SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2014; **7**:53–63
Published online 9 May 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.556

SPECIAL ISSUE PAPER

Algebraic construction of cryptographically good binary linear transformations

Bora Aslan[1] and Muharrem Tolga Sakallı[2]*

[1] Computer Programming Department, Kırklareli University, Kırklareli, Turkey
[2] Computer Engineering Department, Trakya University, Edirne, Turkey

ABSTRACT

Maximum Distance Separable (MDS) and Maximum Distance Binary Linear (MDBL) codes are used as diffusion layers in the design of the well-known block ciphers like the Advanced Encryption Standard, Khazad, Camellia, and ARIA. The reason for the use of these codes in the design of block ciphers is that they provide optimal diffusion effect to meet security of a round function of a block cipher. On the other hand, the constructions of these diffusion layers are various. For example, whereas the Advanced Encryption Standard uses a 4×4 MDS matrix over $GF(2^8)$, ARIA uses a 16×16 involutory binary matrix over $GF(2^8)$. The most important cryptographic property of a diffusion layer is the branch number of that diffusion layer, which represents the diffusion rate and measures security against linear and differential cryptanalysis. Therefore, MDS and Maximum Distance Binary Linear codes, which provide maximum branch number for a diffusion layer, are preferred in the design of block ciphers as diffusion layers. In this paper, we present a new algebraic construction method based on MDS codes for 8×8 and 16×16 involutory and non-involutory binary matrices of branch numbers 5 and 8, respectively. By using this construction method, we also show some examples of these diffusion layers.

KEYWORDS

algebraic construction; binary linear transformations; diffusion layers; MDS codes; MDBL codes; block ciphers

***Correspondence**

Muharrem Tolga Sakallı, Computer Engineering Department, Trakya University, Edirne, Turkey.
E-mail: tolga@trakya.edu.tr

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 [1]. Also, many block ciphers are designed by using two structures: Feistel networks and substitution permutation networks (SPNs). Two important block cipher examples designed by Feistel networks and SPNs can be given as Data Encryption Standard [2] and Advanced Encryption Standard (AES) [3], respectively. An SPN structure consists of a substitution layer followed by a linear transformation, also called diffusion layer. The linear diffusion layer ensures that after a few rounds, all the output bits depend on all the input bits. The substitution layer or nonlinear layer ensures that this dependency is of a complex and nonlinear nature [4], [5]. Popular choices of substitution layers or S-boxes providing good cryptographic properties are based on inversion mapping over $GF(2^8)$ [6], [7], [8], [9]. Such S-boxes are widely used in block ciphers like the AES, Camellia [10], and ARIA [11].

A linear transformation provides diffusion [12] by mixing bits of the fixed size input block to produce the corresponding output block of the same size [13]. Existing techniques of measuring diffusion are as follows:

(1) the avalanche effect [14],
(2) the strict avalanche effect [15],
(3) the completeness property [16],
(4) the branch number [17],
(5) the number of fixed points [13].

Whereas the first two criteria quantify the effects of one-bit change to changes in the output bits, 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 [18] and differential cryptanalysis [19], denotes the minimum number of active S-boxes for any two consecutive rounds. The last technique to measure diffusion proposed in [13] is the number of fixed points. This measure 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 because the input blocks are left unchanged by the linear

transformation [13]. On the other hand, another required property of a diffusion layer affecting its choice is the efficiency in hardware and software implementations.

Many block ciphers use Maximum Distance Separable (MDS) and Maximum Distance Binary Linear codes as diffusion layers. From the well-known ciphers, whereas the AES and Khazad [20] use MDS codes, the Camellia and ARIA use Maximum Distance Binary Linear codes as diffusion layers in their design. These diffusion layers are shown in Table I.

As shown in Table I, Khazad and ARIA use involutional diffusion layers, which transform a 64-bit input to a 64-bit output and a 128-bit input to a 128-bit output, respectively. The reason for using involutional diffusion layers in the design of these block ciphers is that involutional mappings reduce the implementation cost of both encryption and decryption operations, and imply that both transformations have the same cryptographic strength [21]. The Camellia cipher designed by using Feistel structure modifies half of the current 128-bit block and transforms a 64-bit input to a 64-bit output in one round encryption stage. Also, it uses an 8×8 non-involutory binary matrix over $GF(2^8)$ as a diffusion layer. On the other hand, the AES uses a 4×4 non-involutory MDS matrix over $GF(2^8)$, which transforms a 32-bit input to a 32-bit output.

This paper proposes a new algebraic construction method for obtaining cryptographically good binary linear transformations. When constructing binary linear transformations, we concentrate on the two cryptographic properties, which are respectively the branch number and the number of fixed points. Our construction method is based on 2×2 and 4×4 involutory and non-involutory MDS matrices with the elements in $GF(2^4)$. After giving mathematical preliminaries, this construction method is given to determine 8×8 and 16×16 involutory and non-involutory binary matrices over $GF(2^8)$ with branch numbers 5 and 8, respectively, because it is stated in [5] that the maximum branch number of 8×8 and 16×16 binary matrices is respectively upper bounded by 5 and 8. 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 whereas MDS matrices may need XOR operations, table look-ups, and xtime calls [21].

2. FINITE FIELDS

A finite field is commutative ring (with unity) in which all nonzero elements have a multiplicative inverse [21]. 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

$$x_{m-1}\alpha^{m-1} + x_{m-2}\alpha^{m-2} + \ldots + x_1\alpha + x_0 \qquad (1)$$

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 [22], [23].

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 $(x_3, \ldots, x_0) \in GF(2)$ 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 of 2_h or α by any $x \in GF(2^4)$ can be given as

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

where $\otimes$ operation denotes finite field multiplication. By using the idea given in Example 1, where the result of finite field multiplication is given in bits, a table may be constructed for finite field multiplication of all 16 possible values by input x representing 4-bit values. In Table II, the finite field multiplication results obtained by using the same primitive polynomial are given because the results in this table will be used when constructing binary linear transformations.

Table I. 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 $GF(2^8)$

3. MATHEMATICAL PRELIMINARIES

In this section, we present the needed mathematical background for the algebraic construction of cryptographically good binary linear transformations. Because we use MDS matrices in the construction, we also present some important properties of MDS codes.

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

Security Comm. Networks 2014; **7**:53–63 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

Table II. Finite field multiplication results of all 16 possible values by input $x=(x_3, x_2, x_1, x_0)$.

Hexadecimal Values	Polynomial basis			
	α^3	α^2	α	1
1	x_3	x_2	x_1	x_0
2	x_2	x_1	x_3+x_0	x_3
3	x_3+x_2	x_2+x_1	$x_3+x_1+x_0$	x_3+x_0
4	x_1	x_3+x_0	x_3+x_2	x_2
5	x_3+x_1	$x_3+x_2+x_0$	$x_3+x_2+x_1$	x_2+x_0
6	x_2+x_1	$x_3+x_1+x_0$	x_2+x_0	x_3+x_2
7	$x_3+x_2+x_1$	$x_3+x_2+x_1+x_0$	$x_2+x_1+x_0$	$x_3+x_2+x_0$
8	x_3+x_0	x_3+x_2	x_2+x_1	x_1
9	x_0	x_3	x_2	x_1+x_0
A	$x_3+x_2+x_0$	$x_3+x_2+x_1$	$x_3+x_2+x_1+x_0$	x_3+x_1
B	x_2+x_0	x_3+x_1	$x_3+x_2+x_0$	$x_3+x_1+x_0$
C	$x_3+x_1+x_0$	x_2+x_0	x_3+x_1	x_2+x_1
D	x_1+x_0	x_0	x_3	$x_2+x_1+x_0$
E	$x_3+x_2+x_1+x_0$	$x_2+x_1+x_0$	x_1+x_0	$x_3+x_2+x_1$
F	$x_2+x_1+x_0$	x_1+x_0	x_0	$x_3+x_2+x_1+x_0$

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

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 [4]. 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

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

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 [21].

A linear $[n,k,d]$-code over $GF(2^m)$ is a k-dimensional subspace 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 [16]. 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, $d \leq n-k+1$ [21].

Lemma 1. *A code meets the Singleton bound, namely $d \leqslant n-k+1$, which 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 $I_{k\times k}$ is the $k \times k$ identity matrix, and A* is a $k \times (n-k)$ *matrix, is MDS if and only if every square submatrix formed from i rows and i columns, $1 \leq i \leq min\{k, n-k\}$, of A is nonsingular* [21], [24].

In order to check the condition in Lemma 1, one should determine determinants of all square submatrices of an $n \times n$ square matrix with elements in $GF(2^m)$. The number of these determinants is given by Equation (2).

$$\sum_{k=1}^{n-2} \left[C\binom{n}{n-k} \right]^2 \tag{2}$$

For example, from Equation (2), one can determine the number of 3×3 submatrices as 16 and the number of 2×2 submatrices as 36 for a 4×4 matrix and, therefore, in order to check that we have a 4×4 MDS matrix, the total number of the determinants of the submatrices to be searched for is 52.

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 [25]. The third one is related with the use of Hadamard matrices for the construction of involutory MDS matrices. For example, whereas 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 a random construction of MDS and involutory matrices [24]. From the viewpoint of security, 4×4, 8×8, and 16×16 MDS matrices provide the optimal branch numbers of 5, 9, and 17, respectively [17], [20], [21].

$A=circ(a_1, a_2, \ldots, a_n)$ is circulant matrix, where each row vector is rotated one position to the right relative to the preceding row vector and therefore A is an $n \times n$ dimensional matrix as shown in Equation (3).

$$A = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ a_n & a_1 & \cdots & a_{n-1} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ a_2 & a_3 & \cdots & a_1 \end{bmatrix}_{n\times n} \quad (3)$$

Lemma 2. *Let $a_1, a_2, \ldots, a_t$ be elements of $GF(2^m)$. Then*

$$(a_1 + a_2 + \ldots + a_t)^{2^k} = a_1^{2^k} + a_2^{2^k} + \ldots + a_t^{2^k} \quad (4)$$

Lemma 3.

Let $A = Had(a_1, a_2, a_3, a_4) = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix}$ *be a 4×4 Hadamard matrix with the elements of $GF(2^m)$. Also, let the elements be distinct and different from zero. Then, A is an involutory MDS matrix if and only if $\sum_{i=1}^{4} a_i = 1$, where the addition between indices is modulo 2 addition or XOR operation and every square submatrix of A is nonsingular (its determinant is not equal to 0) in $GF(2^m)$.*

Proof. From Lemma 1, if every square submatrix of A is nonsingular, then it is a necessary and sufficient condition to ensure that the matrix A is MDS. On the other hand, as shown in Equation (5), we obtain identity matrix if $\sum_{i=1}^{4} a_i^2 = 1$.

$$A^2 = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix} \cdot \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^{4} a_i^2 = 1 & 0 & 0 & 0 \\ 0 & \sum_{i=1}^{4} a_i^2 = 1 & 0 & 0 \\ 0 & 0 & \sum_{i=1}^{4} a_i^2 = 1 & 0 \\ 0 & 0 & 0 & \sum_{i=1}^{4} a_i^2 = 1 \end{bmatrix} \quad (5)$$

Using Lemma 2, we also obtain $\sum_{i=1}^{4} a_i^2 = \sum_{i=1}^{4} a_i = 1$. Because A is unitary ($A^{-1} = A$) and symmetric ($A = A^T$), the matrix A is involutory and MDS.

Lemma 4. *Let A_{11} be any nonsingular $\frac{n}{2} \times \frac{n}{2}$ matrix with elements from $GF(2^m)$. Then, the $n \times n$ matrix $A_i = \begin{bmatrix} A_{11} & A_{11}^{-1} \\ A_{11}^3 + A_{11} & A_{11} \end{bmatrix}$ is an involutory matrix over $GF(2^m)$* [24].

One can easily show that Lemma 3 is valid for any $n \times n$ matrix over $GF(2^m)$ and if we test matrix A_i given in Lemma 4 for the constraint given in Lemma 1, then we can obtain involutory MDS matrices.

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 [26]

$$\rho(\gamma(A)) = \gamma(\rho(A)) = B \quad (6)$$

Lemma 5. *If two matrices A, B are permutation homomorphic to each other, then A, B are of the same branch number* [26].

By Lemma 5, the branch number is the same for any row or column permutation; thus, many matrices can be constructed by using a binary matrix having the optimal 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.

4. FIXED POINTS IN LINEAR TRANSFORMATIONS

The importance of the number of fixed points in linear transformations is given in [13]. 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 because 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 [13], [27].

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:

$$(A - I)x^T = 0 \quad (7)$$

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

$$F_A = 2^{m(rank(A) - rank(A-I))} = 2^{m(n - rank(A-I))} \quad (8)$$

From Equation (8), 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 [13], the diffusion measure based on the number of fixed points is applied to the linear transformations of several SPN ciphers: the AES, ARIA, PRESENT [28], and Serpent [29]. 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 because the rank of the $A_{ARIA} - I$ matrix is 7. Also, we have found that the 8×8 binary linear transformation of the Camellia includes 2^8 fixed points because the rank of the $A_{Camellia} - I$ matrix is 7.

5. ALGEBRAIC CONSTRUCTION OF 8×8 BINARY LINEAR TRANSFORMATIONS

In this section, for the algebraic construction of 8×8 binary linear transformations, we use 2×2 matrices with the elements in $GF(2^4)$. These linear transformations are constructed by transforming 2×2 matrices into binary form by using Table II, and the constructed binary matrices are both involutory and non-involutory matrices with branch number 5.

When constructing an 8×8 non-involutory binary matrix, we look for 2×2 matrices that satisfy three restrictions simultaneously;

(1) Be MDS,
(2) Be circulant or Hadamard matrix,
(3) The binary matrix, A_{Binary}, transformed from 2×2 matrix should have branch number of 5 and the rank of $A_{Binary} - I$ matrix will be 8.

In Example 2, how we obtain the binary matrix by transforming a possible 2×2 matrix satisfying the restrictions earlier is shown in detail.

Example 2. Let $M = circ(8_h, B_h) = \begin{bmatrix} 8_h & B_h \\ B_h & 8_h \end{bmatrix}$ be MDS and 2×2 circulant type matrix. Consider the matrix multiplication next, where $x_0 = (f_3, f_2, f_1, f_0)$, $x_1 = (f_7, f_6, f_5, f_4)$ represent input vectors and $y_0 = (z_3, z_2, z_1, z_0)$, $y_1 = (z_7, z_6, z_5, z_4)$ represent output vectors formed by 4-bit values and therefore $(f_7, f_6, \ldots, f_0, z_7, z_6, \ldots, z_0) \in GF(2)$.

$$\begin{bmatrix} y_o \\ y_1 \end{bmatrix} = \begin{bmatrix} 8_h & B_h \\ B_h & 8_h \end{bmatrix} . \begin{bmatrix} x_o \\ x_1 \end{bmatrix}.$$

By using Table II, we can write matrix multiplication earlier in binary form as shown next:

$$\begin{aligned} z_0 &= f_1 + f_4 + f_5 + f_7, \\ z_1 &= f_1 + f_2 + f_4 + f_6 + f_7, \\ z_2 &= f_2 + f_3 + f_5 + f_7, \\ z_3 &= f_0 + f_3 + f_4 + f_6, \\ z_4 &= f_0 + f_1 + f_3 + f_5, \\ z_5 &= f_0 + f_2 + f_3 + f_5 + f_6, \\ z_6 &= f_1 + f_3 + f_6 + f_7, \\ z_7 &= f_0 + f_2 + f_4 + f_7, \end{aligned}$$

where + operation denotes modulo 2 addition or XOR operation. Then, the multiplication results can be written in the form of $Z = A.F$, which can be shown as next:

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix},$$

where $Z = (z_7, z_6, \ldots, z_0)$ and $F = (f_7, f_6, \ldots, f_0)$. The binary matrix A shown earlier has branch number of 5 and one fixed point. On the other hand, if the input elements $f_7, f_6, \ldots, f_0$ are in $GF(2^8)$, then we obtain a 64-bit to a 64-bit linear transformation with good cryptographic properties.

When constructing an 8×8 involutory binary matrix, we look for 2×2 matrices that satisfy three restrictions simultaneously;

(1) Be MDS,
(2) Be involutory and random matrix given in Lemma 4,
(3) The binary matrix, A_{Binary}, transformed from 2×2 matrix should have branch number of 5 and the rank of $A_{Binary} - I$ matrix will be 4. Therefore, if it is used as a 64-bit to a 64-bit linear transformation, where each input element is in $GF(2^8)$, the binary linear transformation will have 2^{32} fixed points.

In Example 3, we give a possible 2×2 matrix satisfying the restrictions (1) and (2) earlier and 8×8 binary matrix obtained from that matrix satisfying the restriction (3).

Example 3. Let $M = \begin{bmatrix} E_h & 3_h \\ 6_h & E_h \end{bmatrix}$ be involutory MDS and 2×2 random type matrix. Using the same procedure given in Example 2, we can obtain an involutory 8×8 binary matrix as a diffusion layer, as shown next, with the properties stated in the restriction (3) earlier.

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

On the other hand, by the special permutation 1 given in Section 3, it is possible to obtain new matrices by cyclically rotating l bits, where $l \in \{1, \ldots, 7\}$, to the right of all rows of an 8×8 binary linear transformation. For example, if we rotate cyclically 1 bit to the right of

all rows of the binary linear transformation earlier, we obtain a non-involutory binary linear transformation, $A_{(1)}$ shown next, having one fixed point with branch number 5:

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

6. ON THE PRODUCTIVITY OF THE PROPOSED METHOD FOR 8×8 BINARY LINEAR TRANSFORMATIONS

In order to obtain the productivity of the proposed method for 8×8 binary linear transformations, we have searched for all possible 2×2 nonsingular matrices, which can be transformed into the binary form with branch number 5. In Appendix C, we show the obtained involutory and non-involutory 2×2 matrices with the elements from $GF(2^4)$ defined by the primitive polynomial $x^4 + x + 1$. It should be also noted that, in Appendix C, the matrix (x, y, z, t), where $x, y, z, t \in GF(2^4)$, represents the 2×2 matrix $\begin{bmatrix} x & y \\ z & t \end{bmatrix}$.

By Lemma 5, many binary matrices with optimal branch number can be generated by using any row or column permutation from binary matrices, which are transformed from given 2×2 matrices. For example, if we apply special permutations 1 and 2 together, given in Section 3, which are related with cyclically rotating from 0 to 7 bits to the right of all rows of an 8×8 binary linear transformation and cyclically rotating from 0 to 7 bits downwards of all columns of an 8×8 binary-linear transformation, we can generate 63 matrices more from one binary matrix, and therefore, totally, 1024 matrices (16×64) can be generated. Here, the reason of multiplying 64 by 16 is that we take into account eight involutory matrices and eight out of 16 non-involutory matrices because each two non-involutory matrices, as shown in Appendix C, are related with each other by a combination of special permutations 1 and 2. Moreover, we have noticed that 64 out of these 1024 matrices are involutory. The reason is that any involutory matrix provides seven more involutory matrices when applying special permutations 1 and 2 together to an involutory binary linear transformation.

On the other hand, in $GF(2^4)$, there are two more irreducible polynomials, which are $x^4 + x^3 + 1$ (also primitive polynomial) and $x^4 + x^3 + x^2 + x + 1$. We have also searched for all possible 2×2 matrices, which can be transformed into the binary form with branch number 5, with elements from $GF(2^4)$ defined by primitive polynomial $x^4 + x^3 + 1$ and $x^4 + x^3 + x^2 + x + 1$. We have found new eight involutory and 16 non-involutory matrices for the primitive polynomial $x^4 + x^3 + 1$. But, we could not find any involutory or non-involutory matrices, which can be transformed into the binary form with branch number 5, with elements from $GF(2^4)$ defined by the irreducible polynomial $x^4 + x^3 + x^2 + x + 1$.

7. ALGEBRAIC CONSTRUCTION OF 16 × 16 BINARY LINEAR TRANSFORMATIONS

The ARIA cipher uses a 16×16 binary linear transformation, which is an involution and has branch number of 8 [26]. It has also 2^{72} fixed points because 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 [26], [30]. In this section, by transforming 4×4 matrices with elements in $GF(2^4)$ into binary form by using Table II, we construct 16×16 binary linear transformations having 2^{64} fixed points and satisfying the same cryptographic properties with that of the ARIA.

When constructing involutory 16×16 binary matrices with branch number 8, we look for 4×4 matrices that satisfy four restrictions simultaneously:

(1) Be MDS,
(2) Be involutory in Hadamard matrix form as given in Lemma 3,
(3) The elements of 4×4 matrix in $GF(2^4)$ should be chosen such that each row and each column of the transformed binary matrix should have the Hamming weight equal to 7 or 11.
(4) The binary matrix, A_{Binary}, transformed from 4×4 matrix should have branch number of 8 and the rank of $A_{Binary} - I$ matrix will be 8.

In Examples 4 and 5, we give two examples of involutory 16×16 binary matrices satisfying the restrictions earlier. Also, whereas each row of the 16×16 binary matrix in Example 4 has the Hamming weight equal to 7, each row of the 16×16 binary matrix in Example 5 has the Hamming weight equal to 11.

Example 4.

Let $M = Had(1_h, 5_h, 8_h, D_h) = \begin{bmatrix} 1_h & 5_h & 8_h & D_h \\ 5_h & 1_h & D_h & 8_h \\ 8_h & D_h & 1_h & 5_h \\ D_h & 8_h & 5_h & 1_h \end{bmatrix}$ be involutory MDS and the 4×4 Hadamard type matrix. Using the same procedure given in Example 2, we can transform the 4×4 matrix earlier into the 16×16 binary linear transformation as shown next:

Security Comm. Networks 2014; **7**:53–63 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \\ z_9 \\ z_{10} \\ z_{11} \\ z_{12} \\ z_{13} \\ z_{14} \\ z_{15} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \\ f_{12} \\ f_{13} \\ f_{14} \\ f_{15} \end{bmatrix},$$

where $(f_{15}, f_{14}, \ldots, f_0) \in GF(2)$ denotes the input bits and $(z_{15}, z_{14}, \ldots, z_0) \in GF(2)$ denotes the output bits.

Example 5.

Let $M = Had(B_h, E_h, 7_h, 3_h) = \begin{bmatrix} B_h & E_h & 7_h & 3_h \\ E_h & B_h & 3_h & 7_h \\ 7_h & 3_h & B_h & E_h \\ 3_h & 7_h & E_h & B_h \end{bmatrix}$ be involutory MDS and 4×4 Hadamard-type matrix. Using the same procedure given in Example 4, we can transform the 4×4 matrix earlier into the 16×16 binary linear transformation in which each row has the Hamming weight equal to 11. The obtained linear transformation is shown next:

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

By the special permutation 1 given in Section 3, it is possible to obtain new matrices by cyclically rotating l bits, where $l \in \{1, \ldots, 15\}$, to the right of all rows of a 16×16 binary linear transformation. For example, if we rotate cyclically 1 bit to the right of all rows of the binary linear transformations given in Examples 4 and 5, we obtain non-involutory binary linear transformations having 2^{16} and 2^{24} fixed points, respectively, with branch number 8, when these linear transformations map a 128-bit input to a 128-bit output. The reason is that the rank of $A_{(1)} - I$ matrices of them increases from 8 to 14 and 8 to 13, respectively. Note also that these transformations can process a 128-bit block when each input element is in $GF(2^8)$.

8. 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 type, circulant type, and random type matrices, which can be transformed into the binary form with branch number 8.

For Hadamard type matrices, by using Lemma 3, we show the obtained involutory matrices with the elements from $GF(2^4)$ defined by the primitive polynomial $x^4 + x + 1$ in Appendix D. In Appendix D, each matrix represents one member (representative) of 24 involutory matrices because there are 24 (4!) permutations of any

$Had(x, y, z, t)$, where $x, y, z, t \in GF(2^4)$. We have noticed that all 23 more binary matrices transformed from these permuted elements of 4×4 Hadamard matrices have also branch number of 8. As a result, 576 (24×24) involutory matrices can be generated with branch number 8, and the rank value of $A_{Binary} - I$ matrices of all involutory binary matrices is computed as 8. Moreover, we have searched for all possible 4×4 nonsingular matrices with elements from $GF(2^4)$ defined by $x^4 + x^3 + 1$ and $x^4 + x^3 + x^2 + x + 1$, and found respectively 576 and 672 involutory matrices in the same manner with branch number 8. When combining all results with applying the special permutations 1 and 2 together, given in Section 3, we can determine 29,184 (1824×16) involutory matrices and totally 466,944 (1824×256) both involutory and non-involutory binary matrices with branch number 8 because it is possible to generate 255 more binary matrices, where 15 out of 255 matrices are involutory matrices, from one binary matrix.

For circulant type matrices, on the other hand, we have obtained respectively 96, 96, and 64 non-involutory binary matrices with branch number 8 by transforming 4×4 nonsingular matrices with elements from $GF(2^4)$ defined by $x^4 + x + 1$, $x^4 + x^3 + 1$ and $x^4 + x^3 + x^2 + x + 1$. The rank value of $A_{Binary} - I$ matrices of all non-involutory binary matrices is computed as 12. Moreover, we can increase the number of cryptographically good binary linear transformations by applying any row and column permutation to each of them. With the application special permutations 1 and 2 together; for example, we can generate 65,536 (256×256) binary linear transformations with branch number 8. In Appendix E, we give the list of all 4×4 circulant matrices, which can be transformed into 16×16 non-involutory binary linear transformations with branch number 8, with the elements from $GF(2^4)$ defined by the primitive polynomial $x^4 + x + 1$. Notice that, in Appendix E, the determinants of all circulant matrices are equal to 1 because XOR sum of elements in each row of any circulant matrix is equal to 1. That means they are all nonsingular but not involutory matrices.

For random type matrices, by using Lemma 4, we have obtained several involutory matrices with branch number 8 by transforming 4×4 nonsingular matrices with elements from $GF(2^4)$ defined by $x^4 + x + 1$ and $x^4 + x^3 + 1$. For irreducible polynomial $x^4 + x^3 + x^2 + x + 1$, we could not find any involutory matrix with branch number 8. In addition, we have noticed that these several involutory matrices are also in the list obtained by Hadamard construction. Therefore, in order to provide more accurate information for the productivity of the method, we ignore them.

Above all, we have computed the best rank value of $A_{Binary} - I$ matrices of all obtained matrices as 15, and therefore, if we use a binary linear transformation with this rank value to map a 128-bit input to a 128-bit output value, the linear transformation will only include 2^8 fixed points.

9. SOFTWARE IMPLEMENTATIONS OF THE PROPOSED BINARY LINEAR TRANSFORMATIONS

Because the proposed linear transformations are binary transformations with the input elements of $GF(2^8)$ to be used for processing a 64-bit block or a 128-bit block, the implementation of the binary linear transformations is only based on XOR operations. For example, to implement the binary linear transformation given in Example 4 needs 96 XORs in a straight coding, but it is possible to reduce the needed number of XOR operations by using additional variables. On the other hand, the binary linear transformation given in Example 5 needs 160 XORs to be implemented. But, adding one more variable to the implementation will be enough to reduce the needed number of XOR operations to 95.

10. CONCLUSIONS

In this paper, we have proposed a new algebraic construction method for obtaining cryptographically good binary linear transformations. When constructing these binary linear transformations, we focused on two important cryptographic criteria: the branch number and the number of fixed points. We have obtained 8×8 binary linear transformations with maximum branch number 5 and one fixed point. Moreover, the method for constructing 8×8 involutional binary linear transformations has been given. If the construction of 16×16 binary linear transformations is concerned, we have obtained involutional linear transformations with maximum branch number 8 and 2^{64} fixed points, and therefore, the number of fixed points is reduced by a factor 2^8 than that of the ARIA. On the other hand, by using the proposed method, we have also shown in Section 8 that it is possible to generate 29,184 involutory binary matrices with branch number 8.

The constructed binary linear transformations are resistant against linear and differential cryptanalysis because they are designed to have the optimal branch number value. 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 linear transformations and then the use of these transformations is recommended.

On the other hand, as shown in Section 7, we have used MDS matrices with elements in $GF(2^4)$ for constructing involutory 16×16 binary matrices. But, we have also found non-MDS involutory matrices like $Had(1_h, C_h, 5_h, 9_h)$, which can be transformed into the binary matrix of branch number 8. Therefore, in Appendix D and E, the list of 4×4 matrices is given by considering the property of being MDS or non-MDS. In addition, roughly speaking, the decrease in the number of fixed points for a linear transformation is provided by non-involutory binary matrices. We have found many binary linear transformations, which are not an involution and have the branch number of 8 with considerably reduced fixed points. This idea is also supported by the given examples.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous referees for their valuable comments and are supported by the project TÜBAP (Trakya Üniversitesi Bilimsel Araştırma Projeleri) 2011/163.

REFERENCES

1. Keliher L. Linear cryptanalysis of substitution-permutation networks. *Ph.D. Thesis*, Queen's University, Kingston, Ontario, Canada, 2003.
2. US National Institute of Standards and Technology, Data Encryption Standard. Federal Information Processing Standards Publications, No. 46–3, 1999.
3. US National Institute of Standards and Technology, Advanced Encryption Standard. Federal Information Processing Standards Publications, No. 197, 2001.
4. Aslan FY, Sakallı MT, Aslan B, Bulut S. A new involutory 4×4 MDS matrix for the AES-like block ciphers. *International Review on Computers and Software* 2011; **6**(1):96–103.
5. Kwon D, Sung SH, Song JH, Park S. Design of block ciphers and coding theory. *Trends in Mathematics* 2005; **8**(1):13–20.
6. Nyberg K. Differentially uniform mappings for cryptography. In *Proceedings of EUROCRYPT 93*, Lecture Notes in Computer Science, Vol. **765**, Springer-Verlag, 1994; 55–64.
7. Kavut S, Yücel MD. On Some Cryptographic Properties of Rijndael. In *Proceedings of Information Assurance in Computer Networks, Methods, Models and Architectures for Network Security*, Lecture Notes in Computer Science, Vol. **2052**, Springer-Verlag, 2001; 300–311.
8. Aslan B, Sakallı MT, Buluş E. Classifying 8-bit to 8-bit S-boxes based on power mappings from the point of DDT and LAT distributions. In *Proceedings of International Workshop on the Arithmetic of Finite Fields*, Lecture Notes in Computer Science, Vol. **5130**, Springer-Verlag, 2008; 123–133.
9. Junod P. Statistical cryptanalysis of block ciphers. *Ph.D. Thesis*, EPFL, Lausanne, Switzerland, 2005.
10. Aoki K, Ichikawa T, Kanda M, *et al*. 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, Vol. **2012**, Springer-Verlag, 2001; 39–56.
11. Kwon D, Kim J, Park S, *et al*. New block cipher: ARIA. In *Proceedings of International Conference on Information Security and Cryptology*, Lecture Notes in Computer Science, Vol. **2971**, Springer-Verlag, 2004; 432–445.
12. Shannon CE. Communication theory of secrecy. *Bell System Technical Journal* 1949; **28**(7):656–715.
13. Z'aba MR. Analysis of linear relationships in block ciphers. *Ph.D. Thesis*, Queensland University of Technology, Brisbane, Australia, 2010.
14. Feistel H. Cryptography and computer privacy. *Scientific American* 1973; **228**(5):15–23.
15. Webster AF, Tavares SE. On the design of S-boxes. In *Proceedings of CRYPTO'85*. Lecture Notes in Computer Science, Vol. **218**. Springer-Verlag: Santa Barbara, California, USA, 1986; 523–534.
16. Kam JB, Davida GI. Structured design of substitution-permutation encryption networks. *IEEE Transactions on Computers* 1979; **28**(10):747–753.
17. Daemen J, Rijmen V. *The Design of Rijndael, AES — The Advanced Encryption Standard*. Springer-Verlag, 2002.
18. Matsui M. Linear cryptanalysis method for DES cipher. In *Proceedings of EUROCRYPT 93*. Lecture Notes in Computer Science, Vol. **765**. Springer-Verlag: Lofthus, Norway, 1994; 386–397.
19. Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. In *Proceedings of CRYPTO'90*. Lecture Notes in Computer Science, Vol. **537**. Springer-Verlag: Santa Barbara, California, USA, 1990; 2–21.
20. Barreto PSLM, Rijmen V. The Khazad legacy-level block cipher. In *Proceedings First open NESSIE Workshop*: Leuven, 2000.
21. Nakahara J, Abrahão E. A new involutory MDS matrix for the AES. *International Journal of Network Security* 2009; **9**(2):109–116.
22. McEliece RJ. Finite Fields for Computer Scientists and Engineers. *Kluwer Academic Publishers*, Dordrecht, 1987.
23. Lidl R, Niederreiter H. *Finite Fields, Encyclopedia of Mathematics and its Applications*. Addison-Wesley: Reading, MA, 1983.
24. Youssef AM, Mister S, Tavares SE. On the design of linear transformation for substitution permutation encryption networks. In *Proceedings of Selected Areas in Cryptography (SAC'97)*. Ottawa, Ontario, Canada, 1997; 40–48.
25. Junod P, Vaudenay S. Perfect diffusion primitives for block ciphers. In *Proceedings of Selected Areas in Cryptology (SAC 2004)*, Lecture Notes in Computer Science, Vol. **3357**, Springer-Verlag, 2004; 84–99.
26. Koo BW, Jang HS, Song JH. 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, Vol. **2908**. Springer-Verlag: Jeju Island, Korea, 2003; 489–503.
27. Grinstead CM, Snell JL. Introduction to probability. American Mathematical Society, 2nd Edition, 1997.

28. Bogdanov A, Knudsen LR, Leander G, *et al*. 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, Vol. **4727**, Springer-Verlag, 2007; 450–466.
29. Biham E, Anderson R, Knudsen LR. Serpent: a new block cipher proposal. In *Proceedings of 5th International Workshop of Fast Software Encryption-FSE'98*. Lecture Notes in Computer Science, Vol. **1372**. Springer-Verlag: Paris, France, 1998; 222–238.
30. Koo BW, Jang HS, Song JH. On constructing of a 32×32 binary matrix as a diffusion layer for a 256-bit block cipher. In *Proceedings of International Conference on Information Security and Cryptology*. Lecture Notes in Computer Science, Vol. **4296**. Springer-Verlag: Busan, Korea, 2006; 51–64.

APPENDIX A

The inverse of the given matrix in Example 2

$$A^{-1} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

APPENDIX B

The inverse of the given matrix in Example 3

$$A_{(1)}^{-1} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

APPENDIX C

The list of all 2×2 matrices for the $x^4 + x + 1$, which can be transformed into 8×8 involutory and non-involutory binary matrices with branch number 5. Note that $\diamond$ represents the given 2×2 matrix, which is involutory, and * represents the given 2×2 matrix, which is non-involutory.

(1) $(3_h, 5_h, A_h, 3_h)^{\diamond}$	(13) $(5_h, 3_h, 3_h, A_h)^{*}$
(2) $(3_h, A_h, 5_h, 3_h)^{\diamond}$	(14) $(A_h, 3_h, 3_h, 5_h)^{*}$
(3) $(3_h, 6_h, F_h, 3_h)^{\diamond}$	(15) $(6_h, 3_h, 3_h, F_h)^{*}$
(4) $(3_h, F_h, 6_h, 3_h)^{\diamond}$	(16) $(F_h, 3_h, 3_h, 6_h)^{*}$
(5) $(E_h, 2_h, 5_h, E_h)^{\diamond}$	(17) $(7_h, 9_h, 9_h, 7_h)^{*}$
(6) $(E_h, 5_h, 2_h, E_h)^{\diamond}$	(18) $(9_h, 7_h, 7_h, 9_h)^{*}$
(7) $(E_h, 3_h, 6_h, E_h)^{\diamond}$	(19) $(7_h, C_h, C_h, 7_h)^{*}$
(8) $(E_h, 6_h, 3_h, E_h)^{\diamond}$	(20) $(C_h, 7_h, 7_h, C_h)^{*}$
(9) $(2_h, E_h, E_h, 5_h)^{*}$	(21) $(8_h, B_h, B_h, 8_h)^{*}$
(10) $(5_h, E_h, E_h, 2_h)^{*}$	(22) $(B_h, 8_h, 8_h, B_h)^{*}$
(11) $(3_h, E_h, E_h, 6_h)^{*}$	(23) $(B_h, E_h, E_h, B_h)^{*}$
(12) $(6_h, E_h, E_h, 3_h)^{*}$	(24) $(E_h, B_h, B_h, E_h)^{*}$

(*Continues*)

Table . (Continued)

APPENDIX D

The list of all representatives of 4×4 Hadamard matrices for the $x^4 + x + 1$, which can be transformed into 16×16 involutory binary matrices with branch number 8. Note that, in Appendices D and E, $\star$ represents the given 4×4 matrix, which is MDS, and † represents the given 4×4 matrix, which is not MDS.

(1) $Had(1_h, 2_h, C_h, E_h)^{\star}$	(13) $Had(3_h, 4_h, B_h, D_h)^{\star}$
(2) $Had(1_h, 3_h, 5_h, 6_h)^{\star}$	(14) $Had(3_h, 5_h, 9_h, E_h)^{\star}$
(3) $Had(1_h, 3_h, C_h, F_h)^{\star}$	(15) $Had(3_h, 5_h, A_h, D_h)^{\dagger}$
(4) $Had(1_h, 3_h, D_h, E_h)^{\star}$	(16) $Had(3_h, 7_h, 8_h, D_h)^{\star}$
(5) $Had(1_h, 5_h, 8_h, D_h)^{\star}$	(17) $Had(3_h, 7_h, A_h, F_h)^{\star}$
(6) $Had(1_h, 5_h, 9_h, C_h)^{\dagger}$	(18) $Had(3_h, 7_h, B_h, E_h)^{\star}$
(7) $Had(1_h, 5_h, B_h, E_h)^{\star}$	(19) $Had(4_h, 6_h, C_h, F_h)^{\star}$
(8) $Had(1_h, 7_h, B_h, C_h)^{\star}$	(20) $Had(4_h, 7_h, C_h, E_h)^{\star}$
(9) $Had(2_h, 4_h, 8_h, F_h)^{\star}$	(21) $Had(4_h, 7_h, D_h, F_h)^{\star}$
(10) $Had(2_h, 4_h, A_h, D_h)^{\star}$	(22) $Had(5_h, 7_h, C_h, F_h)^{\dagger}$
(11) $Had(2_h, 6_h, 8_h, D_h)^{\star}$	(23) $Had(9_h, A_h, D_h, F_h)^{\star}$
(12) $Had(3_h, 4_h, A_h, C_h)^{\star}$	(24) $Had(9_h, B_h, C_h, F_h)^{\star}$

APPENDIX E

The list of all 4×4 circulant matrices for the $x^4 + x + 1$, which can be transformed into 16×16 non-involutory binary matrices with branch number 8.

(1) circ $(1_h, 1_h, A_h, B_h)\star$	(49) circ $(3_h, 5_h, A_h, D_h)$ †
(2) circ $(1_h, 1_h, B_h, A_h)\star$	(50) circ $(3_h, D_h, A_h, 5_h)$ †
(3) circ $(1_h, A_h, B_h, 1_h)\star$	(51) circ $(5_h, 3_h, D_h, A_h)$ †
(4) circ $(1_h, B_h, A_h, 1_h)\star$	(52) circ $(5_h, A_h, D_h, 3_h)$ †
(5) circ $(A_h, 1_h, 1_h, B_h)\star$	(53) circ $(A_h, 5_h, 3_h, D_h)$ †
(6) circ $(A_h, B_h, 1_h, 1_h)\star$	(54) circ $(A_h, D_h, 3_h, 5_h)$ †
(7) circ $(B_h, 1_h, 1_h, A_h)\star$	(55) circ $(D_h, 3_h, 5_h, A_h)$ †
(8) circ $(B_h, A_h, 1_h, 1_h)\star$	(56) circ $(D_h, A_h, 5_h, 3_h)$ †
(9) circ $(1_h, 2_h, E_h, C_h)$ †	(57) circ $(3_h, B_h, 4_h, D_h)\star$
(10) circ $(1_h, C_h, E_h, 2_h)$ †	(58) circ $(3_h, D_h, 4_h, B_h)\star$
(11) circ $(2_h, 1_h, C_h, E_h)$ †	(59) circ $(4_h, B_h, 3_h, D_h)\star$
(12) circ $(2_h, E_h, C_h, 1_h)$ †	(60) circ $(4_h, D_h, 3_h, B_h)\star$
(13) circ $(C_h, 1_h, 2_h, E_h)$ †	(61) circ $(B_h, 3_h, D_h, 4_h)\star$
(14) circ $(C_h, E_h, 2_h, 1_h)$ †	(62) circ $(B_h, 4_h, D_h, 3_h)\star$
(15) circ $(E_h, 2_h, 1_h, C_h)$ †	(63) circ $(D_h, 3_h, B_h, 4_h)\star$
(16) circ $(E_h, C_h, 1_h, 2_h)$ †	(64) circ $(D_h, 4_h, B_h, 3_h)\star$
(17) circ $(2_h, 4_h, D_h, A_h)$ †	(65) circ $(4_h, 6_h, E_h, D_h)\star$
(18) circ $(2_h, A_h, D_h, 4_h)$ †	(66) circ $(4_h, D_h, E_h, 6_h)\star$
(19) circ $(4_h, 2_h, A_h, D_h)$ †	(67) circ $(6_h, 4_h, D_h, E_h)\star$
(20) circ $(4_h, D_h, A_h, 2_h)$ †	(68) circ $(6_h, E_h, D_h, 4_h)\star$
(21) circ $(A_h, 2_h, 4_h, D_h)$ †	(69) circ $(D_h, 4_h, 6_h, E_h)\star$
(22) circ $(A_h, D_h, 4_h, 2_h)$ †	(70) circ $(D_h, E_h, 6_h, 4_h)\star$

(*Continues*)

Table . (Continued)

(23) circ $(D_h, 4_h, 2_h, A_h)$ †	(71) circ $(E_h, 6_h, 4_h, D_h)$ ⋆
(24) circ $(D_h, A_h, 2_h, 4_h)$ †	(72) circ $(E_h, D_h, 4_h, 6_h)$ ⋆
(25) circ $(2_h, 7_h, E_h, A_h)$ †	(73) circ $(4_h, 6_h, F_h, C_h)$ ⋆
(26) circ $(2_h, A_h, E_h, 7_h)$ †	(74) circ $(4_h, C_h, 6_h, F_h)$ †
(27) circ $(7_h, 2_h, A_h, E_h)$ †	(75) circ $(4_h, C_h, F_h, 6_h)$ ⋆
(28) circ $(7_h, E_h, A_h, 2_h)$ †	(76) circ $(4_h, F_h, 6_h, C_h)$ †
(29) circ $(A_h, 2_h, 7_h, E_h)$ †	(77) circ $(6_h, 4_h, C_h, F_h)$ ⋆
(30) circ $(A_h, E_h, 7_h, 2_h)$ †	(78) circ $(6_h, C_h, 4_h, F_h)$ †
(31) circ $(E_h, 7_h, 2_h, A_h)$ †	(79) circ $(6_h, F_h, 4_h, C_h)$ †
(32) circ $(E_h, A_h, 2_h, 7_h)$ †	(80) circ $(6_h, F_h, C_h, 4_h)$ ⋆
(33) circ $(2_h, A_h, 5_h, C_h)$ ⋆	(81) circ $(C_h, 4_h, 6_h, F_h)$ ⋆
(34) circ $(2_h, C_h, 5_h, A_h)$ ⋆	(82) circ $(C_h, 4_h, F_h, 6_h)$ †
(35) circ $(5_h, A_h, 2_h, C_h)$ ⋆	(83) circ $(C_h, 6_h, F_h, 4_h)$ †

(*Continues*)

Table . (Continued)

(36) circ $(5_h, C_h, 2_h, A_h)$ ⋆	(84) circ $(C_h, F_h, 6_h, 4_h)$ ⋆
(37) circ $(A_h, 2_h, C_h, 5_h)$ ⋆	(85) circ $(F_h, 4_h, C_h, 6_h)$ †
(38) circ $(A_h, 5_h, C_h, 2_h)$ ⋆	(86) circ $(F_h, 6_h, 4_h, C_h)$ ⋆
(39) circ $(C_h, 2_h, A_h, 5_h)$ ⋆	(87) circ $(F_h, 6_h, C_h, 4_h)$ †
(40) circ $(C_h, 5_h, A_h, 2_h)$ ⋆	(88) circ $(F_h, C_h, 4_h, 6_h)$ ⋆
(41) circ $(3_h, 3_h, 8_h, 9_h)$ ⋆	(89) circ $(5_h, 7_h, C_h, F_h)$ †
(42) circ $(3_h, 3_h, 9_h, 8_h)$ ⋆	(90) circ $(5_h, F_h, C_h, 7_h)$ †
(43) circ $(3_h, 8_h, 9_h, 3_h)$ ⋆	(91) circ $(7_h, 5_h, F_h, C_h)$ †
(44) circ $(3_h, 9_h, 8_h, 3_h)$ ⋆	(92) circ $(7_h, C_h, F_h, 5_h)$ †
(45) circ $(8_h, 3_h, 3_h, 9_h)$ ⋆	(93) circ $(C_h, 7_h, 5_h, F_h)$ †
(46) circ $(8_h, 9_h, 3_h, 3_h)$ ⋆	(94) circ $(C_h, F_h, 5_h, 7_h)$ †
(47) circ $(9_h, 3_h, 3_h, 8_h)$ ⋆	(95) circ $(F_h, 5_h, 7_h, C_h)$ †
(48) circ $(9_h, 8_h, 3_h, 3_h)$ ⋆	(96) circ $(F_h, C_h, 7_h, 5_h)$ †