Algebraic Construction of 16×16 Binary Matrices of Branch Number 7 with One Fixed Point

Speaker M. Tolga Sakallı[1], **Bora Aslan**[2]

[1] *Computer Engineering Department, Trakya University, Edirne, Turkey*
tolga@trakya.edu.tr

[2] *Computer Programming Department, Kırklareli University, Kırklareli, Turkey*
bora.aslan@kirklareli.edu.tr

Extended Abstract

1 Introduction

Most block ciphers are constructed by repeatedly applying a simple function. This approach is known as iterated block cipher. Each iteration is called a round and the repeated function is termed the round function. Also, many block ciphers are designed by using two structures: Feistel Networks and Substitution Permutation Networks (SPNs). Both of these two structures use substitution and linear transformation (also called diffusion layer) to implement Shannon's principles, which are confusion and diffusion. The diffusion layer ensures that after a few rounds all the output bits depend on all the input bits, while the substitution layer or nonlinear layer ensures that this dependency is of a complex and nonlinear nature [1]. Most diffusion layers are linear transformations having matrix representations over $GF(2^m)$ or $GF(2)$. A linear transformation provides diffusion [2] to meet security of a round function of a block cipher by mixing bits of the fixed size input block to produce the corresponding output block of the same size [3]. Existing techniques of measuring diffusion are as follows:

(a) the avalanche effect [4],

(b) the strict avalanche effect [5],

(c) the completeness property [6],

(d) the branch number [7],

(e) the number of fixed points [3].

While the first two criteria quantify the effects of one-bit change to changes in the output bits, the completeness property deals with the dependency of the output bits

on the input bits. On the other hand, the branch number, which represents diffusion rate and measures security against linear [8] and differential cryptanalysis [9], denotes the minimum number of active S-boxes for any two consecutive rounds. The last measure, the number of fixed points, provides an indication of how well the linear transformation effectively changes the value of the input block when producing the output block. The basis of the idea is that there is no diffusion at fixed points since the input blocks are left unchanged by the linear transformation.

Many block ciphers use Maximum Distance Separable (MDS) and Maximum Distance Binary Linear (MDBL) codes as diffusion layers. From the well known ciphers, while the AES [10] and Khazad [11] use MDS codes, the Camellia [12] and ARIA [13] use MDBL codes as diffusion layers in their design. These diffusion layers are shown in Table 1.

Table 1: Diffusion layers of AES, Khazad, Camellia and ARIA

Block Cipher	Diffusion Layer
AES	4×4 MDS matrix over $GF(2^8)$
Khazad	8×8 involutory MDS matrix over $GF(2^8)$
Camellia	8×8 binary matrix over $GF(2^8)$
ARIA	16×16 involutory binary matrix over $GF(2^8)$

Binary matrices, which are linear transformations having matrix representation over GF(2), are used as diffusion layers of block ciphers, such as in Camellia and ARIA. Also, the binary matrices used in Camellia and ARIA have the maximum branch number and therefore are called as MDBL codes [1]. In addition, the maximum branch number of 8×8 and 16×16 binary matrices is respectively upper bounded by 5 and 8 respectively. On the other hand, an advantage of using such binary matrices in the design of block ciphers compared with MDS codes is the implementation phase where only XOR operations are needed while MDS matrices may need XOR operations, table look-ups, and xtime calls [14].

The importance of the number of fixed points in linear transformations is given in [3]. In that study, it is also stated that if the number of fixed points in a linear transformation greatly exceed the expected number for a random linear transformation, then this is an indication of poor diffusion of the linear transformation since the bits in these blocks are left unchanged when producing the output blocks. Note also that the expected number of fixed points in a random permutation is one [3]. In [15], we have presented a new algebraic construction method to obtain 8×8 and 16×16 binary matrices of optimal branch number and our experimental results show that involutional binary matrices provide maximum branch number value for 16×16 binary matrices while they have 2^{64} fixed points at the same time. Note that the number of fixed points in a 16×16 binary transformation is considered when this transformation transforms a 128-bit input a 128-bit output or each input element is a byte value. Also, it is shown that non-involutory 16×16 binary matrices of maximum branch number with 2^8 fixed points can be obtained by us-

ing some special permutations and involutory binary matrices of maximum branch number. As for this paper, the algebraic construction of binary matrices of branch number 7 with one fixed point is presented. Our construction method is based on 4×4 Hadamard or circulant MDS matrices with the elements of $GF(2^4)$. After giving mathematical preliminaries, this construction method is given to determine 16×16 binary matrices of branch number 7 with one fixed point, which can be used to transform a 128-bit input block to a 128-bit output block.

2 Mathematical Background

A finite field is commutative ring (with unity) in which all nonzero elements have a multiplicative inverse [14]. The finite field $GF(2^m)$ has 2^m elements, where m is a nonzero positive integer. Each of the 2^m elements of $GF(2^m)$ can be uniquely represented with a polynomial degree up to $m-1$ with coefficients in $GF(2)$. For example, if x is an element in $GF(2^m)$, then one can have polynomial or standard basis representation of x as

$$x_{m-1}\alpha^{m-1} + x_{m-2}\alpha^{m-2} + ... + x_1\alpha + x_0 \tag{1}$$

where α denotes the primitive element used to construct the finite field $GF(2^m)$. The addition of two field elements of $GF(2^m)$ is simply bitwise XOR operation of the coefficients of the equal powers of α. On the other hand, multiplication in a finite field $GF(2^m)$ is related with multiplying the two polynomials and reducing the product polynomial modulo $p(x)$, which is an irreducible polynomial of degree m. In this paper, we are concerned with the finite field $GF(2^4)$, where the irreducible polynomial over $GF(2)$ is $x^4 + x + 1$. A compact representation of an element $x \in GF(2^4)$ uses hexadecimal digits (denoted with subscript h), expressing the coefficients of the polynomial representation. For example, $\alpha^3 + \alpha = A_h$ in the finite field $GF(2^4)$. For more information on finite fields, the reader is referred to [16, 17].

Example 1. *Let $GF(2^4)$ be defined by the primitive polynomial $p(x) = x^4 + x + 1$. Let α be a root of $p(x)$. Then, for any $x \in GF(2^4)$, we can write $x = x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0$, where $(x_3, ..., x_0) \in GF(2)$ and $\{\alpha_3, \alpha_2, \alpha_1, \alpha_0\}$= $\{\alpha^3, \alpha^2, \alpha^1, 1\}$ is a polynomial basis of $GF(2^4)$ over $GF(2)$. A finite field multiplication (denoted with symbol $\otimes$) of 4_h or α^2 by any $x \in GF(2^4)$ can be given as*

$$\begin{aligned}(4_h \otimes x) mod p(x) &= (\alpha^2 \otimes x) mod p(x) \\ &= (x_3\alpha^5 + x_2\alpha^4 + x_1\alpha^3 + x_0\alpha^2\ ) mod p(x) \\ &= x_1\alpha^3 + (x_3 + x_0)\alpha^2 + (x_3 + x_2)\alpha + x_2,\end{aligned}$$

which corresponds to the 4×4 binary linear transformation

$$\begin{bmatrix} x_0' \\ x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} . \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} .$$

By substituting the elements of $GF(2^4)$ with their corresponding 4×4 binary linear transformations, we can transform 4×4 matrices with the elements of $GF(2^4)$ to their equivalent 16×16 binary matrices. Generally, in the literature, MDS matrices used as diffusion layers are constructed by two types of matrices: circulant and Hadamard matrices. In our study, we also use Hadamard and circulant matrices to construct 16×16 binary matrices of branch number 7 with fixed point one.

Most diffusion layers are linear transformations and represented as matrices and therefore we can define a diffusion layer as $A : (\{0,1\}^m)^n \rightarrow (\{0,1\}^m)^n$ which is a linear transformation as follows:

$$A(x) = A.x^T = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} . \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ \cdots \\ x_n \end{bmatrix},$$

where $x = (x_1, x_2, \cdots, x_n)^T$, $x_i \in \{0,1\}^m$, $i = 1, \ldots, n$. Also, n represents the number of S-boxes in a diffusion layer A, where the size of each input and output of each S-box is m-bit [1]. The elements of matrix A may be in $GF(2^m)$ (especially in $GF(2^8)$ or in $GF(2)$).

The branch number of an $n \times n$ matrix A is defined by

$$\beta(A) = min\{wt(x) + wt(A.x^T) | x \in (\{0,1\}^m)^n, x \neq 0\} \quad (2)$$

The Hamming weight of a code word c is the number of nonzero components in c and denoted by $wt(c)$. In addition, the Hamming distance between two vectors (or code words) from the dimensional vector space is the number of positions (out of) by which the two vectors differ [14].

A linear $[n, k, d]$-code over $GF(2^m)$ is a k-dimensional sub-space of the vector space $(GF(2^m))^n$, where the Hamming distance between two distinct n-element vector is at least d, and d is the largest number with this property. A generator matrix G for a linear $[n, k, d]$-code C is a $k \times n$ matrix whose rows form a basis for C . Linear $[n, k, d]$-codes obey the Singleton bound, $d \leq n - k + 1$ [14].

Lemma 1. *A code meets the Singleton bound, namely $d \leq n - k + 1$, is called a Maximum Distance Separable or MDS code. Alternatively, an $[n, k, d]$-error correcting code with generating matrix $G = [I_{k \times k} | A]$, where I_{kxk} is the $k \times k$ identity matrix, and A is a $k \times (n - k)$ matrix, is MDS if and only if every square sub-matrix formed from i rows and i columns, $1 \leq i \leq min\{k, n - k\}$, of A is nonsingular [14], [15].*

In the literature, generally, there are four approaches for the construction of MDS matrices. The first approach is related with the use of circulant matrices, where each row is a rotated instance (by a single unit) of the neighboring rows in the same direction. The second one is related with the use of some heuristics for the construction of low implementation-cost MDS matrices as stated in [18]. The third one is related with the use of Hadamard matrices for the construction of involutory MDS matrices. For example, while a 4×4 circulant MDS matrix is used in the block cipher AES, an 8×8 involutory MDS matrix (Hadamard matrix) is used in the block cipher Khazad. Finally, the fourth approach is random construction of MDS and involutory matrices [19]. From the viewpoint of security, 4×4, 8×8, and 16×16 MDS matrices provide the optimal branch number of 5, 9, and 17 respectively [7], [14], [20].

A 4×4 circulant matrix with the elements of $GF(2^m)$, $circ(a_1, a_2, a_3, a_4)$, where each row vector is rotated one position to the right relative to the preceding row vector, can be shown as follows:

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_4 & a_1 & a_2 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_2 & a_3 & a_4 & a_1 \end{bmatrix}_{4\times4} \tag{3}$$

A 4×4 Hadamard matrix with the elements of $GF(2^m)$, $had(a_1, a_2, a_3, a_4)$, can be shown as follows:

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix}_{4\times4} \tag{4}$$

Two $n \times n$ binary matrices A, B are permutation homomorphic to each other if there exists a row permutation ρ and a column permutation γ satisfying [21]

$$\rho(\gamma(A)) = \gamma(\rho(A)) = B \tag{5}$$

Lemma 2. *If two matrices A, B are permutation homomorphic to each other, then A, B are of the same branch number [21].*

By Lemma 2, the branch number is the same for any row or column permutation, thus many matrices can be constructed by using a binary matrix having any branch number value. On the other hand, we define two special permutations to be used in the next sections. These are:

1-) to rotate cyclically l bits, where $l \in \{1, \ldots, n-1\}$, to the right of all rows of an $n \times n$ binary linear transformation,

2-) to rotate cyclically l bits, where $l \in \{1, \ldots, n-1\}$, to the downwards of all columns of an $n \times n$ binary linear transformation.

The importance of the number of fixed points in linear transformations is given in [3]. In that study, it is also stated that if the number of fixed points in a linear transformation greatly exceed the expected number for a random linear transformation, then this is an indication of poor diffusion of the linear transformation since the bits in these blocks are left unchanged when producing the output blocks. Note also that the expected number of fixed points in a random permutation is one [3].

Consider an input block to a linear transformation formed by m-bit values in the field $GF(2^m)$ and let the linear transformation matrix be an $n \times n$ matrix and I be an $n \times n$ identity matrix. Then, the set of all fixed points for that linear transformation, which can be represented by a nonsingular matrix A, can be obtained by solving the following equation

$$(A - I)x^T = 0 \tag{6}$$

where 0 is the all-zero vector of length n. Hence, the number of fixed points can be given as

$$F_A = 2^{m(rank(A)-rank(A-I))} = 2^{m(n-rank(A-I))} \tag{7}$$

From Equation 7, it is clear to see that if the $A - I$ matrix has bigger rank, then the linear transformation A has the less number of fixed points. In [3], the diffusion measure based on the number of fixed points is applied to the linear transformations of several SPN ciphers: the AES, ARIA, PRESENT [22], and Serpent [23]. It is shown that the linear transformation of all ciphers except Serpent have more fixed points than the expected number for a random linear transformation. For example, the 16×16 binary linear transformation of the ARIA includes 2^{72} fixed points since the rank of the $A_{ARIA} - I$ matrix is 7.

3 Algebraic Construction of 16×16 Binary Matrices of Branch Number 7 with One Fixed Point

The ARIA block cipher uses a 16×16 binary matrix which is an involution and has branch number of 8 [21]. It has also 2^{72} fixed points since the rank of $A_{ARIA} - I$ matrix is 7. On the other hand, the method for the construction of ARIA type linear transformations can be found in [21], [15]. In this section, by transforming 4×4 matrices with elements in $GF(2^4)$ into the their equivalent binary form, we construct 16×16 binary linear transformations of branch number of 7 with one fixed point. Therefore, we look for 16×16 binary matrices, where the rank of $A_{Binary} - I$ matrices is 16. We call these type of matrices nearly MDBL codes.

When constructing 16×16 binary matrices of branch number 7 with one fixed point, we look for 4×4 matrices that satisfy four restrictions simultaneously;

(i) Be MDS,

(ii) Be Hadamard or circulant matrix,

(iii) The elements of 4×4 matrix in $GF(2^4)$ will be chosen such that the transformed binary matrix should have the Hamming weight equal to 120.

(iv) The binary matrix, A_{Binary}, transformed from 4×4 matrix should have branch number of 7 and the rank of $A_{Binary} - I$ matrix should be 16.

In Examples 2 and 3, we give two examples of 16×16 binary matrices satisfying the restrictions above. Also, while the 16×16 binary matrix in Example 2 is constructed from a 4×4 MDS and Hadamard matrix, the 16×16 binary matrix in Example 3 is constructed from a 4×4 MDS and circulant matrix.

Example 2. *Let* $M = had(8_h, 1_h, B_h, C_h) = \begin{bmatrix} 8_h & 1_h & B_h & C_h \\ 1_h & 8_h & C_h & B_h \\ B_h & C_h & 8_h & 1_h \\ C_h & B_h & 1_h & 8_h \end{bmatrix}$ *be MDS and* 4×4 *Hadamard matrix. Using the idea given in Example 1, the* 4×4 *Hadamard matrix with the elements of* $GF(2^4)$ *can be transformed into the* 16×16 *binary matrix satisfying the criteria above as follows:*

$$A_{Binary} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Example 3. *Let* $M = circ(1_h, 8_h, B_h, C_h) = \begin{bmatrix} 1_h & 8_h & B_h & C_h \\ C_h & 1_h & 8_h & B_h \\ B_h & C_h & 1_h & 8_h \\ 8_h & B_h & C_h & 1_h \end{bmatrix}$ *be MDS and* 4×4 *circulant matrix. Using the idea given in Example 1, the* 4×4 *circulant matrix with the elements of* $GF(2^4)$ *can be transformed into the* 16×16 *binary matrix satisfying the criteria above as follows:*

$$A_{Binary} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

In a straight coding on an 8-bit processor, the binary matrices given in Examples 2 and 3 require 104 byte XORs for the implementation when input elements to the binary matrices are considered as byte values. But, the total number of byte XORs can be reduced by adding more variables to the implementation for both of them. Furthermore, the implementation on a 32-bit and a 64-bit processor can take the advantage of Hadamard and circulant form of the binary matrices. Therefore, the implementation of these matrices are also suitable for these processors.

4 On the Productivity of the Proposed Method for 16×16 Binary Linear Transformations

In order to obtain the productivity of the proposed method for 16×16 binary linear transformations, we have searched for all possible 4×4 nonsingular Hadamard and circulant matrices, which can be transformed into the binary matrices of branch number 7 with one fixed point.

For Hadamard matrices, we have obtained 1584 matrices with the elements from $GF(2^4)$ defined by the primitive polynomial x^4+x+1. Also, we have noticed that the constructed binary matrices from the permutations of the same elements in Hadamard form have identical properties from viewpoint of branch number and

the number of fixed points. Therefore, we have divided 1584 matrices into 66 different classes where each class includes 24 (4!) members, the permutations of the elements of a class. In Appendix A, we show all the obtained classes, which can be transformed into the binary matrices of branch number 7 with one fixed point. Note also that each class with 4 elements is represented by hexadecimal values in ascending order and the Hamming weights of the binary linear transformations constructed from distinct classes may be different from each other. As for circulant matrices, on the other hand, we have obtained 2744 binary matrices with the same cryptographic properties.

In $GF(2^4)$, there are 2 more irreducible polynomials, which are $x^4 + x^3 + 1$ and $x^4 + x^3 + x^2 + x + 1$. For Hadamard type matrices, after searching for all the 4×4 matrices with the elements of $GF(2^4)$ defined by the irreducible polynomial $x^4 + x^3 + 1$, we have again found 66 classes. As for the finite field $GF(2^4)$ defined by the irreducible polynomial $x^4 + x^3 + x^2 + x + 1$, we have found 128 classes with the same cryptographic properties. For circulant type matrices, on the other hand, we have obtained 2786 and 1088 matrices with the elements from $GF(2^4)$ defined by the primitive polynomial $x^4 + x + 1$ and $x^4 + x^3 + x^2 + x + 1$, respectively, which can be transformed into the 16×16 binary matrices of branch number 7 with one fixed point.

5 Conclusions

In this paper, we have proposed a new algebraic construction method to generate 16×16 binary linear transformations of branch number 7 with one fixed point. In this construction method, we have used 4×4 Hadamard and circulant MDS matrices. By using special permutations together given in Section 2, one can generate at least more 15 binary matrices with identical cryptographic properties from one binary matrix. Therefore, totally, we can generate at least 205728 (12858×16) binary matrices of branch number 7 with one fixed point. Moreover, the implementation of these binary matrices on a 32-bit and a 64-bit processor can take the advantage of Hadamard or circulant form of them. Therefore, the implementation of these matrices are also suitable for these processors.

Finally, the given binary matrices for a 128-bit block cipher are resistant against linear and differential cryptanalysis when applying them in a reasonable number of rounds. But, for the other important attacks like truncated differential cryptanalysis and impossible differential cryptanalysis, a further security analysis should be performed on the given binary matrices and then the use of these matrices is recommended.

Appendix: A

The list of all class representatives of 4×4 MDS and Hadamard matrices for the $x^4 + x + 1$, which can be transformed into 16×16 binary matrices of branch number 7 with one fixed point .

1-)$had(1_h,2_h,4_h,E_h)$	34-)$had(3_h,7_h,9_h,E_h)$
2-)$had(1_h,2_h,6_h,A_h)$	35-)$had(3_h,7_h,A_h,D_h)$
3-)$had(1_h,2_h,6_h,B_h)$	36-)$had(3_h,8_h,B_h,F_h)$
4-)$had(1_h,2_h,7_h,9_h)$	37-)$had(3_h,9_h,C_h,D_h)$
5-)$had(1_h,2_h,7_h,F_h)$	38-)$had(4_h,6_h,8_h,F_h)$
6-)$had(1_h,3_h,4_h,E_h)$	39-)$had(4_h,7_h,A_h,C_h)$
7-)$had(1_h,3_h,6_h,C_h)$	40-)$had(4_h,8_h,C_h,E_h)$
8-)$had(1_h,5_h,9_h,E_h)$	41-)$had(4_h,8_h,C_h,F_h)$
9-)$had(1_h,5_h,A_h,D_h)$	42-)$had(4_h,A_h,B_h,E_h)$
10-)$had(1_h,6_h,7_h,B_h)$	43-)$had(5_h,6_h,E_h,F_h)$
11-)$had(1_h,6_h,9_h,B_h)$	44-)$had(5_h,7_h,8_h,F_h)$
12-)$had(1_h,6_h,A_h,B_h)$	45-)$had(5_h,7_h,A_h,D_h)$
13-)$had(1_h,6_h,A_h,F_h)$	46-)$had(5_h,8_h,A_h,E_h)$
14-)$had(1_h,7_h,D_h,E_h)$	47-)$had(5_h,8_h,A_h,F_h)$
15-)$had(1_h,8_h,B_h,C_h)$	48-)$had(5_h,9_h,B_h,F_h)$
16-)$had(1_h,9_h,A_h,B_h)$	49-)$had(5_h,9_h,D_h,F_h)$
17-)$had(1_h,B_h,C_h,E_h)$	50-)$had(5_h,A_h,B_h,F_h)$
18-)$had(2_h,3_h,4_h,E_h)$	51-)$had(5_h,B_h,C_h,E_h)$
19-)$had(2_h,3_h,5_h,B_h)$	52-)$had(6_h,7_h,9_h,C_h)$
20-)$had(2_h,4_h,5_h,C_h)$	53-)$had(6_h,7_h,9_h,F_h)$
21-)$had(2_h,4_h,6_h,E_h)$	54-)$had(6_h,7_h,D_h,E_h)$
22-)$had(2_h,4_h,7_h,C_h)$	55-)$had(6_h,8_h,B_h,E_h)$
23-)$had(2_h,4_h,D_h,E_h)$	56-)$had(6_h,9_h,A_h,C_h)$
24-)$had(2_h,5_h,6_h,A_h)$	57-)$had(6_h,A_h,D_h,E_h)$
25-)$had(2_h,5_h,B_h,E_h)$	58-)$had(7_h,8_h,A_h,C_h)$
26-)$had(2_h,5_h,C_h,E_h)$	59-)$had(7_h,8_h,A_h,D_h)$
27-)$had(2_h,6_h,B_h,C_h)$	60-)$had(7_h,8_h,B_h,D_h)$
28-)$had(2_h,7_h,8_h,9_h)$	61-)$had(7_h,8_h,B_h,F_h)$
29-)$had(3_h,4_h,8_h,C_h)$	62-)$had(7_h,9_h,A_h,E_h)$
30-)$had(3_h,4_h,8_h,D_h)$	63-)$had(7_h,9_h,A_h,F_h)$
31-)$had(3_h,5_h,C_h,F_h)$	64-)$had(7_h,9_h,B_h,D_h)$
32-)$had(3_h,5_h,D_h,E_h)$	65-)$had(7_h,9_h,C_h,F_h)$
33-)$had(3_h,6_h,8_h,9_h)$	66-)$had(7_h,A_h,D_h,E_h)$

References

[1] D. Kwon, S. H. Sung, J. H. Song, S. Park, Design of Block Ciphers and Coding Theory, *Trends in Mathematics*, **8**(1), pp. 13–20, (2005).

[2] C. E. Shannon, Communication Theory of Secrecy, *Bell System Technical Journal*, **28**(7), pp. 656–715, (1949).

[3] M. R. Z'aba, Analysis of Linear Relationships in Block Ciphers. *Ph.D. Thesis*, Queensland University of Technology, Brisbane, Australia, (2010).

[4] H. Feistel, Cryptography and Computer privacy, *Scientific American*, **228**(5), pp. 15–23, (1973).

[5] A. F. Webster, S. E. Tavares, On the Design of S-boxes, In *Proceedings of CRYPTO'85*, Lecture Notes in Computer Science, Springer-Verlag, vol. 218, pp. 523–534, (1986).

[6] J. B. Kam, G. I. Davida, Structured Design of Substitution-Permutation Encryption Networks, *IEEE Transactions on Computers*, **28**(10), pp. 747–753, (1979).

[7] J. Daemen, V. Rijmen, *The Design of Rijndael, AES - The Advanced Encryption Standard*, Springer-Verlag, (2002).

[8] M. Matsui, Linear Cryptanalysis Method for DES Cipher, In *Proceedings of EUROCRYPT 93*, Lecture Notes in Computer Science, Springer-Verlag, vol. 765, pp. 386–397, (1994).

[9] E. Biham, A. Shamir, Differential Cryptanalysis of DES-like Cryptosystems, In *Proceedings of CRYPTO'90*, Lecture Notes in Computer Science, Springer-Verlag, vol. 537, pp. 2–21, (1990).

[10] US National Institute of Standards and Technology, Advanced Encryption Standard. *Federal Information Processing Standards Publications*, **197**, (2001).

[11] P. S. L. M. Barreto, V. Rijmen, The Khazad Legacy-Level Block Cipher, In *Proceedings First open NESSIE Workshop*, Leuven, (2000).

[12] K. Aoki K, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, T. Tokita, Camellia: A 128-bit block cipher suitable for multiple platforms-design and analysis, In *Proceedings of Selected Areas in Cryptography (SAC 2000)*, Lecture Notes in Computer Science, Springer-Verlag, vol. 2012, pp. 39–56, (2001).

[13] 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, J. Hong, New Block Cipher: ARIA, In

Proceedings of International Conference on Information Security and Cryptology, Lecture Notes in Computer Science, Springer-Verlag, vol. 2971, pp. 432–445, (2004).

[14] J. Nakahara Jr., É. Abrahão, A New Involutory MDS Matrix for the AES. *International Journal of Network Security*, **9**(2), pp. 109–116, (2009).

[15] B. Aslan, M. T. Sakallı, Algebraic Construction of Cryptographically Good Binary Linear Transformations, *Special Issue on the Design and Engineering of Cryptographic Solutions for Secure Information Systems-Security and Communication Networks*, In publication process, (2012).

[16] R. J. McEliece, Finite fields for Computer Scientists and Engineers, *Kluwer Academic Publishers*, Dordrecht, (1987).

[17] R. Lidl, H. Niederreiter, *Finite Fields*, Encyclopedia of Mathematics and its Applications, Addison-Wesley, Reading, Massachusetts, (1983).

[18] P. Junod, S. Vaudenay, Perfect Diffusion Primitives for Block Ciphers-Building Efficient MDS Matrices. In *Proceedings of Selected Areas in Cryptology (SAC 2004)*, Lecture Notes in Computer Science, Springer-Verlag, vol. 3357, pp. 84–99, (2004).

[19] A. M. Youssef, S. Mister, S. E. Tavares, On the Design of Linear Transformation for Substitution Permutation Encryption Networks. In *Proceedings of Selected Areas in Cryptography (SAC'97)*, pp. 40–48, (1997).

[20] P. S. L. M. Barreto, V. Rijmen, The Khazad Legacy-Level Block Cipher. In *Proceedings First open NESSIE Workshop*, Leuven, 2000.

[21] B. W. Koo, H. S. Jang, J. H. Song, Constructing and Cryptanalysis of a 16×16 Binary Matrix as a Diffusion Layer, In *Proceedings of Information Security Applications: 4th International Workshop (WISA 2003)*, Lecture Notes in Computer Science, Springer-Verlag, vol.2908, pp. 489–503, (2003).

[22] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, C. Vikkelsoe, PRESENT: An Ultra-Lightweight Block Cipher, In *Proceedings of 9th International Workshop on Cryptographic Hardware and Embedded Systems-CHES 2007*, Lecture Notes in Computer Science, Springer-Verlag, vol. 4727, pp. 450–466, (2007).

[23] E. Biham, R. Anderson, L. R. Knudsen, Serpent: A New Block Cipher Proposal, In *Proceedings of 5th International Workshop of Fast Software Encryption-FSE'98*, Lecture Notes in Computer Science, Springer-Verlag, vol. 1372, pp. 222–238, (1998).