better in this smaller class. Given Boolean functions on even number of input variables, the best possible nonlinearity can be achieved when the magnitude of all the Walsh spectra values are same. However, this is not possible when the number of input variables are odd. In such a scenario, the functions with three valued Walsh spectra $0, \pm\lambda$ may be investigated [1,15], which are known as plateaued functions. It has been noted that there are functions with very good cryptographic properties in this class [1,15].

In [13], two data structures, the matrices ${}_n\mathcal{A}$ and ${}_n\mathcal{B}$, were presented that made the search for RSBFs more efficient. The matrix ${}_n\mathcal{B}$ is used for fast generation of the truth table from its algebraic normal form, and ${}_n\mathcal{A}$ is used for fast calculation of the Walsh transform for the RSBF. In this paper we investigate the matrix ${}_n\mathcal{A}$ in detail. We introduce a new matrix, ${}_n\mathcal{H}$, which is a sub matrix of ${}_n\mathcal{A}$, for *odd* n, after some permutation. This allows us to improve the calculation of the Walsh transform for RSBFs and provides much better combinatorial insight to the problem. Our matrix structure can be used to make a concrete study on plateaued RSBFs on odd number of variables and we could provide necessary conditions on existence of balanced plateaued RSBFs. The construction of 9-variable, 3-resilient Boolean function with nonlinearity 240 is still an unsolved open question in literature [10,11]. In [13] an estimate to search such functions in rotation symmetric class has been presented which needed search of 2^{43} many Boolean functions and could not be completed in [13]. Since such functions are plateaued functions, we apply our results to theoretically show the nonexistence of 9-variable, 3-resilient, nonlinearity 240 functions in the rotation symmetric class. Further, using the matrix ${}_n\mathcal{H}$, we found efficient search strategies for plateaued RSBFs which is much faster than what presented in [13]. We also use efficient implementation strategy in software to make the search faster. As an example of our search efficiency we exhaustively searched for unbalanced 9-variable, 3rd order correlation immune, algebraic degree 5 and nonlinearity 240 RSBFs and found $2 \cdot 8406$ many such functions. The search took only 6064 seconds against the estimated time of 3 years[1] as presented in [13].

2 Preliminaries

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

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

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

[1] Note that, we have attempted to make the search (as explained in [13]) faster using efficient software implementation and found that it is possible to implement optimized code that can search the complete space in 470 hours using a Pentium M 1.6 GHz computer with 512 MB RAM. We have also parallelized the effort over a few computers and searched the complete space as explained in [6].

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

Any Boolean function has a unique representation as a polynomial over F_2, called the *algebraic normal form* (ANF),

$$f(x_1, \ldots, x_n) = a_0 \oplus \bigoplus_{1 \leq i \leq n} a_i x_i \oplus \bigoplus_{1 \leq i < j \leq n} a_{ij} x_i x_j \oplus \ldots \oplus a_{12\ldots n} x_1 x_2 \ldots x_n,$$

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

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

Boolean functions used in ciphers must have high nonlinearity to prevent linear attacks [4, 7].

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

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

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

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

Following the same notation as in [10, 11, 13] we use (n, m, d, σ) to denote an n-variable, m-resilient function with degree d and nonlinearity σ. Further, by $[n, m, d, \sigma]$ we denote an unbalanced n-variable, mth order correlation immune function with degree d and nonlinearity σ.

2.1 Rotation Symmetric Boolean Functions

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

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

Let $(x_1, x_2, \ldots, x_{n-1}, x_n) \in V_n$. Then we extend the definition as

$\rho_n^k(x_1, x_2, \ldots, x_{n-1}, x_n) = (\rho_n^k(x_1), \rho_n^k(x_2), \ldots, \rho_n^k(x_{n-1}), \rho_n^k(x_n))$. Hence, ρ_n^k acts as k cyclic rotation on an n-bit vector.

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

The inputs to a rotation symmetric Boolean function can be divided into partitions so that each partition consists of all cyclic shifts of one input. A partition is generated by $G_n(x_1, x_2, \ldots, x_n) = \{\rho_n^k(x_1, x_2, \ldots, x_n) | 1 \leq k \leq n\}$ and the number of such partitions is denoted by g_n. Thus the number of n-variable RSBFs is 2^{g_n}. Let $\phi(k)$ be Euler's *phi*-function, then it can be shown by Burnside's lemma that (see also [12])

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

By $g_{n,w}$ we denote the number of partitions with weight w. It can also be checked that the number of degree w RSBFs is $(2^{g_{n,w}} - 1)2^{\sum_{i=0}^{w-1} g_{n,i}}$. For the formula of how to calculate $g_{n,w}$ for arbitrary n and w, we refer to [12].

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

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

In [13] it was shown that the Walsh transform takes the same value for all elements belonging to the same group, i.e., $W_f(u) = W_f(v)$ if $u \in G_n(v)$.

In [13], two matrices were introduced, ${}_n\mathcal{A}$ and ${}_n\mathcal{B}$, for efficient search of RSBFs. The matrix ${}_n\mathcal{A}$ is defined as

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

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

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

The notation of ρ_n^k can be extended, in a similar fashion, to monomials. For example, if we have a 4 variable rotation symmetric Boolean function and the term $x_1x_2x_3$ is present in the ANF, then the terms $x_2x_3x_4, x_3x_4x_1$ and $x_4x_1x_2$ must also be present in the ANF. We can associate n-bit pattern $(x_1, x_2, \ldots, x_n)$ of $\Lambda_{n,i}$ with a monomial as well. If there is a '1' in the corresponding position we say that the variable is present in the monomial. Considering this, the $g_n \times g_n$ matrix ${}_n\mathcal{B}$ is defined as [13]

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

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

Note that the ANF of the RSBFs are such that if one monomial from a rotational symmetric group is present in the ANF then all the other monomials of that rotational symmetric group are also present [5,13]. Thus the algebraic normal from of any RSBF possesses a very nice and regular form. The algebraic attack (see [3,8] and the references in these papers) is getting a lot of attention recently. To resist algebraic attacks, the Boolean functions used in the cryptosystems should be chosen properly. It is shown [3] that given any n-variable Boolean function f, it is always possible to get a Boolean function g with degree at most $\lceil \frac{n}{2} \rceil$ such that $f * g$ is of degree at most $\lceil \frac{n}{2} \rceil$. Here the functions are considered to be multivariate polynomials over GF(2) and $f * g$ is the polynomial multiplication. Thus while choosing an f, the cryptosystem designer should be careful that it should not happen that degree of $f * g < \lceil \frac{n}{2} \rceil$ where g is also a low degree function. There is no known result of weakness on cryptographically significant RSBFs yet and we believe that given the algebraic structure of the RSBFs, they will be resistant against the algebraic attacks if the parameters are chosen properly. Though we are not studying this aspect in this paper, we think this could be an important research problem and this gives a good motivation to study the RSBFs for other cryptographic properties.

3 Walsh Spectra of RSBFs

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

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

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

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

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

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

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

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

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

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

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

Let us consider $n = 5$, for which $g_n = 8$. In [13], the group representatives are ordered lexicographically, i.e., $(0,0,0,0,0)$, $(0,0,0,0,1)$, $(0,0,0,1,1)$, $(0,0,1,0,1)$, $(0,0,1,1,1)$, $(0,1,0,1,1)$, $(0,1,1,1,1)$, $(1,1,1,1,1)$. We get the matrix ${}_5\mathcal{A}$. On the other hand if we permute them as $(0,0,0,0,0)$, $(0,0,0,1,1)$, $(0,0,1,0,1)$, $(0,1,1,1,1)$, $(1,1,1,1,1)$, $(0,0,1,1,1)$, $(0,1,0,1,1)$, $(0,0,0,0,1)$, i.e., even weight elements and then the corresponding odd weight elements, we get the matrix ${}_5\mathcal{A}^\pi$ which is of a nice sub matrix structure.

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

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

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

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

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

Theorem 2. *When n is odd, the matrix ${}_n\mathcal{A}^\pi$ is of the form*

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

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

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

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

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

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

Proof. The first column ${}_n\mathcal{A}_{i,0}$ is constructed as ${}_n\mathcal{A}_{i,0} = \sum_{x \in G_n(\Lambda_{n,i})} (-1)^{x \cdot \mathbf{0}} = |G_n(\Lambda_{n,i})|$, since we know that there are $d_{n,t}$ groups with $|G_n(\Lambda_{n,i})| = t$, the first part of the corollary follows.

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

Remark 1. In Subsection 2.1 we defined the RSTT of an RSBF as the g_n-bit string $[f(\Lambda_{n,0}), f(\Lambda_{n,1}), \ldots, f(\Lambda_{n,g_{n-1}})]$, where $\Lambda_{n,0}, \Lambda_{n,1}, \ldots, \Lambda_{n,g_{n-1}}$ are ordered lexicographically. Given Theorem 2, from now on, we consider the RSTT$^\pi$, where we first list the representative elements $\lambda_{n,i}$ with even weights in lexicographical order for $i = 0$ to $\frac{g_n}{2} - 1$. Then we put the elements in the order such that $\Lambda_{n,i} = \overline{\Lambda}_{n,i-\frac{g_n}{2}}$ for $i = \frac{g_n}{2}$ to $g_n - 1$. In the rest of the document, we will use only this ordering (permutation) and by abuse of notations, apply (RSTT, RSTT$^\pi$) and $({}_n\mathcal{A}, {}_n\mathcal{A}^\pi)$ as same thing unless specifically mentioned.

3.2 Improved Walsh Transform Computation

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

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

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

Then we can define

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

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

3.3 Plateaued RSBFs

A Boolean function on odd number of variables is said to be plateaued [1, 15] if its Walsh transform takes only the three values 0 and $\pm\lambda$, where λ is some positive integer. We call λ the *amplitude* of the function.

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

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

There are only 9 valid pairs (w_{1_i}, w_{2_i}) fulfilling (3) and they are listed in Table 1. This means that w_{1_i} and $w_{2_i} \in \{0, \pm\lambda/2, \pm\lambda\}$, i.e., they can only take 5 values. The partition of the matrix ${}_n\mathcal{A}^\pi$ as in Theorem 2 and Table 1 give us the following result.

Proposition 2. *Consider an RSBF on odd number of variables represented by the RSTT* $(\sigma_1 \parallel \sigma_2)$.

1. *If it is plateaued then the functions with RSTT* $(\sigma_2 \parallel \sigma_1)$, $(\overline{\sigma_1} \parallel \overline{\sigma_2})$, $(\overline{\sigma_2} \parallel \overline{\sigma_1})$, $(\sigma_1 \parallel \overline{\sigma_2})$, $(\overline{\sigma_2} \parallel \sigma_1)$, $(\overline{\sigma_1} \parallel \sigma_2)$ *and* $(\sigma_2 \parallel \overline{\sigma_1})$ *are also plateaued.*
2. *If it is correlation immune (respectively resilient) then the functions with RSTT* $(\sigma_2 \parallel \sigma_1)$, $(\overline{\sigma_1} \parallel \overline{\sigma_2})$, *and* $(\overline{\sigma_2} \parallel \overline{\sigma_1})$ *are also correlation immune (respectively resilient).*

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

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

3.4 Necessary condition for balanced plateaued RSBFs

Based on the above discussion, we now present concrete results on necessary conditions for existence of balanced plateaued RSBFs.

Theorem 3. *For n odd, if there exist an n-variable balanced plateaued RSBF with amplitude $\lambda = 2^k$, then the following condition must be satisfied:*

There exist $k'_1 \ldots k'_n$ and $k''_1 \ldots k''_n$, $k^*_i \in [0 \ldots \frac{d_{n,i}}{2}]$, and $\tau \in \{0,1\}$, such that $\sum_{t=1}^{n} t \cdot k'_t = \frac{\tau\lambda + 2^n}{4}, \sum_{t=1}^{n} t \cdot k''_t = \frac{-\tau\lambda+2^n}{4}$.

Proof. If the function $(\sigma_1 \parallel \sigma_2)$ is balanced, then from Table 1, the partial Walsh transform (pWT) for the first column must be $\{0, \frac{\pm\lambda}{2}\}$, i.e., $(\sigma_1^* \cdot {}_n\mathcal{H})[0] = \tau \cdot \frac{\lambda}{2}, (\sigma_2^* \cdot {}_n\mathcal{H})[0] = -\tau \cdot \frac{\lambda}{2}$, for $\tau \in \{0,1\}$. In the first column there are $\frac{d_{n,t}}{2}$ groups of size t. Let for k'_t of them σ_1^* get (+1), and for the rest $(\frac{d_{n,t}}{2} - k'_t)$ it will be (-1). Then pWT for the first column is expressed as $(\sigma_1^* \cdot {}_n\mathcal{H})[0] = \sum_{t=1}^{n}[t \cdot k'_t - (\frac{d_{n,t}}{2} - k'_t) \cdot t] = \tau \cdot \frac{\lambda}{2} \Rightarrow 2\sum_{t=1}^{n} k'_t \cdot t = \frac{\tau\lambda}{2} + \sum_{t=1}^{n} t \cdot \frac{d_{n,t}}{2} \Rightarrow \sum_{t=1}^{n} t \cdot k'_t = \frac{\tau\lambda+2^n}{4}$, for $k'_t = [0 \ldots \frac{d_{n,t}}{2}]$.

The similar expression for $(\sigma_2^* \cdot {}_n\mathcal{H})[0]$ is $\sum_{t=1}^{n} t \cdot k''_t = \frac{-\tau\lambda+2^n}{4}$, for $k''_t = [0 \ldots \frac{d_{n,t}}{2}]$. □

Now we present the result for non existence of $(9,3,5,240)$ RSBF, which has been posed as an open question in [13].

Theorem 4. *A $(9,3,5,240)$ RSBF can not exist.*

Proof. Note that this function is plateaued [11]. Thus we analyze 9-variable balanced plateaued functions for $\lambda = 2^5$ and for this we need to study the ${}_9\mathcal{H}$ matrix. Since $2^9 = 512$, for a balanced function to exist, it must be that $1 \cdot k_1 + 3 \cdot k_3 + 9 \cdot k_9 = \frac{\pm\tau \cdot 2^5 + 512}{4}$ following Theorem 3 and we get the only solution $k'_1 = 1, k'_3 = 0, k'_9 = 15$ and $k''_1 = 0, k''_3 = 1, k''_9 = 13$ for $\tau = 1$.

Let us now consider the value of $W_f(011011011)$, which must be any one of $0, \pm 32$. Let $W_f(011011011) = w_{1_i} + w_{2_i}$. From Table 1 we get that w_{1_i}, w_{2_i} can take values $0, \pm 16$. To get the exact values of w_{1_i}, w_{2_i} one needs to look at the last but one column of the matrix ${}_9\mathcal{H}$. The matrix is as follows.

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
9 & 5 & 1 & 1 & 5 & 1 & 1 & 1 & 1 & 1 & -3 & -3 & 1 & -3 & 1 & 5 & -3 & -3 & -3 & -7 & 1 & 1 & 1 & 1 & 1 & -3 & -3 & -3 & -3 & 5 \\
9 & 1 & 5 & 1 & 1 & 1 & 1 & -3 & 1 & -3 & 1 & 1 & -7 & 1 & -3 & 1 & -3 & -3 & -3 & 5 & 1 & -3 & -3 & -3 & 1 & 5 & 1 & 1 & -3 & 5 \\
9 & 1 & 1 & 5 & -3 & 1 & -3 & 1 & -3 & -3 & -3 & 1 & -3 & -3 & -3 & -3 & 5 & 5 & 1 & -3 & 1 & 1 & -3 & 1 & 1 & -3 & 1 & 1 & 9 & 5 \\
9 & 5 & 1 & -3 & 1 & -7 & -3 & 1 & -3 & -3 & 1 & -3 & 5 & 1 & -3 & 1 & 1 & 1 & 5 & 5 & 1 & 1 & -3 & 1 & 1 & 1 & -3 & -3 & -3 & 1 \\
9 & 1 & 1 & 1 & -7 & 5 & -3 & -3 & -3 & 1 & 1 & -3 & 5 & 1 & 1 & -3 & -3 & -3 & 1 & 1 & -3 & 1 & 5 & 1 & -3 & 1 & 1 & 1 & -3 & 5 \\
9 & 1 & 1 & -3 & -3 & -3 & 1 & -3 & 1 & -3 & -3 & 5 & 1 & 5 & 5 & 1 & 1 & 1 & 1 & -3 & 5 & 1 & 1 & -7 & -3 & -3 & 1 & 1 & -3 & 1 \\
9 & 1 & -3 & 1 & 1 & -3 & -3 & 1 & -3 & 5 & 1 & -3 & -3 & 1 & 5 & 5 & 1 & 1 & -3 & 5 & -3 & -3 & 1 & -3 & -3 & -3 & 1 & 1 & 9 & 1 \\
9 & 1 & 1 & -3 & -3 & -3 & 1 & -3 & 1 & 5 & 5 & 5 & 1 & -3 & -3 & 1 & 1 & 1 & 1 & -3 & -3 & -7 & 1 & 1 & 5 & -3 & 1 & 1 & -3 & 1 \\
9 & 1 & -3 & -3 & -3 & 1 & -3 & 5 & 5 & 1 & 5 & 1 & -3 & -3 & 1 & -3 & 1 & 1 & -3 & 1 & 1 & 5 & 1 & -3 & 1 & 1 & 1 & -7 & -3 & 1 \\
9 & -3 & 1 & -3 & 1 & 1 & -3 & 1 & 5 & 5 & 1 & -3 & -3 & 1 & -3 & 1 & 1 & 1 & 5 & -3 & 1 & 1 & -3 & 1 & -7 & 1 & -3 & 5 & -3 & 1 \\
9 & -3 & 1 & 1 & -3 & -3 & 5 & -3 & 5 & 1 & -3 & 1 & 5 & -3 & 1 & 1 & 1 & 1 & -3 & 1 & -3 & 1 & -3 & 1 & -3 & 5 & -3 & -3 & 9 & 1 \\
9 & 1 & -7 & -3 & 5 & 5 & 1 & -3 & 1 & -3 & -3 & 5 & 1 & -3 & -3 & 1 & 1 & 1 & 1 & 5 & -3 & 1 & 1 & 1 & -3 & -3 & 1 & 1 & -3 & 1 \\
9 & -3 & 1 & -3 & 1 & 1 & 5 & 1 & -3 & -3 & 1 & -3 & -3 & 1 & 5 & 1 & 1 & 1 & 5 & -3 & -7 & 1 & -3 & 1 & 1 & 1 & 5 & -3 & -3 & 1 \\
9 & 1 & -3 & -3 & -3 & 1 & 5 & 5 & -3 & 1 & -3 & 1 & -3 & 5 & 1 & -3 & 1 & 1 & -3 & 1 & 1 & -3 & 1 & 5 & 1 & 1 & -7 & 1 & -3 & 1 \\
9 & 5 & 1 & -3 & 1 & -3 & 1 & 5 & 1 & -3 & 1 & 1 & 1 & 1 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & 1 & 1 & 1 & -3 & 1 & 5 & 5 & 9 & -3 \\
9 & -3 & -3 & 5 & 1 & -3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -3 & -7 & 1 & 1 & 1 & 5 & -3 & -3 & 5 & -3 & -3 & 5 & -3 & -3 & 1 \\
9 & -3 & -3 & 5 & 1 & -3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -3 & 1 & -7 & 1 & 1 & -3 & 5 & -3 & -3 & 5 & -3 & -3 & 5 & -3 & 1 \\
9 & -3 & -3 & 1 & 5 & 1 & 1 & -3 & 1 & -3 & 5 & -3 & 1 & 5 & -3 & -3 & 1 & 1 & 1 & -3 & 1 & -3 & 5 & -3 & 1 & 1 & -3 & -3 & 9 & 1 \\
9 & -7 & 5 & -3 & 5 & 1 & -3 & 5 & -3 & 1 & -3 & 1 & 5 & -3 & 1 & -3 & 1 & 1 & -3 & 1 & 1 & -3 & 1 & -3 & 1 & 1 & 1 & 1 & -3 & 1 \\
9 & 1 & 1 & 1 & 1 & -3 & 5 & -3 & -3 & 1 & 1 & -3 & -3 & -7 & 1 & -3 & 5 & -3 & 1 & 1 & 5 & 1 & 5 & 1 & -3 & 1 & 1 & 1 & -3 & -3 \\
9 & 1 & -3 & 1 & 1 & 1 & 1 & -3 & -7 & 5 & 1 & 1 & 1 & 1 & -3 & 1 & -3 & 5 & -3 & -3 & 1 & 5 & -3 & -3 & 1 & 5 & 1 & 1 & -3 & -3 \\
9 & 1 & -3 & -3 & -3 & 5 & 1 & 1 & 1 & 1 & -3 & -3 & 1 & -3 & 1 & 1 & -3 & -3 & 5 & 1 & 5 & -3 & -3 & -3 & 5 & 1 & 1 & 1 & 9 & -3 \\
9 & 1 & -3 & 1 & 1 & 1 & -7 & -3 & 1 & -3 & 1 & 1 & 1 & 1 & 5 & 1 & 5 & -3 & -3 & -3 & 1 & -3 & -3 & 5 & 1 & 5 & 1 & 1 & -3 & -3 \\
9 & 1 & 1 & 1 & 1 & -3 & -3 & -3 & 5 & 1 & -7 & -3 & -3 & 1 & 1 & -3 & -3 & 5 & 1 & 1 & -3 & 1 & 5 & 1 & 5 & 1 & 1 & 1 & -3 & -3 \\
9 & -3 & 5 & -3 & 1 & 1 & -3 & -3 & -3 & 1 & 1 & 5 & -3 & 1 & 1 & 1 & -3 & -3 & 1 & 1 & 1 & 5 & 1 & 5 & 1 & -3 & -3 & -3 & 9 & -3 \\
9 & -3 & 1 & 1 & -3 & 1 & 1 & 1 & 1 & 1 & -3 & -3 & 1 & 5 & -7 & 5 & 5 & -3 & -3 & 1 & 1 & 1 & 1 & 1 & 1 & -3 & 5 & -3 & -3 & -3 \\
9 & -3 & 1 & 1 & -3 & 1 & 1 & 1 & 1 & -7 & 5 & -3 & 1 & -3 & 1 & 5 & -3 & 5 & -3 & 1 & 1 & 1 & 1 & 1 & 1 & -3 & -3 & 5 & -3 & -3 \\
3 & -1 & -1 & 3 & -1 & -1 & -1 & 3 & -1 & -1 & -1 & 3 & -1 & -1 & -1 & 3 & -1 & -1 & 3 & -1 & -1 & -1 & 3 & -1 & -1 & 3 & -1 & -1 & -1 & -1 \\
9 & 5 & 5 & 5 & 1 & 5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -3 & 1 & 1 & 1 & 1 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -7
\end{pmatrix}$$

Let us represent the last but one column of the matrix ${}_9\mathcal{H}$ as a column matrix C. Thus $w_{1_i} = \sigma_1^* C$ and $w_{2_i} = \sigma_2^* C$, where σ_1^*, σ_2^* are as given in Subsection 3.2. The values k'_1, k''_1 correspond to the top most element of C, which is 1 and the values k'_3, k''_3 correspond to the last but one element of C, which is -1. The values of k'_9, k''_9 correspond to the other 28 values in the column matrix C, where twenty one many values are -3 and seven many values are 9. Let $k'_9 = a' + b'$ and $k''_9 = a'' + b''$, where a', a'' correspond to the values -3 and b', b'' correspond to the values 9. Now $w_{1_i} = \sigma_1^* C = 1 \times 1 + (-1) \times (-1) + (2a' - 21) \times (-3) + (2b' - 7) \times 9 = 2 - 6a' + 18b'$. Also, we have $a' + b' = k'_9 = 15$. Thus the only possible solution is $a' = 12, b' = 3$ and in that case $w_{1_i} = -16$. Similarly, $w_{2_i} = \sigma_2^* C = (-1) \times 1 + 1 \times (-1) + (2a'' - 21) \times (-3) + (2b'' - 7) \times 9 = -2 - 6a'' + 18b''$. Also, we have $a'' + b'' = k''_9 = 13$. Thus the only possible solution is $a'' = 9, b'' = 4$ and in that case $w_{2_i} = 16$. Hence $W_f(011011011) = w_{1_i} + w_{2_i} = 0$. From Theorem 2, it follows that if $W_f(011011011) = w_{1_i} + w_{2_i}$ then $W_f(001001001) = w_{1_i} - w_{2_i}$. Thus, $W_f(001001001) = -32 \neq 0$. Hence, from definition, the function can not be 3-resilient. This proves that there can not be any $(9, 3, 5, 240)$ RSBF. $\Box$

We have checked the necessary condition is satisfied for $\lambda = 2^{\frac{n+1}{2}}$ for odd composite $n = 15, 21$ and 25. For $n = 15$, the solutions are $k'_1 = 1, k'_3 = 0, k'_5 = 1, k'_{15} = 550$ and $k''_1 = 0, k''_3 = 1, k''_5 = 2, k''_{15} = 541$ when $\tau = 1$. For $n = 21$, the solutions are $k'_1 = 0, k'_3 = 1, k'_7 = 1, k'_{21} = 24990$ or $k'_1 = 0, k'_3 = 1, k'_7 = 4, k'_{21} = 24989$ or $k'_1 = 0, k'_3 = 1, k'_7 = 7, k'_{21} = 24988$ and $k''_1 = 1, k''_3 = 0, k''_7 = 2, k''_{21} = 24941$ or $k''_1 = 1, k''_3 = 0, k''_7 = 5, k''_{21} = 24940$ or $k''_1 = 1, k''_3 = 0, k''_7 = 8, k''_{21} = 24939$ when $\tau = 1$. For $n = 25$, the solutions are $k'_1 = 1, k'_5 = 1, k'_{25} = 335626$ and $k''_1 = 0, k''_5 = 2, k''_{25} = 335462$ when $\tau = 1$. Note that there is no solution with $\tau = 0$. It will be interesting to find out some general solution pattern for odd composite n's from the necessary condition of Theorem 3, which is done for odd prime n's in Corollary 3 below. Further we need to study the other columns of the matrix ${}_n\mathcal{H}$ as in the proof of Theorem 4 if we like to prove the non existence results for these cases.

Corollary 3. *For a balanced plateaued RSBF on $n \geq 3$ variables, n prime, τ can only be $(+1)$, i.e., pWT must take the value $\pm\lambda/2$. Further, the necessary condition of Theorem 3 is always satisfied for n prime and $\lambda = 2^{\frac{n+1}{2}}$.*

Proof. For n prime, in the first column of ${}_n\mathcal{H}$ we have 1 row with $(+1)$ and $\frac{2^{n-1}-1}{n}$ rows with values $(+n)$. With $\tau = 0$ we require σ_1^* such that $pWT = 0$, i.e., $(k \cdot n \pm 1)$ must be 0, for some k. For prime $n \geq 3$ there is no such k.

Now we prove the second part. For n prime, $d_{n,1} = 2$ and $d_{n,n} = \frac{2^n-2}{n}$. Thus we get an equation of the form $1 \cdot k_1 + n \cdot k_n = \frac{\tau\lambda + 2^n}{4} = \pm 2^{\frac{n-3}{2}} + 2^{n-2}$, where $k_1 \in [0, 1]$ and $k_n \in [0, \ldots, \frac{2^{n-1}-1}{n}]$. We show that it is always possible to get an integer solution for k_1, k_n.

Note that for $n > 3$ prime, $n | 2^{n-1} - 1$, i.e., $n | (2^{\frac{n-1}{2}} + 1)(2^{\frac{n-1}{2}} - 1)$.

If $n | (2^{\frac{n-1}{2}} + 1)$, then $n | 2^{\frac{n-3}{2}}(2^{\frac{n-1}{2}} + 1)$, i.e., $n | 2^{\frac{n-3}{2}} + 2^{n-2}$. Thus for $\tau = 1$, we take $k_1' = 0$. Also, $n | 2^{\frac{n-3}{2}} + 2^{n-2} - (2^{\frac{n-1}{2}} + 1)$, i.e., $n | -2^{\frac{n-3}{2}} + 2^{n-2} - 1$. Thus for $\tau = -1$, we take $k_1'' = 1$.

If $n | (2^{\frac{n-1}{2}} - 1)$, then $n | 2^{\frac{n-3}{2}}(2^{\frac{n-1}{2}} - 1)$, i.e., $n | -2^{\frac{n-3}{2}} + 2^{n-2}$. Thus for $\tau = -1$, we take $k_1'' = 0$. Also, $n | -2^{\frac{n-3}{2}} + 2^{n-2} + (2^{\frac{n-1}{2}} - 1)$, i.e., $n | 2^{\frac{n-3}{2}} + 2^{n-2} - 1$. Thus for $\tau = 1$, we take $k_1' = 1$. □

Existence of $(n, \frac{n-3}{2}, \frac{n+1}{2}, 2^{n-1} - 2^{\frac{n-1}{2}})$ functions for odd n is an important open question in Boolean function literature [10, 11, 13]. These functions are plateaued with $\lambda = 2^{\frac{n+1}{2}}$. The only results available are for $n = 5, 7$ as described in [9]. Corollary 3 shows that the necessary condition is satisfied for any odd prime n when we search in the class of RSBFs. This gives a partial theoretical justification why such functions were available in the RSBF class for $n = 5, 7$ as observed in [12]. Thus it will be interesting to target the problem for $n = 11$ also.

4 Search Strategy

Based on the theoretical results discussed so far, we present how these results can be used for actual search for RSBFs with certain cryptographic properties. It has been observed in [13] that to search for $(9, 3, 5, 240)$ one needs to check for 2^{43} many RSBFs. Though we have already proved theoretically that such RSBF does not exist, we now show that the search can be reduced to 2^{34} only. This search also produces the $[9, 3, 5, 240]$ functions and we implement the search to get the complete list of $[9, 3, 5, 240]$ RSBFs. Apart from the theoretical results, we exploit nontrivial software implementation to make the search much faster. This is important since the search space becomes larger for higher number of variables and best possible software implementation is required for actual search.

The algorithm uses only the matrix ${}_n\mathcal{H}$ in the search. The idea behind the algorithm is very simple and it can be used to find plateaued RSBFs for a *desired* Walsh transform, e.g., m-resilient or mth order correlation immune.

The first step of the algorithm is to search the complete set of σ_1's such that $w_1 = \sigma_1^* \cdot {}_n\mathcal{H}$ only take values from the set $w_{1_i} \in \{0, \pm\lambda/2, \pm\lambda\}$. Note that in the positions where the Walsh transform must be zero, the corresponding values of the pWT must be $w_{1_i} \in \{0, \pm\lambda/2\}$, three valued only. Let us denote this set of σ_1's by $\mathcal{S}_{\sigma_1}$. From (2) and Table 1 we see that $w_2 = \sigma_2 \cdot {}_n\mathcal{H}$ is calculated in the same way and has the same restrictions, so it means that $\mathcal{S}_{\sigma_2} = \mathcal{S}_{\sigma_1}$.

The second step of the algorithm is to calculate the Walsh transform for $(\sigma_1 \parallel \sigma_2)$ in the space $\mathcal{S}_{\sigma_1} \times \mathcal{S}_{\sigma_2}$. It means that we need to save $\mathcal{S}_{\sigma_1}$ in a list or in a file.

The time complexity for the first step to find $\mathcal{S}_{\sigma_1}$ is $O(2^{g_n/2})$ and the second step has the complexity $O(|\mathcal{S}_{\sigma_1}|^2)$, so the total time complexity is $O(2^{g_n/2}) + O(|\mathcal{S}_{\sigma_1}|^2)$. Note that in this strategy we do not care about what degree we have on the functions, all functions with desired Walsh spectra will be found.

Now we describe how to use the proposed search strategy to implement an exhaustive search for $[9, 3, 5, 240]$ functions. For RSBFs on 9 variables there are $g_9 = 60$ groups and, hence, the total search space for these functions is 2^{60}. However, in the ANF there can not be terms of degree 6, 7, 8 or 9 and, at least one term of degree 5 must be present. Therefore, the search space does not include all RSBFs on 9 variables, instead the search space is of size $2^{\sum_{i=1}^{4} g_{9,i}}(2^{g_{9,5}} - 1) = 2^{29}(2^{14} - 1) \approx 2^{43}$. This is the complexity of the algorithm when one first uses the ${}_n\mathcal{B}$ matrix and then the ${}_n\mathcal{A}$ matrix in the search [13], without considering ${}_n\mathcal{H}$. The term of degree 0 is not considered in the search space.

The restrictions on Walsh spectra for a $[9, 3, 5, 240]$ function are $W_f(\omega) = 0$, for $1 \leq wt(\omega) \leq 3$ and $W_f(\omega) = 0$ or $\pm\, 32$, for $wt(\omega) = 0, wt(\omega) > 3$. We do not use the restriction that the function has a certain degree, instead we only use the matrix ${}_n\mathcal{H}$ to reduce the time complexity. Since $g_9 = 60$, the matrix ${}_n\mathcal{H}$ is of size 30×30. We divide the RSTT into 2 parts, σ_1 and σ_2, each of 30 bits, and generate the set $\mathcal{S}_{\sigma_1}$. By simulation we found that this set is of size $|\mathcal{S}_{\sigma_1}| \approx 2^{17}$ so there is no memory problem with storing the complete set in memory. *This will give us the total search of 2^{34}, which is 2^9 times faster than only using ${}_n\mathcal{A}$ and ${}_n\mathcal{B}$ as done in [13].*

Table 2. Different search strategy complexities.

Boolean functions on 9 variables	2^{512}
RSBFs on 9 variables	2^{60}
Finding [9,3,5,240] using matrices ${}_n\mathcal{A}$, ${}_n\mathcal{B}$ [13]	2^{43}
Finding [9,3,5,240] using our strategy	2^{34}

Although the complexity is reduced it is important to minimize the constant time needed to check each candidate pair. For fast implementation purposes we divide the matrix ${}_n\mathcal{H}$ into two sections, H_1 and H_2 as shown in Figure 1, each containing 15 rows. We divide σ_1 in the same way and denote the two parts

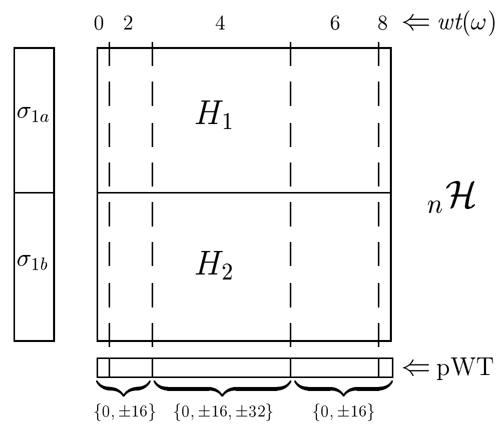

Fig. 1. For fast implementation purposes, the matrix ${}_n\mathcal{H}$ is divided into sections.

$\sigma_1 = (\sigma_{1a} \parallel \sigma_{1b})$. For each section, the sum of the rows is precomputed for each of the 2^{15} possible inputs, and these sums are stored in the table $H_{fast}[2][2^{15}][30]$, having 2 sections with 2^{15} possible inputs for each, and the result is a vector of 30 integers. Now to calculate the partial Walsh transform we only need 2 table look ups and a maximum of 30 integer summations. Unnecessary computation can be avoided by calculating the values of the pWT one by one. If one value is not valid, then we stop and select the next σ_1. Since $W_f(\omega)$ must be 0 for $wt(\omega) \leq 3$, the pWT in these positions must be in $\{0, \pm16\}$. Note that the complement of the representative elements of weight 6 and 8 have weights 1 and 3, so in these positions pWT must also be in $\{0, \pm16\}$. In the rest positions, pWT $\in \{0, \pm16, \pm32\}$. These restrictions can be seen in Table 1. When $\mathcal{S}_{\sigma_1}$ is found, we try all combinations for the cartesian product $(\mathcal{S}_{\sigma_1} \times \mathcal{S}_{\sigma_1})$ and check if the Walsh transform is valid for a $[9, 3, 5, 240]$ function. Since $\mathcal{S}_{\sigma_2} = \mathcal{S}_{\sigma_1}$, we can use the same precomputed tables for fast calculation of $w_2 = \sigma_2^* \cdot {}_n\mathcal{H}$.

The exact search time required is 6064 seconds on a computer with Pentium M 1.6 GHz processor and 512MB RAM using Windows XP operating system. In [13], it was estimated that the search will take almost 3 years to complete the search on a single Pentium 1.6 GHz computer with 256 MB RAM using Linux 7.2 operating system.

Using our strategy we could check that there is no resilient RSBFs with parameters $(9, 3, 5, 240)$ (already proved theoretically) and there are 8406 correlation immune functions with the same parameters $[9, 3, 5, 240]$, when the term of degree 0 is not considered. That is if we also consider the complement of the functions then there are 2×8406 many functions.

Let us denote the autocorrelation value of an n-variable Boolean function f with respect to the vector α as $\Delta_f(\alpha) = \sum_{x \in \{0,1\}^n} (-1)^{f(x) \oplus f(x \oplus \alpha)}$, and the absolute indicator $\Delta_f = \max_{\alpha \in \{0,1\}^n, \alpha \neq \overline{0}} |\Delta_f(\alpha)|$. Low autocorrelation value is important for functions in cryptographic applications [14]. Thus we also check the Δ_f value for these $[9, 3, 5, 240]$ functions.

The Δ_f values of the functions are 80 (4956 many out of 8406), 96 (1020), 112 (312), 136 (180), 152 (1734) and 224 (204). A few correlation immune RSBFs with these parameters have been reported recently using simulated annealing based heuristic search [2]. We execute the search completely and show that the search space can be exhaustively analysed implying that the heuristic method is not required in this case.

It should be noted that we have only exploited the ${}_n\mathcal{H}$ matrix but not used the degree restrictions on the functions. The ${}_n\mathcal{B}$ matrix may also be used for faster search with ${}_n\mathcal{H}$.

Motivated by Corollary 3 and the discussion after it, we also attempted the search for $(11, 4, 6, 992)$ functions. Note that these functions are plateaued. Existence of these functions is not yet known. Since $g_{11} = 188$, the ${}_{11}\mathcal{H}$ matrix is 94×94 and the method of search that we attempt here will not work. Even if using the degree restriction and use of ${}_n\mathcal{B}$ matrix does not come to much help. We attempted some heuristic search and found an $(11, 1, 6, 992)$ plateaued RSBF with Δ_f value 240. Heuristic search, as attempted in [2] may come to help in such a scenario.

5 Conclusion

In this paper we studied the Walsh spectra of rotation symmetric Boolean functions. The set of rotation symmetric Boolean functions is much smaller than the complete space of Boolean functions. Even then complete search of RSBFs is not practical for $n \geq 9$. Our results provide combinatorial insight to the Walsh spectra of the functions and we show that some necessary conditions on existence of certain kinds of functions can be derived from them. In particular, we studied the plateaued RSBFs in this paper. The central result here is to show that the ${}_n\mathcal{A}$ matrix can be written as

$$\left(\begin{array}{c|c} {}_n\mathcal{H} & {}_n\mathcal{H} \\ \hline {}_n\mathcal{H} & -{}_n\mathcal{H} \end{array}\right)$$

after certain permutations when n is odd. Further research in this direction is to study these matrices in more details and to see whether some methods can be explored to analyse functions on higher number of variables. It should also be noted that the matrix structure we present here cannot be extended for n even and studying that case is also an interesting research area.

References

1. C. Carlet and E. Prouff. On plateaued functions and their constructions. In *Fast Software Encryption 2003*, number 2887 in Lecture Notes in Computer Science, pages 54–73. Springer Verlag, 2003.
2. J. Clark, J. Jacob, S. Maitra and P. Stanica. Almost Boolean Functions: The Design of Boolean Functions by Spectral Inversion. In *CEC 2003, the 2003 Congress on Evolutionary Computation*, Volume 3 in the proceedings, page 2173–2180, IEEE Press, December 8–12, 2003, Canberra, Australia.
3. N. Courtois and W. Meier. Algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology - EUROCRYPT 2003*, number 2656 in Lecture Notes in Computer Science, pages 345–359. Springer Verlag, 2003.
4. C. Ding, G. Xiao, and W. Shan. *The Stability Theory of Stream Ciphers*. Number 561 in Lecture Notes in Computer Science. Springer-Verlag, 1991.
5. E. Filiol and C. Fontaine. Highly nonlinear balanced Boolean functions with a good correlation-immunity. In *Advances in Cryptology - EUROCRYPT'98*. Springer-Verlag, 1998.
6. M. Hell, A. Maximov. S. Maitra. On efficient implementation of search strategy for rotation symmetric Boolean functions. In *Ninth International Workshop on Algebraic and Combinatoral Coding Theory, ACCT 2004*, June 19–25, 2004, Black Sea Coast, Bulgaria.
7. M. Matsui. Cryptanalysis method for DES cipher. In *Advances in Cryptology, Eurocrypt 1993*, Lecture Notes in Computer Science, Number 765, Pages 386–397, Springer-Verlag, 1994.
8. W. Meier, E. Pasalic and C. Carlet. Algebraic attacks and decomposition of Boolean functions. In *Advances in Cryptology - EUROCRYPT 2004*, to be published in Lecture Notes in Computer Science. Springer Verlag, 2004.
9. E. Pasalic, S. Maitra, T. Johansson and P. Sarkar. New constructions of resilient and correlation immune Boolean functions achieving upper bound on nonlinearity. In *Workshop on Coding and Cryptography - WCC 2001*, Paris, January 8–12, 2001. Electronic Notes in Discrete Mathematics, Volume 6, Elsevier Science, 2001.
10. P. Sarkar and S. Maitra. Construction of nonlinear Boolean functions with important cryptographic properties. In *Advances in Cryptology - EUROCRYPT 2000*, number 1807 in Lecture Notes in Computer Science, pages 485–506. Springer Verlag, May 2000.
11. P. Sarkar and S. Maitra. Nonlinearity bounds and construction of resilient Boolean functions. In *Advances in Cryptology - Crypto 2000*, number 1880 in Lecture Notes in Computer Science, pages 515–532, Springer-Verlag, 2000.
12. P. Stănică and S. Maitra. Rotation Symmetric Boolean Functions – Count and Cryptographic Properties. In *R. C. Bose Centenary Symposium on Discrete Mathematics and Applications*, December 2002. Electronic Notes in Discrete Mathematics, Elsevier, Volume 15.
13. P. Stănică, S. Maitra and J. Clark. Results on Rotation Symmetric Bent and Correlation Immune Boolean Functions. In *Fast Software Encryption 2004*, to be published in volume 3017 in Lecture Notes in Computer Science, Springer-Verlag, 2004.
14. X-M. Zhang and Y. Zheng. GAC – the criterion for global avalanche characteristics of cryptographic functions. *Journal of Universal Computer Science*, 1(5):316–333, 1995.
15. Y. Zheng and X. M. Zhang. Plateaued Functions. In *ICICS'99*, pages 284-300, volume 1726 in Lecture notes in Computer Science, Springer Verlag.