The ISC Int'l Journal of Information Security
ISeCure

November 2023, Volume 15, Number 3 (pp. 129–138)
http://www.isecure-journal.org

Presented at the ISCISC'2023 in Tehran, Iran.

New Directions in the Design of Binary Matrices for SPN Block Ciphers **

Mahdi Sajadieh [1,*] and Arash Mirzaei [2]

[1] *Department of Electrical Engineering, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran*
[2] *Faculty of Information Technology, Monash University, Australia*

ARTICLE INFO.

Keywords:
Block Cipher, Diffusion Layer, Binary Matrix, Active S-Box

Type:
Research Article

doi: 10.22042/isecure.2024.421696.1035

ABSTRACT

The diffusion layer plays an important role in a block cipher. Some block ciphers, such as ARIA, Camellia, and Skinny use binary matrices as diffusion layers which can be efficiently implemented in hardware and software. In this paper, the goal is to propose some new binary matrices with suitable values for the active S-boxes for R rounds. Firstly, some new 16×16 matrices are proposed whose software implementations are better than the corresponding one for the ARIA block cipher. Also, the values for the minimum active S-boxes for these matrices are greater than the corresponding values for the ARIA block cipher for $R > 5$. To design 32×32 matrices, a structure with a special form is proposed. Using this structure, a 32×32 binary matrix is proposed which guarantees at least 48 active S-boxes for 8 rounds of an SPN structure with this matrix as its diffusion layer. By extending this structure, a 32×32 non-binary matrix is presented which results in at least 60 active S-boxes after 8 rounds.

1 Introduction

Nowadays, huge amounts of information are transmitted in networks which is why information security is a subject that should be paid utmost attention to. To have confidentiality as one of the main aspects of security, encryption is utilized. Encryption algorithms can be divided into symmetric key and public key. Block cipher is one of the most important types of symmetric key algorithms. Modern block ciphers consists of several rounds. In each round input should be confused and diffused for which substitution and permutation (diffusion) layers are used, respectively. The substitution layer is often implemented using several small non-linear substitution boxes (S-boxes) and the diffusion layer combines the output of each S-box.

To represent diffusion layers, an $n \times n$ matrix is used where n is the number of S-boxes in each round. Generally, the matrix can be over finite fields, and in the simplest case, $n \times n$ matrix can be over $GF(2)$ (binary matrix). The E2 [1] and the Camellia block ciphers [2] use 8×8 binary matrices with the branch number of 5 while ARIA [3] uses an involutory 16×16 binary matrix with the branch number of 8 as a diffusion layer. Also, Skinny [4] and Midori [5] are the block ciphers using binary matrices with good properties to implement in hardware. In [6] a 32×32 binary matrix with the branch number 10 was proposed.

* Corresponding author.
**The ISCISC'2023 program committee effort is highly acknowledged for reviewing this paper.
Email addresses: m.sajadieh@khuisf.ac.ir, arash.mirzaei@monash.edu
ISSN: 2008-2045

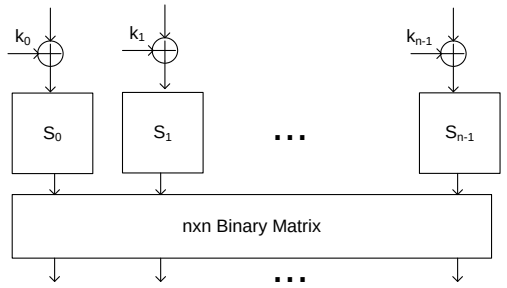

Figure 1. One round of SPN structure with the binary diffusion layer

Some 24×24 binary matrices are introduced in [7] with good implementation properties for lightweight block ciphers.

If the diffusion layer has good properties, then many S-boxes will be active after a few rounds which is why the diffusion layer design has a deep impact on the resistance of block ciphers against linear [8] and differential cryptanalysis [9].

In this paper, to investigate the diffusion properties of proposed binary matrices, we employ them as the diffusion layer of the SPN structure shown in Fig. 1. This structure uses n S-boxes of length m bits. So, the diffusion layer of the considered SPNs is $n \times n$ binary matrices.

We use the following definitions in this paper.

Definition 1. For any given $\triangle x, \triangle y, \Gamma x, \Gamma y \in Z_2^m$, the differential and linear probabilities of S-box S, are defined as:

$$DP(\triangle x \rightarrow \triangle y) = \frac{\#\{x \in Z_2^m | s(x) \oplus s(x \oplus \triangle x) = \triangle y\}}{2^m} \quad (1)$$

$$LP(\Gamma x \rightarrow \Gamma y) = (2 \times \frac{\#\{x \in Z_2^m | x.\Gamma x = s(x).\Gamma y\}}{2^m} - 1)^2 \quad (2)$$

where $a.b$ denotes the parity of the bit-wise product of a and b.

Definition 2. The maximum differential and linear probabilities of an S-box are defined as:

$$p = \max_{\triangle x \neq 0, \triangle y} DP(\triangle x \rightarrow \triangle y) \quad (3)$$

$$q = \max_{\Gamma x, \Gamma y \neq 0} LP(\Gamma x \rightarrow \Gamma y) \quad (4)$$

Definition 3. An S-box is a differential (resp. linear) active S-box if the input difference (resp. output mask) value of the S-box is nonzero.

For a linear diffusion layer which can be represented by a matrix multiplication $\mathbf{y} = \mathbf{A}.\mathbf{x}$, it can be shown that the output difference ($\triangle \mathbf{y}$) is obtained from the input difference ($\triangle \mathbf{x}$) by $\triangle \mathbf{y} = \mathbf{A}. \triangle \mathbf{x}$:

$$\mathbf{y}_1 = \mathbf{A}.\mathbf{x}_1,\ \mathbf{y}_2 = \mathbf{A}.\mathbf{x}_2 \Rightarrow \triangle \mathbf{y} = \mathbf{A}. \triangle \mathbf{x} \quad (5)$$

Also, the input linear mask (Γ_x) is obtained from the output linear mask (Γ_y) by $\Gamma_x = \mathbf{A}^t \Gamma_y$ where $\mathbf{A}^t$ denotes the transpose of matrix $\mathbf{A}$:

$$\mathbf{y} = \mathbf{A}.\mathbf{x} \Rightarrow \Gamma_y^t.\mathbf{y} = \Gamma_y^t.\mathbf{A}.\mathbf{x} = (\mathbf{A}^t.\Gamma_y)^t.\mathbf{x} \Rightarrow \Gamma_x = A^t \Gamma_y \quad (6)$$

Definition 4. The minimum number of differential active S-boxes of two rounds of a linear diffusion layer D is called the differential branch number and is defined as [10]:

$$\beta_d(D) = \min_{\mathbf{x} \neq \mathbf{0}} \{w(\mathbf{x}) + w(D(\mathbf{x}))\} \quad (7)$$

where $w(\mathbf{x})$ is the number of non-zero elements in vector $\mathbf{x}$.

From relations (5) and (6), we can conclude that for a block cipher with matrix $\mathbf{A}$ as its diffusion layer, the minimum number of linear active S-boxes (MNLAS) is equivalent to the minimum number of differential active S-boxes (MNDAS) of that block cipher with matrix $\mathbf{A}^t$ as diffusion layer.

1.1 Notations

In this paper, to avoid the representation of large matrices, some notations are introduced as follows:

Definition 5. $\mathbf{z}_{n \times m}$ shows an all-zeros $n \times m$ matrix. For example

$$\mathbf{z}_{2\times 3} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (8)$$

Definition 6. $\mathbf{P}_{i_0,i_1,i_2,i_3}$ represents a 4×4 binary matrix with one 0 in each row where i_j ($i_j \in 0, 1, 2, 3$) shows the position of zero elements in the j^{th} row. For example:

$$\mathbf{P}_{3,1,2,0} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \quad (9)$$

Definition 7. $\mathbf{\Pi}^{\mathbf{n}}_{(\mathbf{i_0},\mathbf{i_1},\ldots,\mathbf{i_{n-1}})}$ represents an $n \times n$ binary matrix with one 1 in each row where i_j ($i_j \in 0, 1, \cdots, n-1$) shows the position of non-zero element in the j^{th} row. For example:

$$\mathbf{\Pi}^{\mathbf{8}}_{(\mathbf{0,5,4,7,2,1,6,3})} = \begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1\\ 0&0&1&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&1&0&0&0&0 \end{pmatrix} \quad (10)$$

1.2 Our Contribution

Counting the number of (linear or differential) active S-boxes of consecutive rounds of an SPN is often done by considering the truncated values for the block state. It means each all-zero or non-zero S-box input and output is shown by one bit **0** or **1**, respectively. Thus, the block state is shown by an n-bit vector instead of an nm-bit value. In [11] a counting method was proposed that can be used to compute active S-boxes for several rounds of Rijndael. The complexity of this method linearly increases with the rise in the number of rounds.

It is proved that the ratio of MNDAS and MNLAS to number of rounds for binary matrices proposed in[3] never exceeds Br/2 where Br is the matrix branch number. Then some 16×16 binary matrices will be proposed for which the mentioned ratio exceeds Br/2 with an increase in number of rounds. For the new matrices MNDAS for R rounds (e.g. $R = 40$) will be determined using the method mentioned in [11]. To find MNLAS, the method is applied to $\mathbf{A}^t$ instead of $\mathbf{A}$. The proposed matrices have the largest value of MNDAS and MNLAS among existing binary matrices for $R > 5$ rounds.

The mentioned method cannot be used for 32×32 binary matrices because of the method running time and the limitation of the used memory. Thus, some limitations have been applied to the matrices of the mentioned size and the counting method has been modified. In this paper, a new 32×32 binary matrix is also proposed. 8 rounds of an SPN structure using the new matrix as the diffusion layer has at least 48 (linear or differential) active S-boxes. The counting method can be extended for a new 32×32 non-binary matrix which results in at least 60 (linear or differential) active S-boxes after 8 rounds. All of the proposed 16×16 and 32×32 matrices can be efficiently implemented.

The rest of this paper is organized as follows. In Section 2, the counting method mentioned in [11] with some slight changes are explained. Using this method, in Section 2, some 16×16 binary matrices are offered for which the MNDAS of several rounds are determined. In Section 3, a 32×32 binary matrix with at least 48 active S-boxes for 8 rounds and also a 32×32 non-binary matrix with at least 60 active S-boxes for 8 rounds will be proposed and the conclusion is in the last section.

2 New 16×16 Binary Matrices

In this section, some new 16×16 binary matrices are proposed which provide the following properties when used as the diffusion layer in the SPN structure:
− The active S-boxes for R rounds of the proposed SPNs is large.
− The SPN can be efficiently implemented.
According to the mentioned method, to compute the MNDAS for R rounds of the SPN structure using the new matrices, array T must be of size $R \times 2^{16}$.

The main idea to propose a new 16×16 matrix is the usage of the AES structure whose MDS matrix is replaced by binary matrices. By using the mentioned method, we observed that the MNDAS and MNLAS for $4r$ rounds are greater than the expected value which is $16r$. The first proposed matrix is as follows.

$$\mathbf{A}_1 = \begin{pmatrix} 1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1\\ 0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1\\ 1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&0\\ 0&0&0&1&1&0&0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0\\ 0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&1&0&0&0&0&1&1&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1&1&0&0&0&0&1&0&0\\ 0&0&1&0&0&0&0&1&0&0&0&0&0&1&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&1&1&0&0&0\\ 0&1&0&0&0&0&1&0&0&0&0&0&1&0&0&0\\ 0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&0&1&1&0&0&0 \end{pmatrix} \quad (11)$$

This matrix is the result of multiplying $\mathbf{\Pi}^{\mathbf{16}}_{\mathbf{0,5,10,15,4,9,14,3,8,13,2,7,12,1,6,11}}$ by $\mathbf{Q}1$, where $\mathbf{Q}1$ is given below.

$$\mathbf{Q1} = \begin{pmatrix} \mathbf{P_{3,1,0,2}} & \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} \\ \mathbf{z_{4\times4}} & \mathbf{P_{0,2,3,1}} & \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} \\ \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} & \mathbf{P_{1,3,2,0}} & \mathbf{z_{4\times4}} \\ \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} & \mathbf{P_{2,3,0,1}} \end{pmatrix} \quad (12)$$

The differential branch number of $\mathbf{A}_1$ is 4. The MNDAS of r rounds of the SPN structure using $\mathbf{A}_1$ as the diffusion layer has been shown in Table 1. It can be observed from Table 1 that when the number of rounds increases, the ratio of the MNDAS to the number of rounds, also increases. Although there are other options for matrices Q and $\mathbf{\Pi}$, the results for the mentioned one are the best.

To investigate the security of the SPN cipher with $\mathbf{A}_1$ as the diffusion layer against the impossible differential attack, we checked the length of impossible differentials. We looked for truncated differentials of probability 1 in both encryption and decryption directions and found out that after four rounds there is no byte whose difference is zero and there is no set of bytes that the XOR of their differences is zero with probability 1. This ensures us that this SPN does not have any impossible differential in more than 6 rounds. In other words, we found a 6-round impossible differential for the SPN that uses $\mathbf{A}_1$.

It should be noted that the A1 matrix is better than the Skinny matrix in terms of the active S-boxes and the resistance to impossible differential attack, but in terms of hardware implementation in Skinny, 12 XOR operations are needed forthe diffusion layer but for A1 this value requires 32 XOR operations (28 operations with an auxiliary variable). Due to the fact that in order to compare the implementation of two block ciphers, it is necessary to compare the number of rounds of the algorithm as well as the designed S-box, it is not possible to have a detailed review between these diffusion layers of block ciphers in this paper.

To find a binary matrix with at a most 4-round impossible differential, we searched all permutations $\mathbf{\Pi}$, but for all of them, there are impossible differentials on 5 or more rounds. To overcome this problem, another matrix, $\mathbf{A}_2$, is proposed which is the result of multiplying $\mathbf{Q2}$ by $\mathbf{\Pi} = \mathbf{\Pi}^{\mathbf{16}}_{\mathbf{2,7,8,14,3,4,9,15,0,5,10,12,1,6,11,13}}$, where $\mathbf{Q2}$ is of the form:

$$\mathbf{Q2} = \begin{pmatrix} \mathbf{P_{3,0,1,2}} & \mathbf{P_{0,1,2,3}} & \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} \\ \mathbf{P_{3,0,1,2}} \oplus \mathbf{P_{0,1,2,3}} & \mathbf{P_{0,1,2,3}} & \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} \\ \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} & \mathbf{P_{1,2,3,0}} & \mathbf{P_{1,2,3,0}} \\ \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} & \mathbf{P_{1,2,3,0}} \oplus \mathbf{P_{2,3,0,1}} & \mathbf{P_{2,3,0,1}} \end{pmatrix} \quad (13)$$

The form of the upper left 8×8 sub-block of $\mathbf{Q2}$ is like the matrix used in the Camellia block cipher.

The matrix $\mathbf{A_2} = \mathbf{Q2} \times \mathbf{\Pi}$ has at most 4-round impossible differential distinguisher.

The MNDAS of r rounds of the SPN structure using $\mathbf{A}_2$ as the diffusion layer has been shown in Table 2. The inverse of matrix $\mathbf{Q2}$ can be efficiently implemented similar to the implementation of matrix $\mathbf{Q2}$. Software implementation of this matrix has been described in Appendix 1.

3 The 32×32 Matrix

The method proposed in Section 2 cannot be directly used for 32×32 binary matrices to determine the MNDAS of r rounds because of the large running time of the algorithm for each round.
Thus, a new method is used to determine a lower bound for the active S-boxes of r rounds of an SPN structure which uses a 32×32 binary matrix as its diffusion layer. This method is different from the method described in Section 2 in the following aspects:
(1) Here, two m-bit sub-blocks (instead of one m-bit sub-block) are truncated to one bit. Thus, the number of columns of the array T will be $2^{\frac{n}{2}}$
(2) In each iteration of the search algorithm two rounds will be analyzed (instead of one round in each iteration). Thus the array T will be of size $\frac{r}{2} \times 2^{\frac{n}{2}}$ (instead of $r \times 2^n$).

In the following, we introduce a new 32×32 binary matrix with suitable linear and differential properties. Consider the following 32×32 binary matrix $\mathbf{A}_3 = \mathbf{P}.\mathbf{\Pi}$, where:

$$\mathbf{P} = \begin{pmatrix} \mathbf{B_{4\times4}} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} \\ z_{4\times4} & \mathbf{B_{4\times4}} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} \\ z_{4\times4} & z_{4\times4} & \mathbf{B_{4\times4}} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} \\ z_{4\times4} & z_{4\times4} & z_{4\times4} & \mathbf{B_{4\times4}} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} \\ z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & \mathbf{B_{4\times4}} & z_{4\times4} & z_{4\times4} & z_{4\times4} \\ z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & \mathbf{B_{4\times4}} & z_{4\times4} & z_{4\times4} \\ z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & \mathbf{B_{4\times4}} & z_{4\times4} \\ z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & z_{4\times4} & \mathbf{B_{4\times4}} \end{pmatrix} \quad (14)$$

and

$$\mathbf{\Pi} = \mathbf{\Pi}^{\mathbf{32}}_{\mathbf{I}} \quad (15)$$

where

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

The matrix Π is similar to the matrix representation of permutation P_4 introduced in [12]. $\mathbf{B_{4\times4}}$ is a

Table 1. MNDAS of r rounds of the SPN structure using $\mathbf{A}_1$ as the diffusion layer

r	1	2	3	4	5	6	7	8	9	10	12	14	...	20	...	39	40
MNDAS	1	4	7	16	22	28	33	38	43	48	60	70	...	106	...	217	224
MNLAS	1	4	7	16	22	28	33	38	43	48	60	70	...	106	...	217	224
$MNDAS/r$	1.00	2.00	2.33	4.00	4.40	4.66	4.71	4.75	4.77	4.80	5	5	...	5.3	...	5.56	5.60

Table 2. MNDAS of r rounds of the SPN structure using $\mathbf{A}_2$ as the diffusion layer

r	1	2	3	4	5	6	7	8	9	10	12	14	...	20	...	39	40
MNDAS	1	5	8	16	20	25	31	36	41	47	58	68	...	101	...	205	211
$MNDAS/r$	1	2.50	2.66	4	4	4.16	4.42	4.5	4.55	4.7	4.83	4.85	...	5.05	...	5.25	5.27
MNLAS	1	5	8	16	21	25	29	36	40	47	57	69	...	102	...	207	212
$MNLAS/r$	1	2.5	2.66	4	4.2	4.16	4.14	4.5	4.44	4.7	4.66	4.92	...	5.1	...	5.25	5.3

4×4 binary matrix as follows

$$\mathbf{B} = \mathbf{P}_{1,2,3,0} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \tag{16}$$

The sequence of transformations applied to the input of an SPN structure (regardless of the key addition) can be written as

$$\cdots \circ \mathbf{P} \circ \mathbf{\Pi} \circ \mathbf{S} \circ \mathbf{P} \circ \mathbf{\Pi} \circ \mathbf{S} \circ \mathbf{P} \circ \mathbf{\Pi} \circ \mathbf{S} \circ \mathbf{P} \circ \mathbf{\Pi} \circ \mathbf{S} \tag{17}$$

Concerning commutativity of Π and S, this sequence can be rewritten as:

$$\cdots \circ \mathbf{\Pi} \circ \mathbf{P} \circ \mathbf{\Pi} \circ \mathbf{S} \circ \mathbf{P} \circ \mathbf{S} \circ \mathbf{\Pi} \circ \mathbf{P} \circ \mathbf{\Pi} \circ \mathbf{S} \circ \mathbf{P} \circ \mathbf{S} \circ \mathbf{\Pi} \tag{18}$$

Fig 2 shows the SPN structure corresponding to the relation (18) which is composed of two different types of rounds $\mathbf{S} \circ \mathbf{P} \circ \mathbf{S}$ and $\mathbf{\Pi} \circ \mathbf{P} \circ \mathbf{\Pi}$ applied alternately. The input to the r^{th} sub-cipher $\mathbf{S} \circ \mathbf{P} \circ \mathbf{S}$ is shown by $(I_0^r, I_1^r, \cdots, I_{31}^r)$, while the input to the r^{th} sub-cipher $\mathbf{\Pi} \circ \mathbf{P} \circ \mathbf{\Pi}$ is represented by $(O_0^r, O_1^r, \cdots, O_{31}^r)$.

In Section 2, the truncated value of I_i^r was represented by $I_i^{r,t}$ which could be $\mathbf{0}$ or $\mathbf{1}$. Now the double-truncated value for every pair (I_{2i}^r, I_{2i+1}^r), $i = 0, 1, \cdots, 15$ is shown by $\bar{I}_i^{r,t}$ which can be $\bar{\mathbf{0}}$ or $\bar{\mathbf{1}}$. If both of I_{2i}^r and I_{2i+1}^r are all-zero, $\bar{I}_i^{r,t}$ will be shown by $\bar{\mathbf{0}}$. In other cases, it will be shown by $\bar{\mathbf{1}}$. $(I_0^r, I_1^r, \cdots, I_{31}^r)$ can be shown with the double-truncated vector $(\bar{I}_0^{r,t}, \bar{I}_1^{r,t}, \cdots, \bar{I}_{15}^{r,t})$ which corresponds to the double-truncated value $\sum_{i=0}^{i=15} \bar{I}_i^{r,t}.2^i$. This double-truncated value is a number between 0 and 65535. Similar notation can be used for the input to the i^{th} application of $\mathbf{\Pi} \circ \mathbf{P} \circ \mathbf{\Pi}$.

Applying the S-box layer ($\mathbf{S}$) to a double-truncated vector does not change its value, i.e., the S transformation is considered an identity transformation on a double-truncated vector. As can be seen from Fig. 2 the Π is equivalent to a permutation on 16 $2m$-bit values. Therefore Π acts as a permutation on 16 elements of a double truncated vector. This transformation is one-to-one. The application of $\mathbf{P}$ is realized by a parallel application of 8 $\mathbf{B}$ functions. Double-truncated input to a $\mathbf{B}$ can be shown by $(\bar{\mathbf{0}}, \bar{\mathbf{0}})$, $(\bar{\mathbf{0}}, \bar{\mathbf{1}})$, $(\bar{\mathbf{1}}, \bar{\mathbf{0}})$ or $(\bar{\mathbf{1}}, \bar{\mathbf{1}})$. Applying $\mathbf{B}$ to $(\bar{\mathbf{0}}, \bar{\mathbf{1}})$ and $(\bar{\mathbf{1}}, \bar{\mathbf{0}})$ results in $(\bar{\mathbf{1}}, \bar{\mathbf{1}})$. Applying $\mathbf{B}$ to $(\bar{\mathbf{1}}, \bar{\mathbf{1}})$ results in $(\bar{\mathbf{0}}, \bar{\mathbf{1}})$, $(\bar{\mathbf{1}}, \bar{\mathbf{0}})$ or $(\bar{\mathbf{1}}, \bar{\mathbf{1}})$. A function $\mathbf{B}$ is called active if its input (or output) is not all zero. Each active $\mathbf{B}$ in the application of $\mathbf{S} \circ \mathbf{P} \circ \mathbf{S}$ makes at least 4 S-boxes active in its input and its output.

Based on Fig. 2, the analysis of the mentioned structure is as follows. To find a lower bound for MNDAS of R rounds of the mentioned SPN, all of the double-truncated values for the input of the SPN shown in Fig. 2 and all of the possible transitions by application of the $\mathbf{P}$ layers will be considered. In the considered paths, each active $\mathbf{B}$ in $\mathbf{S} \circ \mathbf{P} \circ \mathbf{S}$ sequence of functions, adds four active S-boxes to the lower bound of MNDAS.

Considering a double-truncated vector for the first application of $\mathbf{S} \circ \mathbf{P} \circ \mathbf{S}$, the number of paths is exponentially increased by adding each $\mathbf{P}$ layer in the sequence of applied functions. To overcome this problem, an array T of the size $\frac{R}{2} \times 2^{16}$ is built. The numbering of the rows and the columns of the array starts from 1 and 0, respectively. $T[i][j]$ shows the MNDAS of the characteristics which finished with the double-truncated value j (a number from 0 to 65535) at the end of the i^{th} application of $\mathbf{\Pi} \circ \mathbf{P} \circ \mathbf{\Pi}$. Computation of $i+1^{th}$ row of T is performed by using the i^{th} row of T. $T[r+1][j]$ is computed using the exhaustive search over $T[r][i], i = 0, \cdots, 65535$ and all of the possible transitions from $\bar{I}^{r+1,t} = i$ to $O^{r+1,t} = k$, $k = 0, 1, \cdots, 65535$ by $i+1^{th}$ application of the $\mathbf{S} \circ \mathbf{P} \circ \mathbf{S}$ and all of the possible transitions from $O^{r+1,t} = k$

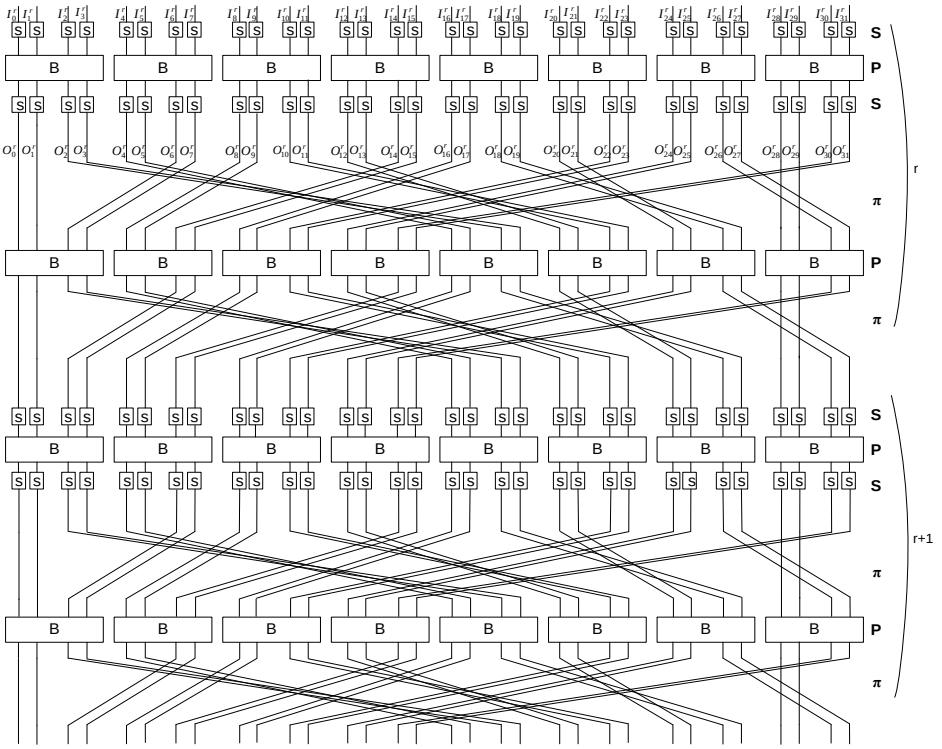

Figure 2. Equivalent form of four rounds of the SPN with $\mathbf{A}_3$ as the diffusion layer

to $I^{r+2,t} = j$ by $i+1^{th}$ application of the $\mathbf{\Pi} \circ \mathbf{P} \circ \mathbf{\Pi}$. For each active $\mathbf{B}$ in the $i+1^{th}$ application of the $\mathbf{S} \circ \mathbf{P} \circ \mathbf{S}$, 4 is added to the lower bound of the number of active S-boxes of the characteristic. Thus we have

$$T[r+1][j] = \min_{i,k} \Big\{ T[r][i] + 4 \times l \;\Big|\; \bar{I}^{r+1,t} = i \rightarrow \bar{I}^{r+2,t} = j \; is \; possible and \; \bar{I}^{r+1,t} \; makes \; l \; different \; \mathbf{B}s \; active \Big\}. \tag{19}$$

For the mentioned structure the minimum non-zero values of the 4^{th} row of the T array is 48 which means that 8 rounds of the structure have at least 48 active S-boxes.

Table 3 and Table 4 show the possible transitions from $\bar{I}^{r,t}$ to $\bar{O}^{r,t}$ and $\bar{O}^{r,t}$ to $\bar{I}^{r+1,t}$, respectively.

Example 1. For computing $T[r+1][153]$, it can be seen from Table 4 that one of the possible values for $\bar{O}^{r+1,t}$ which results in $\bar{I}^{r+2,t} = 153$ is 3. From Table 3 it is concluded that 1,2 and 3 are some values for $\bar{I}^{r+1,t}$ which can be transformed to $\bar{O}^{r+1,t} = 3$. So minimum of $T[r][1] + 1 \times 4$, $T[r][2] + 1 \times 4$ and $T[r][3] + 1 \times 4$ can be a candidate for $T[r+1][153]$. Considering all of the possible transformations results in the final value of $T[r+1][153]$.

The most important properties of $\mathbf{B}$ which affect the resultant bound are as follows:
(1) The minimum active sub-blocks in the input and output of $\mathbf{B}$ takes its maximum possible value.
(2) Double-truncated pairs $(\bar{\mathbf{0}}, \bar{\mathbf{1}})$ and $(\bar{\mathbf{1}}, \bar{\mathbf{0}})$ results in $(\bar{\mathbf{1}}, \bar{\mathbf{1}})$.

To see the effect of the second property on the overall diffusion of the SPN, consider the following matrix $\mathbf{B}_1$:

$$\mathbf{B}_1 = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \tag{20}$$

The minimum positive active sub-blocks in the input and output of $\mathbf{B}_1$ and $\mathbf{B}$ are the same, but if $\mathbf{B}_1$ is used instead of $\mathbf{B}$ in the SPN mentioned in this section, the obtained lower bound of MNDAS for 8 rounds of the SPN would be 16, which is much smaller than the lower bound for SPN using $\mathbf{B}$. This egregious difference is the result of the possible transformations of $(\bar{\mathbf{0}}, \bar{\mathbf{1}})$ to $(\bar{\mathbf{0}}, \bar{\mathbf{1}})$ and $(\bar{\mathbf{1}}, \bar{\mathbf{0}})$ to $(\bar{\mathbf{1}}, \bar{\mathbf{0}})$ for the matrix $\mathbf{B}_1$.

For the introduced method in this section, if the matrix $\mathbf{P}$ uses 8 different matrices $\mathbf{B}$ with the two mentioned properties, the obtained lower bound will not change.

Table 3. Possible transitions from $\bar{I}^{r,t}$ to $\bar{O}^{r,t}$ and the active Bs of each transition

$\bar{I}^{r,t}$	$\bar{O}^{r,t}$	The active **B**s
0	0	0
1	3	1
2	3	1
3	1,2,3	1
4	12	1
5	15	2
6	15	2
7	13,14,15	2
⋮	⋮	⋮
65535	21845,21846,21847,21849,...,65535	8

Table 4. Possible transitions from $\bar{O}^{r,t}$ to $\bar{I}^{r+1,t}$

$\bar{O}^{r,t}$	$\bar{I}^{r+1,t}$
0	0
1	9
2	144
3	153
4	2304
5	2313
6	2448
7	2457
⋮	⋮
15	39321
⋮	⋮
65535	21845,21852,21853,21957,...,65535

The bound can be increased if we use MDS matrices (in $GF(2^8)$) instead of the binary matrix **B** in relation (14). In this case, using the mentioned method, the lower bound of the MNDAS for 8 rounds is increased to 60. For a 4×4 MDS matrix the possible double-truncated transitions would be the same as the ones for matrix **B**. The only difference is that for each active **B**, 5 is added to the lower bound of MNDAS. The SPN structures using the new diffusion layer, have at most 6-round impossible differentials. It can be implemented efficiently similar to Rindael-256 because Π is similar to ShiftRows and **B** is similar to MixColumns. The lower bound of the MNDAS for 8 rounds of Rindael-256 is 50 which is much less than the bound 60 calculated for the proposed SPN. The new SPN can be used to design a block cipher with a block size of 256 bits.

4 Conclusion

In this paper, we concentrated on the design of binary matrices with suitable cryptographic properties. Firstly, using the AES structure, a new 16×16 binary matrix was proposed. Although, this matrix has appropriate results as far as the active S-boxes for R rounds of the Camellia block cipher is concerned, its resistance against impossible differential attack is not preferable. To solve this problem, The Camellia binary matrix idea and byte permutation (similar to ShiftRows in AES) were utilized resulting in a new structure. Using this new structure, a 16×16 matrix was proposed where 10 rounds of an SPN structure using this matrix as its diffusion layer has 47 linear and differential active S-boxes and there is no impossible differential distinguisher for more than 4 rounds.

Due to the limitation of the memory and time for counting the active S-boxes for a 32×32 binary matrix, the counting method mentioned in was modified to provide a lower bound for the active S-boxes. Then based on 4×4 binary matrices, a new structure for 32×32 binary matrices was presented. Using the proposed structure, it was found that the lower bound for the active S-boxes in 8 rounds of the SPN structure employing this matrix as its diffusion layer is 48. Since the counting method used for this structure could be extended to non-binary matrices, the 4×4 binary matrix was replaced by a 4×4 MDS matrix. The lower bound for the number of active S-boxes is 60 for the new non-binary structure. The resultant bound is better than the corresponding lower bound for Rijndael-256 which is 50 [13]. All the proposed diffusion layers proposed in this paper can be efficiently implemented in software.

References

[1] M. Kanda, S. Moriai, K. Aoki, H. Ueda, Y. Takashima, K. Ohta, and T. Matsumoto. E2-A New 128-Bit Block Cipher. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E83-A(1):48–59, 2000.

[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 *SAC 2000*, volume 2012, pages 39–56. Springer-Verlag Berlin Heidelberg, 2001.

[3] D. Kwon, J. Kim, S. Park, S.H. Sung, Y. Sohn, J.H. Song, Y. Yeom, E-J. Yoon, S. Lee, J. Lee, S. Chee, D. Han, and J. Hong. New Block Cipher ARIA. In *ICISC2003*, volume 2971, pages 432–445. Springer-Verlag, 2003.

[4] 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 *Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II 36*, pages 123–153. Springer, 2016.

[5] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In *Advances in Cryptology–ASIACRYPT 2015: 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29–December 3, 2015, Proceedings, Part II 21*, pages 411–436. Springer, 2015.

[6] B. Koo, H. Jang, and J. Song. On constructing of a 32 * 32 binary matrix as a diffusion layer for a 256-bit block cipher. In *ICISC2006*, volume 4296, pages 51–64. Springer-Verlag, 2006.

[7] Muharrem Tolga Sakallı, Sedat Akleylek, Bora Aslan, Ercan Buluş, and Fatma Büyüksaraçoğlu Sakallı. On the construction of and binary matrices with good implementation properties for lightweight block ciphers and hash functions. *Mathematical Problems in Engineering*, 2014, 2014.

[8] M. Matsui. Linear Cryptanalysis Method for DES Cipher. In *EUROCRYPT'93*, volume 765, pages 386–397. Springer-Verlag, 1993.

[9] E. Biham and A. Shamir. Differential Cryptanalysis of DES-like Cryptosystems. In *CRYPTO'90*, volume 537, pages 2–21. Springer-Verlag, 1990.

[10] J. Daemen. *Cipher and Hash Function Design Strategies Based on Linear and Differential Cryptanalysis*. PhD thesis, Elektrotechniek Katholieke Universiteit Leuven, Belgium, 1995.

[11] Mahdi Sajadieh, Arash Mirzaei, Hamid Mala, and Vincent Rijmen. A new counting method to bound the number of active s-boxes in rijndael and 3d. *Designs, Codes and Cryptography*, 83:327–343, 2017.

[12] Hongjun Wu. The hash function jh. *Submission to NIST (round 3)*, 6, 2011.

[13] J. Daemen and V. Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer-Verlag, 2002.

A Software Implementations

Now some points are described about the method of implementing of round function on processors with word length 32 or above. In a round function of the SPN structure after the key addition layer which simply is implemented by XOR operations, the S-box layer and diffusion layer are applied to the input block as follows:

$$Q \circ \Pi \circ S$$

where Q and Π are $n \times n$ binary matrices. The above functions are applied to the input block $X = (x_1, x_2, ..., x_n)$, where each $x_i, i = 1, 2, \cdots, n$ is a byte. Since we can interchange S and Π, $Q \circ \Pi \circ S(X) = Q \circ S \circ \Pi(X)$. Thus, in the following the method of implementation is described. If all of the 8-bit S-boxes used in S-box layer are the same, to implement $Q \circ S$ for all Qs introduced in this paper it suffices to pre-compute and store the following 4 lookup tables each with 256 4-byte word entries:

$$T_{0111}[x] = \begin{pmatrix} 0 \\ s[x] \\ s[x] \\ s[x] \end{pmatrix}, \quad T_{1011}[x] = \begin{pmatrix} s[x] \\ 0 \\ s[x] \\ s[x] \end{pmatrix},$$

$$T_{1101}[x] = \begin{pmatrix} s[x] \\ s[x] \\ 0 \\ s[x] \end{pmatrix}, \quad T_{1110}[x] = \begin{pmatrix} s[x] \\ s[x] \\ s[x] \\ 0 \end{pmatrix}$$

On the other hand, to implement the inverse of the round function the following 4 lookup tables are stored:

$$TI_{0111}[x] = \begin{pmatrix} 0 \\ s^{-1}[x] \\ s^{-1}[x] \\ s^{-1}[x] \end{pmatrix}, \quad TI_{1011}[x] = \begin{pmatrix} s^{-1}[x] \\ 0 \\ s^{-1}[x] \\ s^{-1}[x] \end{pmatrix},$$

$$TI_{1101}[x] = \begin{pmatrix} s^{-1}[x] \\ s^{-1}[x] \\ 0 \\ s^{-1}[x] \end{pmatrix}, \quad TI_{1110}[x] = \begin{pmatrix} s^{-1}[x] \\ s^{-1}[x] \\ s^{-1}[x] \\ 0 \end{pmatrix}$$

The matrix $Q1$ introduced in Section 2 can be divided into 4 independent 4×4 sub-blocks. Thus, applying $Q1 \circ S$ to input vector can be implemented by independent calculation of 4 sub-blocks of the output. Each output sub-block is calculated by 4 table lookups and 3 XOR operations. Calculation of the $Q1^{-1} \circ S^{-1}$ is the same.

Matrix $Q2$ introduced in Section 2 can be divided into two 8×8 sub-blocks which can be implemented independently. Here, the implementation of the first 8×8 sub-block is described. Implementation of the second 8×8 sub-block is performed similarly. To compute $(y_1, y_2, \cdots ..., y_8)$, the following vectors are calculated first:

$$\begin{aligned} z_0 &= P_{3,0,1,2} \times \begin{pmatrix} s[x_1] \\ s[x_2] \\ s[x_3] \\ s[x_4] \end{pmatrix} \\ &= T_{1011}[x_1] \oplus T_{1101}[x_2] \oplus T_{1110}[x_3] \oplus T_{0111}[x_4] \\ z_1 &= P_{0,1,2,3} \times \begin{pmatrix} s[x_5] \\ s[x_6] \\ s[x_7] \\ s[x_8] \end{pmatrix} \\ &= T_{0111}[x_5] \oplus T_{1011}[x_6] \oplus T_{1101}[x_7] \oplus T_{1110}[x_8] \end{aligned}$$

Then (y_1, y_2, y_3, y_4) is computed as follows:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = z_0 \oplus z_1 \tag{A.1}$$

and (y_5, y_6, y_7, y_8) is computed as follows:

$$\begin{pmatrix} y_5 \\ y_6 \\ y_7 \\ y_8 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} \oplus z_0 \lll 8 \tag{A.2}$$

where $z_0 <<< 8$ shows the circular shifting of z_0 by 8 bits. Thus, the calculation of each output sub-block needs 8 table lookups, 8 XOR operations and 1 circular shift operation. To compute $(y_9, y_{10}, \cdots ..., y_{16})$ the following vectors are calculated first:

$$\begin{aligned} z_2 &= P_{1,2,3,0} \times \begin{pmatrix} s[x_9] \\ s[x_{10}] \\ s[x_{11}] \\ s[x_{12}] \end{pmatrix} \\ &= T_{1110}[x_9] \oplus T_{0111}[x_{10}] \oplus T_{1011}[x_{11}] \oplus T_{1101}[x_{12}] \\ z_3 &= P_{1,2,3,0} \times \begin{pmatrix} s[x_{13}] \\ s[x_{14}] \\ s[x_{15}] \\ s[x_{16}] \end{pmatrix} \\ &= T_{1110}[x_{13}] \oplus T_{0111}[x_{14}] \oplus T_{1011}[x_{15}] \oplus T_{1101}[x_{16}] \end{aligned}$$

Then $(y_9, y_{10}, y_{11}, y_{12})$ is computed as follows:

$$\begin{pmatrix} y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{pmatrix} = z_2 \oplus z_3 \tag{A.3}$$

and $(y_{13}, y_{14}, y_{15}, y_{16})$ is computed as follows:

$$\begin{pmatrix} y_{13} \\ y_{14} \\ y_{15} \\ y_{16} \end{pmatrix} = \left(\begin{pmatrix} y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{pmatrix} \right) \lll 8 \oplus z_2 \tag{A.4}$$

Thus, the calculation of each output sub-block needs 8 table lookups, 8 XOR operations, and 1 circular shift operation.

Inverse of $Q2$ is as follows.

$$\mathbf{Q2^{\hat{}}\text{-}1} = \begin{pmatrix} \mathbf{P_{0,1,2,3}} & \mathbf{P_{0,1,2,3}} & \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} \\ \mathbf{P_{3,0,1,2}} \oplus \mathbf{P_{0,1,2,3}} & \mathbf{P_{3,0,1,2}} & \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} \\ \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} & \mathbf{P_{0,1,2,3}} & \mathbf{P_{3,0,1,2}} \\ \mathbf{z_{4\times4}} & \mathbf{z_{4\times4}} & \mathbf{P_{0,1,2,3}} \oplus \mathbf{P_{3,0,1,2}} & \mathbf{P_{3,0,1,2}} \end{pmatrix} \tag{A.5}$$

It is clear that $Q2^{-1} \circ S^{-1}$ can be implemented similar to $Q2 \circ S$.

B Hardware Implementations of 4×4 Binary Matrix

To hardware implementation of binary matrix $\mathbf{P}_{1,2,3,0} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}$ We can implement this matrix directly or recursive. To implement recursive, these relations as below (x_i's are inputs and y_i's are output):

$$\begin{aligned} y_0 &= x_0 \oplus x_2 \oplus x_3 \\ y_1 &= x_1 \oplus x_2 \oplus y_0 \\ y_2 &= x_2 \oplus x_3 \oplus y_1 \\ y_3 &= x_3 \oplus y_0 \oplus y_1 \end{aligned} \tag{B.1}$$

If we want to decrease 8 XOR to 7 XOR, we need to define a temporary variable:

$$\begin{aligned} temp &= x_0 \oplus x_1 \oplus x_2 \oplus x_3 \\ y_0 &= x_0 \oplus temp \\ y_1 &= x_1 \oplus temp \\ y_2 &= x_2 \oplus temp \\ y_3 &= x_3 \oplus temp \end{aligned} \tag{B.2}$$

Mahdi Sajadieh received the B.Sc., M.Sc. and Phd degrees in Communication Engineering from Isfahan University of Technology (IUT), Isfahan, Iran, in 2004, 2007 and 2012, respectively. He joined Islamic Azad University, Isfahan (Khorasgan) Branch in 2011 and at present time is an Assistant Professor in Electrical Engineering Department. His research interests include Cryptography and Channel Coding.

Arash Mirzaei received his B.Sc., M.Sc. degrees in Communication Engineering from Isfahan University of Technology, Isfahan, Iran, in 2007 and 2009, respectively. He then worked in the cybersecurity field for about ten years before starting his Ph.D. at Monash University, Melbourne, Australia in 2019. His research interests include Cryptography and Security.