On the Relationship between Resilient Boolean Functions and Linear Branch Number of S-boxes

Sumanta Sarkar[1], Kalikinkar Mandal[2], and Dhiman Saha[3]

[1] TCS Innovation Labs, Hyderabad, INDIA
sumanta.sarkar1@tcs.com
[2] University of Waterloo, Waterloo, CANADA
kmandal@uwaterloo.ca
[3] Indian Institute of Technology, Bhilai, INDIA
dhiman@iitbhilai.ac.in

Abstract. Differential branch number and linear branch number are critical for the security of symmetric ciphers. The recent trend in the designs like PRESENT block cipher, ASCON authenticated encryption shows that applying S-boxes that have nontrivial differential and linear branch number can significantly reduce the number of rounds. As we see in the literature that the class of 4×4 S-boxes have been well-analysed, however, a little is known about the $n \times n$ S-boxes for $n \geq 5$. For instance, the complete classification of 5×5 affine equivalent S-boxes is still unknown. Therefore, it is challenging to obtain "the best" S-boxes with dimension ≥ 5 that can be used in symmetric cipher designs. In this article, we present a novel approach to construct S-boxes that identifies classes of $n \times n$ S-boxes ($n = 5, 6$) with differential branch number 3 and linear branch number 3, and ensures other cryptographic properties. To the best of our knowledge, we are the first to report 6×6 S-boxes with linear branch number 3, differential branch number 3, and with other good cryptographic properties such as nonlinearity 24 and differential uniformity 4.

Keywords: S-box, Resilient Boolean function, linear branch number, differential branch number, nonlinearity, differential uniformity, lightweight cipher.

1 Introduction

A basic design principle of a block cipher consists of confusion and diffusion as suggested by Shannon [15]. The confusion layer makes the relation between the key and the ciphertext as complex as possible, whereas the diffusion layer spreads the plaintext statistics across the ciphertext. Over the years, several block ciphers have been constructed, and the most notable one is AES [6]. Later on, a lot of interest grew in lightweight cryptography, as the requirement of security of Internet of Things (IoT) was felt. In this regard, lightweight block ciphers like PRESENT [4], CLEFIA [17] were standardized by ISO/IEC 29192. NIST too has taken an initiative to standardize lightweight cryptography algorithms [10]. With the advent of lightweight cryptography, a lot of effort has been devoted to find lightweight S-boxes with good cryptographic properties. There is also a considerable amount of literature available on lightweight MDS matrices which are used to build the diffusion layer.

In practice, S-boxes are used to build the confusion layer. An $n \times m$ S-box is a mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$. In most cases S-boxes with $n = m$ are used, however there are some S-boxes where $n \neq m$, for instance DES [7] uses $(6, 4)$ S-boxes. In order to build a secure block cipher, the S-box should have high nonlinearity, high differential uniformity, high degree. Additionally, to reduce the number of rounds, it is desired that the number of active S-boxes increase as quickly as possible, and to achieve this goal, the S-boxes should have high differential and linear branch numbers. In case of AES, the number of active S-boxes increases due to the choice of ShiftRow and MixColumn operation. However, in the case of PRESENT or ASCON, this depends largely on the branch numbers of the S-box itself. PRESENT has removed the usual diffusion layer that is normally implemented by an MDS matrix. Thus saving a considerable amount of hardware cost. It uses a 4×4 S-box that has the following properties: differential branch number is 3; differential uniformity is 4; nonlinearity is 4; algebraic degree is 3.

The round function of PRESENT is comprised of 16 such S-boxes followed by a bit-permutation $L : \mathbb{F}_2^{64} \to \mathbb{F}_2^{64}$, where the role of the bit-permutation is to mix up the outputs of the S-boxes which become the input to the next round. As a bit-permutation can be implemented by wires only, this reduces the hardware implementation cost (in gates) for the entire design.

In [14], the upper bounds on linear and differential branch numbers were derived. For an $n \times n$ S-box $\mathcal{S}$, its linear branch number, denoted by $\mathcal{LBN}(\mathcal{S})$, satisfies $\mathcal{LBN}(\mathcal{S}) \leq n-1$, and its differential branch number, denoted by $\mathcal{DBN}(\mathcal{S})$, satisfies $\mathcal{DBN}(\mathcal{S}) \leq \lceil \frac{2n}{3} \rceil$. It is also interesting to note that ASCON [8] and SYCON [13] use 5×5 S-boxes that have differential branch number 3 and linear branch number 3. The block cipher SC2000 [16] used a 6×6 S-box, however, it has both linear and differential branch number 2.

1.1 Our Contribution

It is easy to observe that the trivial lower bound for differential and linear branch number is 2. However, constructing an S-box that has both differential and linear branch number greater than 2 along with other cryptographic properties is a nontrivial task. In this article, we investigate the problem of constructing S-boxes with both differential and linear branch number greater than 2. Our idea is to apply the relationship between resilient Boolean functions and the linear branch number to construct S-boxes that ensure linear branch number at least 3. In Section 3, we present Algorithm 1 that produces S-boxes with linear branch number 3. Further, we present Algorithm 2 which produces S-boxes with linear branch number 3 and differential branch number 3. Then, in Section 4, we consider some known classes of permutations over $\mathbb{F}_{2^6}$ with well-known cryptographic properties, and applying Algorithms 1 and 2, we obtain 6×6 S-boxes with both linear and differential branch number 3, nonlinearity 24, and differential uniformity 4. The hardware implementation cost of such S-boxes are also provided. To the best of our knowledge, this is the first time such 6×6 S-boxes with good cryptographic properties are reported. We also show how to construct efficient 5×5 S-boxes that have low hardware implementation overheads.

2 Preliminaries

Denote by $\mathbb{F}_2$, the finite field of two elements $\{0, 1\}$. Let $\mathbb{F}_{2^n}$ be the finite field with 2^n elements and $\mathbb{F}_2^n$ be the n-dimensional vector space over $\mathbb{F}_2$. For any $x \in \mathbb{F}_2^n$, the Hamming weight of x, denoted by $wt(x)$ is the number of 1's in x. Bitwise XOR is denoted by $\oplus$ and for any $x, y \in \mathbb{F}_2^n$ their dot product $x \cdot y$ is simply the usual inner product $x_0y_0 \oplus \cdots \oplus x_{n-1}y_{n-1}$. An $n \times n$ S-box is a permutation $\mathcal{S} : \mathbb{F}_2^n \to \mathbb{F}_2^n$. We denote by $\mathbb{GL}(n, \mathbb{F}_2)$, the set of all linear permutations of $\mathbb{F}_2^n$. Clearly $\mathbb{GL}(n, \mathbb{F}_2)$ is a proper subset of the set of all permutations over $\mathbb{F}_2^n$. The S-box $\mathcal{S}$ can also be viewed as an n-tuple of Boolean functions in n-variable, i.e., $\mathcal{S} = (f_1, \ldots, f_n)$, where $f_i : \mathbb{F}_2^n \to \mathbb{F}_2$, here f_i is called a *coordinate* function of $\mathcal{S}$ and any linear combination of coordinate functions is called a *component* function of $\mathcal{S}$.

For a secure design, S-box needs to satisfy several properties such as high nonlinearity, high differential uniformity, high algebraic degree, etc [5]. Basically the nonlinearity of $\mathcal{S}$ is the minimum nonlinearity that is obtained by any component function of $\mathcal{S}$. The algebraic degree of $\mathcal{S}$ is the maximum degree of its coordinate functions. Let $\mathcal{S}(\delta, \Delta) = \{\#x \in \mathbb{F}_2^n : \mathcal{S}(x) \oplus \mathcal{S}(x \oplus \delta) = \Delta\}$. Then the differential uniformity of $\mathcal{S}$ is defined as

$$\mathcal{DU}_{\mathcal{S}} = \max_{\delta \neq 0, \Delta} \{\mathcal{S}(\delta, \Delta)\}.$$

Lower the differential uniformity, better the resistance is against the differential attack [3]. The least possible differential uniformity is 2, and S-boxes with 2 differential uniformity are called Almost Perfect Nonlinear (APN) functions. The differential distribution table (DDT) of $\mathcal{S}$ is a matrix of order $2^n \times 2^n$ constructed as follows: the (δ, Δ)-th element of DDT is $\mathcal{S}(\delta, \Delta)$. In Table 1, we present the difference distribution table of the S-box $\mathcal{S} = 408235\text{B}719\text{A}6\text{CDEF}$.

Table 1. DDT of S-Box 408235B719A6CDEF

$\delta \backslash \Delta$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	4	0	0	2	0	2	0	2	0	2	0	4	0	0	0
2	0	0	8	0	0	0	0	0	2	0	0	2	2	0	0	2
3	0	0	0	6	2	0	2	2	2	0	0	0	0	0	2	0
4	0	0	0	2	4	4	0	2	0	2	0	0	0	2	0	0
5	0	2	0	2	0	4	0	0	2	2	0	0	2	0	0	2
6	0	0	0	0	0	0	4	4	0	0	0	4	0	0	0	4
7	0	2	0	2	0	0	0	4	0	0	2	2	0	2	2	0
8	0	0	2	0	2	4	0	0	4	2	0	0	0	0	0	2
9	0	2	0	0	2	0	0	0	2	4	0	0	0	2	4	0
A	0	0	0	0	0	0	2	2	0	2	4	2	0	2	2	0
B	0	2	2	2	0	0	2	0	0	0	2	4	2	0	0	0
C	0	4	2	0	0	0	2	0	2	0	0	0	2	4	0	0
D	0	0	0	0	2	0	0	2	0	2	2	0	4	4	0	0
E	0	0	0	2	2	0	0	0	0	2	4	0	0	0	2	4
F	0	0	2	0	0	4	2	0	0	0	0	2	0	0	4	2

We now recall the notions of correlation matrices, linear and differential branch numbers. Consider an $n \times n$ S-box $\mathcal{S}$. For any $\alpha, \beta \in \mathbb{F}_2^n$ the correlation coefficient of $\mathcal{S}$ with respect to (α, β) is given by

$$\mathtt{C}_{\mathcal{S}}(\alpha, \beta) = \sum_{x \in \mathbb{F}_2^n} (-1)^{\beta \cdot \mathcal{S}(x) + \alpha \cdot x}. \tag{1}$$

If $\mathcal{S}(x) = (f_1(x), \ldots, f_n(x))$, then $\beta \cdot \mathcal{S}(x)$ is a Boolean function that is a linear combination of $\{f_1(x), \ldots, f_n(x)\}$, and $\alpha \cdot x$ is a linear Boolean function of the form $\ell_1 x_1 \oplus \ldots \oplus \ell_n x_n$.

It is easy to see that $-2^n \leq \mathtt{C}_{\mathcal{S}}(\alpha, \beta) \leq 2^n$. The correlation matrix $\mathtt{C}_{\mathcal{S}}$ of $\mathcal{S}$ is the $2^n \times 2^n$ matrix indexed by $\alpha, \beta \in \mathbb{F}_2^n$ in which the entry in the cell (α, β) is given by $\mathtt{C}_{\mathcal{S}}(\alpha, \beta)$:

$$\mathtt{C}_{\mathcal{S}} = [C_{\alpha,\beta}]_{2^n \times 2^n} \qquad \text{where } C_{\alpha,\beta} = \mathtt{C}_{\mathcal{S}}(\alpha, \beta) \tag{2}$$

Next we recall some definitions related to the differential branch number and linear branch number.

Definition 1. *For any $n \times n$ S-box $\mathcal{S}$, its differential branch number (respectively linear branch number) is denoted by $\mathcal{DBN}(\mathcal{S})$ (respectively $\mathcal{LBN}(\mathcal{S})$) and defined as*

$$\mathcal{DBN}(\mathcal{S}) := \min_{x,x' \in \mathbb{F}_2^n,\, x \neq x'} \{wt(x \oplus x') + wt(\mathcal{S}(x) \oplus \mathcal{S}(x'))\},$$

and

$$\mathcal{LBN}(\mathcal{S}) := \min_{\alpha,\beta \in \mathbb{F}_2^n,\, \mathtt{C}_{\mathcal{S}}(\alpha,\beta) \neq 0} \{wt(\alpha) + wt(\beta)\},$$

where $\mathtt{C}_{\mathcal{S}}(\alpha, \beta)$ is the correlation coefficient as in (1).

If $\mathcal{S}$ is a linear permutation of $\mathbb{F}_2^n$, then there exists a binary $n \times n$ invertible matrix M such that $\mathcal{S}(x) = \mathrm{M}x$ for every $x \in \mathbb{F}_2^n$. In this case $\mathcal{DBN}(\mathcal{S})$ and $\mathcal{LBN}(\mathcal{S})$ can be simplified as done in the following fact taken from [6, Ch 9].

Fact 1 *Let $\mathcal{S}$ be a linear permutation of $\mathbb{F}_2^n$ given by $\mathrm{M} \in \mathbb{GL}(\mathrm{n}, \mathbb{F}_2)$. Then,*

$$\mathcal{DBN}(\mathcal{S}) = \min_{\alpha \in \mathbb{F}_2^n, \alpha \neq 0} \{wt(\alpha) + wt(\mathrm{M}\alpha)\}$$

$$\mathcal{LBN}(\mathcal{S}) = \min_{\alpha \in \mathbb{F}_2^n, \alpha \neq 0} \{wt(\alpha) + wt(\mathrm{M}^{\mathrm{t}}\alpha)\}.$$

For any S-box $\mathcal{S}$ it is easy to see that $\mathcal{DBN}(\mathcal{S})$ is ≥ 2 and $\mathcal{LBN}(\mathcal{S}) \geq 2$. Also,

$$\mathcal{DBN}(\mathcal{S}) = \mathcal{DBN}(\mathcal{S}^{-1}) \qquad \text{and} \qquad \mathcal{LBN}(\mathcal{S}) = \mathcal{LBN}(\mathcal{S}^{-1}).$$

It is interesting to note that the differential branch number is related to DDT. The differential branch number can be redefined as

$$\mathcal{DBN}(\mathcal{S}) := \min_{\delta \neq 0, \Delta \neq 0, \mathcal{D}_{\mathcal{S}}(\delta,\Delta) \neq 0} \{wt(\delta) + wt(\Delta)\}.$$

For example, it is clear from the DDT (Table 1), the differential branch number of 408235B719A6CDEF is 2.

One important classification of S-boxes is partitioning them into *affine equivalence* classes. For sake of completeness, we define the affine equivalence of S-boxes below.

Definition 2 (Affine Equivalence). *Let $\mathcal{S}, \mathcal{S}'$ be two permutations of $\mathbb{F}_2^n$. We say that $\mathcal{S}$ is affine equivalent to $\mathcal{S}'$ if there exist matrices A, B $\in \mathbb{GL}(n, \mathbb{F}_2)$, and $c, d \in \mathbb{F}_2^n$ such that*

$$\mathcal{S}'(x) = B \cdot \mathcal{S}[A\,x \oplus c] \oplus d, \qquad \text{for all } x \in \mathbb{F}_2^n. \tag{3}$$

Affine equivalence preserves some cryptographic properties of S-boxes, such as differential uniformity, nonlinearity, degree, but it does not preserve branch numbers in general. For instance, the two S-boxes $\mathcal{S} = \texttt{C56B90AD3EF84712}$ and $\mathcal{S}' = \texttt{CD6310A5BE784F92}$ are affine equivalent, but they have different differential branch number: $\mathcal{DBN}(\mathcal{S}) = 3$, whereas $\mathcal{DBN}(\mathcal{S}') = 2$. The S-box $\mathcal{S}$ is used in PRESENT.

On the other hand, if A and B are permutation matrices[4] then the corresponding affine equivalence class preserves the branch number [12]. We state this as the following lemma.

Lemma 1. *If $\mathcal{S}$ and $\mathcal{S}_1$ are two affine equivalent $n \times n$ S-boxes, such that $\mathcal{S}_1(x) = B \cdot \mathcal{S}[A\,x \oplus c] \oplus d$, for all $x \in \mathbb{F}_2^n$, where A and B are $n \times n$ permutation matrices, and $c, d \in \mathbb{F}_2^n$, then $\mathcal{DBN}(\mathcal{S}) = \mathcal{DBN}(\mathcal{S}_1)$ and $\mathcal{LBN}(\mathcal{S}) = \mathcal{LBN}(\mathcal{S}_1)$.*

3 Relation between resilient Boolean function and linear branch number

Let us define the resilient Boolean function first.

Definition 3. *A Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is called r-resilient if*

$$\sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus \alpha \cdot x} = 0,$$

for all $\alpha \in \mathbb{F}_2^n$ such that $0 \leq wt(\alpha) \leq r$.

The relation between resilient Boolean functions and linear branch number was first noticed in [14]. We reiterate it here for the sake of clarity.

Lemma 2. *Let $\mathcal{S} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an S-box. Then all the coordinate functions are ($\mathcal{LBN}(\mathcal{S}) - 2$)-resilient and also the algebraic* $\deg(\mathcal{S}) \leq n - \mathcal{LBN}(\mathcal{S}) + 1$

Proof. Let us assume $\mathcal{LBN}(\mathcal{S}) = r$. Then for all β with $wt(\beta) = 1$ and for all α with $1 \leq wt(\alpha) \leq r - 2$

$$\mathtt{C}_{\mathcal{S}}(\alpha, \beta) = \sum_{x \in \mathbb{F}_2^n} (-1)^{\beta \cdot \mathcal{S}(x) + \alpha \cdot x} = 0.$$

As $wt(\beta) = 1$, so $\beta \cdot \mathcal{S}(x)$ is a coordinate function of $\mathcal{S}$, and every coordinate function of an S-box is necessarily balanced.

The degree of an n-variable r-resilient function is bounded by $n - 1 - r$, which proves the second part of the lemma. □

Suppose the S-box S has $\mathcal{LBN}(S) = 3$, then every coordinate function of S will be 1-resilient. The differential and linear branch numbers are not invariant in an affine equivalence class unlike nonlinearity or differential uniformity. So if by some construction method one can get an S-box with high nonlinearity and high differential uniformity, but with $\mathcal{DBN} = \mathcal{LBN} = 2$, then one naive idea would be to search in the affine equivalent class of that S-box for $\mathcal{DBN} \geq 3$ and $\mathcal{LBN} \geq 3$.

[4] A matrix obtained by permuting rows (or columns) of an identity matrix.

However, this search may not conclude as the size of affine equivalence is huge to exhaust for dimensions more than 4. For instance, cardinality of $\mathbb{GL}(5, \mathbb{F}_2)$ is around 2^{24}. In this case we can apply the necessary condition that every coordinate function should be resilient, to reject many S-boxes without checking their whole affine equivalence class. Based on this, we develop an algorithm which takes an S-box as an input and efficiently checks the possibility of the existence of any S-box with linear branch number 3 in its affine equivalence class.

Algorithm 1 Construction of S-boxes with linear branch number 3

```
    Input: S-box $\mathcal{S} : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$
    Output: $\emptyset$ or S-boxes with linear branch number 3
 1: Construct $\mathcal{B}$ as the set of all possible nonzero component functions of $\mathcal{S}$
 2: Extract the subset $\mathcal{R} \subset \mathcal{B}$ which is the set of 1-resilient component functions of $\mathcal{S}$.
 3: if $|\mathcal{R}| < n$ then
 4:     return $\emptyset$
 5: else
 6:     $\mathcal{T} = \emptyset$                                        ▷ Empty Set
 7:     Form a new set $\{f_1, \ldots, f_n\}$; $f_i \in \mathcal{R}$
 8:     if $\mathcal{U} = (f_1, \ldots, f_n)$ is a permutation of $\mathbb{F}_2^n$ then
 9:         $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathcal{U}\}$ and go to Step 7
10: return $\mathcal{T}$
```

Algorithm 1 is an heuristic one that we apply to construct a collection of affine equivalent S-boxes with linear branch number 3. The input S-box $\mathcal{S}$ could have linear branch number 3 or 2. However, the effectiveness of this algorithm can be realized if we take $\mathcal{S}$ such that $\mathcal{LBN}(\mathcal{S}) = 2$, then we show how it can lead to an affine equivalent S-box(es) with $\mathcal{LBN}(\mathcal{S}) = 3$. First it forms all possible 1-resilient component functions of $\mathcal{S}$ out of all $2^n - 1$ component functions in Step 2. If $\mathcal{R}$ does not have at least n numbers of 1-resilient functions, the algorithm quits as, to ensure the linear branch number 3, all n-coordinate functions must be 1-resilient. On the other hand, if there are at least n numbers of 1-resilient component functions available, then every time n of them are chosen as coordinate functions in Step 7. Then it all remains to check whether $\mathcal{U}$ is a permutation or not. If yes, it is an S-box with $\mathcal{LBN} = 3$. Obviously, $\mathcal{U}$ is an affine equivalent of $\mathcal{S}$ and thus the nonlinearity and differential uniformity are preserved in $\mathcal{U}$.

We now apply the degree bound in order to show a nonexistence result related to 4 × 4 S-boxes.

Theorem 1. *There is no* 4×4 *S-box with* $\mathcal{LBN} = 3$ *and nonlinearity nonzero.*

Proof. If a 4×4 S-box $\mathcal{S}$ has $\mathcal{LBN} = 3$, then by Lemma 2, we know $\deg(\mathcal{S}) \leq 2$. There are 302 affine equivalent 4×4 S-boxes, and among them only 6 classes have degree 2. Each of these 6 S-boxes have nonlinearity zero. Thus the proof. □

The degree bound of S-box with $\mathcal{LBN} = 3$ has been very effective in the above proof. Out of 302 affine equivalent S-boxes, there are 244 S-boxes with nonzero nonlinearity and nontrivial differential uniformity (< 16). Then in order to prove the same result, one had to check the full class of each of these 244 S-boxes.

3.1 Adding $\mathcal{DBN} = 3$ criterion

It is clear that ensuring $\mathcal{LBN}(\mathcal{S}) = 3$ will harden the linear cryptanalysis [9], on the other hand it is also desired that $\mathcal{DBN}(\mathcal{S}) > 2$, which gives better protection against the differential cryptanalysis.

For lightweight ciphers, 4×4 S-boxes have been very popular choice, for example, PRESENT [4], SKINNY [2], and GIFT [1]. In [14], the upper bounds on linear and differential branch number were derived. For an $n \times n$ S-box $\mathcal{S}$, $\mathcal{LBN}(\mathcal{S}) \leq n - 1$, and $\mathcal{DBN}(\mathcal{S}) \leq \lceil \frac{2n}{3} \rceil$. Thus for 4×4 S-boxes the maximum $\mathcal{LBN}$ and $\mathcal{DBN}$ values are exactly 3. However, as per Theorem 1, there is no scope of using 4×4 S-box with $\mathcal{LBN} = 3$. Lightweight ciphers namely ASCON [8] and SYCON [13] have used 5×5 S-boxes with $\mathcal{LBN} = \mathcal{DBN} = 3$.

We now introduce another heuristic in Algorithm 2 which takes an S-box with $\mathcal{LBN} = 3$ and $\mathcal{DBN} = 2$, and then applies linear transformation on both input and output to get an affine equivalent S-box which preserves the linear branch number, however, makes $\mathcal{DBN} = 3$.

Algorithm 2 Construction of S-boxes with linear branch number 3 and differential branch number 3

```
Input: S-box $\mathcal{S} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ with $\mathcal{LBN}(\mathcal{S}) = 3$
Output: $\emptyset$ or S-boxes with linear branch number 3 and differential branch number 3
Take a set of matrices $\mathcal{M}_1 \subset \mathbb{GL}(n, \mathbb{F}_2)$
$\mathcal{A}_S = \emptyset$
for $A$ in $\mathcal{M}_1$ do
    if $\mathcal{LBN}(\mathcal{S} \circ A) = 3$ then
        $\mathcal{A}_S \leftarrow \mathcal{A}_S \cup A$
Take a set of matrices $\mathcal{M}_2 \subset \mathbb{GL}(n, \mathbb{F}_2)$
$\mathcal{B}_S = \emptyset$
for $B$ in $\mathcal{M}_2$ do
    if $\mathcal{DBN}(B \circ \mathcal{S}) = 3$ then
        $\mathcal{B}_S \leftarrow \mathcal{B}_S \cup B$
$\mathcal{T} = \emptyset$
for $A \in \mathcal{A}_S$ do
    for $B$ in $\mathcal{B}_S$ do
        if $\mathcal{LBN}(B \circ \mathcal{S} \circ A) = 3$ and $\mathcal{DBN}(B \circ \mathcal{S} \circ A) = 3$ then
            $\mathcal{T} \leftarrow \mathcal{T} \cup \{B \circ \mathcal{S} \circ A\}$
return $\mathcal{T}$
```

One can apply Algorithm 1 to get an S-box with $\mathcal{LBN} = 3$, which will be the input to Algorithm 2. One naive way to look for S-boxes with $\mathcal{LBN} = \mathcal{DBN} = 3$ is to search in an affine equivalence class of an S-box. As the dimension grows, the size of the affine equivalence class also grows making it impossible to exhaust. So we apply the heuristic that takes a subset of matrices A of $\mathbb{GL}(n, \mathbb{F}_2)$ which acts on the input variables of $\mathcal{S}$ with $\mathcal{LBN}(S) = 3$, and also preserves the $\mathcal{LBN}$. Then it takes another subset of matrices B of $\mathbb{GL}(n, \mathbb{F}_2)$ which acts on the output of $\mathcal{S} \circ A$, and also increases $\mathcal{DBN}$ to $\mathcal{DBN} = 3$. After that combining these two submatrices, if $\mathcal{LBN}(B \circ \mathcal{S} \circ A) = \mathcal{DBN}(B \circ \mathcal{S} \circ A) = 3$, then $B \circ \mathcal{S} \circ A$ is added to the collection $\mathcal{T}$. For a fixed $\mathcal{M}_1$ and $\mathcal{M}_2$, the worst-case time complexity of Algorithm 2 in terms of bit operations is $O(|\mathcal{M}_1| \cdot |\mathcal{M}_2| \cdot (n^2 2^n + 2^{2n} + n 2^{3n})) = O(|\mathcal{M}_1| \cdot |\mathcal{M}_2| \cdot n 2^{3n})$ where $O(n^2 2^n)$ is the time complexity of constructing an affine equivalent S-box, $O(2^{2n})$

is for computing differential branch number, and $O(n2^{3n})$ is for computing linear branch number.

We do not want to consider $\mathcal{M}_1 = \mathcal{M}_2 = \mathbb{GL}(n, \mathbb{F}_2)$, as the complexity will be too high. For instance, $|\mathbb{GL}(5, \mathbb{F}_2)| \approx 2^{24}$, so one can imagine the vastness involved in this case. We carefully choose some subclass of $\mathbb{GL}(n, \mathbb{F}_2)$. We also want to keep the $n \times n$ identity matrix $\mathcal{I}_{n\times n}$ in both $\mathcal{M}_1$ and $\mathcal{M}_2$. If the input S-box is already a lightweight one, then ideally we want a minimum overhead for the input and output linear transformation so that the overall implementation does not scale much. In that case one of A and B being equal to $\mathcal{I}_{n\times n}$ will serve the purpose.

4 Leveraging the known classes of S-boxes with good cryptographic properties

As we aimed at constructing S-boxes with good cryptographic properties along with high branch numbers, we leverage the existing classes of S-boxes that are known to have good cryptographic properties. We achieve this by applying Algorithms 1 and 2 to the known classes of S-boxes with good cryptographic properties. We start with the power functions, which are defined over finite fields $\mathbb{F}_{2^n}$ and are of the form $F(x) = x^d$ for some d such that F is a permutation of $\mathbb{F}_{2^n}$. There are several known classes of power functions which have good cryptographic properties. We consider the simplest one, the Gold functions.

4.1 Gold function

Definition 4. *Let n be odd. The function $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$, defined by*

$$F(x) = x^{2^k+1}$$

is known as Gold function where $\gcd(k, n) = 1$.

Note that the Gold functions are quadratic. For odd n and $\gcd(k, n) = 1$, then it becomes APN [11]. In the case of even n and $\gcd(2^k + 1, 2^n - 1) = 1$, the Gold functions can have the best differential uniform 4. For example, for $k = 2$ and $n = 6$, Gold function $F(x) = x^5$ is 4 differentially uniform, the nonlinearity is 24, which is also high. Therefore, this function is interesting to the cipher designers. However, $\mathcal{LBN}(F) = \mathcal{DBN}(F) = 2$, and that makes it a weaker choice for the design. We now apply Algorithm 1 hoping that it would yield an S-box that has $\mathcal{LBN} = 3$. In order to do that first we consider all the nonzero component functions of F. As this is defined over finite fields, so the set of component functions is $C_F = \{Tr(\lambda F(x)) : \lambda \in \mathbb{F}_{2^n}^*\}$, where $\mathbb{F}_{2^n}^*$ consists of all nonzero elements from $\mathbb{F}_{2^n}$. Note that the Trace function (Tr) is defined by

$$Tr(x) = x + x^2 + \ldots + x^{2^{n-1}},$$

and the range of Tr is in $\mathbb{F}_2$, that means it is a Boolean function.

By computing C_F for $F(x) = x^5$, we notice that there exist only 10 1-resilient component functions. For As there are 6 coordinate functions for a 6×6 S-box, thus enough combinations of 1-resilient functions are available to construct an affine equivalent S-box with $\mathcal{LBN} = 3$ as per Algorithm 1. The only thing that we need to care about is that while taking 6 component functions as the coordinate functions, they should form an S-box, that is a permutation of $\mathbb{F}_2^6$. Then we are ensured to have a 6×6 S-box with linear branch number 3 and nonlinearity 24. Following is an example of a set of 6 1-resilient component functions of x^5 defined over $\mathbb{F}_{2^6}$, which yield the S-box with $\mathcal{LBN} = 3$.

$$\begin{aligned}
y_0 &= x_0x_5 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_4 \oplus x_2x_5 \oplus x_3x_4 \oplus x_3x_5 \oplus x_3 \oplus x_4x_5 \oplus x_4 \oplus x_5\\
y_1 &= x_0x_4 \oplus x_0 \oplus x_1x_2 \oplus x_1x_4 \oplus x_1 \oplus x_2x_4 \oplus x_2x_5 \oplus x_3 \oplus x_4 \oplus x_5\\
y_2 &= x_0x_1 \oplus x_0x_3 \oplus x_1x_3 \oplus x_1x_4 \oplus x_1 \oplus x_2x_3 \oplus x_2x_4 \oplus x_3x_4 \oplus x_4 \oplus x_5\\
y_3 &= x_0x_2 \oplus x_0x_3 \oplus x_0x_4 \oplus x_1x_2 \oplus x_1x_4 \oplus x_1 \oplus x_2x_3 \oplus x_2x_5 \oplus x_3x_4 \oplus x_3x_5\\
&\quad \oplus x_3 \oplus x_4x_5 \oplus x_4 \oplus x_5\\
y_4 &= x_0x_1 \oplus x_0x_2 \oplus x_0 \oplus x_1x_2 \oplus x_1x_4 \oplus x_1x_5 \oplus x_1 \oplus x_2x_3 \oplus x_2x_4 \oplus x_3x_4 \oplus x_4x_5 \oplus x_4\\
y_5 &= x_0x_1 \oplus x_0x_3 \oplus x_0x_4 \oplus x_1 \oplus x_2 \oplus x_3x_4 \oplus x_3x_5 \oplus x_3 \oplus x_4x_5 \oplus x_4
\end{aligned}$$

4.2 6×6 Quadratic S-box with $\mathcal{LBN} = 3$ and $\mathcal{DBN} = 3$

We consider the class $F(x) = x^{10} + \alpha x$ defined over $\mathbb{F}_{2^6}$. First we find by using Algorithm 1, 6×6 S-boxes with $\mathcal{LBN} = 3$ of this form. Then we apply Algorithm 2 to get an S-box with $\mathcal{LBN} = \mathcal{DBN} = 3$. Essentially we will form a subclass of affine equivalent S-boxes. In order to shorten the search effort, we choose a small class of invertible matrices. We choose 6×6 nonsingular Toeplitz matrices.

Definition 5. *A matrix is called Toeplitz if every descending diagonal from left to right is constant.*

Following is an example of a Toeplitz matrix of order $n \times n$

$$\begin{bmatrix} a_0 & a_1 & a_2 & \dots & a_{n-2} & a_{n-1} \\ a_{-1} & a_0 & a_1 & \dots & a_{n-3} & a_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{-(n-1)} & a_{-(n-2)} & a_{-(n-3)} & \dots & a_{-1} & a_0 \end{bmatrix}. \tag{4}$$

A Toeplitz matrix is defined by its first row and first column. For instance $\{a_0, a_1, \dots, a_{n-1}, a_{-1}, a_{-2}, \dots, a_{-(n-1)}\}$ defines the Toeplitz matrix as in (4).

We consider the S-box $\mathcal{S} = [0, 54, 47, 48, 3, 5, 24, 55, 23, 56, 32, 38, 46, 49, 45, 27, 61, 14, 62, 36, 16, 19, 39, 13, 1, 43, 26, 25, 22, 12, 57, 10, 30, 58, 17, 28, 9, 29, 50, 15, 8, 53, 31, 11, 37, 40, 6, 34, 2, 35, 33, 41, 59, 42, 44, 20, 63, 7, 4, 21, 60, 52, 51, 18]$. For this S-box we have $\mathcal{LBN}(\mathcal{S}) = 3$, but $\mathcal{DBN}(\mathcal{S}) = 2$. The hardware cost of this S-box is 79.68 GE.

We take $\mathcal{S}$ as an input to Algorithm 2. Let $\mathbb{T}_6$ denote the set of all nonsingular 6×6 Toeplitz matrices. Then $|\mathbb{T}_6| = 1024$. To keep the search space small, we choose $\mathcal{M}_1 = \mathcal{M}_2 = \mathbb{T}_6$. As a result we see that among the possible Toeplitz matrices, by applying the following Toeplitz matrix M on the output, it is possible to obtain an affine equivalent S-box $\mathcal{S}'$ with $\mathcal{LBN}(\mathcal{S}') = 3$ and $\mathcal{DBN}(\mathcal{S}') = 3$.

$$\mathrm{M} = \begin{bmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1\,0 \\ 1\,0\,0\,0\,0\,1 \end{bmatrix}. \tag{5}$$

In Table 2, we present this S-box $\mathcal{S}'$, and its hardware cost is 86.71 GE. The cost of this derived S-box does not scale much as it comes through by applying such a low cost matrix M in Equation (5).

Table 2. 6×6 S-box $\mathcal{S}'$ with $\mathcal{LBN}(\mathcal{S}') = 3$, $\mathcal{DBN}(\mathcal{S}') = 3$. nonlinearity $= 24$, differential uniformity $= 4$, degree $= 2$

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$\mathcal{S}(x)$	0	58	47	28	3	29	24	15	23	53	32	11	46	40	45	34	61	35	62	41	16	42	39	20	1	7	26	21	22	52	57	18	30	54	17	48	9	5	50	55	8	56	31	38	37	49	6	27	2	14	33	36	59	19	44	13	63	43	4	25	60	12	51	10

We would like to point out that as far as we know the existence of 6×6 S-box that have significant cryptographic properties and at the same time has $\mathcal{LBN} = \mathcal{DBN} = 3$ has never been reported. This function is the first in the literature.

4.3 Cubic function

Next we consider the function $F : \mathbb{F}_{2^6} \to \mathbb{F}_{2^6}$, defined by

$$F(x) = x^{19},$$

S-box derived from this function has degree 3, nonlinearity 24 and differential uniformity 4. We apply the same technique as in Algorithm 1, however, we extend the input by considering the Extended Affine (EA) equivalent[5] class of x^{19}. Then we get a 6×6 S-box $\mathcal{S}$ such that $\mathcal{LBN}(\mathcal{S}) = 3$ and $\mathcal{DBN}(S) = 3$. The S-box is given in Table 3 and its hardware cost is 158.59 GE.

Table 3. 6×6 S-box $\mathcal{S}$ with $\mathcal{LBN}(\mathcal{S}) = 3$, $\mathcal{DBN}(\mathcal{S}) = 3$. nonlinearity $= 24$, differential uniformity $= 4$, degree $= 3$

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$\mathcal{S}(x)$	0	45	48	15	58	32	14	49	13	7	41	12	3	54	55	26	42	25	22	34	60	38	53	31	21	51	4	24	27	28	43	33	39	19	30	63	16	1	59	8	57	62	29	50	6	44	36	17	23	10	56	37	9	47	5	20	40	52	35	2	18	61	46	11

[5] $\mathcal{S}$ and $\mathcal{S}'$ are EA equivalent if $\mathcal{S}' = B \circ \mathcal{S} \circ A + L$ for some linear function L and affine permutations A and B.

Table 4. 5×5 S-box $\mathcal{S}$ with $\mathcal{LBN}(\mathcal{S}) = 3$, $\mathcal{DBN}(\mathcal{S}) = 3$. nonlinearity = 8, differential uniformity = 8, degree = 2

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$\mathcal{S}(x)$	0	14	27	17	22	24	15	5	30	25	7	4	11	12	16	19	3	9	8	6	21	31	28	18	20	23	29	26	1	2	10	13

$y_0 = x_0x_3 \oplus x_1 \oplus x_2x_3 \oplus x_3x_4 \oplus x_4$
$y_1 = x_0 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1 \oplus x_2x_3 \oplus x_2 \oplus x_3 \oplus x_4$
$y_2 = x_0x_1 \oplus x_0x_4 \oplus x_0 \oplus x_2 \oplus x_3$
$y_3 = x_0x_3 \oplus x_0 \oplus x_1 \oplus x_3x_4 \oplus x_3$
$y_4 = x_1x_4 \oplus x_1 \oplus x_2 \oplus x_3$

Hardware cost = 38.28 GE

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$\mathcal{S}(x)$	0	27	22	15	14	17	24	5	30	7	11	16	25	4	12	19	3	8	21	28	9	6	31	18	20	29	1	10	23	26	2	13

$y_0 = x_0 \oplus x_1x_3 \oplus x_2x_3 \oplus x_3x_4 \oplus x_4$
$y_1 = x_0x_1 \oplus x_0x_3 \oplus x_0 \oplus x_1x_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4$
$y_2 = x_0x_2 \oplus x_1 \oplus x_2x_4 \oplus x_2 \oplus x_3$
$y_3 = x_0 \oplus x_2x_3 \oplus x_2 \oplus x_3x_4 \oplus x_3$
$y_4 = x_0x_4 \oplus x_0 \oplus x_1 \oplus x_3$

Hardware cost = 38.28 GE

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$\mathcal{S}(x)$	0	27	14	17	22	15	24	5	30	7	25	4	11	16	12	19	3	8	9	6	21	28	31	18	20	29	23	26	1	10	2	13

$y_0 = x_0 \oplus x_1x_3 \oplus x_2x_3 \oplus x_3x_4 \oplus x_4$
$y_1 = x_0x_2 \oplus x_0x_3 \oplus x_0 \oplus x_1 \oplus x_2x_3 \oplus x_2 \oplus x_3 \oplus x_4$
$y_2 = x_0x_1 \oplus x_1x_4 \oplus x_1 \oplus x_2 \oplus x_3$
$y_3 = x_0 \oplus x_1x_3 \oplus x_1 \oplus x_3x_4 \oplus x_3$
$y_4 = x_0x_4 \oplus x_0 \oplus x_2 \oplus x_3$

Hardware cost = 44.53 GE

5 Lightweight 5×5 S-boxes

In another direction we use the relationship between resilient Boolean functions and S-boxes with linear branch number 3 to come up with lightweight S-boxes with $\mathcal{LBN} = 3$ and $\mathcal{DBN} = 3$. In particular we restrict to 5×5 quadratic S-boxes.

We give a little tweak to Algorithm 1, and start with 1-resilient functions in the first place in order to get S-boxes with $\mathcal{LBN} = 3$. First we enumerate all the 5-variable quadratic 1-resilient Boolean functions, the total number of such functions is 2868. However, if we restrict the quadratic resilient functions to have only 4 terms, then there are 285 such functions; and with only 5 terms, there are 330 such functions. We consider this type of resilient functions and combine them as coordinate functions of S-boxes. If these coordinate functions form an S-box, then $\mathcal{LBN} = 3$ is ensured. Then we apply the idea of Algorithm 2, however with a little tweak. We randomly select matrices from $\mathbb{GL}(5, \mathbb{F}_2)$, and apply on input and output of the S-boxes in order to get S-boxes with $\mathcal{LBN} = 3$ and $\mathcal{DBN} = 3$. We give some examples of S-boxes that are obtained in this way in Table 4. We also measure their hardware cost. Implementation of all the S-boxes in this article are done using Verilog HDL for ASIC. Mentor LeonardoSpectrum Level 3 (2018a.2) is used for synthesis with the UMC 65 nm Low-Power RVT (Regular VT) Standard Performance Generic Core Cell Library from Faraday.

6 Conclusions

We have studied the relationship between the resilient Boolean functions and S-boxes with linear branch number 3. We have shown how efficiently and systematically S-boxes with linear and differential branch number 3 can be constructed. We have presented such 5 × 5 and 6 × 6 S-boxes with good cryptographic properties such as nonlinearity and differential uniformity. The hardware costs of these S-boxes are provided. We think these S-boxes are interesting and can be used in cipher design, for instance, by following the design principle of ASCON. This idea can also be explored further in order to construct new hardware-friendly S-boxes.

References

1. Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Siang Meng Sim, Yosuke Todo, and Yu Sasaki. GIFT: A small present. *IACR Cryptology ePrint Archive*, 2017:622, 2017.
2. Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The skinny family of block ciphers and its low-latency variant mantis. In *Proceedings, Part II, of the 36th Annual International Cryptology Conference on Advances in Cryptology — CRYPTO 2016 - Volume 9815*, pages 123–153, Berlin, Heidelberg, 2016. Springer-Verlag.
3. Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. In *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '90, pages 2–21, London, UK, UK, 1991. Springer-Verlag.
4. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 4727 of *LNCS*, pages 450–466. Springer, 2007.
5. Claude Carlet. Vectorial Boolean functions for cryptography. In P. Hammer Y. Crama, editor, *Boolean Methods and Models*. Cambridge University Press, 2010.
6. Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.
7. DES. Data encryption standard. In *In FIPS PUB 46, Federal Information Processing Standards Publication*, pages 46–2, 1977.
8. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2. Submission to NIST Lightweight Cryptography project, 2019.
9. Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, EUROCRYPT '93, pages 386–397, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.
10. NIST. Nist lightweight cryptography project, 2019.
11. Gold R. Maximal recursive sequences with 3-valued recursive crosscorrelation functions. *IEEE Trans. Inf. Theory*, 14(1):154–156, 1968.
12. Markku-Juhani O. Saarinen. Cryptographic analysis of all 4 × 4-bit S-boxes. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pages 118–133. Springer, 2011.
13. Sumanta Sarkar, Kalikinkar Mandal, and Dhiman Saha. Sycon v1.0. Submission to the NIST Lightweight Cryptography project, 2019.
14. Sumanta Sarkar and Habeeb Syed. Bounds on differential and linear branch number of permutations. In Willy Susilo and Guomin Yang, editors, *Information Security and Privacy*, pages 207–224, Cham, 2018. Springer International Publishing.
15. C. E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal, Vol 28, pp. 656715*, October 1949.
16. Takeshi Shimoyama, Hitoshi Yanami, Kazuhiro Yokoyama, Masahiko Takenaka, Kouichi Itoh, Jun Yajima, Naoya Torii, and Hidema Tanaka. The block cipher sc2000. In *Revised Papers from the 8th International Workshop on Fast Software Encryption*, FSE '01, pages 312–327, Berlin, Heidelberg, 2002. Springer-Verlag.

17. Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata. The 128-Bit Blockcipher CLEFIA (Extended Abstract). In Alex Biryukov, editor, *Fast Software Encryption, 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007, Revised Selected Papers*, volume 4593 of *Lecture Notes in Computer Science*, pages 181–195. Springer, 2007.