CRYPTOGRAPHIC PROPERTIES OF CYCLIC BINARY MATRICES

AKBAR MAHMOODI RISHAKANI
Department of Sciences, Shahid Rajaee Teacher Training University, Tehran, Iran

SEYED MOJTABA DEHNAVI
Department of Mathematics and Computer Sciences, Kharazmi University, Tehran, Iran

MOHAMMADREZA MIRZAEE SHAMSABAD
Department of Mathematics and Computer Sciences, Shahid Beheshti University, Tehran, Iran

NASOUR BAGHERI∗
Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran

(Communicated by Claude Carlet)

Abstract. Many modern symmetric ciphers apply MDS or almost MDS matrices as diffusion layers. The performance of a diffusion layer depends on its diffusion property measured by branch number and implementation cost which is usually measured by the number of XORs required. As the implementation cost of MDS matrices of large dimensions is high, some symmetric ciphers use binary matrices as diffusion layers to trade-off efficiency versus diffusion property. In the current paper, we investigate cyclic binary matrices (CBMs for short), mathematically. Based upon this theoretical study, we provide efficient matrices with provable lower bound on branch number and minimal number of fixed-points. We consider the product of sparse CBMs to construct efficiently implementable matrices with the desired cryptographic properties.

1. Introduction

Designing large S-boxes with good cryptographic properties has its own limitations in both generation and implementation; this leads to the use of diffusion layers along with suitable S-boxes to make symmetric ciphers resistant against statistical attacks. Diffusion layers are usually evaluated based on two factors: diffusion property (branch number) and implementation cost which are inversely related. Matrices with maximum branch number with respect to Singleton bound are called MDS matrices [21]. Diffusion layers of symmetric ciphers AES [7], Twofish [27] and SNOW [9] apply MDS matrices with entries in finite fields. In the papers [23, 24, 5], the authors tried to improve the efficiency of MDS matrices. However, implementation costs of MDS matrices of large dimensions (i.e. 8, 16, 32, ...) are rather high, due to field multiplications.
To overcome the high implementation cost of field multiplications, the ciphers LOISS [10], SMS4 [16] and ZUC [1] apply MDS diffusion layers which do not use field multiplications; instead, they use basic instructions of softwares: rotations and XORs. The $4 \times 4$ MDS diffusion layers mentioned in [13] are characterized on arbitrary input lengths. In spite of the fact that this method reduces the implementation cost of $4 \times 4$ MDS matrices, but for large dimensions, no MDS matrix has been provided yet.

There is a trade-off between the branch number and implementation cost of diffusion layers. Since the implementation cost of MDS matrices with large dimensions is high, the symmetric ciphers E2 [17], Camellia [2] and ARIA [19] apply diffusion layers of large dimensions and lower branch numbers in contrast to MDS matrices. They use $8 \times 8$ and $16 \times 16$ matrices over 8-bit inputs, whose implementations only need the XOR operations. In fact, the diffusion layers which only use the XOR operation are $(0,1)$-matrices over $GF(2^s)$, for some natural number $s$. Cyclic $(0,1)$-matrices (CBMs) are used as diffusion layers of symmetric ciphers, Midori [4], Fides [6], ASCON [8] and HIGHT [15]. In fact, the design strategy of these ciphers is to use iterative lightweight rounds instead of a heavy round.

It is proved that the maximum branch number of $16 \times 16$ $(0,1)$-matrices is 8 [11]. For the case of $32 \times 32$ matrices the best known branch number is 12 [26]. In this respect, some papers present $(0,1)$-matrices with the mentioned branch numbers [3, 25, 18, 14] for use in symmetric cryptography. In this paper, cyclic $(0,1)$-matrices with maximal known branch numbers are studied. In this paper, Section 2 is devoted to notations and definitions and some basic facts are recalled. After that, in Section 3, we study cryptographic properties of CBMs. In this class, matrices of order $2^n$ are invertible if and only if the number of non-zero elements in each row is odd [22]. First, we show that the linear and differential branch numbers of these matrices are equal. This fact halves the time complexity of the search for matrices with desired (optimal) branch numbers, in this class. Then, we prove a theorem concerning construction of CBMs of large dimension with provable lower bound on branch number (where computer search is infeasible). We show that the number of non-zero elements in each row of a CBM imposes an upper bound on its branch number. According to this upper bound, for matrices whose number of non-zero elements in each row is equal to or less than 12, our methodology guarantees to find matrices of large dimensions whose branch number is equal to the mentioned upper bound.

In some recent cryptanalysis of symmetric ciphers, fixed-points of diffusion layers lead to some attacks [12, 20]. So, we verify fixed-points of the proposed CBMs, as well as their branch number and implementation cost. In this regard, we characterize the CBMs of dimension $2^n$ (which are mostly used in cryptography) with minimal fixed-points.

In Section 4, on one hand, we propose techniques to reduce the search complexity for finding CBMs with desired cryptographic properties; on the other hand, we provide techniques for efficient implementation. This leads to $16 \times 16$ and $32 \times 32$ CBMs which can be implemented by at most 49 and 146 XORs, respectively.

2. Notations and definitions

The cardinality of a finite set $A$ is denoted by $|A|$. The finite field with 2 elements is denoted by $\mathbb{F}_2$. The $i$-th component of a vector $x \in \mathbb{F}_2^n$ is denoted by $x_i$; in fact, $x = (x_{n-1}, \ldots, x_0)$. Hamming weight of a vector $x \in \mathbb{F}_2^n$ is denoted by $wt(x)$ and
defined as
\[ \text{wt}(x) = |\{ x_i : x_i \neq 0, 0 \leq i \leq n - 1\}|. \]
The notation \( \land \) is used for AND operation, \( \ll (\gg) \) for left (right) cyclic shifts and \( \oplus \) for XOR operations. The zero vector is denoted by \( 0 \). A function \( f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2 \) is called a Boolean function and a function \( f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m \) with \( n > 1 \) is called a vectorial Boolean function. Such a function can be represented as \( (f_{n-1}, ..., f_0) \): the Boolean function \( f_i : \mathbb{F}_2^n \rightarrow \mathbb{F}_2 \), \( 0 \leq i < n \), is called the \( i \)-th component of \( f \).

We denote the set of all \( n \times n \) matrices over \( \mathbb{F}_2 \) by \( \mathcal{M}_n(\mathbb{F}_2) \). A function \( f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m \) with the property \( f(x \oplus y) = f(x) \oplus f(y), \quad x, y \in \mathbb{F}_2^n \), is called a bitwise linear map. In this case, obviously there exists a matrix in \( \mathcal{M}_n(\mathbb{F}_2) \) which we denote by \( M_f \) such that \( f(x) = x M_f, x \in \mathbb{F}_2^n \).

Let \( f : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n \) be a bitwise linear map. The differential branch number of \( f \) (or \( M_f \)) is defined as
\[ b_f^d = \min \{ \text{wt}(x) + \text{wt}(x M_f) : x \in \mathbb{F}_2^n \}. \]
The linear branch number of \( f \) (or \( M_f \)) is defined as
\[ b_f^l = \min \{ \text{wt}(x) + \text{wt}(x M_f^T) : x \in \mathbb{F}_2^n \}, \]
where, \( M_f^T \) is the transpose of \( M_f \). If we have \( b_f^d = b_f^l \), then we say that the branch number of \( f \) is \( b_f \) and denote it by \( b_f \).

Let \( n \geq 2 \) be a natural number. We define
\[ L_n = \{ f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m : f(x) = \bigoplus_{j=0}^{t} (x \gg i_j), 0 \leq i_0 < \cdots < i_t < n \}. \]
Let \( f \in L_n \); it is straightforward that the rows (columns) of \( M_f \) are cyclic shifts of each other. In fact, the corresponding matrices of elements in \( L_n \) are exactly the \( n \times n \) CBMs. The number of 1's in each row (column) of \( M_f \) is denoted by \( w_f \).

We denote the composition of two mappings \( f \) and \( g \) by \( (f \circ g)(x) = f(g(x)) \); and the \( r \)-times composition of \( f \) by itself is denoted by \( f^{(r)} \). As \( L_n \) is a commutative ring, so for \( f, g \in L_n \), we have
\[ M_{f \circ g} = M_f M_g = M_g M_f. \]
For the sake of simplicity, in tables we denote \( f(x) = \bigoplus_{j=0}^{t} (x \gg i_j) \) by \( \text{circ}(i_0, i_1, ..., i_t) \) and the composition of \( f \) and \( g \) by \( \text{circ}(i_0, i_1, ..., i_t) \cdot \text{circ}(j_0, j_1, ..., j_m) \).

3. Cryptographic properties of \( L_n \)

In this section, we investigate some cryptographic properties of \( L_n \) from mathematical viewpoint and lay a theoretical foundation for applications presented in the next section. The following lemma is a straightforward result of [21, Theorem 4.5.6].

**Lemma 3.1.** For every \( f \in L_n \), we have \( b_f^d = b_f^l \).
According to Lemma 3.1, in the sequel we only calculate $b_f^d$ for every $f \in L_n$.

The following lemma is proved in [22] in details. But, the mentioned proof needs some algebraic preliminaries. For self-completeness, we give an equivalent proof with the notations and definitions of this paper.

**Lemma 3.2.** A boolean function $f \in L_n$ with $n = 2^m$ is invertible if and only if $w_f$ is odd.

**Proof.** Let $f \in L_n$ with $f(x) = \bigoplus_{j=0}^{t}(x \gg i_j)$, $0 \leq i_0 < \cdots < i_t < n$. An easy calculation shows that $f^{(2)}(x) = \bigoplus_{j=0}^{t}(x \gg 2i_j)$. By mathematical induction on $m$ we have,

$$g(x) = f^{(n)}(x) = \bigoplus_{j=0}^{t}(x \gg n i_j) = \bigoplus_{j=0}^{t}(x).$$

Equation (1) shows that for every $x \in \mathbb{F}_2^n$, $g(x) = 0$ if $w_f = t + 1$ is even and $g(x) = x$ if $w_f$ is odd. This implies that $f$ is invertible if and only if $w_f$ is odd. Additionally, in this case,

$$f^{-1}(x) = f^{(n-1)}(x).$$

The following lemma gives an upper bound for the branch number of CBMs in terms of the number of non-zero elements in each row.

**Lemma 3.3.** Let $f \in L_n$ with $f(x) = \bigoplus_{j=0}^{t}(x \gg i_j)$, $0 \leq i_0 < \cdots < i_t < n$, be invertible. Then $b_f \leq w_f + 1$ and $b_f$ is even for $n = 2^m$ ($m \geq 2$).

**Proof.** For $a \in \mathbb{F}_2^n$ with $wt(a) = 1$, we have

$$wt(a) + wt(f(a)) = w_f + 1.$$ 

This leads to $b_f \leq w_f + 1$; and according to Lemma 3.2, $wt(a) + wt(f(a))$ is an even number in the case $n = 2^m$. Now, by mathematical induction on the weights of the inputs, we prove that $wt(x) + wt(f(x))$ is even in the case $n = 2^m$. Suppose that the induction hypothesis is true for $wt(x) = r$, $1 \leq r < n$. For every $x \in \mathbb{F}_2^n$ with $wt(x) = r + 1$, there exist $x', a' \in \mathbb{F}_2^n$ such that $x = x' \oplus a'$, $wt(x') = r$ and $wt(a') = 1$. By the identity (see also [21])

$$wt(x \oplus y) = wt(x) + wt(y) - 2wt(x \land y),$$

we get

$$wt(x \oplus y) = wt(x) + wt(y) \mod 2.$$ 

Now by induction hypothesis, linearity of $f$ and (2), we have

$$wt(f(x)) = wt(f(x' \oplus a'))$$

$$= wt(f(x')) + wt(f(a')) \mod 2$$

$$= wt(x') + wt(a') \mod 2$$

$$= wt(x' \oplus a') \mod 2$$

$$= wt(x) \mod 2.$$ 

This means that $b_f$ is an even number in the case of $n = 2^m$. 

**Example 1.** For every $f \in L_{2^n}$ ($n \geq 2$) with $w_f = 3$, we have $b_f = 4$: According to Lemma 3.2, $f$ is invertible. So, by Lemma 3.3, $b_f$ is even and $b_f \leq 4$; i.e. $b_f = 2$ or 4. To prove that $b_f \neq 2$, we distinguish two cases:
Advances in Mathematics of Communications

Cryptographic matrices 5

- If $\text{wt}(x) = 1$, then $\text{wt}(f(x)) = 3$ according to the proof of Lemma 3.3. So, $\text{wt}(x) + \text{wt}(f(x)) \geq 4$.
- If $\text{wt}(x) \geq 2$, then $\text{wt}(f(x)) \geq 1$; because, $f$ is invertible. So, $\text{wt}(x) + \text{wt}(f(x)) \geq 3$.

This means that, $b_f = 4$. Therefore, all of the diffusion layers applied in ASCON [8] have branch number 4.

According to Lemma 3.3, let $f \in L_n$ be such that $b_f = w_f + 1$. In this case, we call $f$ as well as its corresponding matrix, perfect with respect to $w_f$. Following theorem gives a tool to construct perfect CBMs of large dimensions.

**Theorem 3.4.** Let $f \in L_n$ with $f(x) = \bigoplus_{j=0}^{t}(x \gg i_j), \ 0 \leq i_0 < \cdots < i_t < n$. For the mapping $g \in L_{2n}$ with $g(x) = \bigoplus_{j=0}^{t}(x \gg i_j)$ we have $b_g \geq b_f$. Furthermore, if $b_f = w_f + 1$, then $b_g = b_f$.

**Proof.** Let

$$h_m(x) = x_{i_0} \oplus x_{i_1} \oplus \cdots \oplus x_{i_t}, \ x \in \mathbb{F}_2^n, \ m \geq n.$$  

One can check that for $y = f(x)$, it holds that

$$y_i = h_n(x \gg i), \ 0 \leq i < n.$$  

Now, let $X = (x', x) \in \mathbb{F}_2^{2n}$ and $Z = (z', z) = g(X)$, where $x', x, z, z' \in \mathbb{F}_2^n$. We have

$$Z_i = h_{2n}(X \gg i), \ 0 \leq i < 2n.$$  

For every $x \in \mathbb{F}_2^n (m \geq n)$, $h_m$ acts on the $n$ least significant bits of $x$ and $h_m$ is a bitwise linear mapping. So, for every $0 \leq i < n$, the $n$ least significant bits of $X \gg i$ equals to $(x \gg i \oplus x' \ll (n - i))$. Thus,

$$z_i \oplus z'_i = Z_i \oplus Z_{i+n} = h_{2n}(X \gg i) \oplus h_{2n}(X \gg (i + n))$$

$$= h_n(x \gg i \oplus x' \ll (n - i)) \oplus h_n(x' \gg i \oplus x \ll (n - i))$$

$$= h_n((x \gg i \oplus x') \gg i)$$

which leads to

$$z \oplus z' = f(x \oplus x').$$

To complete the proof, we distinguish two cases:

**Case 1.** $x = x'$.

In this case, by the linearity of $f$ and equation (3), we have $z = z'$. By the proof of equation (3), for $0 \leq i < n$,

$$z_i = h_{2n}(X \gg i) = h_n(x \gg i \oplus x' \ll (n - i))$$

$$= h_n(x \gg i \oplus x \ll (n - i)) = h_n(x \gg i).$$

This means that $z = f(x)$. So, for $X \neq 0$,

$$\text{wt}(X) + \text{wt}(g(X)) = \text{wt}(x) + \text{wt}(z) + \text{wt}(x') + \text{wt}(z') \geq 2b_f.$$  

**Case 2.** $x \neq x'$.

In this case, $x \oplus x' \neq 0$ and obviously $\text{wt}(x', x) \geq \text{wt}(x \oplus x')$. Thus, considering equation (3), we have

$$\text{wt}(X) + \text{wt}(g(X)) \geq \text{wt}(x \oplus x') + \text{wt}(y \oplus y') \geq b_f.$$  

Equations (4) and (5) implies that $b_g \geq b_f$. In the case of $b_f = w_f + 1$, as $w_f = w_g$, we have $b_g = b_f$, by Lemma 3.3.
Example 2. Let \( f(x) = x \oplus (x \gg 1) \oplus (x \gg 2) \oplus (x \gg 3) \oplus (x \gg 6). \) when \( f \in L_8, \) then \( b_f = 4 \) and when \( f \in L_{16}, \) then \( b_f = 6. \) This shows that, the inequality in Theorem 3.4, could be strict.

In the following, we study the number of fixed-points of CBMs. For simplicity, we identify \( \mathbb{Z}_{2^n} \) (the set of integers mod \( 2^n \)) with \( \mathbb{F}_2^n \) through the natural binary representation.

Lemma 3.5. Let \( T \in L_n, \) \( n = 2^m, \) \( m \geq 3 \) with \( T(x) = x \oplus (x \gg 1). \) For \( 1 \leq k \leq n, \) the mapping \( T^{(k)} \) has \( 2^k \) roots.

Proof. The roots of \( T \) are \( X_1 = 0 \) (the all-zero vector) and \( X_2 = 2^n - 1 \) (the all-one vector). This is because of the fact that, taking \( x_0 = 0, \) the solution \( X_1 \) is uniquely determined and taking \( x_0 = 1, \) the solution \( X_2 \) is uniquely determined. We have \( T^{(n)}(a) = a \oplus (a \gg n) = 0 \) for every \( a \in \mathbb{F}_2^n. \) Applying \( T, \) \( n - 1 \) times on \( a = 2^{n-1} \) repeatedly, we have the following sequence of vectors:

\[
\begin{array}{cccc}
\quad & 1 & 0 & 0 \\
\text{n-1 times} & 1 & 0 & 0 \\
\text{n-2 times} & 1 & 0 & 0 \\
\end{array}
\]

This shows that \( T^{(n-1)}(2^{n-1}) \neq 0. \) On the other hand, we have \( T^{(n)}(2^{n-1}) = 0. \) Now, since the equation \( T(x) = 0 \) has only \( X = 2^n - 1 \) as a non-zero solution, so \( T^{(n-1)}(2^{n-1}) = 2^n - 1. \) It follows that, the equation \( T^{(r)}(x) = 2^n - 1, \) \( 1 \leq r \leq n - 1, \) has always a solution. Consider the equation \( T^{(k)}(x) = 0, \) \( 2 \leq k \leq n. \) We have \( T(T^{(k-1)}(x)) = 0. \) So, either \( T^{(k-1)}(x) = 0 \) or \( T^{(k-1)}(x) = 2^n - 1. \) We use induction on \( k. \) By mathematical induction hypothesis, \( T^{(k-1)}(x) = 0 \) has \( 2^{k-1} \) solutions. We have already proved that \( T^{(k-1)}(x) = 2^n - 1 \) has at least one solution, say \( \alpha_{k-1}; \) so, all of the solutions are of the form \( \alpha_{k-1} \oplus \beta \) where \( T^{(k-1)}(\beta) = 0. \) It follows that the equation \( T^{(k)}(x) = 0 \) has \( 2^{k-1} + 2^{k-1} = 2^k \) solutions. \( \square \)

It is worth noting that, if we choose \( T(x) = x \oplus x \ll 1 \) in Lemma 3.5, then \( T^{(k)} \) has \( 2^k \) roots.

The weight of a non-invertible element \( f \in L_{2^m} \) is an even number by Lemma 3.2. So, the mapping \( g(x) = x \oplus f(x) \) would be invertible with only zero as the root. Since the roots of \( g \) are fixed-points of \( f, \) the mapping \( f \) has only zero as the fixed-point, in this case. In the following lemma we characterize the number of fixed-points of invertible CBMs of order \( 2^m. \)

Lemma 3.6. Let \( f \in L_{2^m} \) with \( f(x) = \bigoplus_{j=0}^{t} (x \gg i_j), \) \( 0 \leq i_0 < \cdots < i_t < 2^m \) and \( m \geq 3 \) be invertible and \( f(x) \neq x. \) For a unique value of \( k, \) \( 1 \leq k < 2^m, \) we have \( g(x) = x \oplus f(x) = h(T^{(k)}(x)), \) in which \( h \in L_{2^m} \) is invertible and \( T(x) = x \oplus (x \gg 1); \) and the mapping \( f \) has \( 2^k \) fixed-points.

Proof. One can see that for \( 1 \leq i < j \leq 2^m \) we have \( \text{circ}(i, j) = \text{circ}(0, 1)(i, i + 1, \ldots, i + r - 1), \) \( r = j - i. \) As \( f \) is invertible, by Lemma 3.2, \( t \) is even. Taking \( t = 2k \) and \( r_j = i_{2j} - i_{2j-1}, \) \( 1 \leq j \leq k; \) for \( i_0 > 0, \) we have \( g = \text{circ}(0, 1)(0, 1, \ldots, i_0 - 1) \oplus \text{circ}(0, 1)(i_1, i_1 + 1, \ldots, i_1 + r_1 - 1) \oplus \text{circ}(0, 1)(i_3, i_3 + 1, \ldots, i_3 + r_2 - 1) \oplus \ldots \)
The roots of the mappings $g$ and $h$ also have the same set of fixed-points. By a similar discussion, the case of $i = 0$ could be treated similarly.

In Lemma 3.6, the factor $h$ in the decomposition of $g$ is not necessarily unique, but according to the proof, it has no role in the calculation of the fixed-points of $f$. The set of the fixed-points are exactly the roots of $T$. This occurs because, $g$ is nonzero ($f(x) \neq x$) and $T^{(2^n)}(x) = 0$.

Now, we show the uniqueness of $k$. Let $g(x) = h(T^{(k)}(x))$ and $g(x) = h'(T^{(k')})(x))$, where $h, h'$ are invertible and $1 \leq k, k' < 2^m$. To determine the roots of $g$, we have $g(x) = h(T^{(k)}(x)) = 0$.

Then, $h^{-1}(h(T^{(k)}(x))) = 0$; so, $T^{(k)}(x) = 0$. According to Lemma 3.5, $g$ has $2^k$ roots. By a similar discussion, $g(x) = h'(T^{(k')})(x))$, has $2^{k'}$ roots. This shows that $k = k'$. It is worth noting that, we have determined the number of the fixed-points of $f$ in this process; i.e. the number of the roots of $g$.

The case of $i = 0$ could be treated similarly.

In Lemma 3.6, the factor $h$ in the decomposition of $g$ is not necessarily unique, but according to the proof, it has no role in the calculation of the fixed-points of $f$. The set of the fixed-points are exactly the roots of $T^{(k)}$. Now, let $f_1, f_2 \in L_{2^m}$ have the same number of fixed-points, say $k$. By Lemma 3.6, there exist invertible mappings $h_1, h_2 \in L_{2^m}$, for which

$$g_1(x) = x \oplus f_1(x) = h_1(T^{(k)}(x)),$$

$$g_2(x) = x \oplus f_2(x) = h_2(T^{(k)}(x)).$$

The roots of the mappings $g_1$ and $g_2$ are exactly the roots of $T^{(k)}$. Therefore, $f_1$ and $f_2$ also have the same set of fixed-points.

The following example illustrates the application of Lemma 3.6, in calculating the number of fixed-points of every element in $L_n$, $n = 2^m$, $m \geq 3$; i.e. every $n \times n$ CBM.

**Example 3.** The stream cipher LOISS [10] applies the mapping $f : \mathbb{F}_2^{32} \rightarrow \mathbb{F}_2^{32}$,

$$f(x) = x \oplus (x \ll 2) \oplus (x \ll 10) \oplus (x \ll 18) \oplus (x \ll 24),$$

as the diffusion layer. To use the notations of the current paper, we can represent $f$ as a member of $L_{32}$ with

$$f(x) = x \oplus (x \gg 8) \oplus (x \gg 14) \oplus (x \gg 22) \oplus (x \gg 30).$$

We have,

$$g(x) = x \oplus f(x) = (x \gg 8) \oplus (x \gg 14) \oplus (x \gg 22) \oplus (x \gg 30).$$

By the decomposition procedure presented in the proof of Lemma 3.6, we have $g(x) = h(T^{(2^n)}(x))$, where,

$$h(x) = (x \gg 8) \oplus (x \gg 10) \oplus (x \gg 12) \oplus (x \gg 22) \oplus (x \gg 24) \oplus (x \gg 26) \oplus (x \gg 28)$$

and $T(x) = x \oplus (x \gg 1)$. Thus, by Lemma 3.6 the diffusion layer of LOISS has 4 fixed-points.
Note that, the all-zero and all-one vectors are fixed-points of every invertible $f \in L_n$, $n = 2^m$, $m \geq 3$. The following lemma characterizes invertible CBMs of order $2^m$ with minimal number of fixed-points.

**Theorem 3.7.** Let $f \in L_{2^m}$ with $f(x) = \bigoplus_{j=0}^{t}(x \gg i_j)$, $0 \leq i_0 < \cdots < i_t < 2^m$ and $m \geq 3$ be invertible. The mapping $f$ has two fixed-points if and only if $\sum_{j=0}^{t} i_j$ is odd.

**Proof.** As $f$ is invertible, by Lemma 3.2, $t$ is even ($w_f = t + 1$). Take $t = 2k$ and $r_j = i_{2j} - i_{2j-1}$, $1 \leq j \leq k$. According to the equation (6), for $g(x) = x \oplus f(x)$ in the case of $i_0 > 0$, we have

$$g = \text{circ}(0,1)(0,1,...,i_0 - 1,i_1,...,i_2 - 1,i_3,...,i_4 - 1,i_{2k-1},...,i_{2k} - 1).$$

Now, if we put $h = \text{circ}(0,1,...,i_0 - 1,i_1,...,i_2 - 1,i_3,...,i_4 - 1,i_{2k-1},...,i_{2k} - 1)$, then, $w_h = i_0 + r_1 + r_2 + \cdots + r_k$. One can check that, $w_h$ is odd if and only if $\sum_{j=0}^{t} i_j$ is odd. So, by Lemma 3.2 $h$ is invertible if and only if $\sum_{j=0}^{t} i_j$ is odd. Thus, by Lemma 3.6, $f$ has exactly two fixed-points if and only if $\sum_{j=0}^{t} i_j$ is odd.

The case of $i_0 = 0$ could be treated similarly.

The following lemma shows that the number of fixed-points of CBMs of order $2^m$ does not increase, doubling the dimension.

**Lemma 3.8.** Let $f_1 \in L_n$ with $f_1(x) = \bigoplus_{j=0}^{t}(x \gg i_j)$, $0 \leq i_0 < \cdots < i_t < n$, $n = 2^m$ and $m \geq 3$. The mapping $f_2 \in L_{2n}$ with $f_2(x) = \bigoplus_{j=0}^{t}(x \gg i_j)$ has the same number of fixed-points as $f_1$.

**Proof.** Considering Lemma 3.6, it is straightforward to see that the decomposition of $g_1(x) = x \oplus f_1(x) \in L_n$ and $g_2(x) = x \oplus f_2(x) \in L_{2n}$ to the presented form is the same. Thus, the number of fixed-points of $f_1$ and $f_2$ are equal.

**Example 4.** The block cipher HIGHT [15] and authenticated encryption design ASCON Ver 1.2 [8] apply some members of $L_n (n = 8, 64)$ with branch number 4 as diffusion layers. These diffusion layers along with their number of fixed-points are listed in Table 1. The number of fixed-points are calculated using Theorem 3.7 and Lemma 3.6.

The diffusion layer of ASCON is illustrated in Figure 1. It consists of $f_1, f_2, f_3, f_4, f_5$ which are listed in Table 1. These mappings act on 64 5-bit words. Since the branch of the mappings $f_1, f_2, f_3, f_4, f_5$ is 4 ($b_{f_1} = b_{f_2} = b_{f_3} = b_{f_4} = b_{f_5} = 4$), so, in each linear or differential characteristic, there are at least 4 (input and output) active words. It is straightforward to see that, the number of fixed-points of this diffusion layer is $2 \times 2 \times 2 \times 4 \times 4 = 128$.

**Remark 1.** Let $f \in L_n$ with $f(x) = \bigoplus_{j=0}^{t}(x \gg i_j)$ and $F \in L_{2n}$ with $F(x) = \bigoplus_{j=0}^{t}(x \gg i_j)$ be invertible. According to Theorem 3.4, $b_F \geq b_f$ and by Lemma 3.8, $F$ and $f$ have the same number of fixed-points, say $r$.

Consider the diffusion layer illustrated in Figure 2, which acts on $n m$-bit words (similar to the case of ASCON). In a differential or linear characteristic, at least
Table 1. Diffusion layers of HIGHT and ASCON

| n  | Source | The mapping | Number of fixed-points |
|----|--------|-------------|------------------------|
| 8  | HIGHT  | circ(1,6,7) | 8                      |
| 8  | HIGHT  | circ(2,4,5) | 2                      |
| 64 | ASCON  | \( f_1 = \text{circ}(0,19,28) \) | 2                      |
| 64 | ASCON  | \( f_2 = \text{circ}(0,39,61) \) | 4                      |
| 64 | ASCON  | \( f_3 = \text{circ}(0,1,6) \)  | 2                      |
| 64 | ASCON  | \( f_4 = \text{circ}(0,10,17) \) | 2                      |
| 64 | ASCON  | \( f_5 = \text{circ}(0,7,41) \)  | 4                      |

Figure 1. The corresponding diffusion layer of ASCON.

Figure 2. An ASCON-like diffusion layer.

Let \( b_f \) number of input and output words would be active; Moreover, it has \( r^m \) fixed-points. In this case, concatenating the inputs together and the outputs together, the mentioned diffusion layer could be represented as

\[
g : \mathbb{F}_2^{mn} \rightarrow \mathbb{F}_2^{mn}, \quad g(x) = \bigoplus_{j=0}^{r^m} (x \gg mi_j).
\]
Also, $M_g$ could be constructed from $M_f$ as follows: replace every 0 by a $0$ and every 1 by $I$. Here, $0$ and $I$ stand for the zero and identity matrices of order $m$, respectively.

Now, suppose that the number of input words as well as the output words are increased to $2n$. In this case, a common parallel application is illustrated in Figure 3. This diffusion layer is such that, there are at least $b_f$ active input and output $m$-bit words. Further, it has $r^{2m}$ fixed-points.

Instead of the diffusion layer in Figure 3, we propose the diffusion layer in Figure 4 which has the same implementation cost, but with better cryptographic properties. The proposed diffusion layer, activates at least $b_F$ ($b_F \geq b_f$) input and output words. The interesting point is that, it has $r^m$ ($\ll r^{2m}$) fixed-points by Lemma 3.8.

### 4. Efficient Verification and Implementation

In this section, we focus on efficient methods of verification and implementation of CBMs with desired cryptographic properties.

To verify the branch number of the elements of $f \in L_n$ with $f(x) = \bigoplus_{i=0}^{t}(x \gg i_j)$, it suffices to check the ones with $i_0 = 0$; because, $\text{wt}(x \gg r) = \text{wt}(x)$, $x \in F_2^n$, $1 \leq r \leq n$ and $f(x \gg r) = f(x) \gg r$. For example, let $f \in L_{32}$ with $f(x) = (x \gg 1) \oplus (x \gg 4) \oplus (x \gg 5)$. Then, for

$$g_1(x) = f(x \gg 31) = x \oplus (x \gg 3) \oplus (x \gg 4),$$
we have \( b_f = b_{g_1} = b_{g_2} = b_{g_3} \). In the sequel we suppose that \( i_0 = 0 \).

For every invertible \( f \in L_n \), \( b_f = b_{f^{-1}} \) so, Lemma 3.3 could be improved to

\[
b_f \leq \min\{w_f, w_{f^{-1}}\} + 1.
\]

This means that, to calculate the branch number of an invertible \( f \in L_n \) it is sufficient to verify all \( x \in \mathbb{F}_2^n \) with \( \text{wt}(x) \leq \min\{w_f, w_{f^{-1}}\} - 1 \).

**Example 5.** Let \( f \in L_{32} \) with
\[
f(x) = x \oplus (x \gg 1) \oplus (x \gg 3) \oplus (x \gg 4) \oplus (x \gg 5) \oplus (x \gg 6) \oplus (x \gg 7) \oplus (x \gg 8) \\
\quad \oplus (x \gg 9) \oplus (x \gg 10) \oplus (x \gg 12).
\]

Then,
\[
f^{-1}(x) = (x \gg 1) \oplus (x \gg 3) \oplus (x \gg 4) \oplus (x \gg 7) \oplus (x \gg 8) \oplus (x \gg 13) \oplus (x \gg 15).
\]

So, \( b_f \leq \min\{11, 7\} + 1 = 8 \). Thus, it suffices to check all \( x \in \mathbb{F}_2^{32} \), with \( \text{wt}(x) \leq 6 \), which are \( \binom{16}{1} + \binom{16}{2} + \cdots + \binom{16}{8} = 14892 \) numbers of vectors. Note that, we don’t need to check the values of \( f(x) \) for \( \text{wt}(x) = 7 \); because \( f \) is invertible.

As we know that \( b_f \leq 8 \), so, to calculate the minimum value of \( \text{wt}(x) + \text{wt}(f(x)) \) in the case that \( \text{wt}(x) = 6 \), it suffices to calculate \( \text{wt}(f^{-1}(a)) \) for \( \text{wt}(a) = 1, 2 \). Because, for every vector \( x \in \mathbb{F}_2^{16} \) with \( \text{wt}(x) = 6 \) and \( \text{wt}(f(x)) \geq 3 \), we have \( \text{wt}(x + \text{wt}(f(x)) \geq 9 \); this means that, such vectors have no influence on the calculation of \( b_f \). Thus, calculating \( f(x) \) and \( f^{-1}(x) \) with \( \text{wt}(x) = 1, 2, 3, 4 \), determines \( b_f \). Therefore, we should calculate \( f(x) \) and \( f^{-1}(x) \) for \( \binom{16}{1} + \binom{16}{2} + \binom{16}{3} + \binom{16}{4} = 2516 \) vectors, which are much less than 14892.

According to the improvements given in Example 5 for the process of calculating \( b_f \), \( f \in L_n \), we have exhaustively searched for the case \( n = 16 \). The results are reported in Table 2. Among the 3240 mappings with largest branch number, i.e. 8, 1372 are perfect with respect to \( w_f \) and 1600 have minimal number of fixed-points.

Some invertible mappings \( f \in L_{16} \) with \( b_f = 8 \) and two numbers of fixed-points are listed below

| \( \text{circ}(0, 1, 2, 3, 4, 7, 10) \) | \( \text{circ}(0, 1, 4, 8, 10, 13, 15) \) |
| \( \text{circ}(0, 4, 6, 7, 9, 11, 14) \) | \( \text{circ}(0, 8, 10, 11, 13, 14, 15) \) |
| \( \text{circ}(0, 1, 2, 7, 9, 13, 15) \) | \( \text{circ}(0, 2, 8, 10, 11, 13, 15) \) |
| \( \text{circ}(0, 3, 6, 7, 8, 9, 10) \) | \( \text{circ}(0, 4, 10, 11, 12, 13, 15) \) |

By Theorem 3.4 and Theorem 3.7, the corresponding matrices of the above list, considered as elements of \( L_{2^n} \), \( n \geq 5 \), have branch number 8 and just two fixed-points. In this regard, we list some \( 2^n \times 2^n \), \( n \geq 5 \) CBMs with branch number 12 and two fixed-points:

| \( \text{circ}(0, 1, 2, 3, 4, 5, 6, 8, 11, 15, 20) \) | \( \text{circ}(0, 1, 2, 3, 4, 7, 8, 14, 21, 24, 27) \) |
| \( \text{circ}(0, 1, 2, 3, 4, 9, 10, 12, 15, 17) \) | \( \text{circ}(0, 1, 2, 3, 4, 9, 10, 11, 13, 16, 30) \) |
| \( \text{circ}(0, 1, 2, 3, 4, 13, 15, 18, 20, 24, 31) \) | \( \text{circ}(0, 1, 2, 3, 4, 13, 16, 17, 18, 22, 25) \) |
| \( \text{circ}(0, 1, 2, 3, 5, 6, 8, 9, 15, 21, 29) \) | \( \text{circ}(0, 1, 2, 3, 5, 6, 7, 14, 16, 29, 30) \) |

Now, we focus on efficient implementation of CBMs using the method of matrix decomposition. By Lemma 3.3, for every invertible \( f \in L_{16} \) with \( b_f = 8 \), we have \( w_f \geq 7 \); so, a naive implementation of \( f \) (without reusing resources or intermediate variables) needs 96 XORs. If we could find \( f_1, f_2 \in L_{16} \) such that \( w_{f_1} = w_{f_2} = 3 \)
and $M_f = M_{f_1}M_{f_2}$, then the implementation of $M_f$ using $M_{f_1}M_{f_2}$ is reduced to 64 XORs. Of course, in some cases the implementation cost could be further reduced. We illustrate this improvement through the following lemma.

**Lemma 4.1.** Let $f \in L_n$ with $f(x) = x \oplus (x \gg i_0) \oplus (x \gg i_1)$. If the sequences $(0, i_0, i_1)$ or $(i_0, i_1, n)$ follow an arithmetic progression with common difference of $d$, then $M_f$ requires at most $3dS_0 + 3\lceil \frac{S_1}{d} \rceil (S_1 - d) + 2S_2$ XORs. Here, 

$$S_0 = \lfloor \frac{S_1}{d} \rfloor, \quad S_1 = n - 2dS_0, \quad S_2 = S_1 - 2\lceil \frac{S_1}{d} \rceil (S_1 - d).$$

**Proof.** Since the sequences $(0, i_0, i_1)$ or $(i_0, i_1, n)$ follow an arithmetic progression with common difference of $d$, so $2d < n$. For $0 \leq i < d$, the implementation cost of both $y_i$ and $y_{i+d}$ is 3 XORs. Because, taking $f(x_{n-1}, \ldots, x_1, x_0) = (y_{n-1}, \ldots, y_1, y_0)$

- either $(0, i_0, i_1)$ follow an arithmetic progression; i.e. $f(x) = x \oplus (x \gg d) \oplus (x \gg 2d)$.
- $r = x_{i+d} \oplus x_{i+2d}$, $y_i = x_i \oplus r$, $y_{i+d} = r \oplus x_{i+3d}$,

- or $(i_0, i_1, n)$ follow an arithmetic progression; i.e. with common difference of $d$, then $f = \text{circ}(0, n-2d, n-d)$. So, $f(x) = x \oplus (x \gg n-2d) \oplus (x \gg n-d)$. So,

\begin{align}
  r &= x_i \oplus x_{i+n-d}, \quad y_i = x_{i+n-2d} \oplus r, \quad y_{i+d} = r \oplus x_{i+d}.
\end{align}

Note that, all the indices are calculated mod $n$. The above discussion means that $y_0, y_1, \ldots, y_{2d-1}$ could be implemented by $3d$ XORs. Now, let $S_0$ be the largest number such that $2S_0d < n$; i.e. $S_0 = \lfloor \frac{n}{d} \rfloor$. The same argument shows that, for $0 < j < S_0$, the implementation cost of $y_{2jd}, y_{2jd+1}, \ldots, y_{2jd+2d-1}$ is $3d$ XORs. Thus, $y_0, y_1, \ldots, y_{2dS_0-1}$ need $3dS_0$ XORs for implementation. It remains calculating the implementation cost of the remaining $S_1 = n - 2dS_0$ outputs. Clearly, $S_1 < 2d$.

- If $S_1 < d$, then $y_{2dS_0}, y_{2dS_0+1}, \ldots, y_n$ takes $2S_1 = 2S_2$ XORs (in this case $\lceil \frac{S_1}{d} \rceil = 0$).
- If $d \leq S_1 < 2d$, then there $S_1 - d$ pairs of the form $y_i, y_{i+d}$, with 3 XOR implementation cost among $y_{2dS_0}, y_{2dS_0+1}, \ldots, y_n$, $2dS_0 \leq i < n - d$. The remaining $S_2 = S_1 - 2(S_1 - d)$ outputs take $2S_2$ XORs. Note that, $\lceil \frac{S_1}{d} \rceil = 1$.

So, overall, $M_f$ requires at most $3dS_0 + 3\lceil \frac{S_1}{d} \rceil (S_1 - d) + 2S_2$ XORs for implementation. □

**Example 6.** Let $f_1, f_2 \in L_{16}$ with

\begin{align*}
  f_1(x) &= x \oplus (x \gg 1) \oplus (x \gg 2), \\
  f_2(x) &= x \oplus (x \gg 7) \oplus (x \gg 14).
\end{align*}
and, \( f(x) = f_1 \circ f_2(x) \). So,
\[
f(x) = (x \gg 1) \oplus (x \gg 2) \oplus (x \gg 7) \oplus (x \gg 8) \oplus (x \gg 9) \oplus (x \gg 14) \oplus (x \gg 15).
\]
We have \( b_f = 8 \) which is reported in Table 3. Let
\[
(y_{15}, \ldots, y_1, y_0) = f_2(x_{15}, \ldots, x_1, x_0),
\]
\[
(z_{15}, \ldots, z_1, z_0) = f_1(y_{15}, \ldots, y_1, y_0),
\]
and
\[
(z_{15}, \ldots, z_1, z_0) = f_1 \circ f_2(x_{15}, \ldots, x_1, x_0).
\]
According to Lemma 4.1, the implementation of \( f \) is as follows:
\[
\begin{align*}
y_0 &= x_0 \oplus x_7 \oplus x_{14}, & y_7 &= x_7 \oplus x_{14} \oplus x_5, \\
y_1 &= x_1 \oplus x_8 \oplus x_{15}, & y_8 &= x_8 \oplus x_{15} \oplus x_6, \\
y_2 &= x_2 \oplus x_9 \oplus x_0, & y_9 &= x_9 \oplus x_0 \oplus x_7, \\
y_3 &= x_3 \oplus x_{10} \oplus x_1, & y_{10} &= x_{10} \oplus x_1 \oplus x_8, \\
y_4 &= x_4 \oplus x_{11} \oplus x_2, & y_{11} &= x_{11} \oplus x_2 \oplus x_9, \\
y_5 &= x_5 \oplus x_{12} \oplus x_3, & y_{12} &= x_{12} \oplus x_3 \oplus x_{10}, \\
y_6 &= x_6 \oplus x_{13} \oplus x_4, & y_{13} &= x_{13} \oplus x_4 \oplus x_{11}, \\
y_{14} &= x_{14} \oplus x_5 \oplus x_{12}, & y_{15} &= x_{15} \oplus x_6 \oplus x_{13}, \\
z_0 &= y_0 \oplus y_1 \oplus y_2, & z_1 &= y_1 \oplus y_2 \oplus y_3, \\
z_2 &= y_2 \oplus y_3 \oplus y_4, & z_3 &= y_3 \oplus y_4 \oplus y_5, \\
z_4 &= y_4 \oplus y_5 \oplus y_6, & z_5 &= y_5 \oplus y_6 \oplus y_7, \\
z_6 &= y_6 \oplus y_7 \oplus y_8, & z_7 &= y_7 \oplus y_8 \oplus y_9, \\
z_8 &= y_8 \oplus y_9 \oplus y_{10}, & z_9 &= y_9 \oplus y_{10} \oplus y_{11}, \\
z_{10} &= y_{10} \oplus y_{11} \oplus y_{12}, & z_{11} &= y_{11} \oplus y_{12} \oplus y_{13}, \\
z_{12} &= y_{12} \oplus y_{13} \oplus y_{14}, & z_{13} &= y_{13} \oplus y_{14} \oplus y_{15}, \\
z_{14} &= y_{14} \oplus y_{15} \oplus y_0, & z_{15} &= y_{15} \oplus y_0 \oplus y_1.
\end{align*}
\]
Firstly, we calculate the values of \( r_0, r_1, \ldots, r_6 \) and use them to determine \( y_0, y_1, \ldots, y_{13} \). Then, we calculate the values of \( r_7, r_8, \ldots, r_{14} \) and use them to determine \( z_0, z_1, \ldots, z_{15} \). Therefore, the implementation cost of \( M_{f_1} \) and \( M_{f_2} \) would be 24 and 25 XORs, respectively. Thus, \( M_f \) needs 49 XORs using \( M_f = M_{f_1} \cdot M_{f_2} \).
is twofold:

(a) According to Lemma 3.3, to attain maximal branch number, we need at least two factors with weight three (i.e. with three 1’s in each row of their corresponding matrices) in the case of \( f \in L_{16} \); i.e. \( f = circ(0, i_0, i_1)(0, j_0, j_1) \), and at least three factors with weight three for \( f \in L_{32} \); i.e. 
\[
f = circ(0, i_0, i_1)(0, j_0, j_1)(0, r_0, r_1).
\]

(b) By Lemma 4.1, the best implementation (by our approach) pertains to the cases that the rotation values follow an arithmetic progression.

Some instances with minimum implementation cost are reported in Table 3.

Another method to attain efficiently implemented \( 32 \times 32 \) CBMs with maximal branch number is to search among mappings which are decomposed into 
\[
f = circ(0, i_0, i_1)(0, j_0, j_1)(j_2, j_3).
\]

In this case, the programming shows that there are no mappings of this form with maximal branch number if \( 0, j_0, j_1, j_2, j_3 \) or \( j_0, j_1, j_2, j_3, 32 \) follow an arithmetic progression. So, we have searched among the mappings in which \( (0, i_0, i_1) \) or \( (i_0, i_1, 32) \) or \( (0, j_0, j_1, j_2, j_3) \) follow an arithmetic progression. Some of the results with 160 XOR implementation cost are listed below 
\[
circ(0, 1, 2)(0, 1, 2, 7, 24), \quad circ(0, 1, 2)(0, 1, 2, 10, 15),
\]
\[
circ(0, 2, 4)(0, 8, 12, 30, 31), \quad circ(0, 2, 4)(0, 18, 22, 30, 31),
\]

### Table 3. \( n \times n \) CBMs with maximal branch number and efficient implementation.

| \( n \) | mapping | \#XORs | \#fixed-points |
|---|---|---|---|
| 16 | \( circ(0, 1, 2)(0, 2, 9) \) | 49 | 4 |
| 16 | \( circ(0, 14, 15)(0, 7, 14) \) | 49 | 4 |
| 16 | \( circ(0, 1, 2)(0, 7, 14) \) | 49 | 16 |
| 16 | \( circ(0, 14, 15)(0, 2, 9) \) | 49 | 16 |
| 16 | \( circ(0, 3, 6)(0, 5, 10) \) | 51 | 16 |
| 16 | \( circ(0, 3, 6)(0, 6, 11) \) | 51 | 4 |
| 32 | \( circ(0, 3, 6)(0, 4, 8)(0, 5, 10) \) | 146 | 64 |
| 32 | \( circ(0, 3, 6)(0, 4, 8)(0, 22, 27) \) | 146 | 4 |
| 32 | \( circ(0, 26, 29)(0, 24, 28)(0, 22, 27) \) | 146 | 64 |
| 32 | \( circ(0, 4, 8)(0, 5, 10)(0, 13, 26) \) | 148 | 4 |
| 32 | \( circ(0, 4, 8)(0, 22, 27)(0, 6, 19) \) | 148 | 4 |
| 32 | \( circ(0, 1, 2)(0, 7, 14)(0, 12, 24) \) | 150 | 64 |
| 32 | \( circ(0, 30, 31)(0, 18, 25)(0, 8, 20) \) | 150 | 64 |
| 32 | \( circ(0, 1, 2)(0, 9, 18)(0, 12, 24) \) | 150 | 4 |
| 32 | \( circ(0, 30, 31)(0, 14, 23)(0, 8, 20) \) | 150 | 4 |
| 32 | \( circ(0, 3, 6)(0, 4, 8)(0, 11, 22) \) | 150 | 4 |
| 32 | \( circ(0, 7, 14)(0, 12, 24)(0, 15, 30) \) | 151 | 4 |
| 32 | \( circ(0, 9, 18)(0, 12, 24)(0, 15, 30) \) | 151 | 64 |
| 32 | \( circ(0, 4, 8)(0, 11, 22)(0, 13, 26) \) | 152 | 64 |
Table 4. $2^m \times 2^m$, $m \geq 6$ CBMs with provable branch numbers and determined number of fixed-points.

| $f$ | $b_f$ | #XORs | #fixed-points |
|-----|-------|--------|--------------|
| circ$(0, 1, 2)(0, 2, 9)$ | 8 | $49 \times 2^{m-4}$ | 4 |
| circ$(0, 14, 15)(0, 7, 14)$ | $\geq 8$ | $49 \times 2^{m-4}$ | 4 |
| circ$(0, 3, 6)(0, 4, 8)(0, 22, 27)$ | $\geq 12$ | $146 \times 2^{m-5}$ | 4 |
| circ$(0, 4, 8)(0, 5, 10)(0, 13, 26)$ | $\geq 12$ | $148 \times 2^{m-5}$ | 4 |
| circ$(0, 1, 2)(0, 9, 18)(0, 12, 24)$ | $\geq 12$ | $150 \times 2^{m-5}$ | 4 |

According to Theorem 3.4 and Lemma 3.8, Table 3 provides efficient binary cyclic matrices of large dimensions ($2^m \times 2^m$, $m \geq 6$) with a determined number of fixed-points and a lower bound on branch number (independent of dimension) which are reported in Table 4.

5. Conclusion

In this paper, we study CBMs (cyclic binary matrices) which offer an alternative to MDS matrices to reduce the implementation cost. Firstly, we prove a theorem which gives a tool to construct $2n \times 2n$ CBMs with provable lower bound on branch number from $n \times n$ ones. Then, an efficient technique for computing the number of fixed-points of CBMs with dimension $2^m$ is given and the ones with minimal number of fixed-points are characterized. Finally, we propose $16 \times 16$ and $32 \times 32$ CBMs with maximal known branch number and minimal number of fixed-points via searching among the product of sparse CBMs.

We believe that, in the continuation of the studies of this paper, exploring the maximum branch number of $32 \times 32$ CBMs could be a good line of research. Besides, in this class of matrices, finding to what extent implementation costs can be reduced remains an open problem.

References

[1] Specification of the 3GPP confidentiality and integrity algorithms 128-EEA3 and 128-EIA3. document 2: ZUC specification, Online available at https://www.gsma.com/aboutus/wp-content/uploads/2014/12/eea3eia3zucv16.pdf.
[2] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima and T. Tokita, Camellia: A 128-bit block cipher suitable for multiple platforms - design and analysis, in Selected Areas in Cryptography (Waterloo, ON, 2000), 39–56, Lecture Notes in Comput. Sci., 2012, Springer, Berlin, 2001.
[3] B. Aslan and M. T. Sakalli, Algebraic construction of cryptographically good binary linear transformations, Security and Communication Networks, 7 (2014), 53–63.
[4] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita and F. Regazzoni, Mi-dori: A block cipher for low energy (extended version), Advances in Cryptology - ASIACRYPT 2015. Part II, 9453 (2015), 411–436.
[5] C. Beierle, T. Kranz and G. Leander, Lightweight multiplication in $gf(2^n)$ with applications to MDS matrices, in Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I, 9814 (2016), 625–653.
[6] B. Bilgin, A. Bogdanov, M. Knezevic, F. Mendel and Q. Wang, FIDES: Lightweight authenticated cipher with side-channel resistance for constrained hardware, Cryptographic Hardware and Embedded Systems - CHES 2013, 8086 (2013), 142–158.
[7] J. Daemen and V. Rijmen, Rijndael for AES, in AES Candidate Conference, 2000, 343–348.
[8] C. Doerbraunig, M. Eichlseder, F. Mendel and M. Schl"{o}fer, Ascon v1.2, Online available at http://competitions.cr.yp.to/round3/asconv12.pdf.

[9] P. Ekdahl and T. Johansson, A new version of the stream cipher SNOW, in Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John’s, Newfoundland, Canada, August 15-16, 2002. Revised Papers (eds. K. Nyberg and H. M. Heys), vol. 2595 of Lecture Notes in Computer Science, Springer, 2003, 47–61.

[10] D. Feng, X. Feng, W. Zhang, X. Fan and C. Wu, Loiss: A byte-oriented stream cipher, in Coding and Cryptology - Third International Workshop, IWCC 2011, Qingdao, China, May 30–June 3, 2011. Proceedings, 6639 (2011), 109–125.

[11] J. Guo, J. Jean, I. Nikolic, K. Qiao, Y. Sasaki and S. M. Sim, Invariant subspace attack against midori64 and the resistance criteria for s-box designs, IACR Trans. Symmetric Cryptol., 2016 (2016), 33–56.

[12] Z. Guo, R. Liu, S. Gao, W. Wu and D. Lin, Direct construction of optimal rotational-xor diffusion primitives, IACR Trans. Symmetric Cryptol., 2017 (2017), 169–187.

[13] Z. Guo, W. Wu and S. Gao, Constructing lightweight optimal diffusion primitives with feistel structure, in Selected Areas in Cryptography - SAC 2015 - 22nd International Conference, Sackville, NB, Canada, August 12-14, 2015, Revised Selected Papers (eds. O. Dunkelman and L. Keliher), vol. 9566 of Lecture Notes in Computer Science, Springer, 2016, 352–372.

[14] Z. Guo, W. Wu and S. Gao, Constructing lightweight optimal diffusion primitives with feistel structure, in Selected Areas in Cryptography - SAC 2015 - 22nd International Conference, Sackville, NB, Canada, August 12-14, 2015, Revised Selected Papers (eds. O. Dunkelman and L. Keliher), vol. 9566 of Lecture Notes in Computer Science, Springer, 2016, 352–372.

[15] D. Feng, X. Feng, W. Zhang, X. Fan and C. Wu, Loiss: A byte-oriented stream cipher, in Coding and Cryptology - Third International Workshop, IWCC 2011, Qingdao, China, May 30–June 3, 2011. Proceedings, 6639 (2011), 109–125.

[16] M. Grassl, Code Tables: Bounds on the parameters of various types of codes, Online available at http://www.codetables.de/main.html, last access Jan. 2020.

[17] J. Guo, J. Jean, I. Nikolic, K. Qiao, Y. Sasaki and S. M. Sim, Invariant subspace attack against midori64 and the resistance criteria for s-box designs, IACR Trans. Symmetric Cryptol., 2016 (2016), 33–56.

[18] Z. Guo, R. Liu, S. Gao, W. Wu and D. Lin, Direct construction of optimal rotational-xor diffusion primitives, IACR Trans. Symmetric Cryptol., 2017 (2017), 169–187.

[19] Z. Guo, W. Wu and S. Gao, Constructing lightweight optimal diffusion primitives with feistel structure, in Selected Areas in Cryptography - SAC 2015 - 22nd International Conference, Sackville, NB, Canada, August 12-14, 2015, Revised Selected Papers (eds. O. Dunkelman and L. Keliher), vol. 9566 of Lecture Notes in Computer Science, Springer, 2016, 352–372.

[20] M. Kanda, S. Moriai, K. Aoki, H. Ueda, Y. Takashima, K. Ohta and T. Matsumoto, E2 - a new 128-bit block cipher, E83-A, (2000), 48–59.

[21] B. Koo, H. S. Jang and J. H. Song, On constructing of a 32 × 32 binary matrix as a diffusion layer for a 256-bit block cipher, in Information Security and Cryptology - ICISC 2006, 8th International Conference, Yokohama, Japan, October 10-13, 2006, Proceedings, 2006, 46–59.

[22] G. Leander, B. Minaud and S. Renjom, A generic approach to invariant subspace attacks: Cryptanalysis of robin, iscream and zorro, in Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26–30, 2015, Proceedings, Part I, 2015, 254–283.

[23] S. Ling and C. Xing, Coding Theory: A First Course, Cambridge University Press, 2004, https://books.google.com/books?id=Wh8sV31SwC.

[24] A. M. Rishakani, S. M. Dehnavi, M. M. Shamsabad, H. Maimani and E. Pasha, New concepts in design of lightweight mds diffusion layers, 2014 11th International ISC Conference on Information Security and Cryptology, 2014.

[25] M. Sajadieh, M. Dakhilalian and H. Mala, Perfect involutory diffusion layers based on invertibility of some linear functions, IET Information Security, 5 (2011), 228–236.

[26] M. Sajadieh, M. Dakhilalian, H. Mala and P. Sepehrdad, Efficient recursive diffusion layers for block ciphers and hash functions, J. Cryptology, 28 (2015), 240–256.

[27] M. T. Sakalli and B. Aslan, On the algebraic construction of cryptographically good 32×32 binary linear transformations, J. Computational Applied Mathematics, 259 (2014), 485–494.

[28] R. Sch{"u}rer and W. C. Schmid, MinT: A Database for Optimal Net Parameters, In: Niederreiter H., Talay D. (eds) Monte Carlo and Quasi-Monte Carlo Methods 2004. Springer, Berlin, Heidelberg.
[27] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall and N. Ferguson, Twofish: A 128-bit block cipher.

Received September 2019; 1st revision October 2019; final revision December 2019.

E-mail address: am.rishakani@sru.ac.ir
E-mail address: dehnavism@ipm.ir
E-mail address: m_mirzaee@sbu.ac.ir
E-mail address: nbagheri@sru.ac.ir