A New Construction of Resilient Boolean Functions with High Nonlinearity

Soumen Maity
Indian Institute of Science Education and Research
Pashan, Pune 411021, INDIA
soumen@iiserpune.ac.in

Chrisil Arackaparambil
Department of Computer Science
6211 Sudikoff Laboratory Dartmouth College Hanover.
cja@cs.dartmouth.edu

Kezhasono Meyase
Tata Consultancy Services Limited, Whitefield
Bangalore - 560066, Karnataka, India
kezhasono.meyase@tcs.com

Abstract

In this paper we develop a technique that allows us to obtain new effective construction of 1-resilient Boolean functions with very good nonlinearity and autocorrelation. Our strategy to construct a 1-resilient function is based on modifying a bent function, by toggling some of its output bits. Two natural questions that arise in this context are "at least how many bits and which bits in the output of a bent function need to be changed to construct a 1-resilient Boolean function". We present an algorithm which determines a minimum number of bits of a bent function that need to be changed to construct a 1-resilient Boolean function. We also present a technique to compute points whose output in the bent function need to be modified to get a 1-resilient function. In particular, the technique is applied upto 14-variable functions and we show that the construction provides 1-resilient functions reaching currently best known nonlinearity and achieving very low autocorrelation absolute indicator values which were not known earlier.

Keywords: Autocorrelation, Bent Function, Boolean Function, Nonlinearity, Resiliency.

1 Introduction

One of the most general types of stream cipher systems is several Linear Feedback Shift Registers (LFSRs) combined by a nonlinear Boolean function. This function must satisfy certain criteria to resist different attacks (in particular, correlation attack suggested by Siegenthaler [17] and different types of linear attacks). Important properties of Boolean functions for use in stream cipher are balancedness, high order resiliency, high algebraic degree, and high nonlinearity. Constructions of Boolean functions possessing a good combination of these properties have been proposed in [14, 15, 18, 19, 6, 21, 1, 2]. In [16, 5], it had been shown how bent functions can be modified to construct highly nonlinear balanced Boolean functions. A recent construction method [10, 11] presents modification of some output points of a bent function to construct highly nonlinear 1-resilient functions. In [10, 11], a lower bound on the minimum number of bits of a bent function that need to be modified is given. However the bound is not tight for functions with more than 10 variables. In this paper, we give a better lower bound on the minimum number of bits of a bent function that need to be changed. The bound is proved to be tight for functions up to 14 variables. Further [11] does not provide any technique to select the points whose output in the bent function need to be modified and the points are selected by computer simulation. Our main contribution here is a construction to select those points whose output in the bent function need to be modified to get a 1-resilient function. For the first time, we give a combinatorial construction which can be used to obtain a 1-resilient function for any n. In particular, we concentrate on construction of 1-resilient Boolean functions up to 14 variables with best known nonlinearity and autocorrelation. *Throughout the paper we consider the number of input variables (n) is even.* Here, we identify Maiorana-McFarland type bent functions which can be modified to get 1-resilient functions with currently best known parameters. We get 1-resilient functions with better nonlinearity and autocorrelation absolute indicator values that were not known earlier for $n = 12, 14$ variables.

1.1 Preliminaries

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

The *Hamming distance* between two binary strings S_1, S_2 is denoted by $d(S_1, S_2)$, i.e., $d(S_1, S_2) = \#(S_1 \neq S_2)$. Also the *Hamming weight* or simply the weight of a binary string S is the number of ones in S. This is denoted by $wt(S)$. An n-variable function f is said to be *balanced* if its output column in the truth table contains equal number of 0s and 1s (i.e., $wt(f) = 2^{n-1}$).

Denote addition operator over $GF(2)$ by $\oplus$. An n-variable Boolean function $f(x_1, \ldots, x_n)$ can be considered to be a multivariate polynomial over $GF(2)$. This polynomial can be expressed as a sum of product representation of all distinct k-th order products $(0 \leq k \leq n)$ of the variables. More precisely, $f(x_1, \ldots, x_n)$ can be written as

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

where the coefficients $a_0, a_{ij}, \ldots, a_{12\ldots n} \in \{0, 1\}$. This representation of f is called the *algebraic normal form* (ANF) of f. The number of variables in the highest order product term with nonzero coefficient is called the *algebraic degree*, or simply the degree of f and denoted by $deg(f)$.

Functions of degree at most one are called *affine* functions. An affine function with constant term equal to zero is called a *linear* function. The set of all n-variable affine (respectively linear) functions is denoted by $A(n)$ (respectively $L(n)$). The nonlinearity of an n-variable function f is

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

i.e., the distance from the set of all n-variable affine functions.

Let $x = (x_1, \ldots, x_n)$ and $\omega = (\omega_1, \ldots, \omega_n)$ both belong 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

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

In terms of Walsh spectrum, the nonlinearity of f is given by

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

For n even, the maximum nonlinearity of a Boolean function can be $2^{n-1} - 2^{\frac{n}{2}-1}$ and the functions possessing this nonlinearity are called bent functions [13]. Further, for a bent function f on n variables, $W_f(\omega) = \pm 2^{\frac{n}{2}}$ for all ω.

In [7], an important characterization of correlation immune and resilient functions has been presented, which we use as the definition here. A function $f(x_1, \ldots, x_n)$ is m-resilient (respectively m-th order correlation immune) iff its Walsh transform satisfies

$W_f(\omega) = 0$, for $0 \le wt(\omega) \le m$ (respectively $W_f(\omega) = 0$, for $1 \le wt(\omega) \le m$).

We use the notation used in [14, 15], by an (n, m, d, σ) function we denote an n-variable, m-resilient function with degree d and nonlinearity σ.

We will now define *restricted Walsh transform* which will be frequently used in this text. The *restricted Walsh transform* of $f(x)$ on a subset S of $\{0,1\}^n$ is a real valued function over $\{0,1\}^n$ which is defined as

$$W_f(\omega)|_S = \sum_{x \in S} (-1)^{f(x) \oplus x \cdot \omega}.$$

Now we present the following technical result.

Proposition 1 *[11] Let $S \subset \{0,1\}^n$ and $b(x), f(x)$ be two n-variable Boolean functions such that $f(x) = 1 \oplus b(x)$ when $x \in S$ and $f(x) = b(x)$ otherwise. Then $W_f(\omega) = W_b(\omega) - 2W_b(\omega)|_S$.*

Propagation Characteristics (PC) and Strict Avalanche Criteria (SAC) [12] are important properties of Boolean functions to be used in S-boxes. Further, Zhang and Zheng [22] identified related cryptographic measures called Global Avalanche Characteristics (GAC).

Let $\alpha \in \{0,1\}^n$ and f be an n-variable Boolean function. Define the autocorrelation value of 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

$$\Delta_f = \max_{\alpha \in \{0,1\}^n, \alpha \neq \overline{0}} |\Delta_f(\alpha)|.$$

A function is said to satisfy PC(k), if $\Delta_f(\alpha) = 0$ for $1 \le wt(\alpha) \le k$. Note that, for a bent function f on n variables, $\Delta_f(\alpha) = 0$ for all nonzero α, i.e., $\Delta_f = 0$.

Analysis of autocorrelation properties of correlation immune and resilient Boolean functions has gained substantial interest recently as evident from [20, 23, 8, 3]. In [8, 3], it has been identified that some well known constructions of resilient Boolean functions are not good in terms of autocorrelation properties. Since the present construction is modification of bent functions which possess the best possible autocorrelation properties, we get very good autocorrelation properties of the 1-resilient functions.

2 Main Results

In this section, we present an algorithm which determines a minimum number of bits of a bent function that need to be changed to construct a 1-resilient Boolean function. We also provide a construction that computes the points whose output in the bent function need to be modified to get a 1-resilient function. Initially we start with a simple technical result.

Proposition 2 *Let $\ell(n)$ be the minimum distance between n-variable bent and 1-resilient functions, that is*

$$\ell(n) = \min\{d(b, f) : b \text{ is a bent function }, f \text{ is a 1-resilient function }\}.$$

Then $\ell(n) \geq 2^{\frac{n}{2}-1}$.

Proof: The weight of a bent function b on n variables can take two values: $2^{n-1} + 2^{\frac{n}{2}-1}$ or $2^{n-1} - 2^{\frac{n}{2}-1}$. Note that for a balanced function f, $wt(f) = 2^{n-1}$. So we need to change at least $2^{\frac{n}{2}-1}$ points of b to get a balanced function. This shows that the distance of a bent function from the balanced functions is at least $2^{\frac{n}{2}-1}$. The 1-resilient functions are balanced by definition and hence the result. ■

For a bent function b on n variables the Walsh spectrum values are $+2^{\frac{n}{2}}$ or $-2^{\frac{n}{2}}$. *In this paper, we consider the bent functions b with $W_b(\omega) = +2^{\frac{n}{2}}$ for $0 \leq wt(\omega) \leq 1$.* Let S be a subset of $\{0,1\}^n$ and $f(x)$ be an n-variable Boolean function obtained by modifying the $b(x)$ values for $x \in S$ and keeping the other bits unchanged. That is,

$$\begin{aligned} f(x) &= 1 \oplus b(x), && \text{if } x \in S \\ &= b(x), && \text{otherwise.} \end{aligned}$$

Then from Proposition 1,

$$W_f(\omega) = W_b(\omega) - 2W_b(\omega)|_S \ \forall \ \omega,$$

and in particular,

$$W_f(\omega) = 2^{\frac{n}{2}} - 2W_b(\omega)|_S \text{ for } 0 \leq wt(\omega) \leq 1.$$

It is known that f is 1-resilient iff $W_f(\omega) = 0$ for $0 \leq wt(\omega) \leq 1$, i.e., iff

$$W_b(\omega)|_S = 2^{\frac{n}{2}-1} \text{ for } 0 \leq wt(\omega) \leq 1.$$

Thus the problem is to find a subset S of $\{0,1\}^n$ of minimum cardinality and a suitable bent function $b(x)$ that satisfy the following conditions:

$$W_b(\omega)|_S = 2^{\frac{n}{2}-1} \qquad \text{for} \quad 0 \leq wt(\omega) \leq 1 \qquad (1)$$

$$W_b(\omega) = +2^{\frac{n}{2}} \qquad \text{for} \quad 0 \leq wt(\omega) \leq 1 \qquad (2)$$

2.1 Determining a minimum number of bits of an n variable bent function that need to be modified to construct a 1-resilient function

For the convenience of the reader, we would like to write subset S as matrix $\mathbf{S}$ whose rows are the elements of S. Formally, given $S = \{x^{i_1}, x^{i_2}, \ldots, x^{i_k}\} \subseteq \{0,1\}^n$, consider the matrices

$$\mathbf{S}^{k\times n} = (x^{i_1}, x^{i_2}, \ldots, x^{i_k})^T,\ b(\mathbf{S})^{k\times 1} = (b(x^{i_1}), b(x^{i_2}), \ldots, b(x^{i_k}))^T, \text{ and}$$

$$(\mathbf{S} \oplus b(\mathbf{S}))^{k\times n} = (x^{i_1} \oplus b(x^{i_1}), x^{i_2} \oplus b(x^{i_2}), \ldots, x^{i_k} \oplus b(x^{i_k}))^T.$$

By A^T we mean transpose of a matrix A. Also by abuse of notation, $x^{i_j} \oplus b(x^{i_j})$ means the GF(2) addition (XOR) of the bit $b(x^{i_j})$ with each of the bits of x^{i_j}.

Consider Condition 1 with $wt(\omega) = 0$. If k_0 is the number of 0s in $b(S)$ and k_1 is the number of 1s in $b(S)$, then we have that, $k_0 - k_1 = 2^{\frac{n}{2}-1}$. Also, $k_0 + k_1 = k$. Solving these two equations, $k_0 = \frac{k}{2} + 2^{\frac{n}{2}-2}$ and $k_1 = \frac{k}{2} - 2^{\frac{n}{2}-2}$. Note that k is always even since k_0 and k_1 are integers. Now consider Condition 1 with $wt(\omega) = 1$. Let ω be the unit vector having a 1 in position j and 0 in all other places. Then the jth column $\left(x_j^{i_1} \oplus b(x^{i_1}), x_j^{i_2} \oplus b(x^{i_2}), \ldots x_j^{i_k} \oplus b(x^{i_k})\right)^T$ of $\mathbf{S} \oplus b(\mathbf{S})$ has $\frac{k}{2} + 2^{\frac{n}{2}-2}$ 0s and $\frac{k}{2} - 2^{\frac{n}{2}-2}$ 1s. Thus by Condition 1, we have that there are exactly $\frac{k}{2} + 2^{\frac{n}{2}-2}$ many 0's and $\frac{k}{2} - 2^{\frac{n}{2}-2}$ many 1's in $b(\mathbf{S})$ and in each column of $\mathbf{S} \oplus b(\mathbf{S})$ respectively.

Without loss of generality we assume that the first $\frac{k}{2} + 2^{\frac{n}{2}-2}$ entries of $b(\mathbf{S})$ are 0s and the last $\frac{k}{2} - 2^{\frac{n}{2}-2}$ entries are 1s. Denote the sub-matrix consisting of the first $\frac{k}{2} + 2^{\frac{n}{2}-2}$ rows of $\mathbf{S}$ as block $\mathbf{S}_0$ (the corresponding elements are in S_0) and the sub-matrix consisting of the last $\frac{k}{2} - 2^{\frac{n}{2}-2}$ rows of $\mathbf{S}$ as block $\mathbf{S}_1$ (the corresponding elements are in S_1).

$$S = S_0 \cup S_1 \text{ and } \mathbf{S} = (\mathbf{S}_0, \mathbf{S}_1)^T.$$

Since S is a set, all the rows of $\mathbf{S}$ are distinct and furthermore, as the first $\frac{k}{2} + 2^{\frac{n}{2}-2}$ entries of $b(\mathbf{S})$ are 0s and the last $\frac{k}{2} - 2^{\frac{n}{2}-2}$ entries are 1s, the rows of $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$ are distinct among themselves as are the rows of $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$. Further, the Boolean complement of any row of $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$ is not a row in $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$.

Our problem is now to construct a matrix $\mathbf{S} \oplus b(\mathbf{S}) = (\mathbf{S}_0 \oplus b(\mathbf{S}_0), \mathbf{S}_1 \oplus b(\mathbf{S}_1))^T$ satisfying the conditions (Condition 1 in matrix notation):

(a) No. of rows in $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$ is $\frac{k}{2} + 2^{\frac{n}{2}-2}$, and no. of rows in $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$ is $\frac{k}{2} - 2^{\frac{n}{2}-2}$.

(b) Weight of each column of $\mathbf{S} \oplus b(\mathbf{S})$ is $\frac{k}{2} - 2^{\frac{n}{2}-2}$.

(c) Rows of $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$ are distinct among themselves and so are the rows of $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$. Further, the Boolean complement of any row of $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$ is not in $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$.

Note that, for the above condition to be satisfied, weight of each column of $\mathbf{S} \oplus b(\mathbf{S})$ must be at least one for, if it is zero all rows of $\mathbf{S} \oplus b(\mathbf{S})$ will be zero row vectors and hence identical. So, $\frac{k}{2} - 2^{\frac{n}{2}-2} \geq 1$ which gives $k \geq 2^{\frac{n}{2}-1} + 2$.

Suppose that one such matrix $\mathbf{S} \oplus b(\mathbf{S})$ is constructed. Note that the minimum number of 1s required for the distinct rows of $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$ is at least

$$\sum_{i=1}^{r_0} i \binom{n}{i} + (r_0 + 1)\left(\frac{k}{2} + 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_0} \binom{n}{i}\right),$$

where r_0 is such that

$$\sum_{i=0}^{r_0} \binom{n}{i} \leq \frac{k}{2} + 2^{\frac{n}{2}-2} < \sum_{i=0}^{r_0+1} \binom{n}{i}$$

is satisfied (using all the rows upto weight r_0 and some of the rows with weight r_0+1). Similarly the minimum number of 1s required for the distinct rows of $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$ is at least

$$\sum_{i=1}^{r_1} i \binom{n}{i} + (r_1 + 1)\left(\frac{k}{2} - 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_1} \binom{n}{i}\right),$$

where r_1 is such that

$$\sum_{i=0}^{r_1} \binom{n}{i} \leq \frac{k}{2} - 2^{\frac{n}{2}-2} < \sum_{i=0}^{r_1+1} \binom{n}{i}$$

is satisfied.
So the minimum number of 1s required to form $\mathbf{S} \oplus b(\mathbf{S})$ is at least

$$\sum_{i=1}^{r_0} i \binom{n}{i} + (r_0+1)\left(\frac{k}{2} + 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_0} \binom{n}{i}\right) + \sum_{i=1}^{r_1} i \binom{n}{i} + (r_1+1)\left(\frac{k}{2} - 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_1} \binom{n}{i}\right)$$

where r_0 and r_1 are as above.

On the other hand, Condition 1b says there would be exactly $n \times (\frac{k}{2} - 2^{\frac{n}{2}-2})$ many 1s in $\mathbf{S} \oplus b(\mathbf{S})$ as each column contains exactly $\frac{k}{2} - 2^{\frac{n}{2}-2}$ many

1s and there are n columns. If using rows of lower weight we obtain columns of weight less than $\frac{k}{2} - 2^{\frac{n}{2}-2}$ then we may increase the weight of our rows. However, if the weight of some column is greater than $\frac{k}{2} - 2^{\frac{n}{2}-2}$ then we cannot do with k rows and must increase k. This is the basis of the next algorithm which computes a lower bound on $\ell(n)$.
The above arguments tell us that k must satisfy the following condition.

$$\sum_{i=1}^{r_0} i\binom{n}{i} + (r_0+1)\left(\frac{k}{2} + 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_0}\binom{n}{i}\right) + \sum_{i=1}^{r_1} i\binom{n}{i} + \\ (r_1+1)\left(\frac{k}{2} - 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_1}\binom{n}{i}\right) \leq n \times \left(\frac{k}{2} - 2^{\frac{n}{2}-2}\right).$$

Here is an algorithm to compute the minimum k satisfying this condition.

Algorithm 1
Input: number of variables n.
Output: number of points required k, r_0 and r_1.

1. Set $k = 2^{\frac{n}{2}-1} + 2$ and $w = 1$ where w is the weight of columns of $\mathbf{S} \oplus b(\mathbf{S})$. ($w = \frac{k}{2} - 2^{\frac{n}{2}-2}$)

2. Compute r_0 and r_1 such that

$$\sum_{i=0}^{r_0}\binom{n}{i} \leq \frac{k}{2} + 2^{\frac{n}{2}-2} < \sum_{i=0}^{r_0+1}\binom{n}{i} \text{ and } \sum_{i=0}^{r_1}\binom{n}{i} \leq \frac{k}{2} - 2^{\frac{n}{2}-2} < \sum_{i=0}^{r_1+1}\binom{n}{i}$$

   are satisfied.

3. Set minimum number of 1s in $\mathbf{S} \oplus b(\mathbf{S})$,

$$z = \sum_{i=0}^{r_0} i\binom{n}{i} + (r_0+1)\left(\frac{k}{2} + 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_0}\binom{n}{i}\right) \\ + \sum_{i=0}^{r_1} i\binom{n}{i} + (r_1+1)\left(\frac{k}{2} - 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_1}\binom{n}{i}\right).$$

4. If $z \leq n \cdot w$, stop. k is the required number of points.

5. $k = k + 2$, $w = w + 1$. ($w = \frac{k}{2} - 2^{\frac{n}{2}-2}$, so that when k increases by 2, w increases by 1.)

6. Go to step 2.

The following table illustrates the number of points k, as computed by the above algorithm for different values of n.

n	8	10	12	14	16	18	20	22	24	26
r_0+1	2	2	2	2	3	3	3	3	4	4
r_1+1	1	1	1	1	2	2	2	2	3	3
k	10	22	44	86	168	342	684	1350	2662	5430

Theorem 1 *The above algorithm gives a lower bound on the number of bits of an n-variable bent function that need to be modified to construct a 1-resilient function, that is $k \leq \ell(n)$.*

Proof: Let b be a bent function and f, a 1-resilient function such the distance between b and f is $\ell(n)$, that is the number of points where b and f give different values is $\ell(n)$.

Let S be the set of points where b and f give different values ($|S| = \ell(n)$).

$$S = \{x \in \{0,1\}^n : b(x) \neq f(x)\}.$$

Then by modifying the bits of b corresponding to elements of S we obtain f. Hence $\ell(n)$ must satisfy the necessary Conditions 1a, 1b & 1c for k. The above algorithm computes the minimum k that satisfies Conditions 1a, 1b & 1c, and hence $k \leq \ell(n)$. ■

The algorithm in the next section computes k points satisfying Conditions 1a, 1b & 1c. Then, to get a 1-resilient function, we will need a bent function b which has the desired output values at the points given by the next algorithm, namely, b must be such that

$$b(x) = \begin{cases} 0, & \text{for } x \in S_0 \\ 1, & \text{for } x \in S_1 \end{cases}$$

Since the class of bent functions is very large, it may be conjectured that we can find a bent function satisfying the above condition. Then, the above algorithm gives us the minimum distance since it is already a lower bound.

For $n = 8, 10, 12, 14$ we identify Maiorana-McFarland type bent functions which can be modified to get 1-resilient functions, using the points given by the next algorithm. This shows that the bound given by the above algorithm is tight and is the minimum distance for these values of n.

2.2 Finding points whose output bits in the bent function need to be modified to get 1-resilient function

To find an n variable 1-resilient function from a bent function, we modify output for certain points of the bent function. Essentially, we look only at the $S \oplus b(S)$ matrix where S is the set of points to be modified and b is the bent function. Our aim is to find a set of points S satisfying Condition 1. Here we give a construction of S with the number of rows k given by Algorithm 1, $\frac{k}{2} + 2^{\frac{n}{2}-2}$ rows in $S_0 \oplus b(S_0)$ and $\frac{k}{2} - 2^{\frac{n}{2}-2}$ in $S_1 \oplus b(S_1)$. Our technique is as suggested by Algorithm 1.

Since we want S satisfying Conditions 1a, 1b & 1c with minimum number of rows, we use rows of minimum weight. If we use rows of higher weight, column weight $\frac{k}{2} - 2^{\frac{n}{2}-2}$ also increases so that we need more number of points k. First we construct the matrix $S_0 \oplus b(S_0)$. Matrix $S_1 \oplus b(S_1)$ is also constructed in a similar manner. As in Algorithm 1, to construct $S_0 \oplus b(S_0)$ we use all points of weight $\leq r_0$. These rows will have a uniform column weight $\frac{\sum_{i=0}^{r_0} i\binom{n}{i}}{n}$. Further we need $m_0 = \frac{k}{2} + 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_0} \binom{n}{i}$ rows of weight $r_0 + 1$.
We want to select the remaining rows of weight $r_0 + 1$ such that the weight of all n columns is more or less uniform to keep the total weight of each column in $S \oplus b(S)$ as $\frac{k}{2} - 2^{\frac{n}{2}-2}$. Let $w_0 = \left\lfloor \frac{m_0 \times (r_0+1)}{n} \right\rfloor$ and t_0 be the remainder so that $m_0 \times (r_0 + 1) = n \times w_0 + t_0$. By a careful selection of m_0 rows of weight $r_0 + 1$, we can get t_0 columns of weight $w_0 + 1$ and $n - t_0$ columns of weight w_0, that is the column weights do not differ by more than 1. We now need a few definitions.

The *circular shift operator* ROT rotates the Boolean vector x by d positions. That is, if $y = \mathrm{ROT}(x, d)$ then y is the vector obtained by a circular shift of the bits in x by d positions. For example, $\mathrm{ROT}(x, 2) = (0,0,0,1,0,1,0,0)$, where $x = (0,0,0,0,0,1,0,1)$.
A set C of Boolean row vectors is called a *circular block* if for any $x \in C$, $C = \{\ \mathrm{ROT}(x, d) : d \text{ is an integer}\}$. That is, the vectors in C are identical up to circular shifting and C is closed under circular shifting.

Example 1 *When* $x = (0,0,0,0,0,0,1,1)$ *in the above definition, we get the circular block* $C =$
$\{(0,0,0,0,0,0,1,1), (0,0,0,0,0,1,1,0), (0,0,0,0,1,1,0,0), (0,0,0,1,1,0,0,0), (0,0,1,1,0,0,0,0), (0,1,1,0,0,0,0,0), (1,1,0,0,0,0,0,0), (1,0,0,0,0,0,0,1)\}$.

An important characteristic of circular blocks is that all columns are of equal weight. Note that $|C| \leq n$. Also, a circular block may not have n vectors.

Example 2 *When* $x = (0,0,0,1,0,0,0,1)$, *we get the circular block* $C = \{(0,0,0,1,0,0,0,1), (0,0,1,0,0,0,1,0), (0,1,0,0,0,1,0,0), (1,0,0,0,1,0,0,0)\}$. *with* $|C| = 4$.

A *generator* of a circular block C is the vector $g \in C$ which appears first in lexicographic order. In other words, it is the smallest number when the

vectors in C are interpreted as binary numbers. For the circular block in Example 1 the generator is $(0,0,0,0,0,0,1,1)$ while for that in Example 2 the generator is $(0,0,0,1,0,0,0,1)$. Note that the Least Significant Bit (LSB) of a generator is always 1. We will obtain circular block C by $\leq n$ circular shifts of it's generator g. That is $C = \{\ \mathrm{ROT}(g,d) : d \leq n\}$
The next algorithm constructs a matrix T with m rows of weight r.

We can represent a point x by a set containing, the positions of the r 1s in the point. For example, $x = (0,0,1,0,0,0,1,0)$ is represented by the set $\{2,6\}$, which we denote by the ordered list $\hat{x} = [2,6]$.

Since the LSB of a generator is always 1, the number of generators is $\leq \binom{n-1}{r-1}$ and their ordered list representations will be a selection of $r-1$ positions from the set $\{2,3,\ldots,n\}$ in addition to the LSB. However, all such selections will not give a generator. Still, we can easily check if such a selection gives a generator or not.

First, note that the ROT operation for the ordered list representation is just addition modulo n to each of the list elements.

To check if a selection $[p_1,p_2,\ldots,p_{r-1}]$ gives a generator, note that the corresponding vector with the LSB is $\hat{x} = [1,p_1,p_2,\ldots,p_{r-1}]$. Now, if this vector is not a generator then it must have a generator g in it's circular block. Then $g < x$ in the lexicographic ordering. Also, g can be obtained from x by circular shift. Since LSB in g is 1, when rotating x to get g, a 1 in x initially at position $p_i, 1 \leq i \leq (r-1)$ will come at the LSB. This corresponds to a rotation by $n - p_i + 1$ bit shifts.

So to check if a vector x given by the selection is a generator or not we just have to rotate each of the $(r-1)$ 1s at positions $p_i, 1 \leq i \leq (r-1)$ by $n-p_i+1$ and check if the resulting vector $y_i < x$ in lexicographic ordering. If no $y_i < x$ then x is a generator. Thus checking if x is a generator requires $O(r)$ operations.

Now to get a circular block, we get a selection $[p_1,p_2,\ldots,p_{r-1}]$ from $\{2,3,\ldots,n\}$ and check if the vector x corresponding to $\hat{x} = [1,p_1,p_2,\ldots,p_{r-1}]$ is a generator. Then we construct a circular block using the generator.

The issue is, when constructing m rows of weight r, after using some number of circular blocks, we may find that the number of rows required is less than the number of rows in the next circular block. If we use only some rows of the next circular block to complete m rows we may find that the column weights differ by more than 1. To overcome this, we reserve a generator g_r and do not use it at first. Only when we find that the remaining number of rows to be constructed is $\leq n$ we use g_r. g_r is chosen as follows:

1. **If r divides n:** $\hat{g_r} = [1,2,\ldots,r]$. We generate points from this generator as shown in the next example:

   Example 3 *For* $n = 8$ *with* $r = 2$, $\hat{g_r} = [1,2] = (0,0,0,0,0,0,1,1)$

and the points are generated in the following sequence:
$(0,0,0,0,0,0,1,1)$, $(0,0,0,0,1,1,0,0)$, $(0,0,1,1,0,0,0,0)$, $(1,1,0,0,0,0,0,0)$, $(0,0,0,0,0,1,1,0)$, $(0,0,0,1,1,0,0,0)$, $(0,1,1,0,0,0,0,0)$, $(1,0,0,0,0,0,0,1)$.

The following pseudocode generates these n rows, $\{x[1], x[2], \ldots, x[n]\}$

```
for i in {0,1, ..., r-1}:
    for j in {0, 1, ..., n/r-1}:
        x[i+j] = ROT(g, j*r + i)
```

2. **If r does not divide n:** g_r is chosen by distributing the $n - r$ 0s equally among the 1s. Points are generated by successive circular shifts as shown below:

   Example 4 *For $n = 8$ with $r = 3$, $\hat{g_r} = [1,3,6]$ and the points are generated in the following sequence:*
   $(0,0,1,0,0,1,0,1)$, $(0,1,0,0,1,0,1,0)$, $(1,0,0,1,0,1,0,0)$, $(0,0,1,0,1,0,0,1)$, $(0,1,0,1,0,0,1,0)$, $(1,0,1,0,0,1,0,0)$, $(0,1,0,0,1,0,0,1)$, $(1,0,0,1,0,0,1,0)$.

Note that in each case, after every row, column weights do not differ by more than 1.

Algorithm 2
Input: number of variables n, number of rows m, row weight r.
Output: $m \times n$ matrix T having t columns of weight $w + 1$ and $n - t$ columns of weight w, with $w = \lfloor \frac{m \times r}{n} \rfloor$ and t the remainder so that $m \times r = n \times w + t$.

1. Initialize T as the empty matrix. $T = ()$.
2. Compute the reserved generator g_r accordingly as r divides n or not.
3. Initialize $m' = 0$, the number of rows of T constructed so far.
4. If $m - m' \leq n$, go to Step (8).
5. Compute a new generator g, $g \neq g_r$.
6. Construct the circular block C by repeatedly circular shifting g (Let the number of vectors in C be d).
7. $T = (T, C)^T$, $m' = m' + d$ and go to Step (4).
8. Use the reserved generator g_r to construct the partial block D with $m - m'$ rows.
9. $T = (T, D)^T$.

Theorem 2 *Algorithm 2 runs correctly in $O(r \cdot \binom{n-1}{r-1})$ time.*

Proof: Since we ensure that at the end of the algorithm, column weights do not differ by more than 1, and we use rows of minimum possible weights, we get the column weights as desired and the algorithm runs correctly.

Further, the number of generators is $\leq \binom{n-1}{r-1}$, obtained by a selection of $r-1$ positions from the set $\{2, 3, \ldots, n\}$ in addition to the LSB. Checking if a selection gives a generator or not requires $O(r)$ operations. So Algorithm 2 requires $O(r \cdot \binom{n-1}{r-1})$ time. ■

Now, we construct $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$ by first including all points of weight upto r_0 and then using Algorithm 2 to find the remaining $m_0 = \frac{k}{2} + 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_0} \binom{n}{i}$ points of weight $r_0 + 1$. Similarly for $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$, include all points of weight upto r_1 and then use Algorithm 2 to find the remaining $m_1 = \frac{k}{2} - 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_1} \binom{n}{i}$ points of weight $r_1 + 1$.

After constructing $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$ and $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$ in this manner, column weights in the two matrices do not differ by more than 1. But the $\mathbf{S} \oplus b(\mathbf{S})$ matrix thus obtained may have column weights differing by more than 1. To avoid this we permute the columns of $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$ so that the columns of higher weight are identified with the columns of lower weight of $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$. Then in the resulting $\mathbf{S} \oplus b(\mathbf{S})$ matrix, column weights do not differ by more than 1.

Now to satisfy Conditions 1a, 1b & 1c we need only that the columns weights are equal. To do this we need to add exactly one 1 in certain columns, z' in number (note that $z' < n$). This is not too difficult since we have a large number of rows ($> 2^{\frac{n}{2}-1}$).

Construction 1

Input: number of variables n, number of points k, r_0 and r_1 from Algorithm 1.

Output: $k \times n$ matrix $\mathbf{S}$ satisfying Conditions 1a, 1b & 1c.

1. Add all rows of weight r_0 and r_1 to the matrices $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$ and $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$ respectively.

2. Compute $m_0 = \frac{k}{2} + 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_0} \binom{n}{i}$ and $m_1 = \frac{k}{2} - 2^{\frac{n}{2}-2} - \sum_{i=0}^{r_1} \binom{n}{i}$.

3. Use Algorithm 2 with inputs n, m_0, $r_0 + 1$ to get matrix T_0.
   $\mathbf{S}_0 \oplus b(\mathbf{S}_0) = (\mathbf{S}_0 \oplus b(\mathbf{S}_0), T_0)^T$.

4. Use Algorithm 2 with inputs n, m_1, $r_1 + 1$ to get matrix T_1.
   $\mathbf{S}_1 \oplus b(\mathbf{S}_1) = (\mathbf{S}_1 \oplus b(\mathbf{S}_1), T_1)^T$.

5. Permute columns of $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$ suitably so that columns of higher weight of the $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$ matrix are identified with that of lower weight columns of $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$.

6. Accommodate the remaining z' ones in the two matrices in a suitable manner.

7. $\mathbf{S}_0 = \mathbf{S}_0 \oplus b(\mathbf{S}_0)$ and $\mathbf{S}_1 = 1 \oplus (\mathbf{S}_1 \oplus b(\mathbf{S}_1))$.

8. $\mathbf{S} = (\mathbf{S}_0, \mathbf{S}_1)^T$.

Theorem 3 *Construction 1 finds inputs whose output in the bent function need to be modified to get* 1*-resilient function.*

Proof: To show that Conditions 1a, 1b & 1c hold for the matrix $\mathbf{S}$ constructed as above, note that k is obtained from Algorithm 1, at the end of which $z \leq w \cdot n$. Algorithm 2 ensures that column weights of $\mathbf{S} \oplus b(\mathbf{S})$ do not differ by more than 1, using rows of minimum possible weights. So in Construction 1 after adding the remaining z' ones, each column has weight exactly $\frac{k}{2} - 2^{\frac{n}{2}-2}$.

Algorithm 2 constructs the matrices using distinct rows. We now only need to show that the Boolean complement of any row of $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$ is not in $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$. Weight of any row in $\mathbf{S}_0 \oplus b(\mathbf{S}_0)$ is $\leq r_0 + 1$ so it's complement must have weight $\geq n - r_0 - 1$. So if the rows in $\mathbf{S}_1 \oplus b(\mathbf{S}_1)$ are of weight $< n - r_0 - 1$ then we are through. Here we assume that $r_1 < n - r_0$ or equivalently $r_0 + r_1 < n$. That this is a reasonable assumption can be observed from the table giving values for r_0 and r_1 up to $n = 26$. We can see that r_0 and r_1 grow very slowly as compared to n.

Since $r_1 \leq r_0$, the next theorem holds.

Theorem 4 *Construction 1 requires* $O(r_0 \cdot \binom{n-1}{r_0-1})$ *time.*

3 Construction of the 1-resilient function

Now that we have the set S we need to construct the bent function b satisfying Conditions 1 & 2.

The original Maiorana-McFarland class of bent function is as follows [4]. Consider n-variable Boolean functions on (X, Y), where $X, Y \in \{0,1\}^{\frac{n}{2}}$ of the form $b(X, Y) = X \cdot \pi(Y) + g(Y)$ where π is a permutation on $\{0,1\}^{\frac{n}{2}}$ and g is any Boolean function on $\frac{n}{2}$ variables. Then b is a bent function. For a fixed value of Y, $X \cdot \pi(Y)$ can be seen as a linear function on X and $g(Y)$ is constant either 0 or 1 over all X. So that the function b can be seen as a concatenation of $2^{\frac{n}{2}}$ distinct (upto complementation) affine function on $\frac{n}{2}$ variables.

We require a bent function $b(x)$ on n variables satisfying the condition that $b(x) = 0$ for $x \in S_0$ and $b(x) = 1$ for $x \in S_1$. We have to decide what permutations π on $\{0,1\}^{\frac{n}{2}}$ and what kind of functions g on $\{0,1\}^{\frac{n}{2}}$ we can take such that the conditions on b are satisfied. Let us fix the notation and ordering of input variables as $x = (x_1, x_2, \ldots, x_n)$, $X = (X_1, X_2, \ldots, X_{\frac{n}{2}}) = (x_1, x_2, \ldots, x_{\frac{n}{2}})$, and $Y = (Y_1, Y_2, \ldots, Y_{\frac{n}{2}}) = (x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n)$.

Now, we look at Condition 2. It is easy to see that for $0 \leq wt(\omega) \leq 1$ a bent function will have the restricted Walsh spectrum value

$W_b(\omega)|_{(X,Y),X\in\{0,1\}^{\frac{n}{2}}} = 0$ for all values of Y except for one Y where it is $\pm 2^{\frac{n}{2}}$. We want $W_b(\omega) = +2^{\frac{n}{2}}$ at that Y. This will happen only when $X\cdot\pi(Y)\oplus g(Y)\oplus x\cdot\omega = 0$ or $X\cdot\pi(Y)\oplus g(Y) = x\cdot\omega$ at that Y. We ensure this by conditions as below:

1. For $wt(\omega) = 0$ we want for one Y, $X\cdot\pi(Y)\oplus g(Y) = x\cdot\omega$. That is $X\cdot\pi(Y)\oplus g(Y) = 0$. Not that here, X is variable and takes all possible values. Equating the constant parts, we get $g(Y) = 0$. Equating the variable parts, we get $X\cdot\pi(Y) = 0$ so that $\pi(Y) = (0,0,\ldots,0)$.

   So we require for a particular Y, $\pi(Y) = (0,0,\ldots,0)$ and $g(Y) = 0$.

2. For ω having a 1 in the latter half, we want for one Y, $X\cdot\pi(Y)\oplus g(Y) = x\cdot\omega$. But $x\cdot\omega = x_i$, with $\frac{n}{2} < i \le n$, which is constant. So $X\cdot\pi(Y)$ must be constant giving $\pi(Y) = (0,0,\ldots,0)$. This must hold for each such value of ω so that $g(Y) = x_{\frac{n}{2}+1} = x_{\frac{n}{2}+2} = \ldots = x_n$

   So we require either $Y = (0,0,\ldots 0)$ with $\pi(Y) = (0,0,\ldots,0)$ and $g(Y) = 0$ OR $Y = (1,1,\ldots 1)$ with $\pi(Y) = (0,0,\ldots,0)$ and $g(Y) = 1$.

3. The last case is for ω having a 1 in the former half, we want for one Y, $X\cdot\pi(Y)\oplus g(Y) = x\cdot\omega$. But $x\cdot\omega = x_i$, with $1\le i\le\frac{n}{2}$. Equating constant parts, $g(Y) = 0$, so that $X\cdot\pi(Y) = x_i$. We get $wt(\pi(Y)) = 1$ with the 1 in the i'th position.

   So our condition is: for $\pi(Y)\in\{(1,0,\ldots,0),(0,1,\ldots,0),(0,0,\ldots,1))\}$, $g(Y) = 0$.

We can combine the first two parts above to give the following two conditions:

$$\pi(0,0,\ldots,0) = (0,0,\ldots,0) \text{ and } g(0,0,\ldots,0) = 0. \tag{3}$$

$$\text{For} \quad \pi(Y)\in\{(1,0,\ldots,0),(0,1,\ldots,0),(0,0,\ldots,1))\}, g(Y) = 0. \tag{4}$$

3.1 The 8-variable 1-resilient Functions

Algorithm 1 gives us $k = 10$. We compute $r_0 = 1$ and $r_1 = 0$ so that $m_0 = 0$ and $m_1 = 0$. Using all points of weight ≤ 1 (as $r_0 = 1$) we get,

$$\mathbf{S_0} = \mathbf{S_0}\oplus b(\mathbf{S_0}) = \begin{pmatrix} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0 \end{pmatrix}$$

Using the point of weight 0 as $r_1 = 0$ we get,

$$\mathbf{S_1} \oplus b(\mathbf{S_1}) = (\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ )$$

so that

$$\mathbf{S_1} = (\ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ ).$$

There are no remaining ones so $z' = 0$. We define g and π below which satisfies the conditions for construction of the required bent function.

1. $g(Y) = \begin{cases} 1 & \text{if } Y = (1,1,1,1) \\ 0 & \text{if } (X,Y) \in S \text{ and } Y \neq (1,1,1,1). \end{cases}$

2. $\pi(Y) = Y$

If we take π and g as above then we get the value of $b(x) = b(X,Y) = X \cdot \pi(Y) + g(Y)$ to be zero when $x \in S_0$ and one when $x \in S_1$. Also π and g satisfy Conditions 3 & 4. A 1-resilient function $f(x)$ is obtained as follows.

$$\begin{aligned} f(x) &= 1 \oplus b(x), \text{ if } x \in S \\ &= b(x), \text{ otherwise.} \end{aligned}$$

We checked that the nonlinearity of f is 116, algebraic degree is 6 and $\Delta_f = 24$. A count on the number of bent functions satisfying Conditions 1 & 2 is given in [11].

3.2 The 10-variable 1-resilient Functions

Algorithm 1 gives us $k = 22$. We compute $r_0 = 1$ and $r_1 = 0$ so that $m_0 = 8$ and $m_1 = 2$.
Using all points of weight ≤ 1 (as $r_0 = 1$) and the reserved generator $[12] = (0000000011)$ (as $m_0 - m' = 8 < 10$ and 2 divides n in Algorithm 2) we get

$$\mathbf{S_0} = \mathbf{S_0} \oplus b(\mathbf{S_0}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Using the point of weight zero and the reserved generator $[1] = (0000000001)$ (as $m_1 - m' = 2 < 10 = n$ and 1 divides n) we get

$$\mathbf{S_1} \oplus b(\mathbf{S_1}) = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&0&0&0 \end{pmatrix}.$$

We find that $z' = 2$. We add these in the rows of $\mathbf{S_1} \oplus b(\mathbf{S_1})$ to get

$$\mathbf{S_1} \oplus b(\mathbf{S_1}) = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&1 \\ 0&1&1&0&0&0&0&0&0&0 \end{pmatrix}.$$

$$\mathbf{S_1} = \mathbf{1} \oplus \mathbf{S_1} \oplus b(\mathbf{S_1}) = \begin{pmatrix} 1&1&1&1&1&1&1&1&1&1 \\ 0&1&1&1&1&1&1&1&1&0 \\ 1&0&0&1&1&1&1&1&1&1 \end{pmatrix}.$$

Taking $g(Y) = 0$ and $\pi(Y) = Y$. we get the value of $b(x) = b(X,Y) = X \cdot \pi(Y) + g(Y)$ to be zero when $x \in S_0$ and one when $x \in S_1$. Also π and g satisfy Conditions 3 & 4. A 1-resilient function $f(x)$ is obtained as before. The nonlinearity of f is 488, algebraic degree is 8 and $\Delta_f = 48$.

3.3 The 12-variable 1-resilient Functions

Algorithm 1 gives us $k = 44$. We compute $r_0 = 1$ and $r_1 = 0$ so that $m_0 = 25$ and $m_1 = 5$.

Using all points of weight ≤ 1, generators $[1,3]$, $[1,4]$ and the reserved generator $[1,2]$ (2 divides n) for points of weight 2 we get $\mathbf{S_0} = \mathbf{S_0} \oplus b(\mathbf{S_0})$ with 38 rows. Using the point of weight zero and the reserved generator $[1] = (000000000001)$ (1 divides n) we get after permutation

$$\mathbf{S_1} \oplus b(\mathbf{S_1}) = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \end{pmatrix}.$$

We find that $z' = 5$. We add these in the rows of $\mathbf{S_1} \oplus b(\mathbf{S_1})$ to get

$$\mathbf{S_1} \oplus b(\mathbf{S_1}) = \begin{pmatrix} 0&0&1&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 1&1&0&0&0&1&0&0&0&0&0&0 \end{pmatrix}.$$

$$\mathbf{S_1} = 1 \oplus \mathbf{S_1} \oplus b(\mathbf{S_1}) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Taking $g(Y) = 0$ and $\pi(Y) = Y$. we get the value of $b(x) = b(X, Y) = X \cdot \pi(Y) + g(Y)$ to be zero when $x \in S_0$ and one when $x \in S_1$. Also π and g satisfy Conditions 3 & 4. A 1-resilient function $f(x)$ is obtained as before. The nonlinearity of f is 2000 and algebraic degree is 10. The function f we constructed here has $\Delta_f = 104$ and this is the best known value which is achieved for the first time here.

3.4 The 14-variable 1-resilient Functions

Algorithm 1 gives us $k = 86$. We compute $r_0 = 1$ and $r_1 = 0$ so that $m_0 = 60$ and $m_1 = 10$.

Using all points of weight ≤ 1, generators [1,3], [1,4], [1,5], [1,6] and the reserved generator [1,2] (2 divides n) for points of weight 2 we get $\mathbf{S_0} = \mathbf{S_0} \oplus b(\mathbf{S_0})$ with 75 rows.

Using the point of weight zero and the reserved generator [1] (1 divides n) we get after permutation

$$\mathbf{S_1} \oplus b(\mathbf{S_1}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

We find that $z' = 3$. We add these in the rows of $\mathbf{S_1} \oplus b(\mathbf{S_1})$ to get

$$\mathbf{S_1} \oplus b(\mathbf{S_1}) = \begin{pmatrix} 0&0&0&0&0&0&0&1&1&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0 \end{pmatrix}.$$

$$\mathbf{S_1} = 1 \oplus \mathbf{S_1} \oplus b(\mathbf{S_1}) = \begin{pmatrix} 1&1&1&1&1&1&1&0&0&0&1&1&1&1 \\ 1&1&1&1&1&0&1&1&1&1&1&1&1&1 \\ 1&1&1&1&0&1&1&1&1&1&1&1&1&1 \\ 1&1&1&0&1&1&1&1&1&1&1&1&1&1 \\ 1&1&0&1&1&1&1&1&1&1&1&1&1&1 \\ 1&0&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&0 \\ 1&1&1&1&1&1&1&1&1&1&1&1&0&1 \\ 1&1&1&1&1&1&1&1&1&1&1&0&1&1 \\ 1&1&1&1&1&1&1&1&1&1&0&1&1&1 \end{pmatrix}.$$

We define g and π below which satisfies the conditions for construction of the required bent function.

1. $g(Y) = \begin{cases} 1, & if\ Y \in A \\ 0, & otherwise. \end{cases}$
   where $A =$
   $\{(0,0,0,1,1,1,1)\ (1,1,1,1,1,1,0), (1,1,1,1,1,0,1), (1,1,1,1,0,1,1), (1,1,1,0,1,1,1), (1,1,0,1,1,1,1), (1,0,1,1,1,1,1), (0,1,1,1,1,1,1), (1,1,1,1,1,1,1)\}$

2. $\pi(Y) = Y$.

If we take π and g as above then we get the value of $b(x) = b(X,Y) = X \cdot \pi(Y) + g(Y)$ to be zero when $x \in S_0$ and one when $x \in S_1$. Also π and g satisfy Conditions 3 & 4. A 1-resilient function $f(x)$ is obtained as before. The function f we constructed here has nonlinearity 8098 and this is the best known nonlinearity value which is achieved for the first time here.

4 Conclusions

In this paper we present a strategy to construct highly nonlinear 1-resilient functions by modifying some output bits of a bent function. We present a good lower bound on the minimum number of bits of a bent function needed to be modified. We have shown that the bound is tight for functions upto 14-input variables. One interesting problem is to study whether Algorithm 1 provides the minimum distance between n-variable bent and 1-resilient functions for all even n. We present an algorithm to generate the points whose output in the bent function require to be modified. For $n = 8, 10, 12, 14$ we identify Maiorana-McFarland type bent functions which can be modified to get 1-resilient functions, using the points given by Construction 1. This shows that the bound given by Algorithm 1 is tight and is the minimum distance for these values of n. Further our construction is superior to [11] in terms of the nonlinearity (we get better nonlinearity for 14 variables) and autocorrelation absolute indicator (we get 1-resilient functions with absolute indicator value that was not known earlier for 12 variable). Since the class of bent functions is very large, it may be conjectured that it is always possible to identify bent functions which can be modified to get 1-resilient functions, using the points given by Construction 1.

References

[1] C. Carlet. A larger Class of Cryptographic Boolean Functions via a Study of the Maiorana-McFarland Constructions. In *Advances in Cryptology - CRYPTO 2002*, number 2442 in Lecture Notes in Computer Science, pages 549–564. Springer Verlag, 2002.

[2] C. Carlet and P. Sarkar. Spectral domain analysis of correlation immune and resilient Boolean functions. *Finite Fields and Its Applications*, 8(1):120–130, January 2002.

[3] P. Charpin and E. Pasalic. On Propagation Characteristics of Resilient Functions. In *SAC 2002*, number 2595 in Lecture Notes in Computer Science, pages 175–195. Springer-Verlag, 2003.

[4] J. F. Dillon. Elementary Hadamard Difference sets. PhD Thesis, University of Maryland, 1974.

[5] H. Dobbertin. Construction of bent functions and balanced Boolean functions with high nonlinearity. In *Fast Software Encryption-FSE 1994*, number 1008 in Lecture Notes in Computer Science, pages 61–74. Springer-Verlag, 1994.

[6] M. Fedorova and Y. V. Tarannikov. On the constructing of highly nonlinear resilient Boolean functions by means of special matrices. In *Progress in Cryptology - INDOCRYPT 2001*, number 2247 in Lecture Notes in Computer Science, pages 254–266. Springer Verlag, 2001.

[7] X. Guo-Zhen and J. Massey. A spectral characterization of correlation immune combining functions. *IEEE Transactions on Information Theory*, 34(3):569–571, May 1988.

[8] S. Maitra. Autocorrelation Properties of correlation immune Boolean functions. In *INDOCRYPT 2001*, number 2247 in Lecture Notes in Computer Science, pages 242-253. Springer Verlag, 2001.

[9] Maity, S., Chrisil Arackaparambil and Kezhasono Meyase, Construction of 1-resilient Boolean functions with very good nonlinearity, *Sequences and Their Applications - SETA 2006*, Lecture Notes in Computer Science, Vol 4086, Springer-Verlag, Vol. 4086, 2006, pp. 417-431.

[10] S. Maity and T. Johansson. Construction of Cryptographically Important Boolean Functions. In *Progress in Cryptology - INDOCRYPT 2002*, number 2551 in Lecture Notes in Computer Science, pages 234–245. Springer Verlag, 2002.

[11] S. Maity and S. Maitra. Minimum distance between bent and 1-resilient functions. In *Fast Software Encryption-FSE 2004*, number 3017 in Lecture Notes in Computer Science, pages 143-160 . Springer-Verlag, 2004.

[12] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle. Propagation characteristics of Boolean functions. In *Advances in Cryptology - EUROCRYPT'90*, number 473 in Lecture Notes in Computer Science, pages 161–173. Springer-Verlag, 1991.

[13] O. S. Rothaus. On bent functions. *Journal of Combinatorial Theory, Series A*, 20:300–305, 1976.

[14] 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, 2000.

[15] P. Sarkar and S. Maitra. Nonlinearity bounds and constructions of resilient Boolean functions. In *Advances in Cryptology - CRYPTO 2000*, number 1880 in Lecture Notes in Computer Science, pages 515–532. Springer Verlag, 2000.

[16] J. Seberry, X. M. Zhang, and Y. Zheng. Nonlinearly balanced Boolean functions and their propagation characteristics. In *Advances in Cryptology - CRYPTO'93*, number 773 in Lecture Notes in Computer Science, pages 49–60. Springer-Verlag, 1994.

[17] T. Siegenthaler. Decrypting a Class of Straem Ciphers Using Ciphertext Only *IEEE Transactions on Computer* Vol. C-34 (1), 81–85, 1985.

[18] Y. V. Tarannikov. On resilient Boolean functions with maximum possible nonlinearity. In *Progress in Cryptology - INDOCRYPT 2000*, number 1977 in Lecture Notes in Computer Science, pages 19–30. Springer Verlag, 2000.

[19] Y. V. Tarannikov. New constructions of resilient Boolean functions with maximal nonlinearity. In *FSE 2001*, number 2355 in Lecture Notes in Computer Science, pages 66–77. Springer Verlag, 2001.

[20] Y. V. Tarannikov, P. Korolev and A. Botev. Autocorrelation coefficients and correlation immunity of Boolean functions. In *ASIACRYPT 2001*, number 2248 in Lecture Notes in Computer Science, pages 460–479. Springer Verlag, 2001.

[21] Y. Zheng and X. M. Zhang. Improved upper bound on the nonlinearity of high order correlation immune functions. In *SAC 2000*, number 2012 in Lecture Notes in Computer Science, pages 264–274. Springer Verlag, 2000.

[22] 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.

[23] Y. Zheng and X. M. Zhang. On relationships among propagation degree, nonlinearity and correlation immunity. In *Advances in Cryptology - ASIACRYPT 2000*, number 1976 in Lecture Notes in Computer Science, pages 470–482. Springer Verlag, 2000.