Efficient Search for Symmetric Boolean Functions under Constraints on Walsh Spectrum Values*

Sumanta Sarkar and Subhamoy Maitra
Applied Statistics Unit, Indian Statistical Institute,
203 B T Road, Kolkata 700 108, INDIA,
Email: {sumanta_r, subho}@isical.ac.in

Abstract

In this paper we present an efficient exhaustive search strategy on symmetric Boolean functions having the Walsh spectrum values constrained in a range at certain points. Exploiting the structure in Walsh spectrum of a symmetric Boolean function and its relationship with Krawtchouk matrix, we extend the concept of folded vectors and pruning introduced by Gathen and Roche in 1997. The strategy is applied to search for highly nonlinear symmetric Boolean functions and nonlinear symmetric resilient and correlation immune functions. We also experimentally justify that our method provides further efficiency than the search strategy presented by Gathen and Roche.

Keywords: Boolean Function, Correlation Immunity, Efficient Exhaustive Search, Nonlinearity, Resiliency, Symmetry, Walsh Spectrum.

1 Introduction

A standard model of stream cipher, called Nonlinear Combiner Model [22, 23, 6], combines LFSR sequences using a nonlinear Boolean function. While

*This is a substantially revised and extended version of the paper presented in the "Second International Workshop on Boolean Functions: Cryptography and Applications, BFCA 06", March 13–15, 2006, LIFAR, University of Rouen, France.

JCMCC **68** (2009), pp. 163-191

selecting the Boolean function one has to maintain some constraints for crytographic purposes. The function should be balanced to maintain the pseudo-randomness of the generated key-stream. Moreover, the function should be highly nonlinear since a function with low nonlinearity is weak with respect to the linear approximation attack [6]. Further, to reduce the vulnerability to the correlation attack we have to choose the combining function with correlation immunity of high order [22, 23]. High algebraic degree is one of the necessary conditions for high linear complexity [6, 19]. So far there have been lots of research to achieve Boolean functions with good cryptographic properties together.

The advantage of studying symmetric Boolean functions is that the size of this class is much less as compared to the general Boolean functions. The total number of distinct n-variable symmetric Boolean functions is 2^{n+1}, whereas that of general Boolean functions is 2^{2^n}. Moreover, an n-variable symmetric Boolean function can be expressed by an $(n+1)$ length binary vector (called its value vector) which requires less amount of memory to be stored. However, symmetric functions with good cryptographic properties have not yet been exhibited and its use in stream cipher is still not encouraging. Even then, the study on symmetric functions with certain cryptographic properties is continuing mainly due to their nice combinatorial properties [3, 18, 13, 10, 21, 24, 8, 17, 20, 12, 1, 25, 4, 5] related to binomial coefficients and Krawtchouk polynomials.

An interesting question was raised in [3] on the existence of nonlinear, resilient, symmetric Boolean functions. The existence was later shown in [10] giving the example of nonlinear 1-resilient symmetric functions on even number of input variables $4t^2 - 2$ as well as 2-resilient nonlinear symmetric functions on odd number of input variables $4t^2 - 1$ ($t \geq 2$, integer). Later in [8] the problem has been studied independently, where the authors could study the resiliency of nonlinear symmetric functions till 128 variables with a nice search technique. Apart from the classes presented in [10], another class of 2-resilient nonlinear symmetric functions has been identified in [8] for input variables $n = F_{2i+2}F_{2i+3} + 1$, where $i \geq 2$ and $i \not\equiv 1 \bmod 3$ and $\{F_i\}$ is the Fibonacci sequence ($F_0 = 0$, $F_1 = 1$ and $F_{i+2} = F_i + F_{i+1}$, $i \geq 0$). Clearly this will provide 1-resilient nonlinear symmetric functions on $n - 1$ many input variables. In [25], it has been claimed that new classes of nonlinear resilient symmetric functions have been discovered. However, we find that these are only the classes presented in [10, 11] (see Section 4 for detailed discussion). The correspondence between the work [8] and the resiliency of symmetric Boolean functions can be found in good details in [1].

In [8], an efficient search method to get balanced nonlinear symmetric

Boolean functions has been proposed. In this paper we extend their strategy to search nonlinear symmetric Boolean functions having constrained Walsh spectrum values at certain points. Since resiliency and nonlinearity directly depend on Walsh spectrum values, by choosing the range of the Walsh spectrum values properly, our strategy can be exploited to search resilient or correlation immune symmetric functions more efficiently than [8].

The organization of the paper is as follows. We start with some preliminary discussion in the next section. Our contribution in expediting the exhaustive search for symmetric Boolean functions with constrained Walsh spectrum values are presented in Section 3. The initial idea presented in Subsection 3.1 is taken from [8], but we make substantial inroad in extending the idea which is presented in Subsections 3.2, 3.3. In Subsection 3.3.2, we use these techniques to search efficiently for highly nonlinear symmetric functions. Section 4 begins with some theoretical results related to existence of nonlinear resilient symmetric functions. A sufficient condition on the non existence of nonlinear symmetric 2-resilient functions is presented in Theorem 1 for even number of input variables relating the elements of Krawtchouk matrix. Using this, a sufficient condition on the non existence of nonlinear symmetric 3-resilient functions on any number of input variables (even or odd) is shown (see Corollary 2). Based on these theoretical results and the efficient search ideas, in Subsection 4.1 we compare our results with [8] in searching symmetric nonlinear 2-resilient functions. Experimental evidences show that in searching 2-resilient nonlinear symmetric functions on n variables, our method is of time complexity $O(2^{\frac{n}{8}})$ that is asymptotically better in comparison to $O(2^{\frac{n}{4}})$ presented in [8].

We could check the 2-resilient nonlinear symmetric functions till $n = 261$ variables. This is a noticeable improvement than the search of 2-resilient functions till $n = 128$ variables presented in [8]. Note that in Section 7 of an earlier draft version [9] of the published paper [8], the enumeration of 2-resilient nonlinear symmetric function till $n = 500$ variables has been claimed using a different method (though this is not claimed in the published paper [8]) which does not exploit folding and pruning. This [9, Section 7] has been studied only for a special case when the Walsh spectra values are zero at the points of weight $\leq m$ (for m-resilient functions), i.e., the technique can not be exploited to efficiently search nonlinear symmetric functions with nonzero Walsh spectra values. The main motivation of our paper is to study the efficiency of searching nonlinear symmetric Boolean functions having constrained Walsh spectra value and hence our experiment of enumerating 2-resilient functions is not to compete with any technique that only finds such functions, but to study the experimental time complexity of our improved strategy using folding and pruing in comparison to

that of [8].

We have also searched till $n = 128$ variables for unbalanced nonlinear symmetric correlation immune functions. We could find correlation immune functions of order 3, but there are no order 4 functions till $n = 128$. In [20], these functions have been studied only till $n = 20$ variables.

2 Preliminaries

Denote the set of n-variable Boolean functions $f : \{0,1\}^n \rightarrow \{0,1\}$ by B_n. One of the standard representation of a Boolean function $f(x_1, \ldots, x_n)$ is by the output column of its *truth table*, i.e., a binary string of length 2^n,

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

Any Boolean function f has a unique representation as a multivariate polynomial over $GF(2)$, called the algebraic normal form (ANF),

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

where the coefficients $a_0, a_i, a_{ij}, \ldots, a_{12\ldots n} \in \{0,1\}$. 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 n-variable affine functions is denoted by $A(n)$. An affine function with constant term equal to zero is called a linear function.

Definition 1 *A Boolean function is called symmetric if its output depends only on the Hamming weight (the number of 1's) of the input vectors.*

Thus a Boolean function $f \in B_n$ is symmetric if $f(\alpha) = f(\beta)$, where $wt(\alpha) = wt(\beta)$ for $\alpha, \beta \in \{0,1\}^n$. It is clear that one can represent an n-variable symmetric Boolean function $f(x_1, \ldots, x_n)$ in a reduced form by $n+1$ bits string re_f such that $re_f(i) = f(x_1, \ldots, x_n)$ when $wt(x_1, \ldots, x_n) = i$. The notation re_f is well known as the value vector of a symmetric Boolean function.

Walsh transform is a very useful tool in analyzing Boolean functions.

Definition 2 *Let $x = (x_1, \ldots, x_n)$ and $\omega = (\omega_1, \ldots, \omega_n)$ both belonging to $\{0,1\}^n$ and $x \cdot \omega = x_1\omega_1 + \ldots + x_n\omega_n$. Let $f(x)$ be a Boolean function on*

n-variables. Then the Walsh transform *of $f(x)$ is an integer valued function over $\{0,1\}^n$ which is defined as $W_f(\omega) = \sum_{x\in\{0,1\}^n}(-1)^{f(x)+x\cdot\omega}$.*

Definition 3 *A Boolean function f is balanced iff $W_f(0) = 0$.*

A Boolean function f is m-th order correlation immune iff $W_f(\omega) = 0$, for all ω with $1 \leq wt(\omega) \leq m$.

A Boolean function f is m-resilient if it is m-th order correlation immune and balanced as well. The necessary and sufficient condition for f to be m-resilient is that $W_f(\omega) = 0$, for all ω with $0 \leq wt(\omega) \leq m$.

Further the nonlinearity of f is given by

$$nl(f) = 2^{n-1} - \frac{1}{2}\max_{\omega\in\{0,1\}^n}|W_f(\omega)|.$$

The Walsh spectrum of symmetric Boolean functions have very nice combinatorial properties related to Krawtchouk polynomial [21]. Krawtchouk polynomial [15, 14] of degree i is given by $K_i(x,n) = \sum_{j=0}^{i}(-1)^j\binom{x}{j}\binom{n-x}{i-j}$. It is known that for a fixed ω, such that $wt(\omega) = k$, $\sum_{wt(x)=i}(-1)^{\omega\cdot x} = K_i(k,n)$. Thus it can be checked that if $f \in B_n$ is a symmetric function with value vector $re_f = (f_0,\ldots,f_n)$, then for $wt(\omega) = k$, $W_f(\omega) = \sum_{i=0}^{n}(-1)^{f_i}K_i(k,n)$. It is also known that for a symmetric function $f \in B_n$ and $\alpha,\beta \in \{0,1\}^n$, $W_f(\alpha) = W_f(\beta)$, if $wt(\alpha) = wt(\beta)$. Thus it is enough to calculate the Walsh spectrum for the inputs of $n+1$ different weights. Keeping this in mind, given a symmetric Boolean function $f \in B_n$, we denote $rw_f(i) = W_f(\omega)$, such that $wt(\omega) = i$. Thus rw_f can be seen as a mapping from $\{0,\ldots,n\}$ to $\mathbb{Z}$. It is clear that if we want to determine all the Walsh spectrum values for f it is enough to multiply $((-1)^{f_0},\ldots,(-1)^{f_n})$ with the the matrix $K(n)$, where the (i,k)-th element is $K_i(k,n)$. The matrix $K(n)$ is referred as Krawtchouk matrix [7]. Let us now revisit a few important existing results in this area [15, 14].

Proposition 1

1. $K_0(k,n) = 1, K_1(k,n) = n - 2k$,
2. $(i+1)K_{i+1}(k,n) = (n-2k)K_i(k,n) - (n-i+1)K_{i-1,n}(k,n)$,
3. $K_i(k,n) = (-1)^k K_{n-i}(k,n)$,
4. $\binom{n}{k}K_i(k,n) = \binom{n}{i}K_k(i,n)$,

5. $K_i(k,n) = (-1)^i K_i(n-k,n)$,

6. $(n-k)K_i(k+1,n) = (n-2i)K_i(k,n) - kK_i(k-1,n)$,

7. $(n-i+1)K_i(k,n+1) = (3n-2i-2k+1)K_i(k,n) - 2(n-k)K_i(k,n-1)$.

As example, let us present the Krawtchouk matrix for $n = 14, 15$. For brevity, we write the top-left $\frac{1}{4}$-th part of the matrix, the rest can be obtained using property 3 and 5 of Proposition 1. The matrix for $n = 14$ is as follows.

$$\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
14 & 12 & 10 & 8 & 6 & 4 & 2 & 0 \\
91 & 65 & 43 & 25 & 11 & 1 & -5 & -7 \\
364 & 208 & 100 & 32 & -4 & -16 & -12 & 0 \\
1001 & 429 & 121 & -11 & -39 & -19 & 9 & 21 \\
2002 & 572 & 22 & -88 & -38 & 20 & 30 & 0 \\
3003 & 429 & -165 & -99 & 27 & 45 & -5 & -35 \\
3432 & 0 & -264 & 0 & 72 & 0 & -40 & 0
\end{bmatrix}$$

Here is the matrix for $n = 15$.

$$\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
15 & 13 & 11 & 9 & 7 & 5 & 3 & 1 \\
105 & 77 & 53 & 33 & 17 & 5 & -3 & -7 \\
455 & 273 & 143 & 57 & 7 & -15 & -17 & -7 \\
1365 & 637 & 221 & 21 & -43 & -35 & -3 & 21 \\
3003 & 1001 & 143 & -99 & -77 & 1 & 39 & 21 \\
5005 & 1001 & -143 & -187 & -11 & 65 & 25 & -35 \\
6435 & 429 & -429 & -99 & 99 & 45 & -45 & -35
\end{bmatrix}$$

Detailed discussion on Krawtchouk Polynomial and Walsh spectrum of a symmetric function can be found in [5]. We now present the following known technical result that will be used frequently in this paper.

Proposition 2 *Let $lin = (lin_0, \ldots, lin_n) = (0,1,0,1,\ldots)$ be the n-variable linear symmetric function and $add = (add_0, \ldots, add_n)$ be another n-variable symmetric function. Then the function $f = (lin \oplus add)$ follows the inequality $|W_f(w)| \leq W$ where $wt(w) = k < n$ iff*

$$|\sum_{i=0}^{n} (-1)^i add_i K_i(k,n)| \leq \frac{W}{2} \tag{1}$$

Proof: We have

$$|\sum_{i=0}^{n}(-1)^{(lin_i \oplus add_i)} K_i(k,n)| \leq W$$

iff $|\sum_{i=0}^{n}\{(-1)^{lin_i}(1-2add_i)\}K_i(k,n)| \leq W$,

(since $(-1)^a = 1-2a$, for $a \in \{0,1\}$)

iff $|\sum_{i=0}^{n} 2(-1)^{lin_i} add_i K_i(k,n)| \leq W$

(since $\sum_{i=0}^{n}(-1)^{lin_i} K_i(k,n) = 0$ for $k < n$)

iff $|\sum_{i=0}^{n}(-1)^i add_i K_i(k,n)| \leq \frac{W}{2}$. ■

Corollary 1 *The function $f = (lin \oplus add)$ is balanced iff*

$\sum_{i=0}^{n}(-1)^i add_i K_i(0,n) = 0$.

Proof: This follows easily by putting $k = 0$ and $W = 0$ in (1). ■

3 Search with constrained Walsh spectrum

We start this section with the idea presented in [8] towards searching balanced symmetric Boolean functions on n variables. Then we extend the idea towards the search of symmetric Boolean functions where there are constraints at certain Walsh spectrum points.

3.1 Method proposed in [8]

In [8], Gathen and Roche made an exhaustive search for n-variable nonlinear balanced symmetric Boolean functions f till $n = 128$ variables. Since the search was for n-variable nonlinear symmetric balanced functions f, they concentrated on searching n-variable symmetric functions $add = (add_0, \ldots, add_n)$ such that $f = (lin \oplus add)$ becomes balanced, where $lin = (lin_0, \ldots, lin_n) = (0,1,0,1,\ldots)$ is the n-variable linear symmetric Boolean function. From Corollary 1, it is clear that the search for the balanced symmetric functions in [8] was the search for the patterns add satisfying $\sum_{i=0}^{n}(-1)^i add_i K_i(0,n) = 0$, i.e., they considered the following search problem if one notes that $K_i(0,n) = \binom{n}{i}$.

Problem 1

Find the n-variable symmetric function add such that

$$\sum_{i=0}^{n}(-1)^i add_i\binom{n}{i} = 0.$$

The trivial search space consisting of all the symmetric functions would be 2^{n+1}. The concept of searching over the folded symmetric functions [8] reduced the search space down to $3^{\lceil \frac{n}{2} \rceil}$ for the initial search. This is described below.

First consider the n odd case. As $K_i(0,n) = K_{n-i}(0,n)$ (by Proposition 1 property 3), Problem 1 can be written as $\sum_{i=0}^{\frac{n-1}{2}}(-1)^i(add_i - add_{n-i})\binom{n}{i} = 0$, i.e.,

$$\sum_{i=0}^{\frac{n-1}{2}}(-1)^i M_i\binom{n}{i} = 0, \tag{2}$$

where $M_i = add_i - add_{n-i}$. So in this case instead of searching the full pattern *add*, initial search can be restricted over the folded patterns $M = (M_0, \ldots, M_{\frac{n-1}{2}})$. As $M_i \in \{-1, 0, 1\}$ for $0 \leq i \leq \frac{n-1}{2}$, the search space size over all folded patterns is $3^{\frac{n+1}{2}}$. It is worth noting that $3^{\frac{n+1}{2}} << 2^{n+1}$ (asymptotically smaller).

Similarly if we consider n even and $add_{\frac{n}{2}} = 0$, then Problem 1 can be written as

$$\sum_{i=0}^{\frac{n}{2}-1}(-1)^i P_i\binom{n}{i} = 0, \tag{3}$$

where $P_i = add_i + add_{n-i}$. Also in this case the search can be executed over the folded patterns $P = (P_0, \ldots, P_{\frac{n}{2}-1})$. Again as $P_i \in \{0, 1, 2\}$ for $0 \leq i \leq \frac{n}{2} - 1$, the search space is of size $3^{\frac{n}{2}}$.

Remark 1 *If $add_{\frac{n}{2}} = 1$, then in the function $add' = (1 \oplus add)$ we have $add'_{\frac{n}{2}} = 0$. So if we perform the search over add' we will get the folded patterns which were already found in the case of add. Thus after getting the folded patterns for $add_{\frac{n}{2}} = 0$, complementing them we can get the folded patterns for $add_{\frac{n}{2}} = 1$. It is easy to verify that complement of a folded pattern P can be obtained by interchanging the places of the 0's and 2's keeping the places for the 1's unchanged.*

3.1.1 Pruning

The search in [8] has been made efficient further by an interesting pruning idea for odd n. The search is initiated from $M_{\frac{n-1}{2}}$. At the r-th step down one needs to check whether

$$\sum_{i=r}^{\frac{n-1}{2}}(-1)^i M_i\binom{n}{i} = -\left[\sum_{i=0}^{r-1}(-1)^i M_i\binom{n}{i}\right],$$

i.e., $|\sum_{i=r}^{\frac{n-1}{2}}(-1)^i M_i\binom{n}{i}| \leq \sum_{i=0}^{r-1}|(-1)^i M_i\binom{n}{i}|$, i.e.,

$$|\sum_{i=r}^{\frac{n-1}{2}}(-1)^i M_i\binom{n}{i}| \leq \sum_{i=0}^{r-1}\binom{n}{i}, \tag{4}$$

since the maximum value that $|M_i|$ can take is 1. So if (4) is not satisfied, then the sub pattern $(M_r, \ldots, M_{\frac{n-1}{2}})$ can not be a part of the folded pattern $(M_0, \ldots, M_{\frac{n-1}{2}})$ satisfying (2), in which case the remaining 3^r possibilities $(M_0, \ldots, M_{r-1})$ can be pruned from the search tree.

The pruning idea for even n is quite similar. The necessary condition for the sub pattern $(P_r, \ldots, P_{\frac{n}{2}-1})$ to be part of a pattern $(P_0, \ldots, P_{\frac{n}{2}-1})$ satisfying (3) is

$$|\sum_{i=r}^{\frac{n}{2}-1}(-1)^i P_i\binom{n}{i}| \leq 2\sum_{i=0}^{r-1}\binom{n}{i}. \tag{5}$$

For this search, the idea of pruning works very efficiently. By empirical evidence in [8], it is claimed that the number of steps required is of the order $2^{\frac{n}{4}}$ which is much less than $3^{\lceil\frac{n}{2}\rceil}$.

3.1.2 Unfolding

After getting the folded pattern the symmetric function *add* can be obtained by unfolding the folded pattern. Unfolding from the pattern $M = (M_0, \ldots, M_{\frac{n-1}{2}})$ is as follows.

- $M_i = 0 \rightarrow add_i = add_{n-i} = 0$ or $add_i = add_{n-i} = 1$;
- $M_i = 1 \rightarrow add_i = 1$ and $add_{n-i} = 0$;
- $M_i = -1 \rightarrow add_i = 0$ and $add_{n-i} = 1$.

When we unfold the pattern M, number of symmetric functions obtained is 2^s where $s = \#$ of 0's in M.

The unfolding for $P = (P_0, \ldots, P_{\frac{n}{2}-1})$ is as follows.

- $P_i = 0 \rightarrow add_i = add_{n-i} = 0$;
- $P_i = 1 \rightarrow$ "$add_i = 1$ and $add_{n-i} = 0$" or "$add_i = 0$ and $add_{n-i} = 1$";
- $P_i = 2 \rightarrow add_i = 1$ and $add_{n-i} = 1$.

When unfolding the folded pattern P, the number of symmetric functions obtained is 2^t, where t is the number of 1's in P. In [8, Algorithm 5.1], the search was for all folded patterns for odd n, satisfying (2).

The function *add* is XORed with the linear function *lin* to yield the balanced function. Note that we only count the functions in complement free manner, i.e., if we count a symmetric function then we will not count its complement.

During the search (with pruning), one can keep track with the number of steps it requires to yield the folded patterns (some steps will not really produce a feasible folded pattern as they may die without reaching a complete folded pattern). One can set a COUNTER initialized to 0, and each time during the search the COUNTER is increased by 1 as one component of the folded vector chooses one option from $3^{\lceil \frac{n}{2} \rceil}$ possible options. Thus the COUNTER value at the end of the search will reveal the search effort given for a particular n.

Example 1 *As example, for $n = 34$, we can find following folded vectors P for the add patterns:*
`00000022012112100 (four 1's), 00000022010222100 (two 1's),`
`00000022012110110 (five 1's), 00000022010220110 (three 1's),`
`00000000000001210 (two 1's).`

For each of the patterns we can get 2^t many unfolded vectors where $t =$ number of 1's in P. Thus we can get $(2^4+2^2+2^5+2^3+2^2) = 64$ many add patterns and when XORed with lin, $(lin \oplus add)$ will provide the total class of nonlinear balanced symmetric functions for $n = 34$. The total search effort required to find the folded patterns is $COUNTER = 4221 \approx 2^{12}$.

For $n = 35$, as it is odd we always get the trivial (all zero) folded pattern `00000000000000000`. *After unfolding, this pattern will provide 2^{17} many (complement free) nonlinear balanced symmetric functions. In addition to that, we get two more folded patterns*
`11111111111̄01111̄00 (three 0's), 11111111111111111̄00 (two 0's),`
where $\bar{1}$ *denotes* -1. *Thus we will get $2^3 + 2^2 = 12$ more such functions.*

The total search effort required to find the folded patterns is COUNTER $= 886 \approx 2^{10}$. *Once again note that we enumerate the symmetric functions in complement free manner.*

3.2 Searching nonlinear symmetric functions with constrained value at a single point in the Walsh spectrum

The idea of [8] can be extended beyond finding balanced function. Suppose we want to search some nonlinear symmetric function $f = (lin \oplus add)$ on n variables, where *lin* and *add* are as described in the Subsection 3.1, with some constraint at a point ω with $wt(\omega) = k$ such that its Walsh spectrum value at that point lies in the range $[-W, W]$, $W > 0$. So by Proposition 2, it is enough to consider the following search problem.

Problem 2

Find the n-variable symmetric functions add such that

$$|\sum_{i=0}^{n}(-1)^i add_i K_i(k,n)| \leq \frac{W}{2}.$$

Now instead of searching for the full pattern *add*, we can search over the folded pattern of *add* to reduce the search space.

CASE 1a. n odd, k even.

By Proposition 1(3), $K_i(k,n) = K_{n-i}(k,n)$. Thus,

$|\sum_{i=0}^{n}(-1)^i add_i K_i(k,n)| \leq \frac{W}{2}$ is equivalent to

$$|\sum_{i=0}^{\frac{n-1}{2}}(-1)^i M_i K_i(k,n)| \leq \frac{W}{2}, \tag{6}$$

where $M_i = add_i - add_{n-i}$ for $i = 0$ to $\frac{n-1}{2}$. For each add_i we have options 0 or 1. So in this case, the size of the search space is 2^{n+1}. However, for each M_i we have three options $\{-1, 0, 1\}$, in which case the size of the search space becomes $3^{\frac{n+1}{2}}$. It is worth noting that $3^{\frac{n+1}{2}} << 2^{n+1}$.

CASE 1b. n odd, k odd.

By Proposition 1(3), $K_i(k,n) = -K_{n-i}(k,n)$. Therefore,

$|\sum_{i=0}^{n}(-1)^i add_i K_i(k,n)| \leq \frac{W}{2}$ is equivalent to

$$|\sum_{i=0}^{\frac{n-1}{2}}(-1)^i P_i K_i(k,n)| \leq \frac{W}{2}, \tag{7}$$

where $P_i = add_i + add_{n-i}$ for $i = 0$ to $\frac{n-1}{2}$. Here the options for each P_i are $\{0,1,2\}$ and the search space size is also $3^{\frac{n+1}{2}}$.

CASE 2a. n even, k even.

Consider $add_{\frac{n}{2}} = 0$. By Proposition 1(3), $K_i(k,n) = K_{n-i}(k,n)$.

Therefore, $|\sum_{i=0}^{n}(-1)^i add_i K_i(k,n)| \leq \frac{W}{2}$ is equivalent to

$|\sum_{i=0}^{\frac{n}{2}-1}(-1)^i P_i K_i(k,n)| \leq \frac{W}{2}$, where $P_i = (add_i + add_{n-i})$ for $i = 0$ to $\frac{n}{2} - 1$. For this the search space size is $3^{\frac{n}{2}}$.

CASE 2b. n even, k odd.

Consider $add_{\frac{n}{2}} = 0$. By Proposition 1(3), $K_i(k,n) = -K_{n-i}(k,n)$.

Therefore, $|\sum_{i=0}^{n}(-1)^i add_i K_i(k,n)| \leq \frac{W}{2}$, is equivalent to

$|\sum_{i=0}^{\frac{n}{2}-1}(-1)^i M_i K_i(k,n)| \leq \frac{W}{2}$, where $M_i = (add_i - add_{n-i})$ for $i = 0$ to $\frac{n}{2} - 1$. In this situation the search space size is $3^{\frac{n}{2}}$ too.

Remark 2 *For n even, we generally consider $add_{\frac{n}{2}} = 0$ as it is already discussed in Remark 1 in Subsection 3.1. This means that when we construct $f = (lin \oplus add)$, then the value $f_{\frac{n}{2}}$ will be the same as $lin_{\frac{n}{2}}$. Further, for even n and odd k, $K_{\frac{n}{2}}(k,n) = 0$ and hence the value of $add_{\frac{n}{2}}$ does not participate in Walsh spectrum computation. However, for even n and even k, $K_{\frac{n}{2}}(k,n) \neq 0$ in general. From the discussion in Remark 1, it is clear that the patterns with $add_{\frac{n}{2}} = 1$ will be taken care of by the complement patterns of the cases when $add_{\frac{n}{2}} = 0$. We only search the patterns which are complement free and thus it is enough to consider $add_{\frac{n}{2}} = 0$ only.*

After getting the folded patterns of *add* of the form P or M, the full patterns of *add* can be obtained by unfolding P or M and the exact function f is then obtained by XORing *add* with linear function *lin*.

So far we have seen the initial search space being reduced to $3^{\lceil \frac{n}{2} \rceil}$. Slightly modified idea of pruning introduced in [8] can be used to prune the folded patterns while considering the Problem 2.

Let us discuss this idea of for odd n. For k even, we have to search for the folded pattern $M = (M_0, \ldots, M_{\frac{n-1}{2}})$ satisfying (6), i.e.,

$|\sum_{i=0}^{\frac{n-1}{2}} (-1)^i M_i K_i(k,n)| \leq \frac{W}{2}$, i.e.,

$|\sum_{i=r}^{\frac{n-1}{2}} (-1)^i M_i K_i(k,n)| - |\sum_{i=0}^{r-1} (-1)^i M_i K_i(k,n)| \leq \frac{W}{2}$, i.e.,

$|\sum_{i=r}^{\frac{n-1}{2}} (-1)^i M_i K_i(k,n)| \leq \frac{W}{2} + |\sum_{i=0}^{r-1} (-1)^i M_i K_i(k,n)|$, i.e.,

$|\sum_{i=r}^{\frac{n-1}{2}} (-1)^i M_i K_i(k,n)| \leq \frac{W}{2} + \sum_{i=0}^{r-1} |(-1)^i M_i K_i(k,n)|$, i.e.,

$$|\sum_{i=r}^{\frac{n-1}{2}} (-1)^i M_i K_i(k,n)| \leq \frac{W}{2} + \sum_{i=0}^{r-1} |K_i(k,n)|, \tag{8}$$

since maximum value that $|M_i|$ can take is 1. Clearly, if the sub pattern $(M_r, \ldots, M_{\frac{n-1}{2}})$ does not satisfy (8), then it cannot be in any of the folded pattern $(M_0, \ldots, M_{\frac{n-1}{2}})$ which satisfy (6). So at once we can prune all the 3^r patterns from the search space which contain $(M_r, \ldots, M_{\frac{n-1}{2}})$ as a sub pattern.

For k odd, we have to search for the patterns $P = (P_0, \ldots, P_{\frac{n-1}{2}})$ satisfying (7). Necessary condition for a sub pattern $(P_r, \ldots, P_{\frac{n-1}{2}})$ to be a part of these pattern P would be

$$|\sum_{i=r}^{\frac{n-1}{2}} (-1)^i P_i K_i(k,n)| \leq \frac{W}{2} + 2\sum_{i=0}^{r-1} |K_i(k,n)|. \tag{9}$$

So the same idea of pruning can be applied here also. The even variable case is very much similar.

Example 2 *Here we consider* $n = 35$ *and* $W = 0$ *for Problem 2. First for* $k = 0$, *we get the folded patterns*
11111111111$\bar{1}$01$\bar{1}\bar{1}$00, 1111111111111$\bar{1}\bar{1}$00, 00000000000000000.
The search effort is COUNTER $= 886 < 2^{10}$.

For $k = 1$ *we get the folded patterns*
00000000000000000, 00000000211210000, 00000020011011000,
00000020222221000, 00000000000001100, 00000000211211100,
00000020011012100, 00000020222222100, 00000000000002200,
00000000211212200, 00000010111110120, 00000010111111220,
11111111111110011, 020202020202022011, 111111111111111111.

The search effort is COUNTER $= 6915 < 2^{13}$.

For $k = 2$ we get the folded patterns
$10101\bar{1}\bar{1}\bar{1}0\bar{1}00\bar{1}\bar{1}10\bar{1}\bar{1}$, $1\bar{1}1\bar{1}1\bar{1}110\bar{1}0000 10\bar{1}\bar{1}$, $1\bar{1}1\bar{1}1\bar{1}10\bar{1}01\bar{1}0110\bar{1}\bar{1}$,
$10101\bar{1}\bar{1}\bar{1}0\bar{1}00\bar{1}\bar{1}01\bar{1}\bar{1}$, $1\bar{1}1\bar{1}1\bar{1}110\bar{1}00000 1\bar{1}\bar{1}$, $1\bar{1}1\bar{1}1\bar{1}10\bar{1}01\bar{1}0101\bar{1}\bar{1}$,
$\bar{1}1\bar{1}1\bar{1}1\bar{1}010\bar{1}10\bar{1}11\bar{1}\bar{1}$, $\bar{1}1\bar{1}1\bar{1}1\bar{1}\bar{1}0100001 1\bar{1}\bar{1}$, $\bar{1}0\bar{1}0\bar{1}11101001111\bar{1}\bar{1}$,
$1\bar{1}1\bar{1}1\bar{1}1\bar{1}1\bar{1}1\bar{1}1\bar{1}1\bar{1}1\bar{1}$, $1\bar{1}1\bar{1}1\bar{1}1\bar{1}1\bar{1}1\bar{1}1\bar{1}001\bar{1}$, $1\bar{1}1\bar{1}1\bar{1}1\bar{1}1\bar{1}1\bar{1}1\bar{1}111\bar{1}$,
$1100\bar{1}00\bar{1}111001\bar{1}\bar{1}\bar{1}0$, $1100\bar{1}00100\bar{1}0011\bar{1}\bar{1}0$, $1100\bar{1}00100\bar{1}00100\bar{1}0$,
$\bar{1}\bar{1}001001\bar{1}001\bar{1}010\bar{1}0$, $1100\bar{1}00100\bar{1}001\bar{1}1\bar{1}0$, $\bar{1}\bar{1}001001\bar{1}001\bar{1}001\bar{1}0$,
$000000011\bar{1}\bar{1}10\bar{1}1\bar{1}00$, $000000000000001\bar{1}00$, $0000000\bar{1}\bar{1}11\bar{1}011\bar{1}00$,
$000000011\bar{1}\bar{1}10\bar{1}0000$, 000000000000000000.

The search effort is COUNTER $= 8314 < 2^{14}$.

Following Example 2, one can check that the values of the COUNTER (i.e., search effort) differs according to different values of k, which is clear as the efficiency of pruning depends on the values $K_i(k,n)$ we are considering and that is different for different columns. It will be clearer if one looks at the Krawtchouk matrix. The distribution of the numbers in the column $k = 0$ of the Krawtchouk matrix is nicely set, the bigger numbers are at the middle of that column. As we are considering the folded vectors, those bigger numbers will be associated with the values at the end of the folded vectors. That is why starting the search method by growing the folded vector from the end provides a good chance to prune early. For pruning in the other columns (say $k = 1, 2$) it is not as good as the case $k = 0$. However, the pruning is always very effective as in this case the exhaustive search space for folded patterns is as large as $3^{18} > 2^{28}$.

Example 3 *Now we present a practical example for large $n = 101$, $k = 2$ and $W = 0$ in Problem 2. We find four folded patterns as follows:*
$\bar{1}1\bar{1}$,
$\bar{1}\bar{1}00100\bar{1}00100\bar{1}00100\bar{1}00100\bar{1}00100\bar{1}00100\bar{1}00100\bar{1}00100\bar{1}0$,
$\bar{1}00000$,
$000000000000000000000000000000000000000000000000000$

The search effort is $COUNTER = 431044816 < 2^{29}$ that is much less than $3^{51} (> 2^{80})$, which highlights the advantage of pruning. The exact time taken by a C program in Fedora Core 3 operating system is 58 seconds on a PC having 3.6 Ghz Intel Xeon and 4 GB RAM.

3.3 Searching nonlinear symmetric functions with constrained Walsh spectrum values at more than one points

From the earlier discussions we find that we need to concentrate on the cases n even or odd and k even or odd, i.e., four cases.

Suppose we are searching for nonlinear symmetric functions $f = (lin \oplus add)$ where lin and add are as described in Subsection 3.1, such that $W_f(\omega_j) \in [-W_j, W_j]$, for $wt(\omega_j) = k_j$, where k_j's are either all odd or all even. Then by Proposition 2, it is enough to consider the following search problem.

Problem 3

Find the n-variable symmetric functions add such that

$$|\sum_{i=0}^{n}(-1)^i add_i K_i(k_j, n)| \leq \frac{W_j}{2},$$

for k_j's either all odd or all even.

Let us first analyze this problem for odd n in the following two cases.

CASE 1a. All k_j's are even.
By Proposition 1(3), $K_i(k_j, n) = K_{n-i}(k_j, n)$. So Problem 3 can be written as

$$|\sum_{i=0}^{\frac{n-1}{2}}(-1)^i M_i K_i(k_j, n)| \leq \frac{W_j}{2}, \tag{10}$$

for all given k_j's, where $M_i = add_i - add_{n-i}$ for $i = 0, \ldots, \frac{n-1}{2}$. Note that the search space size over the folded pattern $M = (M_0, \ldots, M_{\frac{n-1}{2}})$ is $3^{\frac{n+1}{2}}$.

CASE 1b. All k_j's are odd.
By Proposition 1(3), $K_i(k_j, n) = -K_{n-i}(k_j, n)$. Problem 3 can be written as

$$|\sum_{i=0}^{\frac{n-1}{2}}(-1)^i P_i K_i(k_j, n)| \leq \frac{W_j}{2}, \tag{11}$$

for all given k_j's, where each $P_i = (add_i + add_{n-i})$ for $i = 0, \ldots, \frac{n-1}{2}$. So the search space size is now $3^{\frac{n+1}{2}}$.

Next consider n even and $add_{\frac{n}{2}} = 0$ (for $add_{\frac{n}{2}} = 1$, refer to Remark 2).

CASE 2a. All k_j's are even.
By Proposition 1(3), $K_i(k_j, n) = K_{n-i}(k_j, n)$. So Problem 3 can be written as

$$|\sum_{i=0}^{\frac{n}{2}-1}(-1)^i P_i K_i(k_j, n)| \leq \frac{W_j}{2}, \tag{12}$$

for all given k_j's, where each $P_i = (add_i + add_{n-i})$ for $i = 0, \ldots, \frac{n}{2} - 1$. So the search space size is now reduced to $3^{\frac{n}{2}}$.

CASE 2b. All k_j's are odd.
By Proposition 1(3), $K_i(k_j, n) = -K_{n-i}(k_j, n)$. So Problem 3 can be written as

$$|\sum_{i=0}^{\frac{n}{2}-1}(-1)^i M_i K_i(k_j, n)| \leq \frac{W_j}{2}, \tag{13}$$

for all given k_j's, where $M_i = add_i - add_{n-i}$ for $i = 0, \ldots, \frac{n}{2} - 1$. The required search space size is now $3^{\frac{n}{2}}$.

We can also apply same idea of pruning as described in Subsection 3.2 to expedite the search process. Let us discuss it for odd n and even k_j's. The necessary condition for a sub pattern $(M_r, \ldots, M_{\frac{n-1}{2}})$ to be part of a pattern $M = (M_0, \ldots, M_{\frac{n-1}{2}})$ which is a solution to (10) is $|\sum_{i=r}^{\frac{n-1}{2}}(-1)^i M_i K_i(k_j, n)| \leq \frac{W_j}{2} + \sum_{i=0}^{r-1} |K_i(k_j, n)|$, for all k_j's. So once the sub pattern $(M_r, \ldots, M_{\frac{n-1}{2}})$ violates this condition we can discard all the 3^r patterns from the search tree, which contain $(M_r, \ldots, M_{\frac{n-1}{2}})$ as a sub pattern. The idea of pruning is quite similar in case of even variables.

Example 4 *As example, we consider* $n = 101, k_1 = 2, k_2 = 4, W_1 = 0, W_2 = 0$. *We get the following folded patterns:*
$\bar{1}1\bar{1}$,
000000000000000000000000000000000000000000000000000

The search effort is COUNTER $= 5076601 < 2^{23}$ *which takes 1 second on the same platform as explained in Example 3.*

The folded patterns that we find from this search need to be unfolded to get the full patterns for *add* which are next to be XORed with the linear function *lin* to produce the exact function f. While unfolding, there may be many options corresponding to a folded pattern. This is to note that for odd n, the symmetric functions generated from all zero folded pattern will have Walsh spectrum values zero at even weight input points.

3.3.1 Searching nonlinear symmetric functions with constrained Walsh spectrum values at both odd and even weight points

Suppose we are searching for a nonlinear symmetric function on n variables $f = (lin \oplus add)$, where lin and add are as described in Subsection 3.1, such that $W_f(\omega_j) \in [-W_j, W_j]$, given $wt(\omega_j) = k_j$ taking both odd and even values. Then by Proposition 2 this search problem can be written as follows.

Problem 4

Find the n-variable symmetric functions add such that

$$|\sum_{i=0}^{n}(-1)^i add_i K_i(k_j, n)| \leq \frac{W_j}{2},$$

with k_j's taking both even and odd values.

Let $k_{e_1}, \ldots, k_{e_l}$ are even and $k_{o_1}, \ldots, k_{o_p}$ are odd with the corresponding bound on Walsh spectrum values respectively are W_{e_i} for $1 \leq i \leq l$ and W_{o_i} for $1 \leq i \leq p$. Then search for solutions for Problem 4 can be divided into two search problems. Note that in Problem 4, both of the two cases of Problem 3, i.e., all even k_j and all odd k_j have been considered altogether. So first search for the patterns which are the solutions of Problem 3 considering $k_j = k_{e_j}$, for $1 \leq j \leq l$. Next find the patterns which are also solutions of Problem 3 but considering $k_j = k_{o_j}$, for $1 \leq j \leq p$. Obviously the patterns which are the solutions in both of the cases are the required patterns for add. Let us discuss this search problem for odd n. First we consider Problem 3 with $k_j = k_{e_j}$, for $1 \leq j \leq l$, which is exactly **CASE 1a**, i.e., we have to search for the folded patterns $M = (M_0, \ldots, M_{\frac{n-1}{2}})$ such that

$$|\sum_{i=0}^{\frac{n-1}{2}}(-1)^i M_i K_i(k_{e_j}, n)| \leq \frac{W_{e_j}}{2}, \tag{14}$$

for all j, $1 \leq j \leq l$.

Next we consider Problem 3 again with $k_j = k_{o_j}$, for $1 \leq j \leq p$, which is **CASE 1b**. We search for the folded patterns P such that

$$|\sum_{i=0}^{\frac{n-1}{2}}(-1)^i P_i K_i(k_{o_j}, n)| \leq \frac{W_{o_j}}{2} \tag{15}$$

for all j, $1 \leq j \leq p$. So our desired symmetric functions for *add* satisfy both of (14) and (15) in two different kinds of folding.

The most interesting issue here is one can find the exact functions *add* not by unfolding but by solving the patterns $M = (M_0, \ldots, M_{\frac{n-1}{2}})$ and $P = (P_0, \ldots, P_{\frac{n-1}{2}})$. For this we present the following technical result.

Proposition 3 *Let* $a_0, a_1 \in \{0, 1\}$. *The equations* $a_0 + a_1 = x$, $a_0 - a_1 = y$ *are solvable iff* $(x + y)$ *is* 0 mod 2.

Proof: Solutions of these two equations are $a_0 = \frac{x+y}{2}$ and $a_1 = \frac{x-y}{2}$. Now a_0 and a_1 belong to $\{0, 1\}$, iff $(x + y)$ and $(x - y)$ are either 0 or 2. ■

Based on Proposition 3, we consider the folded patterns

$M = (M_0, \ldots, M_{\frac{n-1}{2}})$ and $P = (P_0, \ldots, P_{\frac{n-1}{2}})$ and directly solve them (when possible) to get the exact function *add*.

For even n, it is clear that we will obtain folded patterns of the type P for all even k_j's. On the other hand folded M patterns will be obtained for all odd k_j's. The rest treatment is similar to the odd n case.

Here also the same idea of pruning can be applied. Following same argument we can say that if the sub pattern $(M_r, \ldots, M_{\frac{n-1}{2}})$ does not satisfy

$$|\sum_{i=r}^{\frac{n-1}{2}} (-1)^i M_i K_i(k_{e_j}, n)| \leq \frac{W_{e_j}}{2} + \sum_{i=0}^{r-1} |K_i(k_{e_j}, n)|, \tag{16}$$

for $1 \leq j \leq l$, then it cannot be a part of any $M = (M_0, \ldots, M_{\frac{n-1}{2}})$ which satisfies (14). So all 3^r patterns containing $(M_r, \ldots, M_{\frac{n-1}{2}})$ as a sub pattern can be pruned from the search tree. Similarly if the sub pattern $(P_r, \ldots, P_{\frac{n-1}{2}})$ does not satisfy

$$|\sum_{i=r}^{\frac{n-1}{2}} (-1)^i P_i K_i(k_{o_j}, n)| \leq \frac{W_{o_j}}{2} + 2\sum_{i=0}^{r-1} |K_i(k_{o_j}, n)|, \tag{17}$$

for $1 \leq j \leq p$, then it cannot be a part of any $P = (P_0, \ldots, P_{\frac{n-1}{2}})$ which satisfies (15). So all the 3^r patterns containing $(P_r, \ldots, P_{\frac{n-1}{2}})$ as a sub pattern can be pruned from the search tree.

Example 5 *We now apply our strategy to search for* balanced *nonlinear symmetric functions on* 101 *variables having some constraints on the Walsh*

spectrum values. The constraints are at the input points of weights $1, 2, 3, 4$ *in the ranges* $[-2^{20}, 2^{20}]$, $[-2^6, 2^6]$, $[-2^{20}, 2^{20}]$, $[-2^9, 2^9]$ *respectively. We could find only all zero folded pattern* M *for the constraints on even weight points. The search effort is* $COUNTER = 60220 < 2^{16}$. *For the constraint on odd weights, we get* 202 *folded patterns of the type* P. *The required search effort is* $COUNTER = 4591342 < 2^{23}$. *Then after solving among the patterns of the type* M *and* P *we could find* 11 *many nonlinear symmetric functions. While solving these patterns we require* $2 \times 202 \times 1 \times 51 < 2^{15}$ *more addition/subtraction operations. As a whole it requires* $< 2^{24}$ *steps to produce the required functions.*

3.3.2 Searching symmetric functions with high nonlinearity

The bent functions attain the maximum possible nonlinearity $2^{n-1} - 2^{\frac{n}{2}-1}$ for even number of input variables n (Walsh spectrum values $\pm 2^{\frac{n}{2}}$) and the value vectors of symmetric bent functions [21, 16] are characterized as the $n+1$ length substring of the infinite string $(0011)^*$. These functions are of degree 2. Further, we know from [17] that the highest nonlinearity for n (odd) variable symmetric functions is $2^{n-1} - 2^{\frac{n-1}{2}}$ and once again its value vector is the $n+1$ length substring of the infinite string $(0011)^*$. These functions are of degree 2 and the Walsh spectrum of these functions are 3-valued which are $0, \pm 2^{\frac{n+1}{2}}$. Characterization of the Walsh spectrum values of degree 2 symmetric functions can be found in [1, Table 1].

Let us consider the search for n-variable nonlinear function $f = (lin \oplus add)$, where lin and add are as described earlier, with nonlinearity greater or equal to $2^{n-1} - \frac{W}{2}$, i.e., the maximum absolute value in the Walsh spectrum is W. This problem is exactly framed as Problem 4, where we will consider k_j's taking values from 0 to $n-1$. After getting the P, M patterns, we solve them according to Proposition 3. The patterns add obtained as the solutions are then XORed with the n-variable linear symmetric function lin to get the function f.

Then the function f *is tested for the column* n *(as Proposition 2 takes care of all* $k < n$, *but not* n*) in the Krawtchouk matrix and the functions whose Walsh value at that point are in the range* $[-W, W]$ *are the symmetric function with the given lower bound on nonlinearity.*

Further, if one likes to get more constraints, say $W_j < W$ for some specific points, then that can also be managed in a similar manner. For a balanced function, we need that the Walsh spectrum value at all-zero point will be zero.

We now mention the experimental result for $n = 101$. For $n = 101$ the highest nonlinearity would be $2^{100} - 2^{50}$. Let us search for balanced symmetric functions having nonlinearity $\geq 2^{100} - \frac{2^{51}+2^{10}}{2}$. To achieve these functions first we search for patterns considering Problem 3 for all even weight $k_j \notin \{0, n\}$ with $W = 2^{51} + 2^{10}$ and further $W = 0$ for $k_j = 0$. We find only all zero folded vector M with search effort COUNTER = $838 < 2^{10}$. After that we find patterns considering Problem 3 for all odd $k_j < n$ with $W = 2^{51} + 2^{10}$. In this case we find $2922935 < 2^{22}$ many folded patterns P with search effort COUNTER = $4408153 < 2^{23}$. Then we unfold the patterns by solving P, M patterns. The unfolded patterns are then XORed with the linear symmetric function *lin* and the new functions are checked for the Walsh spectrum value at the point having weight n. The functions we obtain finally are as follows:
01011001100110011001100110011001100110011001100110011001100
11001100110011001100110011001100110011001100110011001100101,
nonlinearity = $2^{100} - 2^{50} - 204$,

11011001100110011001100110011001100110011001100110011001100
11001100110011001100110011001100110011001100110011001100100,
nonlinearity = $2^{100} - 2^{50} - 202$,

00011001100110011001100110011001100110011001100110011001100
11001100110011001100110011001100110011001100110011001100111,
nonlinearity = $2^{100} - 2^{50} - 2$,

and the maximum nonlinear functions, $n+1$ length substring of $(0011)^*$, as explained in [17].

The total effort taken is less than $2^{10} + 2^{23} + 2 \times 1 \times 2^{23} < 2^{25}$.

4 Finding nonlinear resilient and correlation immune symmetric functions

In [3], it was conjectured that nonlinear, resilient, symmetric Boolean functions do not exist. This conjecture was disproved in [10], where the construction of nonlinear 1-resilient symmetric functions on even number of input variables $4t^2 - 2$ as well as 2-resilient nonlinear symmetric functions on odd number of input variables $4t^2 - 1$ ($t \geq 2$, integer) have been provided.

When $n = 4t^2 - 2$, $t \geq 2$, the 1-resilient nonlinear symmetric function is the symmetric linear function complemented in the value vector at the places $k, k+1, n-k, n-k+1$, where $k = 2t^2 - t - 1$, keeping rest of the

positions unchanged. The smallest member of this class is available for $n = 14$ when $k = 5$ and the value vector is $(0,1,0,1,0,0,1,1,0,0,1,1,0,1,0)$. When $n = 4t^2 - 1$, $t \geq 2$, the 2-resilient nonlinear symmetric function is given by the symmetric linear function complemented in the value vector at the places $k, k+1, n-k-1, n-k$, where $k = 2t^2 - t - 1$, keeping rest of the places unchanged. The smallest member of this class is for $n = 15$ when $k = 5$ and the value vector $(0,1,0,1,0,0,1,1,0,0,1,1,0,1,0,1)$.

Recently, in [25], it has been claimed that a new class of nonlinear 2-resilient symmetric functions have been discovered on n variables, where $n = 4t^2 - 1$, $t \geq 2$. However, we find that these are actually the classes presented in [10]. In fact, there was a minor typographical error regarding the construction of 2-resilient functions in [10], which has been corrected in [11, Pages 144–146] and that happens to be the same recently rediscovered class presented in [25].

Further, in [25], a class (claimed to be new) of 1-resilient function has been presented on n variables, where $n = 4t^2 - 2$, $t \geq 2$. This 1-resilient nonlinear symmetric function is the symmetric linear function complemented in the value vector at the places $k-1, k, n-k-1, n-k$, where $k = 2t^2 - t - 1$, keeping rest of the positions unchanged. If one considers the value vector of the 2-resilient functions given in [10] for n odd and then by removing the first element considers the value vector of $(n-1)$-variable function, then the nonlinear, symmetric, 1-resilient function constructed in [25] is immediately available.

In [8] the problem has been studied independently. They have experimented till $n = 128$ variables. Apart from the classes presented in [10], they have identified another class of 2-resilient nonlinear symmetric functions for input variables $n = F_{2i+2}F_{2i+3} + 1$ where $i \geq 2$ and $i \not\equiv 1 \bmod 3$ and $\{F_i\}$ is the Fibonacci sequence ($F_0 = 0$, $F_1 = 1$ and $F_{i+2} = F_i + F_{i+1}$, $i \geq 0$). Clearly this will provide 1-resilient nonlinear symmetric functions on $n - 1$ many input variables. The first (minimum) n in this series is 105.

4.1 Improvement in complexity over the method proposed in [8] in finding 2-resilient functions

In [8], first the folded patterns corresponding to $add = (add_0, \ldots, add_n)$ are considered such that $\sum_{i=0}^{n}(-1)^i add_i \binom{n}{i} = 0$. That is, from such a pattern add (neither all zero nor all one) one can get a balanced nonlinear symmetric function $f = (lin \oplus add)$ where $lin = (lin_0, \ldots, lin_n) = (0,1,0,1,\ldots)$, is the n-variable symmetric linear function. In [8] each nonlinear symmetric value vector $add = (add_0, \ldots, add_n)$ has been studied to calculate a term

called "*gap*" [8, Theorem 2.2]. One can check that if $gap \geq m+1$, $(m \geq 0)$ then $f = (lin \oplus add)$ is a nonlinear symmetric m-resilient function. So the function *add* with positive gap implies that the function f is balanced since 0-resiliency means balancedness. One should also refer to [2, Proposition 1] for the relationship between degree in Numerical Normal Form (NNF) and the order of resiliency.

With our strategy 2-resilient nonlinear symmetric Boolean functions can be found with much less computational effort than that of [8]. We search for m-resilient nonlinear symmetric functions on n-variables, i.e., the motive of the search is to find the functions $f = (lin \oplus add)$ such that $W_f(\omega) = 0$, for all ω such that $wt(\omega) \leq m$. It is clear that the search for m-resilient symmetric functions can be performed by considering Problem 4 with the parameters $W_j = 0$ for $0 \leq k_j \leq m$. The folded patterns of the types M, P are obtained and then solved according to Proposition 3. Finally the patterns *add* obtained as the solutions are XORed with the linear symmetric function *lin* and hence we obtain n-variable m-resilient symmetric functions if at all they exist.

We now present the exact strategy in finding 2-resilient nonlinear symmetric functions till 261-variables.

4.1.1 n odd

From [8] one can observe that there cannot be any 2-resilient nonlinear symmetric function on n-variables when n is prime. Thus we will exclude this case in the search. Next we present another important necessary condition for existence of 2-resilient functions which reduces the search effort to a significant level.

We start the search for nonlinear 2-resilient symmetric functions considering Problem 3, for $W_1 = 0, W_2 = 0$ and $k_1 = 0, k_2 = 2$. The folded patterns we obtain in this case are of M type. Next considering Problem 3 once again with $W_1 = 0$ and $k_1 = 1$, we obtain folded patterns of type P. These M, P patterns are solved according to Proposition 3. The solutions *add* are then XORed with *lin* to get nonlinear 2-resilient symmetric functions on n variables. The following result helps in reducing the search further.

Lemma 1 *Let n be an odd integer, $M = (M_0, \ldots, M_{\frac{n-1}{2}})$, where $M_i \in \{-1, 0, 1\}$ and $P = (P_0, \ldots, P_{\frac{n-1}{2}})$, $P_i \in \{0, 1, 2\}$. Let the solution to the equations $\sum_{i=0}^{\frac{n-1}{2}} (-1)^i M_i K_i(0, n) = 0$ and $\sum_{i=0}^{\frac{n-1}{2}} (-1)^i M_i K_i(2, n) = 0$ be*

the all zero pattern M only. Then to find 2-resilient nonlinear symmetric functions, it is enough to search for the patterns $P = (P_0, \ldots, P_{\frac{n-1}{2}})$ with the constraints $P_i \in \{0, 2\}$ satisfying $\sum_{i=0}^{\frac{n-1}{2}} (-1)^i P_i K_i(1, n) = 0$.

Proof: It is clear that 2-resilient functions can be obtained by solving given M and P patterns according to Proposition 3. As the components of the M patterns are all equal to zero, according to the Proposition 3, a pattern P having 1 as any one component P_i can never give a solution and hence P patterns having only 0 and 2 as the components may produce 2-resilient symmetric functions. ■

Given Lemma 1 the search space size for finding folded patterns regarding Problem 3 considering all odd k_j, now reduces to $2^{\frac{n+1}{2}}$ down from $3^{\frac{n+1}{2}}$. As $P_i = 1$ for any i can not provide a valid solution with the given M pattern, we have to only search for the patterns P where $P_i \in \{0, 2\}$ for all i such that $0 \leq i \leq \frac{n-1}{2}$. This result provides a theoretical bound on complexity which is $2^{\frac{n+1}{2}}$ as opposed to the theoretical value $3^{\frac{n+1}{2}}$ given in [8, Page 358]. The theoretical bound in [8] is only due to folding. Our better theoretical bound is due to folding and further characterization of symmetric resilient functions. Later we also show that using pruning technique we achieve better empirical complexity compared to the empirical complexity presented in [8, Page 358].

We have searched till $n = 261$ for odd n's and found that only all zero M patterns as solutions. Note that our search is complement free. So using Lemma 1, the search space for folded P patterns for Problem 3 can be restricted to $2^{\frac{n+1}{2}}$ for odd n till 261. Above this the pruning idea will also work to provide a faster search.

As an example, let us mention the search effort for $n = 105$. Finding patterns M for Problem 3 with the parameters $W_1 = 0, W_2 = 0$ and $k_1 = 0, k_2 = 2$ requires the search effort COUNTER = 202757 and we got only all zero M pattern. Then to find P patterns for Problem 3 for $W_1 = 0$ and $k_1 = 1$, the required search effort is COUNTER = 115874. Here we consider $P_i \in \{0, 2\}$ as P_i cannot be 1 to have a valid solution. The length of both M and P patterns is 53. Then solving them according to Proposition 3, we found only one 2-resilient nonlinear symmetric function. The solution step requires 53×2 many addition/subtraction steps. Thus our total search effort is $< 2^{18}$ which is significantly better than the initial 26926322 many comparisons ($2^{24} < 26926322 < 2^{25}$) in pruning plus analyzing 2^{53} many unfolded choices (for the folded all zero pattern of length 53) to calculate *gap* as explained in [8].

To experimentally analyze our computational effort, we consider the cases for odd n, $129 \leq n \leq 261$, (n not prime as there cannot be any 2-resilient functions). We collect both of the COUNTER values for searching M, P patterns; add them and further add the number of steps to solve them. We denote this value as ϕ_n. We list the $\lceil \frac{\phi_n}{2^{\frac{n}{8}}} \rceil$ and then note the maximum (respectively minimum) value of $\lceil \frac{\phi_n}{2^{\frac{n}{8}}} \rceil$ in the range which is 30 (respectively 11) for $n = 129$ (respectively $n = 261$). Further the value of $\frac{\phi_n}{2^{\frac{n}{8}}}$ is decreasing in that range with n. That is the reason we can experimentally estimate the complexity for searching 2-resilient nonlinear symmetric function on odd number of variables n as $O(2^{\frac{n}{8}})$, where we get the functions by first finding the folded vectors of type M, P and then solving them to get the unfolded functions. Our strategy is significantly better than [8], as in [8] finding the the folded patterns corresponding to the balanced functions requires $O(2^{\frac{n}{4}})$ time complexity.

4.1.2 n even

We have already noted in [8, Theorem 2.6] that there is no nonlinear balanced symmetric function on $(p-1)$-variables, where p is a prime and hence there is no 2-resilient symmetric nonlinear function on $(p+1)$-variables. In our search we will exclude these two cases.

Theorem 1 *Let n be even and $P = (P_0, \ldots, P_{\frac{n}{2}-1})$, where $P_i \in \{0, 1, 2\}$. If the solution to the equations*

$\sum_{i=0}^{\frac{n}{2}-1} (-1)^i P_i K_i(0, n) = 0$ *and* $\sum_{i=0}^{\frac{n}{2}-1} (-1)^i P_i K_i(2, n) = 0$ *is the all zero vector P only, then there is no nonlinear symmetric 2-resilient functions on n-variables.*

Proof: We know that $add_i + add_{n-i}$ can take the values from $\{0, 1, 2\}$. We assume that the solutions to the equation $\sum_{i=0}^{\frac{n}{2}-1} (-1)^i P_i K_i(k, n) = 0$ for $k = 0, 2$ provides only the all zero vector for P. Thus the solution to $\sum_{i=0}^{\frac{n}{2}-1} (-1)^i (add_i + add_{n-i}) K_i(k, n) = 0$ for $k = 0, 2$ provides only the all zero vector as a solution for $(add_0 + add_n, add_1 + add_{n-1}, \ldots, add_{\frac{n}{2}-1} + add_{\frac{n}{2}+1})$. This means $add_i + add_{n-i} = 0$, i.e., $add_i = add_{n-i} = 0$. If we consider $add_{\frac{n}{2}} = 0$, then add is the all zero function and hence $f = (lin \oplus add)$ would be simply the linear symmetric function. So in this case we cannot have nonlinear symmetric 2-resilient functions on n-variables.

On the other hand, if we consider $add_{\frac{n}{2}} = 1$, then we have

$\sum_{i=0}^{n}(-1)^i add_i K_i(2,n) = -K_{\frac{n}{2}}(2,n) = \frac{2}{n-1}\binom{n-1}{\frac{n}{2}} \neq 0$. Thus, in this case $lin \oplus add$ cannot be a 2-resilient function as the Walsh spectrum value at input points of weight 2 is not zero. ■

Corollary 2 *Let n be even and $P = (P_0, \ldots, P_{\frac{n}{2}-1})$, where $P_i \in \{0,1,2\}$. If the solution to the equations*

$\sum_{i=0}^{\frac{n}{2}-1}(-1)^i P_i K_i(0,n) = 0$ *and* $\sum_{i=0}^{\frac{n}{2}-1}(-1)^i P_i K_i(2,n) = 0$ *is the all zero vector P only, then there is no nonlinear symmetric m-resilient ($m \geq 3$) functions on n or $n+1$ variables.*

Proof: By the previous theorem it is clear that for an even variable n with the given conditions there can not be any nonlinear 2-resilient symmetric functions. So there can not be any nonlinear $m \geq 3$-resilient symmetric functions on n-variables.

Under the assumption of the only all zero solution on vector P, for even n the linear symmetric function and its complement are the only 2-resilient symmetric functions. Thus there can not be any nonlinear 3-resilient symmetric function on $(n+1)$ variables. Consequently no m-resilient ($m \geq 3$) nonlinear symmetric function on $(n+1)$-variables exists. ■

To find the 2-resilient symmetric functions we first consider Problem 3 with $W_1 = 0, W_2 = 0$ and $k_1 = 0, k_2 = 2$ to get P patterns. If we find only all zero vector P after the search then we immediately conclude that there is no 2-resilient nonlinear symmetric function following Theorem 1. Our experiment shows this is what that happens till $n = 260$.

If it at all happens that one gets some non zero vector P, then the Problem 3 with $W_1 = 0$ and $k_1 = 1$ should be considered to get M patterns. After that one may solve P and M patterns according to Proposition 3. The solutions are then XORed with the symmetric lincar function to get all 2-resilient symmetric functions on n-variables. In our experiments we do not require to consider Problem 3 for $W_1 = 0$ and $k_1 = 1$ any more till $n = 260$.

To give an idea of the computational effort, we present the case for $n = 212$. Considering $W_1 = 0, W_2 = 0$ and $k_1 = 0, k_2 = 2$ for Problem 3, the search effort is COUNTER $= 6220967502 < 2^{33}$. The time taken by a C program in Fedora Core 3 operating system is 37 minutes and 50 seconds on a PC having 3.6 Ghz Intel Xeon 4 GB RAM.

We experimentally analyze our computational effort for even n, $128 \leq n \leq 260$, (n and $n-1$ not prime as there cannot be any 2-resilient functions).

We calculate the COUNTER values to find the pattern P and as we only find all zero vector P in all the cases, following Theorem 1 we do not need for searching M patterns further. We denote this COUNTER value by ϕ_n; calculate $\lceil \frac{\phi_n}{2^{\frac{n}{8}}} \rceil$ and then note the maximum (respectively minimum) value of $\lceil \frac{\phi_n}{2^{\frac{n}{8}}} \rceil$ in the range which is 97 (respectively 49) for $n = 128$ (respectively $n = 260$). Further the value of $\frac{\phi_n}{2^{\frac{n}{8}}}$ is decreasing in that range with n. So we can experimentally estimate the complexity for searching 2-resilient nonlinear symmetric function on odd number of variables n as $O(2^{\frac{n}{8}})$. So we see for the even n case also, our strategy is significantly better than [8], as in [8] finding only the the folded patterns corresponding to the balanced functions requires $O(2^{\frac{n}{4}})$ time complexity.

In [9, Section 7], the nonlinear symmetric 2-resilient functions have been studied till 500 variables. However, the technique used there seems unsuitable for searching nonlinear symmetric Boolean functions when Walsh spectra values are nonzero. Our experimental result in this section is not to show on how much we can enumerate for nonlinear 2-resilient symmetric functions but to explain better search complexity than [8]. Note that in the following section we present the enumeration strategy for unbalanced symmetric nonlinear correlation immune functions. Since these functions have nonzero Walsh spectrum value at the zero point, the strategy of [9, Section 7] can not be applied directly and we explain how our extension of folding and pruning strategy over [8] works in this scenario.

4.2 Unbalanced 3-rd order correlation immune nonlinear symmetric functions

The question of discovering 3-rd order unbalanced correlation immune nonlinear symmetric Boolean functions was raised in [20] and this was studied only till 30-variables in that paper. Here we use our technique to extend this till 128-variables. The motive of the search is to find the n-variable functions $f = (lin \oplus add)$ such that $W_f(\omega) = 0$ for $1 \leq wt(\omega) \leq 3$, So this problem is actually Problem 4 with the parameters $1 \leq k_j \leq 3$ with the corresponding $W_j = 0$.

As shown in Subsection 3.3.1, we consider Problem 3 and first search for folded patterns with the constraint $W_1 = 0, W_2 = 0$ and $k_1 = 1, k_2 = 3$. Further we search for folded patterns for the same problem with $W_1 = 0$ and $k_1 = 2$. The patterns obtained from these two searches are then solved to find the patterns *add* which are then XORed with the linear function symmetric function to produce nonlinear symmetric functions with 3-rd

order correlation immunity. Below we mention only the number of 3-rd order correlation immune functions starting from 10-variables, by the pair (n, c) where n means the number of variables and c means the number of such functions (up to complementation). The list is as follows: (10, 1), (14, 1), (15, 1), (16, 4), (20, 2), (21, 2), (22, 2), (24, 1), (26, 3), (27, 1), (28, 1), (32, 3), (33, 2), (34, 2), (35, 1), (36, 2), (38, 1), (39, 2), (40, 3), (44, 4), (45, 1), (48, 1), (49, 1), (50, 2), (51, 1), (52, 1), (56, 3), (57, 1), (58, 1), (62, 1), (63, 3), (64, 6), (68, 1), (69, 1), (70, 1), (74, 1), (75, 2), (76, 1), (80, 4), (81, 3), (82, 2), (86, 1), (87, 1), (88, 1), (92, 1), (93, 1), (94, 1), (96, 1), (98, 1), (99, 2), (100, 4), (104, 1), (105, 1), (106, 1), (110, 1), (111, 1), (116, 1), (117, 1), (118, 1), (120, 2), (121, 1), (122, 1), (123, 1), (124, 1), (128, 1).

In [20], it is mentioned that there is no 4-th order Correlation Immune n-variable nonlinear symmetric Boolean function for $6 \leq n \leq 20$. We have checked the 3-rd order correlation immune functions and found that none of them are 4-th order. It reveals that indeed there is no 4-th order correlation immune nonlinear symmetric functions till 128-variables.

5 Conclusion

In this paper we make a systematic study in searching nonlinear symmetric functions with constraints on Walsh spectrum values. We concentrate on the folded structure of the value vectors of symmetric functions that have been exploited in [8] and explore it further using the relationship between Walsh spectrum of a symmetric Boolean function and Krawtchouk polynomial. Experimental results reveal the advantage of our technique over the method presented in [8]. We have also come up with some theoretical results that provide better understanding in studying nonlinear symmetric resilient functions.

Acknowledgments: The authors like to acknowledge the anonymous reviewer for the comments that improved the technical and editorial quality of this paper.

References

[1] A. Canteaut and M. Videau. Symmetric Boolean Function. *IEEE Transaction On Information Theory*, Volume 51, 2791–2811, 2005.

[2] C. Carlet and P. Guillot. Bent, resilient functions and the Numerical Normal Form. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Volume 56, 87–96, 2001.

[3] B. Chor, O. Goldreich, J. Hastad, J. Friedman, S. Rudich and R. Smolensky. The bit extraction problem or t-resilient functions. In *26th IEEE Symposium on Foundations of Computer Science*, 396–407, 1985.

[4] T. W. Cusick and Y. Li. k-th order symmetric SAC Boolean functions and bisecting binomial coefficients. *Discrete Applied Mathematics*, Volume 149, 73–86, 2005.

[5] D. K. Dalai, S. Maitra and S. Sarkar, *Basic Theory in Construction of Boolean Functions with Maximum Possible Algebraic Immunity.* In *Design, Codes and Cryptography*, Volume 40(1), 41–58, 2006.

[6] C. Ding, G. Xiao, and W. Shan. The Stability Theory of Stream Ciphers. Number 561 in Lecture Notes in Computer Science. Springer-Verlag, 1991.

[7] P. Feinsiver and R. Fitzgerald, The Spectrum of Symmetric Krawtchouk Matrices. *Linear Algebra & Applications*, Volume 235, 121–139, 1996.

[8] J. von zur Gathen and J. R. Roche. Polynomials with Two Values. *Combinatorica*, Volume 17(3), 345–362, 1997.

[9] J. von zur Gathen and J. R. Roche. Polynomials with Two Values. Preprint, 23 September, 1993. Available at http://www-math.uni-paderborn.de/preprints/preprints_pages/preprints_Gathen.html [last accessed February 2, 2007].

[10] K. Gopalakrishnan, D. G. Hoffman and D. R. Stinson. A Note on a Conjecture Concerning Symmetric Resilient Functions. *Information Processing Letters*, Volume 47(3), 139–143, 1993.

[11] K. Gopalakrishnan. A study of Correlation-immune, resilient and related cryptographic functions. *PhD thesis*, University of Nebraska, 1994.

[12] A. Gouget. On the propagation criterion of Boolean functions. In *Proceedings of the Workshop on Coding, Cryptography and Combinatorics 2003*, published by Birkhäuser Verlag, K. Feng, H. Niederreiter and C. Xing Eds., 153–168, 2004.

[13] N. Jefferies. Sporadic partitions of binomial coefficients. *Electronics Letters*, Volume 27, 1334–1336, 1991.

[14] I. Krasikov. On Integral Zeros of Krawtchouk Polynomials. *Journal of Combinatorial Theory, Series A*, Volume 74, 71–99, 1996.

[15] F. J. MacWillams and N. J. A. Sloane. The Theory of Error Correcting Codes. North Holland, 1977.

[16] S. Maitra and P. Sarkar. Characterization of symmetric bent functions – An elementary proof. *Journal of Combinatorial Mathematics and Combinatorial Computing*, Volume 43, 227–230, 2002.

[17] S. Maitra and P. Sarkar. Maximum Nonlinearity of Symmetric Boolean Functions on Odd Number of Variables. *IEEE Transactions on Information Theory*, Volume 48(9), 2626–2630, 2002.

[18] C. J. Mitchell. Enumerating Boolean functions of cryptographic significance. *Journal of Cryptology*, Volume 2(3), 155–170, 1990.

[19] R. A. Rueppel and O. J. Staffelbach. Products of Linear Recurring Sequences with Maximum Complexity. *IEEE transaction on Information Theory*, Volume IT-33, 124–131, 1987.

[20] P. Sarkar and S. Maitra. Balancedness and Correlation Immunity of Symmetric Boolean Functions. To be published in *Discrete Mathematics*. Available at: http://dx.doi.org/10.1016/j.disc.2006.08.008 [last accessed February 2, 2007].

[21] P. Savicky. On the Bent Boolean Functions that are Symmetric. *European Journal of Combinatorics*, Volume 15, 407–410, 1994.

[22] T. Siegenthaler. Correlation-Immunity of Nonlinear Combining Functions for Cryptographic Applications. *IEEE Transactions on Information theory*, Volume IT-30(5), 776–780, 1984.

[23] T. Siegenthaler. Decrypting a Class of Stream Ciphers Using Ciphertext Only. *IEEE Transaction on Computers*, Volume C-34(1), 81–85, 1985.

[24] Y. X. Yang and B. Guo. Further enumerating Boolean functions of cryptographic significance. *Journal of Cryptology*, Volume 8(3), 115–122, 1995.

[25] C. K. Wu and E. Dawson. Correlation Immunity and Resiliency of Symmetric Boolean Functions. *Theoretical Computer Science*, Volume 312, 321–335, 2004.