The ISC Int'l Journal of Information Security

January 2023, Volume 15, Number 1 (pp. 111–124)

http://www.isecure-journal.org

$4n \times 4n$ Diffusion Layers Based on Multiple 4×4 MDS Matrices

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

[1] *Department of Electrical Engineering, Khorasgan (Isfahan) Branch, Islamic Azad University, Isfahan, Iran.*

[2] *Faculty of Information Technology, Monash University, Melbourne, Australia.*

ARTICLE INFO.

Article history:
Received: November 20, 2021
Revised: August 12, 2022
Accepted: September 1, 2022
Published Online: September 4, 2022

Keywords:
Active S-box, Block Cipher, Diffusion Layer, MDS Matrix

Type: Research Article

doi: 10.22042/isecure.2022.316014.724

dor: 20.1001.1.20082045.2023.15.1.5.2

ABSTRACT

In terms of security, MDS matrices are one of the best choices for the diffusion layer of block ciphers. However, as these matrices grow in size, their software implementation becomes a challenge. In this paper, to benefit from the properties of MDS matrices and avoid the mentioned challenge, we use 4×4 MDS matrices to build some 16×16 matrices with a low number of zero elements. We show that if these matrices are used as diffusion layers of software-based SPN structures, the resulting block ciphers have similar properties as AES in software implementation complexity (i.e. the number of required CPU instructions) and resistance against linear and differential attacks. Moreover, the best impossible differential and square distinguishers for the proposed 16×16 structures have a similar length as SPN structures with 16×16 MDS matrices. Thus, the new structures outperform AES concerning the impossible differential and square attacks. Additionally, we show that if the proposed SPN structure uses the AES key schedule, its results for the differential related-key attacks are better than those for AES. We also extend the idea and use 4×4 MDS matrices to design 24×24 and 32×32 matrices with acceptable properties for SPN structure design. Finally, we extend the idea to propose some matrices for Feistel structures with SP-type F-functions. We show that the resulting structures are more secure than the improved Type-II GFS.

1 Introduction

Block cipher is a type of symmetric key encryption algorithm whose goal is to provide data confidentiality using a secret shared between communicating parties. Block ciphers are designed based on the principles of confusion and diffusion, introduced by Shannon [1]. It means that the block cipher is divided into some rounds where each round consists of some confusion and diffusion layers (transformations). The confusion layer of a block cipher is usually provided by n small-sized S-boxes. The diffusion layer is deployed using some linear transformations that mix the outputs of the S-boxes. The diffusion layer guarantees that after a few rounds, each output bit is dependent on all input bits and the confusion layer ensures that this dependency is highly non-linear.

There are two primary approaches to designing a block cipher depending on the implementation environment which the cipher is optimized for: (1) Conventional ciphers are optimized for desktop and server environments with no strict resource constraints, and (2) Lightweight ciphers which perform remarkably better in constrained environments, i.e. hardware and

* Corresponding author.
Email addresses: m.sajadieh@khuisf.ac.ir, arash.mirzaei@monash.edu

ISSN: 2008-2045

embedded software platforms [2].

The structure of a block cipher can be categorized into two types: Substitution-Permutation Network (SPN) and the Feistel network. Each round of an SPN structure consists of a substitution layer (representing the confusion transformation) followed by a permutation layer (representing the diffusion transformation) where the combination is called Substitution-Permutation (SP) transformation. Rijndael [3], is one of the most important SPN-based block ciphers. A basic Feistel cipher divides input into two sub-blocks, x and y, and in each round of the cipher, the transformation $(x, y) \longrightarrow (f(x) \oplus y, x)$ is performed where the function f is called the round function. A generalized Feistel structure (GFS) divides the input block into more than 2 sub-blocks [4]. The most popular form of GFS is called Type-II [5] where the output of a single round of Type-II GFS for the input $(m_0, m_1, \ldots, m_{2k-1})$ is $\pi(m_0, f(m_0) \oplus m_1, \ldots, f(m_{2k-2}) \oplus m_{2k-1})$ and π is a permutation on $2k$ sub-blocks. If

$$\pi(m_0, m_1, \ldots, m_{2k-1}) = (m_{2k-1}, m_0, m_1, \ldots, m_{2k-2})$$

holds, the Type-II GFS is called the standard Type-II GFS. This type of GFS with SP-type round function has the problem of difference cancellation caused by the XOR operation. Two methods have been proposed to resolve this issue: (i) Using multiple MDS matrices as diffusion layer of the round functions [6–8], (ii) Modifying permutations of the Type-II GFS connections [9, 10]. CLEFIA is the most famous example of a Type-II GFS [11] with an SP-type round function.

One of the main goals of a block cipher designer is to maximize the minimum number of linear and differential active S-boxes (MNLAS [1] and MNDLS [2], respectively [12]) for several rounds of the block cipher. The faster the number of active S-boxes increases, the fewer rounds the block cipher requires to resist linear and differential family attacks [13]. Thus, diffusion layer plays an important role in the design of a block cipher.

Due to having the maximum branch number, MDS matrices are appropriate options to be used as diffusion layers of block ciphers to efficiently increase the number of active S-boxes [14]. Since the implementation of large MDS matrices (e.g. of size 16×16, 24×24 and so on) is inefficient, 4×4 MDS matrices are often used to construct practical diffusion layers. For instance, for Rijndael with 128-bit, 192-bit and 256-bit block lengths, four, six and eight 4×4 MDS matrices are used, respectively [3].

For a block cipher, MNDAS and MNLAS of consecutive rounds are directly related to how the cipher is resistant to differential and linear attacks, respectively. For instance, there are at least 25 active S-boxes in 4 rounds of AES which makes it resistant to mentioned attacks. However, resistance against other attacks including impossible differential [15] and square attacks [16] is also of importance. The largest impossible differential distinguisher for AES is 3.5 rounds which results in an attack on 6.5 rounds of this cipher [15]. The corresponding values for square distinguisher are 4 and 6 rounds, respectively [17]. It intuitively seems that the sparseness of a diffusion layer matrix may result in weaknesses of the corresponding cipher in resistance against such attacks because the denser a diffusion matrix is, the faster its byte differences diffuse and hence the shorter its longest distinguishers are [18, 19]. Thus, this paper targets designing some new classes of dense $4n \times 4n$ matrices using 4×4 MDS matrices that achieve:

- acceptable results concerning the security against differential, linear, impossible differential and square attacks,
- high performance in modern general-purpose computers.

1.1 Our Contribution

The contributions of this paper are as follows:

- We introduce some new classes of 16×16 matrices with a low number of zero elements for software-based block ciphers. The SPN structure that uses one of these matrices as its diffusion layer, can be efficiently implemented in software using lookup tables with 4-byte entries and achieves similar performance as AES. Furthermore, it has appropriate properties in terms of resistance against linear, differential, impossible differential and square attacks. In other words, if the AES diffusion layer is replaced with one of the matrices proposed in this paper, the number of active S-boxes for several rounds of the resulting SPN structure would be slightly less than the corresponding values for AES. Nevertheless, there is no impossible differential distinguisher and square distinguisher for more than 2.5 rounds and 3 rounds of the new SPN structure, respectively. The corresponding values for AES are 3.5 rounds and 4.5 rounds, respectively. Moreover, it is shown that if AES key schedule is used in the proposed SPN struc-

[1] Minimum Number of Linear Active S-boxes
[2] Minimum Number of Differential Active S-boxes

ture, the minimum number of active S-boxes in differential-related key attacks on different rounds of this structure is higher than the corresponding values for AES.
- We extend the idea and introduce 24×24 and 32×32 matrices for 192-bit and 256-bit diffusion layers, respectively.
- We extend the idea and design some new matrices for Feistel structures with SP-type F-function. We compare the security of the proposed Feistel structure with that of the existing Feistel structures and show that new structures with 2 sub-blocks have better cryptographic properties than the improved Type-II GFS.

1.2 Related Work

Block cipher design started in the 1970s but has entered a new stage since the AES competition. In this competition, some ciphers (e.g. Rijndael [3]) used MDS matrices as part of their diffusion layers. The paper [20] uses multiple MDS matrices in the Feistel structure and the block cipher CLEFIA [11] is designed based on this idea. Permutations of Type-II GFS with multiple MDS matrices are studied in [21]. The paper [13] suggests the design of perfect diffusion layers based on linear operations. Also, based on this idea, [22] presented the idea of Toeplitz Matrices and [23] presented the idea of lightweight linear diffusion layers from Near-MDS matrices. The necessary and sufficient conditions to construct recursive MDS matrices from non-singular diagonable companion are provided in [24] and an efficient class of lightweight 4×4 MDS matrices is presented in [25]. Recently, [26] investigated the construction of MDS matrices with generalized Feistel structures.

1.3 Preliminaries and Notations

In this paper, SPN structures comprise of $4n$ m-bit S-boxes and hence the state vector $\mathbf{x}$ is represented by a $4n$-tuple vector $[x_0, \ldots, x_{4n-1}]^t$ of m-bit values. The state vector, $\mathbf{x}$ is also represented by a vector, $[\hat{\mathbf{x}}_\mathbf{0}, \ldots, \hat{\mathbf{x}}_{\mathbf{n-1}}]^\mathbf{t}$ where each $\hat{\mathbf{x}}_\mathbf{j}$ is a 4×1 vector of m-bit elements ($\hat{\mathbf{x}}_\mathbf{j} = [x_{4j}, x_{4j+1}, x_{4j+2}, x_{4j+3}]^t$). Each 4×1 vector also represents a word with $4m$-bit length. Then, the left shift and the circular left shift of the vector $\hat{\mathbf{x}}$ by i elements (denoted by $\hat{\mathbf{x}} \ll i$ and $\hat{\mathbf{x}}$ n i, respectively) is equivalent to left shift and circular left shift of the corresponding word by $i \times m$ bits, respectively. Similar notations can be used for the right shift and circular right shift.

In each round of the SPN structure, regardless of the key addition layer, two different transformations are applied to the state. In the first transformation which is the S-box layer, an m-bit substitution is independently applied to each element of the state. Applying the S-box layer on the state vector $\mathbf{x}$ (or similarly on $\hat{\mathbf{x}}_\mathbf{j}$ and x_j, respectively) is denoted by $S(\mathbf{x})$ (or similarly $S(\hat{\mathbf{x}}_\mathbf{j})$ and $S(x_j)$, respectively). In the second transformation which is the diffusion layer, the state vector is multiplied on the left by the $4n \times 4n$ matrix $\mathcal{A}$ over a finite field $GF(2^m)$. Thus, in this paper, the terms diffusion transformation and diffusion matrix are used interchangeably.

The diffusion layer $\mathcal{A}$ is a $4n \times 4n$ matrix that is also represented by an $n \times n$ matrix of 4×4 sub-matrices where each sub-matrix is either an all zeros matrix, denoted by $\mathbf{Z}$, or an MDS matrix (see Property 1). In other words, if the diffusion matrix $\mathcal{A}$ is applied on $\mathbf{x}$, then the output $\mathbf{y}$ ($\mathbf{y} = [\hat{\mathbf{y}}_\mathbf{0}, \ldots, \hat{\mathbf{y}}_\mathbf{n-1}]^\mathbf{t}$) is computed as follows:

$$\begin{pmatrix} \hat{\mathbf{y}}_0 \\ \hat{\mathbf{y}}_1 \\ \vdots \\ \hat{\mathbf{y}}_{n-1} \end{pmatrix} = \begin{pmatrix} \mathbf{A_{0,0}} & \mathbf{A_{0,1}} & \ldots & \mathbf{A_{0,n-1}} \\ \mathbf{A_{1,0}} & \mathbf{A_{1,1}} & \ldots & \mathbf{A_{1,n-1}} \\ \vdots & & & \vdots \\ \mathbf{A_{n-1,0}} & \mathbf{A_{n-1,1}} & \ldots & \mathbf{A_{n-1,n-1}} \end{pmatrix} \cdot \begin{pmatrix} \hat{\mathbf{x}}_0 \\ \hat{\mathbf{x}}_1 \\ \vdots \\ \hat{\mathbf{x}}_{n-1} \end{pmatrix}$$
$$\Rightarrow \hat{\mathbf{y}}_\mathbf{i} = \sum_{\mathbf{j=0}}^{\mathbf{n-1}} \mathbf{A_{i,j}} \cdot \hat{\mathbf{x}}_\mathbf{j} \tag{1}$$

where multiplications are performed over a finite field.

Definition 1. For a 4×1 vector $\hat{\mathbf{x}}$, the VWT (Vector-Wise Truncated) value x^C is a value from 0 to 4 representing the number of non-zero elements of $\hat{\mathbf{x}}$.

Definition 2. For a $4n \times 1$ vector $\mathbf{x}$ of m-bit elements, the VWT vector $\mathbf{x^C}$ is a vector of length n, where the j^{th} element of $\mathbf{x^C}$ (i.e. x_j^C) takes a value from 0 to 4 representing the VWT value of the vector $\hat{\mathbf{x}}_\mathbf{j}$ $= [x_{4j}, x_{4j+1}, x_{4j+2}, x_{4j+3}]^t$.

To distinguish the VWT vector of different states of a cipher from each other, the round number is used as the subscript, e.g. the VWT vector for round R is denoted by $\mathbf{x_R^C}$. Table 1 summarizes the mentioned notations.

Property 1. For two 4×1 vectors $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, if $\hat{\mathbf{y}} = \mathbf{A_{i,j}} \cdot \hat{\mathbf{x}}$ and $\mathbf{A_{i,j}}$ is a 4×4 MDS matrix, the following relation holds between x^C and y^C (VWT value of $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, respectively):

$$y^C = \begin{cases} 0 & for\ x^C = 0 \\ q,\ \ 5 - x^C \leq q \leq 4 & for\ x^C > 0 \end{cases} \tag{2}$$

The differentially or linearly active S-box as well as the branch number will be defined, hereinafter.

Definition 3. An S-box is differentially or linearly active if its input difference or output mask value is

Table 1. Notations

Notation	Description
$\hat{\mathbf{x}}$	A 4×1 vector of m-bit elements
$\hat{\mathbf{x}} \ll i$	The left shift of the vector $\hat{\mathbf{x}}$ by i elements
$\hat{\mathbf{x}}$ n i	The circular left shift of the vector $\hat{\mathbf{x}}$ by i elements
x^C	The VWT value of $\hat{\mathbf{x}}$
$\mathbf{x}$	The state vector $[x_0, \ldots, x_{4n-1}]^t$ of m-bit elements
$\hat{\mathbf{x}}_\mathbf{j}$	The 4×1 vector $[x_{4j}, x_{4j+1}, x_{4j+2}, x_{4j+3}]^t$
x_j^C	The VWT value of $\hat{\mathbf{x}}_\mathbf{j}$
$\mathbf{x^C}$	The VWT vector $[x_0^C, x_1^C, \ldots, x_{n-1}^C]^t$
$\mathcal{A}$	A $4n \times 4n$ matrix, representing the diffusion matrix
$\mathbf{Z}$	An all zeros 4×4 matrix
$\mathbf{A}$	A 4×4 MDS matrix

non-zero.

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

$$\mathbf{y}_1 = \mathcal{A} \cdot \mathbf{x}_1,\ \mathbf{y}_2 = \mathcal{A} \cdot \mathbf{x}_2 \Rightarrow \triangle\mathbf{y} = \mathcal{A} \cdot \triangle\mathbf{x} \quad (3)$$

Also, input linear mask (Γ_x) is obtained from output linear mask (Γ_y) by $\Gamma_x = \mathcal{A}^t \cdot \Gamma_y$ where $\mathcal{A}^t$ denotes transpose of matrix $\mathcal{A}$ [27]:

$$\begin{aligned} &\mathbf{y} = \mathcal{A} \cdot \mathbf{x} \Rightarrow \Gamma_y^t \cdot \mathbf{y} = \Gamma_y^t \cdot \mathcal{A} \cdot \mathbf{x} = (\mathcal{A}^t \cdot \Gamma_y)^t \cdot \mathbf{x} \\ &\Rightarrow \Gamma_x = \mathcal{A}^t \cdot \Gamma_y \end{aligned} \quad (4)$$

Definition 4. The branch number of a linear mapping D is given by:

$$\beta_d(D) = \min_{\mathbf{x} \neq \mathbf{0}} \{w(\mathbf{x}) + w(D(\mathbf{x}))\} \quad (5)$$

where $w(\mathbf{x})$ is the number of non-zero elements in the vector $\mathbf{x}$ [28].

1.4 Outline of Paper

The rest of this paper is organized as follows. Section 2 introduces a new family of 16×16 matrices based on multiple 4×4 matrices where the proposed matrices are modified in Section 2.1. Section 2.2 discusses properties of new matrices regarding the related key attacks. In Section 2.3, some new conditions are imposed on the introduced 4×4 matrices to enhance the software implementation properties of the new matrices. In Section 2.4, the idea is extended to design 24×24 and 32×32 matrices. Section 2.5 discusses implementation properties of the inverse of the new matrices. In Section 3, the introduced idea is used to design some new diffusion matrices for Feistel structures with SP-type F-function. Finally, the conclusion is represented in Section 4.

2 New 16×16 Matrices Based on 4×4 MDS Matrices

A 16×16 MDS diffusion layer outperforms the AES diffusion layer with respect to resistance against linear, differential, impossible differential, and square attacks. If S-boxes are designed to be injective and near-ideal (for example by setting $S(x) = (Ax + B)^{-1}\ mod\ p(x)$ where $p(x)$ is a primitive polynomial in $GF(2^n)$) and the length of entries in the MDS matrix equals the S-box length, the largest impossible differential and square distinguisher for an SPN structure with a 16×16 MDS diffusion matrix would be 2.5 and 3.5 rounds, respectively. This structure would also have at least 17 active S-boxes in every 2 consecutive rounds. However, usage of 16×16 MDS matrices is impractical as the implementation of such matrices is inefficient.

Therefore, our goal in this section is to design some 16×16 diffusion matrices which satisfy the following three conditions:

(1) The SPN structure with new matrices must be as resistant against impossible differential and square attacks as MDS matrices.
(2) The SPN structure with the proposed matrices should not be less secure than AES from the aspect of resistance against differential or linear attacks.
(3) Unlike an SPN structure with a 16×16 MDS matrix, software implementation of the proposed structure should be practical.

To find the matrices with the best results regarding impossible differential and square distinguishers, we searched all 16×16 matrices comprising sub-matrices $\mathbf{Z}$ (i.e. an all zeros 4×4 matrix) and $\mathbf{A}$ (i.e. a 4×4 MDS matrix). To find the longest distinguishers, we used the method introduced in [29]. Regardless of the shifted versions, the matrices with the best results are as follows:

$$\mathcal{P}\mathbf{1} = \begin{pmatrix} \mathbf{Z} & \mathbf{A} & \mathbf{A} & \mathbf{A} \\ \mathbf{A} & \mathbf{Z} & \mathbf{A} & \mathbf{A} \\ \mathbf{A} & \mathbf{A} & \mathbf{Z} & \mathbf{A} \\ \mathbf{A} & \mathbf{A} & \mathbf{A} & \mathbf{Z} \end{pmatrix}, \mathcal{P}\mathbf{2} = \begin{pmatrix} \mathbf{Z} & \mathbf{A} & \mathbf{A} & \mathbf{A} \\ \mathbf{A} & \mathbf{Z} & \mathbf{A} & \mathbf{A} \\ \mathbf{A} & \mathbf{A} & \mathbf{Z} & \mathbf{A} \\ \mathbf{A} & \mathbf{A} & \mathbf{A} & \mathbf{A} \end{pmatrix} \quad (6)$$

Since MNLAS and MNDAS for $\mathcal{P}\mathbf{1}$ are greater than those for $\mathcal{P}\mathbf{2}$, we analyze $\mathcal{P}\mathbf{1}$, hereinafter.

Figure 1 shows the longest impossible and square distinguishers for the SPN structure that uses $\mathcal{P}\mathbf{1}$ as its diffusion layer. In Figure 1, A is an active byte, C

is a constant byte, B is a summation zero byte and U is a non-predicable byte in a square attack. As Figure 1 shows, the length of these distinguishers is the same as those for an SPN structure with a 16×16 MDS matrix. Thus, $\mathcal{P}\mathbf{1}$ meets the first condition of the previously stated conditions. Now we discuss the resistance of the mentioned structure against linear and differential cryptanalysis (corresponding with the second stated condition).

The branch number of $\mathcal{P}\mathbf{1}$ is 7 and the following differential characteristic for this SPN structure shows that the minimum number of differential active S-boxes for every $2r$ round cannot exceed $7r$.

$$\begin{aligned} &\mathbf{x}_R^C = (0,1,1,1) \longmapsto \mathbf{x}_{R+1}^C = (4,0,0,0) \\ &\longmapsto \mathbf{x}_{R+2}^C = (0,1,1,1) \end{aligned} \tag{7}$$

where R in $\mathbf{x}_R^C$ shows the round number.

Therefore, $\mathcal{P}\mathbf{1}$ does not provide satisfactory results regarding MNDAS and MNLAS values. Thus, our aim in Section 2.1 will be to design a new class of matrices for which 1) impossible differential and square distinguishers are similar to those for $\mathcal{P}\mathbf{1}$, and 2) the results for MNDAS and MNLAS are better than those for $\mathcal{P}\mathbf{1}$.

2.1 Modification of the Proposed Matrix

Difference cancellation caused by the XOR operation is the reason for the low number of active S-boxes of $\mathcal{P}\mathbf{1}$. As mentioned in [6], difference cancellations also occur in Feistel structures. The paper [6] uses multiple MDS matrices to increase the number of active S-boxes in Feistel structures. In this section, we extend this idea to resolve the issue of difference cancellations in SPN structures. It means that instead of using one matrix $\mathbf{A}$, 4 distinct MDS matrices $\mathbf{A_0}$ to $\mathbf{A_3}$ are used where the branch number of the 4×16 matrices $[\mathbf{A_0}\ \mathbf{A_1}\ \mathbf{A_2}\ \mathbf{A_3}]$ and $[(\mathbf{A_0^t})^{-1}\ (\mathbf{A_1^t})^{-1}\ (\mathbf{A_2^t})^{-1}\ (\mathbf{A_3^t})^{-1}]$ is 5. To meet the third condition of the condition set, stated in the previous section, each column of the new matrix must not have more than one unique non-zero 4×4 matrix. Thus, for a 16×16 matrix, four distinct 4×4 matrices suffice. The modified version of $\mathcal{P}\mathbf{1}$ is denoted by $\mathcal{Q}\mathbf{1}$ which is as follows:

$$\mathcal{Q}\mathbf{1} = \begin{pmatrix} \mathbf{Z} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} \\ \mathbf{A_0} & \mathbf{Z} & \mathbf{A_2} & \mathbf{A_3} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{Z} & \mathbf{A_3} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{Z} \end{pmatrix} \tag{8}$$

It is to be noted that the impossible differential distinguisher and square distinguisher of $\mathcal{P}\mathbf{1}$ and $\mathcal{Q}\mathbf{1}$ are the same. Since the utilized non-zero 4×4 matrices are MDS, the methods presented in [12, 30] can be utilized to count the number of active S-boxes. In the following, some properties of the matrix $\mathcal{Q}\mathbf{1}$ and matrices $\mathbf{A_i}$ are stated. These properties are used in the counting method.

Property 2. If two vectors $\mathbf{x}$ and $\mathbf{y}$ are XORed, and the resulting vector is shown by $\mathbf{t}$, we have:

$$\mathbf{x} \oplus \mathbf{y} = \mathbf{t} \Longrightarrow |x^C - y^C| \leq t^C \leq x^C + y^C \tag{9}$$

Property 3. If multiple vectors with VWT values of x_i^C $(0 \leq i \leq n-1)$ are XORed, i.e. $\mathbf{t} = \oplus_{i=0}^{n-1} \mathbf{x_i}$, then we have:

$$x_l^C - \sum_{j=0, j\neq l}^{n-1} x_j^C \leq t^C \leq \sum_{j=0}^{n-1} x_j^C \tag{10}$$

where x_l^C is the maximum value among x_i^C.

Another important property, resulting from the fact that the branch number of $[\mathbf{A_0}\ \mathbf{A_1}\ \mathbf{A_2}\ \mathbf{A_3}]$ is 5, is as follows:

Property 4. Based on relation (1), assume that $\hat{\mathbf{y}}_i = \oplus_{j=0}^{n-1} \mathbf{A_j} \cdot \hat{\mathbf{x}}_\mathbf{j}$ and the branch number of $\left(\mathbf{A_0}\ \mathbf{A_1}\ \cdots\ \mathbf{A_{n-1}}\right)$ is 5, then we have the following inequality:

$$\begin{aligned} &\hat{\mathbf{y}}_i = \left(\mathbf{A_0}\ \mathbf{A_1}\ \cdots\ \mathbf{A_{n-1}}\right) \cdot \begin{pmatrix} \hat{\mathbf{x}}_0 \\ \hat{\mathbf{x}}_1 \\ \vdots \\ \hat{\mathbf{x}}_{n-1} \end{pmatrix} \\ &\Longrightarrow y_i^C + \textstyle\sum_{j=0}^{n-1} x_j^C \geq 5 \quad if \quad \sum_{j=0}^{n-1} x_j^C \geq 1 \end{aligned} \tag{11}$$

Example 2.1. Assume that $\hat{\mathbf{y}}_\mathbf{0} = \mathbf{A_1} \cdot \hat{\mathbf{x}}_\mathbf{1} \oplus \mathbf{A_2} \cdot \hat{\mathbf{x}}_\mathbf{2} \oplus \mathbf{A_3} \cdot \hat{\mathbf{x}}_\mathbf{3}$ where $x_1^C = 1$, $x_2^C = 1$ and $x_3^C = 2$. Due to properties of MDS matrices, $(A_1 \cdot \hat{x}_1)^C$ and $(A_2 \cdot \hat{x}_2)^C$ equal 4 and $(A_3 \cdot \hat{x}_3)^C$ equals 3 or 4. Although Property 3 implies that we have $y_0^C \leq 4$ and $y_0^C \geq 0$, based on Property 4 $y_0^C + 1 + 1 + 2 \geq 5$ holds. Thus, the lower bound of y_0^C is 1.

A combination of outputs can also impose some new conditions, one of which is shown in 2.2.

Example 2.2. Assume that

$$\begin{aligned} \hat{\mathbf{y}}_\mathbf{2} &= \mathbf{A_0} \cdot \hat{\mathbf{x}}_\mathbf{0} \oplus \mathbf{A_1} \cdot \hat{\mathbf{x}}_\mathbf{1} \oplus \mathbf{A_3} \cdot \hat{\mathbf{x}}_\mathbf{3} \\ \hat{\mathbf{y}}_\mathbf{3} &= \mathbf{A_0} \cdot \hat{\mathbf{x}}_\mathbf{0} \oplus \mathbf{A_1} \cdot \hat{\mathbf{x}}_\mathbf{1} \oplus \mathbf{A_2} \cdot \hat{\mathbf{x}}_\mathbf{2} \end{aligned} \tag{12}$$

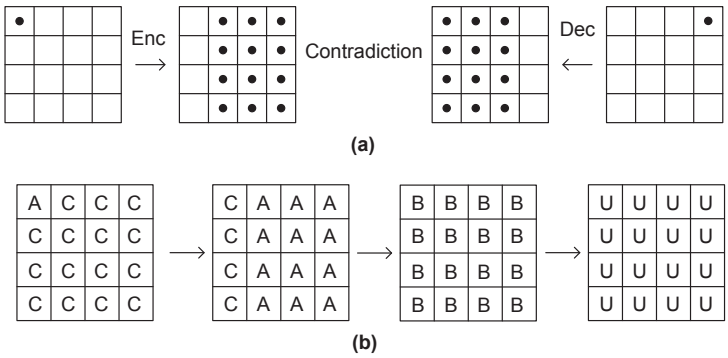

Figure 1. Impossible differential (a) and square distinguisher (b) of SPN structure with $\mathcal{P}\mathbf{1}$

Table 2. MNDAS of r rounds of the SPN structure with $\mathcal{Q}\mathbf{1}$

r	1	2	3	5	8	10	12	14	20	24
MNDAS	1	7	13	24	40	51	62	73	106	128

It holds $\hat{\mathbf{y}}_\mathbf{2} \oplus \hat{\mathbf{y}}_\mathbf{3} = \mathbf{A_2} \cdots \hat{\mathbf{x}}_\mathbf{2} \oplus \mathbf{A_3} \cdot \hat{\mathbf{x}}_\mathbf{3}$ and using property 4, we have

$$\begin{cases} y_2^C + y_3^C + x_2^C + x_3^C \geq 5 \;\; if \;\; x_2^C + x_3^C \geq 1 \\ y_2^C = y_3^C \qquad\qquad\qquad\quad if \;\; x_2^C = x_3^C = 0 \end{cases} \tag{13}$$

and hence, if $y_2^C = x_2^C = x_3^C = 1$, then $y_3^C \geq 2$.

To count the number of active S-boxes, the method mentioned in [12] was used. This method is briefly explained in Appendix A. Table 2 displays the number of active S-boxes for consecutive rounds of an SPN structure with $\mathcal{Q}\mathbf{1}$ as its diffusion layer.

Although the branch number of the matrix $\mathcal{Q}\mathbf{1}$ is 7, the number of differential active S-boxes of $2r$ rounds, is more than $7r$ for $r > 1$ and this number closes to the corresponding values for AES as the number of rounds increases. To compute the number of linear active S-boxes, the counting method must be applied on $(\mathcal{Q}\mathbf{1^t})^{-1}$ which has the below form:

$$(\mathcal{Q}\mathbf{1^t})^{-1} = \begin{pmatrix} \mathbf{Z} & (\mathbf{A_1^t})^{-1} & (\mathbf{A_2^t})^{-1} & (\mathbf{A_3^t})^{-1} \\ (\mathbf{A_0^t})^{-1} & \mathbf{Z} & (\mathbf{A_2^t})^{-1} & (\mathbf{A_3^t})^{-1} \\ (\mathbf{A_0^t})^{-1} & (\mathbf{A_1^t})^{-1} & \mathbf{Z} & (\mathbf{A_3^t})^{-1} \\ (\mathbf{A_0^t})^{-1} & (\mathbf{A_1^t})^{-1} & (\mathbf{A_2^t})^{-1} & \mathbf{Z} \end{pmatrix} \tag{14}$$

According to the similarity of the structure of the matrix $(\mathcal{Q}\mathbf{1^t})^{-1}$ with that of matrix $\mathcal{Q}\mathbf{1}$, if the branch number of $[(\mathbf{A_0^t})^{-1}(\mathbf{A_1^t})^{-1}(\mathbf{A_2^t})^{-1}(\mathbf{A_3^t})^{-1}]$ is 5, then the minimum number of linear active S-boxes for consecutive rounds is also as Table 2 shows.

Table 3. Comparison between AES structure and SPN structure with $\mathcal{P}\mathbf{1}$ and $\mathcal{Q}\mathbf{1}$

Structure	Lin. & Diff. Dist. (Attack)	Imp. Diff. Dist. (Attack)	square Dist. (Attack)
AES	3 (4) [3]	3.5 (6.5) [15]	4.5 (6) [3, 16]
SPN with $\mathcal{P}\mathbf{1}$	5 (6)	2.5 (4.5)	3.5 (5)
SPN with $\mathcal{Q}\mathbf{1}$	4 (5)	2.5 (4.5)	3.5 (5)

An interesting point, observed during the computation of the minimum number of active S-boxes for the matrix $\mathcal{Q}\mathbf{1}$, is that if the branch number of all 4×8 matrices $[\mathbf{Ai\ Aj}]$ $(0 \leq i, j \leq 4 \; and \; i \neq j)$ is 5, the values in Table 2 do not change.

Table 3 compares the SPN structure that uses $\mathcal{Q}\mathbf{1}$ ($\mathcal{P}\mathbf{1}$) as its diffusion layer with AES. The values in parentheses show the largest number of rounds for which the complexity of a given attack is less than the exhaustive search. Table 3 shows that 7 rounds of AES and 6 rounds of the proposed structure are resistant to mentioned attacks, respectively.

Also, using the relation between plaintext and subkeys of the first round, the length of the impossible differential distinguisher for AES can be increased by 1 round [31]. The SPN structures with $\mathcal{P}\mathbf{1}$ or $\mathcal{Q}\mathbf{1}$ have a more complicated form than AES because each byte in $\mathcal{P}\mathbf{1}$ or $\mathcal{Q}\mathbf{1}$ affects 12 bytes in the next round but each byte in AES affects 4 bytes. Thus, it seems that the proposed structure is more secure than AES regarding this attack. We can state the same argument for the integral attack where the length of the square distinguisher for AES can increase by one round.

Implementation of 10 rounds of the SPN structure with $\mathcal{Q}\mathbf{1}$ and full-round AES (10-round AES) on a Pentium II PC with 3.4 GHz CPU shows that the former is 8% slower.

Table 4. MNDAS for related key attack on r rounds of AES and SPN structure with $\mathcal{Q}\mathbf{1}$

r	4	5	6	7	8	9	10
AES	9	11	13	15	21	23	25
SPN structure with $\mathcal{Q}\mathbf{1}$	10	12	16	18	22	24	27

2.2 Related Key Analysis

Resistance of a block cipher against related-key attacks cannot be analyzed without considering its key-schedule details. However, the focus of this paper is on the diffusion layer design and hence no new key-scheduling is proposed. Nevertheless, according to the importance of related-key attacks, it must be shown that the proposed matrices do not have any particular weaknesses that make them vulnerable to related-key cryptanalysis. To show this, we use a key-schedule algorithm with the new diffusion layer to build a complete structure and then analyze it from the aspect of related-key attacks. Since all the cryptographic properties of new matrices were compared with those for Rijndael, Rijndael's key schedule is selected to be used with new diffusion layers. The Figure 2 and Figure 3 illustrate the best truncated differential characteristics for the resulting SPN structure for r=5 and r=8 rounds, respectively. Table 4 shows the minimum number of active S-boxes in the best truncated differential characteristics for the resulting SPN structure for a different number of rounds. The corresponding values for AES are also shown in this table [32].

As Table 4 shows, the results for the new structure are slightly better than those for AES and it provides evidence that the new proposed non-sparse diffusion layer does not cause any particular weaknesses regarding related-key attacks. According to this evidence and since the subject of this paper is not designing key-schedule algorithms, we will not investigate the related-key attack against other diffusion layers that will be proposed in the next sections.

2.3 Criteria for Selection of Matrices $\mathbf{A_i}$s

If $\mathcal{Q}\mathbf{1}$ is used as the diffusion layer of an SPN structure, then the lookup table implementation of each round of the structure requires only 6 XORs and 6 temporary variables for more than one AES round. Each round is represented as follows:

$$\begin{pmatrix} \hat{\mathbf{y}}_0 \\ \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \hat{\mathbf{y}}_3 \end{pmatrix} = \mathcal{Q}\mathbf{1} \cdot \begin{pmatrix} S(\hat{\mathbf{x}}_0) \\ S(\hat{\mathbf{x}}_1) \\ S(\hat{\mathbf{x}}_2) \\ S(\hat{\mathbf{x}}_3) \end{pmatrix} \tag{15}$$

Implementation of $\mathcal{Q}\mathbf{1}$ (and $\mathcal{P}\mathbf{1}$) with lookup table have two phases. In the first phase, 6 temporary variables are computed:

$$\begin{aligned} \hat{\mathbf{u}}_i &= \mathbf{A_i} \cdot \mathbf{S}(\hat{\mathbf{x}}_\mathbf{i}), \mathbf{i} = \{\mathbf{0}, \ldots, \mathbf{3}\}, \\ \hat{\mathbf{v}}_4 &= \hat{\mathbf{u}}_0 \oplus \hat{\mathbf{u}}_1, \\ \hat{\mathbf{v}}_5 &= \hat{\mathbf{u}}_2 \oplus \hat{\mathbf{u}}_3 \end{aligned} \tag{16}$$

Then, $\hat{\mathbf{y}}_i$s are computed as below:

$$\begin{aligned} \hat{\mathbf{y}}_0 &= \hat{\mathbf{v}}_5 \oplus \hat{\mathbf{u}}_1 \\ \hat{\mathbf{y}}_1 &= \hat{\mathbf{v}}_5 \oplus \hat{\mathbf{u}}_0 \\ \hat{\mathbf{y}}_2 &= \hat{\mathbf{v}}_4 \oplus \hat{\mathbf{u}}_3 \\ \hat{\mathbf{y}}_3 &= \hat{\mathbf{v}}_4 \oplus \hat{\mathbf{u}}_2 \end{aligned} \tag{17}$$

There are many methods that can be adopted in order to design the matrices $\mathbf{A_0}$ to $\mathbf{A_3}$, one of which is selecting four 4×4 matrices from a 16×16 MDS matrix. Generally, software implementation of the proposed diffusion layer with 4 distinct $\mathbf{A_i}$ and 16 lookup tables (corresponding to $\mathbf{A_0}$ to $\mathbf{A_3}$) requires 16 table lookups and 20 XOR operations. To decrease the number of tables, a 4×4 matrix is selected as $\mathbf{A_0}$. Then, $\mathbf{A_1}$, $\mathbf{A_2}$ and $\mathbf{A_3}$ are obtained by shifting the rows of $\mathbf{A_0}$. So, instead of 16 lookup tables for $\mathbf{A_0}$, $\mathbf{A_1}$, $\mathbf{A_2}$ and $\mathbf{A_3}$, only 4 lookup tables will be required. One of the sets (with low weight) with the mentioned property in $GF(2^8)$ with primitive polynomial $x^8 + x^4 + x^3 + x^2 + 1$ is:

$$\begin{aligned} \mathbf{A_0} &= \begin{pmatrix} 8 & 3 & 8 & 14 \\ 9 & 13 & 1 & 4 \\ 3 & 12 & 11 & 3 \\ 10 & 7 & 14 & 2 \end{pmatrix} \mathbf{A_1} = \begin{pmatrix} 9 & 13 & 1 & 4 \\ 3 & 12 & 11 & 3 \\ 10 & 7 & 14 & 2 \\ 8 & 3 & 8 & 14 \end{pmatrix} \\ \mathbf{A_2} &= \begin{pmatrix} 3 & 12 & 11 & 3 \\ 10 & 7 & 14 & 2 \\ 8 & 3 & 8 & 14 \\ 9 & 13 & 1 & 4 \end{pmatrix}, \mathbf{A_3} = \begin{pmatrix} 10 & 7 & 14 & 2 \\ 8 & 3 & 8 & 14 \\ 9 & 13 & 1 & 4 \\ 3 & 12 & 11 & 3 \end{pmatrix} \end{aligned} \tag{18}$$

One can verify that if $\hat{\mathbf{y}}_\mathbf{0} = \mathbf{A_0} \cdot \hat{\mathbf{x}}$, we have $\hat{\mathbf{y}}_\mathbf{1} = \mathbf{A_1} \cdot \hat{\mathbf{x}} = \hat{\mathbf{y}}_\mathbf{0}$ n 1, $\hat{\mathbf{y}}_\mathbf{2} = \mathbf{A_2} \cdot \hat{\mathbf{x}} = \hat{\mathbf{y}}_\mathbf{0}$ n 2 and $\hat{\mathbf{y}}_\mathbf{3} = \mathbf{A_3} \cdot \hat{\mathbf{x}} = \hat{\mathbf{y}}_\mathbf{0}$ n 3. It is noteworthy that there are some issues regarding implementing the inverse of the matrix $\mathcal{Q}\mathbf{1}$ using the mentioned $\mathbf{A_i}$ matrices. Thus, if the inverse

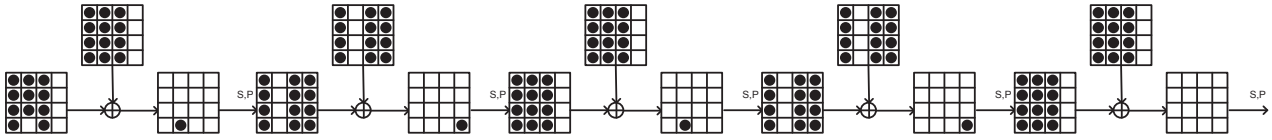

Figure 2. The best truncated differential characteristics for 5 rounds of resulting SPN structure with $\mathcal{Q}\mathbf{1}$

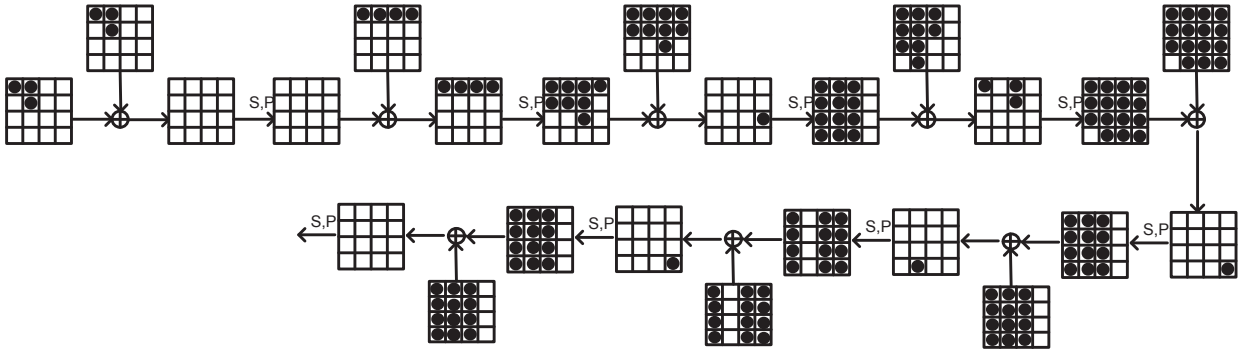

Figure 3. The best truncated differential characteristics for 8 rounds of resulting SPN structure with $\mathcal{Q}\mathbf{1}$

of the matrix is required to be implemented (e.g. when the used mode of operation is ECB [3] or CBC [4]) the matrices $\mathbf{A_i}$s must be selected using a different method. More details will be stated in Section 2.5.

2.4 24×24 and 32×32 Matrices Based on Multiple 4×4 MDS Matrices

In this section, some 24×24 and 32×32 matrices are proposed to be used as diffusion layers of SPN structures with block lengths of 192 bits and 256 bits, respectively. We searched among all the possible 24×24 matrices comprising sub-matrices $\mathbf{Z}$ and $\mathbf{A}$ to find the ones with the best results concerning the impossible differential and square attacks. To find the longest distinguishers, the method introduced in [29] was used. One of the best found matrices is as follows ($\mathbf{A}$ in the i^{th} column of the found matrix has been replaced by $\mathbf{A_i}$):

$$\mathcal{Q}\mathbf{2} = \begin{pmatrix} \mathbf{Z} & \mathbf{A_1} & \mathbf{A2} & \mathbf{Z} & \mathbf{A_4} & \mathbf{A_5} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} & \mathbf{Z} & \mathbf{A_5} \\ \mathbf{Z} & \mathbf{A_1} & \mathbf{Z} & \mathbf{A_3} & \mathbf{A_4} & \mathbf{A_5} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} & \mathbf{A_4} & \mathbf{Z} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{Z} & \mathbf{A_3} & \mathbf{Z} & \mathbf{A_5} \\ \mathbf{A_1} & \mathbf{Z} & \mathbf{A_2} & \mathbf{A_3} & \mathbf{A_4} & \mathbf{A_5} \end{pmatrix} \quad (19)$$

[3] Electronic Codebook
[4] Cipher Block Chaining

The Mixed-integer Linear Programming (MILP) method, mentioned in [33], is used to count MNDAS for the SPN structure with $\mathcal{Q}\mathbf{2}$. However, since the number of inequalities is very large, the number of active S-boxes can be computed for a limited number of rounds (i.e. at most 12 rounds).

MNDAS for 12 rounds of $\mathcal{Q}\mathbf{2}$ is 74 and the best impossible differential (square) distinguisher for an SPN structure using $\mathcal{Q}\mathbf{2}$ is 2.5 (3.5) rounds which are one round shorter than the best distinguisher of Rijndale-192.

An exhaustive search for 32×32 matrices is infeasible. Thus, some of such matrices were searched among which the following matrix has the best results with respect to impossible differential, square, differential and linear distinguishers.

$$\mathcal{Q}\mathbf{3} = \begin{pmatrix} \mathbf{Z} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} & \mathbf{A_4} & \mathbf{Z} & \mathbf{Z} & \mathbf{A_7} \\ \mathbf{A_0} & \mathbf{Z} & \mathbf{A_2} & \mathbf{A_3} & \mathbf{A_4} & \mathbf{A_5} & \mathbf{Z} & \mathbf{Z} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{Z} & \mathbf{A_3} & \mathbf{Z} & \mathbf{A_5} & \mathbf{A_6} & \mathbf{Z} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{Z} & \mathbf{Z} & \mathbf{Z} & \mathbf{A_6} & \mathbf{A_7} \\ \mathbf{Z} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} & \mathbf{A_4} & \mathbf{A_5} & \mathbf{A_6} & \mathbf{Z} \\ \mathbf{A_0} & \mathbf{Z} & \mathbf{A_2} & \mathbf{A_3} & \mathbf{Z} & \mathbf{A_5} & \mathbf{A_6} & \mathbf{A_7} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{Z} & \mathbf{A_3} & \mathbf{A_4} & \mathbf{Z} & \mathbf{A_6} & \mathbf{A_7} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{Z} & \mathbf{A_4} & \mathbf{A_5} & \mathbf{Z} & \mathbf{A_7} \end{pmatrix} \quad (20)$$

To count MNDAS for 32×32 matrices, MILP method was used [33]. However, since the number

Table 5. MNDAS of r rounds of the SPN structure with $\mathcal{Q}_2$ and $\mathcal{Q}_3$ and Rijndael family

r	1	2	3	5	8	10	12	14	20	24
SPN structure with $\mathcal{Q}2$	1	8	13	29	50	64	78	?	?	?
Rijndael-192 [12]	1	5	9	34	50	72	87	103	150	180
SPN structure with $\mathcal{Q}3$	1	9	18	36	60	?	?	?	?	?
Rijndael-256 [12]	1	5	9	41	65	85	105	120	175	210

Table 6. Comparison between Rijndael structure and SPN structure with $\mathcal{Q}2$ and $\mathcal{Q}3$

Structure	Lin. & Diff. Dist. (Attack)	Imp. Diff. Dist. (Attack)	Square Dist. (attack)
Rijndael-192	5 (6)	4.5 (7.5)	5 (7)
SPN with $\mathcal{Q}2$	6 (7)	2.5 (4.5)	3 (5)
Rijndael-256	6 (7)	4.5 (7.5)	5 (7)
SPN with $\mathcal{Q}3$	6 (7)	2.5 (4.5)	3 (5)

of inequalities is very large, the number of active S-boxes can be computed at most for 8 rounds.

MNDAS for 8 rounds of $\mathcal{Q}3$ is 60 and the best impossible differential (square) distinguisher for an SPN structure using $\mathcal{Q}3$ is 2.5 (3.5) rounds which is two rounds shorter than the best distinguisher of Rijndale-256.

The best impossible differential distinguishers for SPN structure with matrices $\mathcal{Q}2$ and $\mathcal{Q}3$ are as Figure 4 and Figure 5 show, respectively. To find the best distinguishers, we used the method mentioned in [19].

Table 5 compares the MNDAS values for SPN structures that use $\mathcal{Q}_2$ and $\mathcal{Q}_3$ with the corresponding values for Rijndael-192 and Rijndael-256. Table 6 compares the SPN structure that uses $\mathcal{Q}2$ ($\mathcal{Q}3$) as its diffusion layer with Rijndael-192 (Rijndael-256). The values in parentheses show the largest number of rounds for which the complexity of a given attack is less than an exhaustive search.

2.5 Implementation of the Inverse Matrix

Implementation of the matrix $\mathcal{P}\mathbf{1}^{-1}$ is similar to that of $\mathcal{P}\mathbf{1}$ (only $(\mathbf{A})^{-1}$ is replaced by $(\mathbf{A})$). However, the inverse of the matrix $\mathcal{Q}\mathbf{1}$ is as follows:

$$\mathcal{Q}\mathbf{1}^{-1} = \begin{pmatrix} \mathbf{Z} & (\mathbf{A_0})^{-1} & (\mathbf{A_0})^{-1} & (\mathbf{A_0})^{-1} \\ (\mathbf{A_1})^{-1} & \mathbf{Z} & (\mathbf{A_1})^{-1} & (\mathbf{A_1})^{-1} \\ (\mathbf{A_2})^{-1} & (\mathbf{A_2})^{-1} & \mathbf{Z} & (\mathbf{A_2})^{-1} \\ (\mathbf{A_3})^{-1} & (\mathbf{A_3})^{-1} & (\mathbf{A_3})^{-1} & \mathbf{Z} \end{pmatrix} \quad (21)$$

Since, elements of each column of $\mathcal{Q}\mathbf{1}^{-1}$, are different from each other, software implementation of $\mathcal{Q}\mathbf{1}^{-1}$ using the lookup table method is inefficient. However, if $\mathbf{A_0}^{-1}$, $\mathbf{A_1}^{-1}$, $\mathbf{A_2}^{-1}$ and $\mathbf{A_3}^{-1}$ are related to each other, then a better implementation for $\mathcal{Q}\mathbf{1}^{-1}$ is probably obtained. Suppose that $\mathbf{A_0}$, $\mathbf{A_1}$, $\mathbf{A_2}$ and $\mathbf{A_3}$ are selected such that their inverses are as follows. One can verify that $\mathbf{A_0}, \ldots, \mathbf{A_3}$ have the desired properties, stated in Section 2.1.

$$\mathbf{A_0^{-1}} = \begin{pmatrix} 3 & 14 & 21 & 15 \\ 20 & 7 & 9 & 25 \\ 8 & 19 & 4 & 27 \\ 2 & 13 & 7 & 14 \end{pmatrix} \mathbf{A_1^{-1}} = \begin{pmatrix} 3 & 9 & 28 & 22 \\ 20 & 20 & 13 & 2 \\ 8 & 30 & 3 & 21 \\ 2 & 3 & 18 & 1 \end{pmatrix}$$
$$\mathbf{A_2^{-1}} = \begin{pmatrix} 3 & 13 & 7 & 1 \\ 20 & 14 & 21 & 22 \\ 8 & 7 & 9 & 2 \\ 2 & 19 & 4 & 21 \end{pmatrix}, \mathbf{A_3^{-1}} = \begin{pmatrix} 3 & 14 & 9 & 25 \\ 20 & 7 & 17 & 20 \\ 8 & 19 & 14 & 23 \\ 2 & 13 & 4 & 27 \end{pmatrix} \quad (22)$$

To implement $\mathcal{Q}\mathbf{1}^{-1} \cdot \mathrm{S}^{-1}(\mathbf{x})$, firstly, the below tables are built:

$$\begin{aligned} &T_{00}(x) = \\ &[3 \cdot S^{-1}(x) \;\; 20 \cdot S^{-1}(x) \;\; 8 \cdot S^{-1}(x) \;\; 2 \cdot S^{-1}(x)]^t \\ &T_{01}(x) = \\ &[14 \cdot S^{-1}(x) \;\; 7 \cdot S^{-1}(x) \;\; 19 \cdot S^{-1}(x) \;\; 13 \cdot S^{-1}(x)]^t \\ &T_{02}(x) = \\ &[21 \cdot S^{-1}(x) \;\; 9 \cdot S^{-1}(x) \;\; 4 \cdot S^{-1}(x) \;\; 7 \cdot S^{-1}(x)]^t \\ &T_{03}(x) = \\ &[15 \cdot S^{-1}(x) \;\; 25 \cdot S^{-1}(x) \;\; 27 \cdot S^{-1}(x) \;\; 14 \cdot S^{-1}(x)]^t \end{aligned} \quad (23)$$

Then, for computation of $\mathbf{A_0^{-1}} \cdot \hat{\mathbf{x}}$, $\mathbf{A_1^{-1}} \cdot \hat{\mathbf{x}}$, $\mathbf{A_2^{-1}} \cdot \hat{\mathbf{x}}$ and $\mathbf{A_3^{-1}} \cdot \hat{\mathbf{x}}$, we have ($\hat{\mathbf{x}} = [\mathbf{x_0}, \mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}]$):

$$\begin{aligned} \hat{\mathbf{u}}_\mathbf{0} &= T_{00}(x_0) \\ \hat{\mathbf{u}}_\mathbf{1} &= T_{01}(x_1) \end{aligned}$$

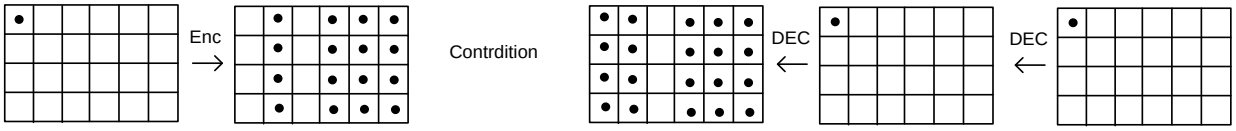

Figure 4. Impossible differential distinguisher of SPN structure with $\mathcal{Q}\mathbf{2}$

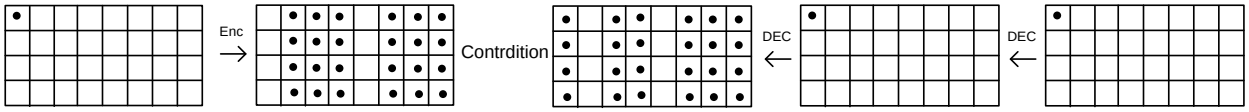

Figure 5. Impossible differential distinguisher of SPN structure with $\mathcal{Q}\mathbf{3}$

$$
\begin{aligned}
\hat{\mathbf{u}}_\mathbf{2} &= T_{02}(x_2)\\
\hat{\mathbf{u}}_\mathbf{3} &= T_{03}(x_3)\\
\hat{\mathbf{t}}_\mathbf{1} &= \hat{\mathbf{u}}_\mathbf{2} + \hat{\mathbf{u}}_\mathbf{3}\\
\hat{\mathbf{t}}_\mathbf{2} &= \hat{\mathbf{u}}_\mathbf{1} + \hat{\mathbf{t}}_\mathbf{1}\\
\hat{\mathbf{w}}_\mathbf{0} = \mathbf{A_0^{-1}} \cdot \hat{\mathbf{x}} &= \hat{\mathbf{u}}_\mathbf{0} + \hat{\mathbf{t}}_\mathbf{2}\\
\hat{\mathbf{w}}_\mathbf{1} = \mathbf{A_1^{-1}} \cdot \hat{\mathbf{x}} &= (\mathbf{t}_2 \text{ n } 1) + \hat{\mathbf{w}}_\mathbf{0}\\
\hat{\mathbf{w}}_\mathbf{2} = \mathbf{A_2^{-1}} \cdot \hat{\mathbf{x}} &= \hat{\mathbf{u}}_\mathbf{0} + (\hat{\mathbf{t}}_\mathbf{2} \text{ n } \mathbf{3}) + \hat{\mathbf{u}}_\mathbf{3}\\
\hat{\mathbf{w}}_\mathbf{3} = \mathbf{A_3^{-1}} \cdot \hat{\mathbf{x}} &= \hat{\mathbf{u}}_\mathbf{0} + \hat{\mathbf{u}}_\mathbf{1} + (\hat{\mathbf{t}}_\mathbf{1} \gg \mathbf{1}) + (\hat{\mathbf{t}}_\mathbf{1} \ll \mathbf{1})
\end{aligned}
\tag{24}
$$

The most important property of the mentioned matrices is that the branch number for the matrices $[\mathbf{A_i A_j}]$ and $[(\mathbf{A_i^t})^{-1}(\mathbf{A_j^t})^{-1}]$ for $i \neq j$ is 5. For an SPN structure with 8-bit S-boxes and a diffusion matrix of $\mathcal{Q}\mathbf{1}$, the encryption speed is about 35% more than the decryption speed in software implementation. Inverse of $\mathcal{Q}\mathbf{2}$ and $\mathcal{Q}\mathbf{3}$ can also be implemented similar to the inverse of $\mathcal{Q}1$.

It is worth mentioning that if the mentioned structures are used in stream modes of operation such as CTR [5] or OFB [6] or if they are used to design hash functions or MAC [7] functions, implementation of inverse matrices is not required. In the next section, the proposed idea is utilized to design the diffusion layer of a Feistel structure with an SP-type F-function.

3 The Proposed Matrices in a Feistel Structure

As stated before, the proposed SP transformations have acceptable software implementation and appropriate security properties. However, due to their inverse properties, their usage in SPN structures may be inefficient in some applications. Therefore, since Feistel structures have the benefit that their encryption and decryption operations are very similar, even the same in some cases, the introduced diffusion layer can be utilized in Feistel structures. To have a 64-bit SP transformation as a round function F with 8-bit S-boxes, 8×8 matrices must be utilized. The 8×8 matrices should be invertible. Due to implementation-based reasons, the following form is proposed to be used in the round function.

$$
\mathcal{Q}\mathbf{1_F} = \begin{pmatrix} \mathbf{Z} & \mathbf{A_1} \\ \mathbf{A_0} & \mathbf{A_1} \end{pmatrix} \tag{25}
$$

Based on the MILP method [33], the minimum number of active S-boxes in 24 rounds of the corresponding Feistel structure is computed as 46. Also, using the method introduced in [29], one can verify that the introduced class of matrices results in long impossible differential distinguishers whose reason is a large number of zeros in the diffusion matrix. To solve the problem, the following matrix is proposed where $\tilde{\mathbf{A}}_\mathbf{0}$ and $\tilde{\mathbf{A}}_\mathbf{1}$ can be computed from $\mathbf{A_0}$ and $\mathbf{A_1}$ using lightweight operations (e.g. shift and XOR), respectively.

$$
\tilde{\mathcal{Q}}\mathbf{1_F} = \begin{pmatrix} \mathbf{A_0} & \mathbf{A_1} \\ \tilde{\mathbf{A}}_\mathbf{0} & \tilde{\mathbf{A}}_\mathbf{1} \end{pmatrix} \tag{26}
$$

The number of active S-boxes for 24 rounds of this Feistel structure increases to 54 given that the branch number of the matrices $[\mathbf{A_0}\ \ \mathbf{A_1}]$, $[\tilde{\mathbf{A}}_\mathbf{0}\ \ \tilde{\mathbf{A}}_\mathbf{1}]$ and $\tilde{\mathcal{Q}}\mathbf{1_F}$ is 5, 5 and 6, respectively. Table 7 compares the results for the proposed Feistel structure with similar structures introduced in [7, 34] and the block cipher CLEFIA [11]. It is worth mentioning that implementing an 8×8 MDS matrix (for structure with $\beta = 9$) using 32-bit registers is about two times more complex than that of the proposed matrix. As Table 7 shows, the MNDAS values for CLEFIA are slightly

[5] Counter
[6] Output Feedback
[7] Message Authentication Code

better than the results for the proposed structure. Nevertheless, the length of the best known impossible differential distinguisher for CLEFIA is 9 [35] which is 2 rounds more than that of the proposed structure.

One example for $\tilde{\mathcal{Q}}\mathbf{1_F}$ is as follows:

$$\tilde{\mathcal{Q}}\mathbf{1_F} = \begin{pmatrix} \mathbf{A_0} & \mathbf{A_1} \\ \tilde{\mathbf{A}}_\mathbf{0} & \tilde{\mathbf{A}}_\mathbf{1} \end{pmatrix} \tag{27}$$

where

$$\mathbf{A_0} = \begin{pmatrix} 12 & 13 & 15 & 11 \\ 3 & 9 & 13 & 2 \\ 1 & 1 & 11 & 9 \\ 12 & 5 & 3 & 7 \end{pmatrix} \tilde{\mathbf{A}}_\mathbf{0} = \begin{pmatrix} 15 & 4 & 2 & 9 \\ 2 & 8 & 6 & 11 \\ 13 & 4 & 8 & 14 \\ 12 & 5 & 3 & 7 \end{pmatrix}$$
$$\mathbf{A_1} = \begin{pmatrix} 1 & 1 & 11 & 9 \\ 12 & 5 & 3 & 7 \\ 12 & 13 & 15 & 11 \\ 2 & 8 & 6 & 11 \end{pmatrix} \tilde{\mathbf{A}}_\mathbf{1} = \begin{pmatrix} 1 & 1 & 11 & 9 \\ 12 & 5 & 3 & 7 \\ 12 & 13 & 15 & 11 \\ 3 & 9 & 13 & 2 \end{pmatrix} \tag{28}$$

To implement $\begin{pmatrix} \hat{\mathbf{y}}_\mathbf{0} \\ \hat{\mathbf{y}}_\mathbf{1} \end{pmatrix} = \tilde{\mathcal{Q}}\mathbf{1_F} \cdot \begin{pmatrix} \hat{\mathbf{x}}_\mathbf{0} \\ \hat{\mathbf{x}}_\mathbf{1} \end{pmatrix}$, if $\hat{\mathbf{u}}_\mathbf{0} = \mathbf{A_0} \cdot \hat{\mathbf{x}}_\mathbf{0}$, $\hat{\mathbf{u}}_\mathbf{1} = \tilde{\mathbf{A}}_\mathbf{1} \cdot \hat{\mathbf{x}}_\mathbf{1}$ and $\hat{\mathbf{t}}$ is a temporary variable, then:

$$\begin{aligned} \hat{\mathbf{t}} &= \hat{\mathbf{u}}_\mathbf{0} \oplus \hat{\mathbf{u}}_\mathbf{1} \\ \hat{\mathbf{y}}_\mathbf{0} &= \hat{\mathbf{t}} \oplus \hat{\mathbf{u}}_\mathbf{1} \gg \mathbf{3} \\ \hat{\mathbf{y}}_\mathbf{1} &= \hat{\mathbf{t}} \oplus \hat{\mathbf{u}}_\mathbf{0} \ll \mathbf{1} \end{aligned} \tag{29}$$

For 192-bit Feistel structures, the following 12×12 matrix has been found using an exhaustive search:

$$\mathcal{Q}\mathbf{2_F} = \begin{pmatrix} \mathbf{Z} & \mathbf{A_1} & \mathbf{A_2} \\ \mathbf{A_0} & \mathbf{Z} & \mathbf{A_2} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{A_2} \end{pmatrix} \tag{30}$$

To have maximum MNDAS and MNLAS, the branch number of the matrix $[\mathbf{A_0} \quad \mathbf{A_1} \quad \mathbf{A_2}]$ and $[((\mathbf{A_0})^\mathbf{t})^{-1} \quad ((\mathbf{A_1})^\mathbf{t})^{-1} \quad ((\mathbf{A_1})^\mathbf{t})^{-1}]$ must be 5. MNDAS and MNLAS for the mentioned structure are shown in Table 8 (they are computed using the method introduced in [12]). Also, Table 8 compares the results for the proposed Feistel structure with similar structures of Feistle Type-II in [10].

It must be pointed out that 18 rounds of a standard Type-II GFS and an improved Type-II GFS with 4×4 MDS matrices have 47 and 50 active S-boxes, respectively [10]. Thus, the introduced structure is as secure as the structures mentioned in [10] from the aspect of resistance against linear and differential attacks and more secure than those structures in terms of resistance against impossible differential attacks because the longest impossible differential distinguisher of the Feistel structure with $\mathcal{Q}\mathbf{2_F}$ as diffusion layer of SP function is 8 rounds which are 3 and 5 rounds shorter than that of improved Type-II GFS and standard Type-II GFS, respectively.

Regarding 256-bit Feistel structures, the following 16×16 matrix has the best results from the aspect of MNDAS.

$$\mathcal{Q}\mathbf{3_F} = \begin{pmatrix} \mathbf{Z} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} \\ \mathbf{A_0} & \mathbf{Z} & \mathbf{A_2} & \mathbf{A_3} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{Z} & \mathbf{A_3} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} \end{pmatrix} \tag{31}$$

However, as Table 9 shows, MNLAS values for $\mathcal{Q}\mathbf{3_F}$ are less than the mentioned MNDAS values. For the matrix, $\mathcal{Q}\mathbf{4_F}$ MNDAS and MNLAS values are the same and can be seen in Table 9.

$$\mathcal{Q}\mathbf{4_F} = \begin{pmatrix} \mathbf{Z} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} \\ \mathbf{A_0} & \mathbf{Z} & \mathbf{A_2} & \mathbf{A_3} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{Z} & \mathbf{A_3} \\ \mathbf{A_0} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{Z} \end{pmatrix} \tag{32}$$

Eighteen rounds of a standard Type-II GFS and an improved Type-II GFS with 4 MDS matrices have 48 and 56 active S-boxes, respectively [9, 10]. Also, the longest impossible differential distinguisher of the Feistel structure with $\mathcal{Q}\mathbf{3_F}$ (and $\mathcal{Q}\mathbf{4_F}$) as diffusion layer of SP function is 3 and 5 rounds shorter than that of improved Type-II GFS and standard Type-II GFS, respectively.

4 Conclusion

In this paper, we firstly used one 4×4 MDS matrix to design non-sparse 16×16 diffusion layers. Although new matrices had appropriate properties regarding resistance against impossible differential and square attacks, their results for the number of differentials and linear active S-boxes were unacceptable. To increase the number of active S-boxes, the idea of using multiple MDS matrices was proposed. The results for the proposed matrices were appropriate from the aspects of resistance against square, impossible differential, related key, linear and differential attacks. Also, the used idea was extended to design matrices of 24×24 and 32×32 sizes.
Then, the proposed SP structure was used as the

Table 7. Comparison of MNDAS values for r rounds of several Feistel structures

r	1	2	4	6	8	12	14	20	24	30
MNDAS for two sub-block Feistel with $\beta = 6$ [34]	0	1	6	8	12	20	24	32	41	50
MNDAS for two sub-block Feistel with $\beta = 9$ [34]	0	1	10	12	21	32	34	49	59	71
MNDAS for two sub-block Feistel with switching and $\beta = 6$ [7]	0	1	6	12	14	24	26	38	48	60
MNDAS for two sub-block Feistel with $\tilde{\mathcal{Q}}\mathbf{1_F}$	0	1	7	12	17	26	31	45	54	68
MNDAS for CLEFIA [11]	0	1	6	12	18	28	34	50	59	76

Table 8. MNDAS and MNLAS of r rounds of two sub-block Feistel with $\mathcal{Q}\mathbf{2_F}$ and Type-II 6-GFS

r	1	2	4	6	8	12	18	24	28	36
MNDAS and MNLAS for Feistel structure using $\mathcal{Q}\mathbf{2_F}$	0	1	10	15	20	31	50	70	80	105
MNDAS and MNLAS of standard Type-II 6-GFS [10]	0	1	6	12	18	30	47	?	?	?
MNDAS and MNLAS of improved Type-II 6-GFS [10]	0	1	6	12	22	32	50	?	?	?

Table 9. MNDAS and MNLAS of r rounds of two sub-block Feistel with $\mathcal{Q}\mathbf{3_F}$ and $\mathcal{Q}\mathbf{4_F}$ and Type-II 8-GFS

r	1	2	4	6	8	12	18	24	28	32
MNDAS for $\mathcal{Q}\mathbf{3_F}$	0	1	11	18	25	40	63	85	100	114
MNLAS for $\mathcal{Q}\mathbf{3_F}$	0	1	10	18	24	36	57	77	90	104
MNDAS and MNLAS for $\mathcal{Q}\mathbf{4_F}$	0	1	11	15	24	34	54	76	89	102
MNDAS and MNLAS of standard Type-II 8-GFS [10]	0	1	6	12	18	36	48	?	?	?
MNDAS and MNLAS of improved Type-II 8-GFS [10]	0	1	6	12	23	39	56	?	?	?

round function of Feistel structures. The new structures are more secure than the Type-II GFS from the aspect of resistance against linear, differential and impossible differential attacks and have desirable implementation properties, making them appropriate options to design future symmetric key ciphers.

References

[1] C. Shannon. Communication theory of secrecy systems. *Bell System Technical Journals*, 28(4):656–715, 1949.

[2] Lawrence Bassham, Çağdaş Çalık, Kerry McKay, and Meltem Sönmez Turan. Submission requirements and evaluation criteria for the lightweight cryptography standardization process. *US National Institute of Standards and Technology*, 2018.

[3] Joan Daemen and Vincent Rijmen. *The design of Rijndael*, volume 2. Springer, 2002.

[4] Kaisa Nyberg. Generalized feistel networks. In *International conference on the theory and application of cryptology and information security*, pages 91–104. Springer, 1996.

[5] Yuliang Zheng, Tsutomu Matsumoto, and Hideki Imai. On the construction of block ciphers provably secure and not relying on any unproved hypotheses. In *Conference on the Theory and Application of Cryptology*, pages 461–480. Springer, 1989.

[6] T. Shirai and K. Shibutani. Improving immunity of feistel ciphers against differential cryptanalysis by using multiple MDS matrices. In *FSE 2004*, volume 3017, pages 260–278. Springer-Verlag, 2004.

[7] Taizo Shirai and Kyoji Shibutani. On feistel structures using a diffusion switching mechanism. In *International Workshop on Fast Software Encryption*, pages 41–56. Springer, 2006.

[8] T. Shirai and K. Araki. On generalized feistel structures using the diffusion switching mechanism. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E91-A(8):2120–2129, 2008.

[9] Yukiyasu Tsunoo, Etsuko Tsujihara, Maki Shigeri, Teruo Saito, Tomoyasu Suzaki, and Hiroyasu Kubo. Impossible differential cryptanalysis of CLEFIA. In *International Workshop on Fast Software Encryption*, pages 398–411. Springer, 2008.

[10] K. Shibutani. On the diffusion of generalized feistel structures regarding differential and linear cryptanalysis. In *SAC 2010*, volume 6544, pages 211–228. Springer-Verlag, 2011.

[11] Taizo Shirai, Kyoji Shibutani, Toru Akishita,

Shiho Moriai, and Tetsu Iwata. The 128-bit blockcipher CLEFIA. In *International workshop on fast software encryption*, pages 181–195. Springer, 2007.

[12] 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(2):327–343, 2017.

[13] Mahdi Sajadieh, Mohammad Dakhilalian, Hamid Mala, and Pouyan Sepehrdad. Efficient recursive diffusion layers for block ciphers and hash functions. *Journal of Cryptology*, 28(2):240–256, 2015.

[14] Serge Vaudenay. On the need for multipermutations: Cryptanalysis of MD4 and SAFER. In *International Workshop on Fast Software Encryption*, pages 286–297. Springer, 1994.

[15] Hamid Mala, Mohammad Dakhilalian, Vincent Rijmen, and Mahmoud Modarres-Hashemi. Improved impossible differential cryptanalysis of 7-round AES-128. In *International Conference on Cryptology in India*, pages 282–291. Springer, 2010.

[16] Joan Daemen, Lars Knudsen, and Vincent Rijmen. The block cipher Square. In *International Workshop on Fast Software Encryption*, pages 149–165. Springer, 1997.

[17] Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Mike Stay, David Wagner, and Doug Whiting. Improved cryptanalysis of Rijndael. In *International Workshop on Fast Software Encryption*, pages 213–230. Springer, 2000.

[18] Ting Cui, Chenhui Jin, and Jing Ma. A new method for finding impossible differentials of generalized feistel structures. *Chinese Journal of Electronics*, 27(4):728–733, 2018.

[19] Bing Sun, Meicheng Liu, Jian Guo, Vincent Rijmen, and Ruilin Li. Provable security evaluation of structures against impossible differential and zero correlation linear cryptanalysis. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 196–213. Springer, 2016.

[20] Taizo Shirai and Kyoji Shibutani. Improving immunity of feistel ciphers against differential cryptanalysis by using multiple MDS matrices. In *International Workshop on Fast Software Encryption*, pages 260–278. Springer, 2004.

[21] Mahdi Sajadieh and Mohammad Vaziri. Using MILP in analysis of feistel structures and improving type II GFS by switching mechanism. In *International Conference on Cryptology in India*, pages 265–281. Springer, 2018.

[22] Sumanta Sarkar and Habeeb Syed. Lightweight diffusion layer: Importance of toeplitz matrices. *Cryptology ePrint Archive*, 2016.

[23] Chaoyun Li and Qingju Wang. Design of lightweight linear diffusion layers from near-MDS matrices. *Cryptology ePrint Archive*, 2017.

[24] Kishan Chand Gupta, Sumit Kumar Pandey, and Ayineedi Venkateswarlu. Towards a general construction of recursive MDS diffusion layers. *Designs, Codes and Cryptography*, 82(1):179–195, 2017.

[25] Akbar Mahmoodi Rishakani, Mohammad Reza Mirzaee Shamsabad, Seyed Mojtaba Dehnavi, Mohammad Amin Amiri, Hamidreza Maimani, and Nasour Bagheri. Lightweight 4x4 MDS matrices for hardware-oriented cryptographic primitives. *The ISC International Journal of Information Security*, 11(1):35–46, 2019.

[26] Mahdi Sajadieh and Mohsen Mousavi. Construction of MDS matrices from generalized feistel structures. *Designs, Codes and Cryptography*, 89(7):1433–1452, 2021.

[27] Christophe De Cannière. Trivium: A stream cipher construction inspired by block cipher design principles. In *International Conference on Information Security*, pages 171–186. Springer, 2006.

[28] Joan Daemen. *Cipher and hash function design strategies based on linear and differential cryptanalysis*. PhD thesis, Doctoral Dissertation, March 1995, KU Leuven, 1995.

[29] Jongsung Kim, Seokhie Hong, Jaechul Sung, Sangjin Lee, Jongin Lim, and Soohak Sung. Impossible differential cryptanalysis for block cipher structures. In *Progress in Cryptology - INDOCRYPT 2003*, pages 82–96. Springer Berlin Heidelberg, 2003.

[30] Taizo Shirai, Shoji Kanamaru, and George Abe. Improved upper bounds of differential and linear characteristic probability for camellia. In *International Workshop on Fast Software Encryption*, pages 128–142. Springer, 2002.

[31] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. A new structural-differential property of 5-round AES. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 289–317. Springer, 2017.

[32] Pierre-Alain Fouque, Jérémy Jean, and Thomas Peyrin. Structural evaluation of AES and chosen-key distinguisher of 9-round AES-128. In *Advances in Cryptology – CRYPTO 2013*, pages 183–203. Springer Berlin Heidelberg, 2013.

[33] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In *International Conference on Information Security and Cryptology*, pages 57–76. Springer, 2011.

[34] Masayuki Kanda. Practical security evaluation

against differential and linear cryptanalyses for feistel ciphers with SPN round function. In *International Workshop on Selected Areas in Cryptography*, pages 324–338. Springer, 2000.

[35] Y. Tsunoo, E. Tsujihara, M. Shigeri, T. Saito, T. Suzaki, and H. Kubo. Impossible differential cryptanalysis of CLEFIA. In *FSE 2008*, volume 5086, pages 398–411. Springer-Verlag, 2008.

Mahdi Sajadieh received the B.Sc., M.Sc. and Ph.D. 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., and 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.

A Counting Method for MNADS of $\mathcal{Q}1$

This section describes the counting method of [12]. The first step of the counting method is to build a state transformation array with $(5)^4$ rows named ST. The j^{th} row of the array corresponds to the round input with VWT value of j (for $\begin{pmatrix}\hat{\mathbf{x}}_0\\ \hat{\mathbf{x}}_1\\ \hat{\mathbf{x}}_2\\ \hat{\mathbf{x}}_3\end{pmatrix}$, $j = x_0^C \times 125 + x_1^C \times 25 + x_2^C \times 5 + x_3^C$). This row includes the possible VWT values for the round output. To find the possible VWT round output values for j^{th} row, S and $\mathcal{Q}\mathbf{1}$ are applied to the input with VWT value of j. Applying S on the input does not change its VWT value. Thus, only $\mathcal{Q}\mathbf{1}$ is applied on input with VWT value of j and possibly results in several VWT values. All of these outputs build j^{th} row of the array.

In addition to the state transformation array, a vector of size 5^4 is built. This vector is notated with Wh. Indexing of the elements of Wh starts from 0. The j^{th} element of Wh is the number of active S-boxes of a round for round input with VWT value of j (for $\begin{pmatrix}\hat{\mathbf{x}}_0\\ \hat{\mathbf{x}}_1\\ \hat{\mathbf{x}}_2\\ \hat{\mathbf{x}}_3\end{pmatrix}$, $Wh(j) = x_0^C + x_1^C + x_2^C + x_3^C$).

Now the method of computing a lower bound for the number of active S-boxes for R rounds of the mentioned SPN structure is described. Consider an array T of size $R \times 5^4$. The element $T[r][i]$ shows the lower bound for a number of active S-boxes of the r-round characteristics finishing with i at the end of the r^{th} round. The most important property of this storage method is that its computational complexity linearly increases with each round (having the r^{th} row, the run time of computing the $r+1^{th}$ row is equal to the run time of computing the $r+2^{th}$ row using the $r+1^{th}$ row). Given the r^{th} row of the array, $T[r+1][j]$ is computed by exhaustive search over $T[r][i], i = 0, \cdots, 5^4 - 1$ and all of the possible transitions from i in round r to j in round $r+1$. Thus, we have:

$$T[r+1][j] = \min_i\{T[r][i] + Wh(i) | i \rightarrow j \textit{ is possible in } ST \textit{ table}\} \quad \text{(A.1)}$$

To compute the first row of the array T using Equation A.1, it is assumed that $T[0][i]$ is 0 for all values of i. The minimum non-zero element of the r^{th} row is the lower bound of the number of active S-boxes for r rounds of the structure.