Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 540253, 12 pages
http://dx.doi.org/10.1155/2014/540253

Research Article

On the Construction of 20×20 and 24×24 Binary Matrices with Good Implementation Properties for Lightweight Block Ciphers and Hash Functions

**Muharrem Tolga Sakallı,[1] Sedat Akleylek,[2,3] Bora Aslan,[4]
Ercan Buluş,[5] and Fatma Büyüksaraçoğlu Sakallı[1]**

[1] *Department of Computer Engineering, Trakya University, 22030 Edirne, Turkey*
[2] *Department of Computer Engineering, Ondokuz Mayis University, 55139 Samsun, Turkey*
[3] *Institute of Applied Mathematics, Middle East Technical University, 06531 Ankara, Turkey*
[4] *Software Engineering Department, Kirklareli University, 39000 Kırklareli, Turkey*
[5] *Department of Computer Engineering, Namık Kemal University, 59860 Çorlu, Turkey*

Correspondence should be addressed to Sedat Akleylek; akleylek@gmail.com

Received 16 June 2014; Revised 9 October 2014; Accepted 13 October 2014; Published 2 November 2014

Academic Editor: Kwok-Wo Wong

We present an algebraic construction based on state transform matrix (companion matrix) for $n \times n$ (where $n \neq 2^k$, k being a positive integer) binary matrices with high branch number and low number of fixed points. We also provide examples for 20×20 and 24×24 binary matrices having advantages on implementation issues in lightweight block ciphers and hash functions. The powers of the companion matrix for an irreducible polynomial over GF(2) with degree 5 and 4 are used in finite field Hadamard or circulant manner to construct 20×20 and 24×24 binary matrices, respectively. Moreover, the binary matrices are constructed to have good software and hardware implementation properties. To the best of our knowledge, this is the first study for $n \times n$ (where $n \neq 2^k$, k being a positive integer) binary matrices with high branch number and low number of fixed points.

1. Introduction

Modern block ciphers are made of several rounds. Each of these consists of confusion and diffusion layers. Confusion and diffusion are two principles of the operation of a secure cipher as identified by Shannon [1]. Many block ciphers use linear transformations together with nonlinear substitution boxes (S-boxes) to implement Shannon's principles. In addition, many block ciphers use S-boxes based on the inversion mapping in a finite field [2, 3]. In a block cipher, a linear transformation is employed to provide the required diffusion. The linear transformation guarantees all the output bits to depend on all the input bits after few rounds. The substitution layer or nonlinear layer provides the necessary confusion making this dependency complex and nonlinear [4]. A linear transformation provides diffusion by mixing bits of the fixed size input block to produce the corresponding output block of the same size [5]. The two existing techniques of measuring diffusion for linear transformations are the branch number [6] and the number of fixed points [5]. The branch number denotes the minimum number of active S-boxes for any two consecutive rounds and represents diffusion rate and measures security against linear and differential cryptanalysis. To achieve better diffusion property, many modern ciphers use linear transformations with high branch number. On the other hand, 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 at these points are left intact by the linear transformation. Note that the expected number of fixed points in a random linear transformation is one [5].

Many block ciphers use maximum distance separable (MDS) and maximum distance binary linear (MDBL) codes as diffusion layers in their round function. The AES [7] and Khazad [8] use MDS codes; the Camellia [9] and ARIA [10] use MDBL codes. It is known that MDS matrices do not give a compact implementation in hardware, for example, AES. Most diffusion layers are linear transformations having matrix representations over $GF(2^m)$ or $GF(2)$. The binary matrices, having matrix representation over GF(2), are employed as diffusion layers in block ciphers like Camellia and ARIA. 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 [11]. Furthermore, the 8×8 and 16×16 binary matrices used in Camellia and ARIA have the maximum branch numbers 5 and 8, respectively, and are therefore called MDBL codes [4]. In [12, 13], an algebraic construction method to generate 8×8, 16×16, and 32×32 binary matrices of maximum branch number was given. There is no general method for $n \times n$ binary matrices where $n \neq 2^k$, k being a positive integer. Constructing diffusion layers with high branch numbers, low number of fixed points, and low-cost hardware/software implementations is an open problem for lightweight block ciphers and hash functions.

In recent years, lightweight cryptography has attracted a lot of attention from the crypto community since the use of resource constraint devices has been increasing. There are several lightweight block cipher constructions with 80-bit and 96-bit block sizes in the literature [14–17]. However, these proposals neglect important real-world constraints except a small chip area and they have different deficiencies as listed below:

(i) the lack of efficiency on low-cost processors,

(ii) a vast amount of program memory storage,

(iii) high execution times due to the high number of rounds,

(iv) the lack of security assessment in detail.

The wide-trail strategy is one of the important approaches to design round transformations of block ciphers that combine efficiency and resistance against linear and differential cryptanalysis. It results in simple and strong security arguments. However, this approach does not help in designing efficient diffusion layers (with a suitable number of active S-boxes). In this respect, the diffusion layers constructed and the method given in this study aim to provide an alternative structure for the block ciphers with input size different than 2^k.

In this study, an algebraic method based on state transform matrix (companion matrix) to construct binary matrices with good implementation properties for lightweight block ciphers and hash functions is given. The emphasis is given to $n \times n$ binary matrices where $n \neq 2^k$ and k is a positive integer. The proposed method can also be considered as a generalization and different interpretation of the methods given in [12, 13] since it works for any n. This method uses 4×4 finite field Hadamard (FFHadamard) matrices with the powers of the companion matrix for an irreducible polynomial over GF(2) of degree 5 and 6×6 circulant matrices with the powers of the companion matrix for an irreducible polynomial over GF(2) of degree 4 to generate 20×20 (involutory and noninvolutory) and 24×24 binary matrices (noninvolutory) of branch numbers 8 and 10 with low number of fixed points, respectively. Also, the binary matrices are constructed to have suitable software and hardware implementation properties for lightweight block ciphers. Note that the binary matrices with these sizes have not been studied in the literature well enough, which may allow us to design a lightweight block cipher with 80-bit and 96-bit block sizes if these matrices are used with 4-bit S-boxes.

This paper is organized as follows: Section 2 describes the required mathematical background and an introduction to the proposed method. In Section 3, the proposed method is given and examples are provided with good cryptographic properties. Security assessment of lightweight block cipher using the proposed diffusion layer is analyzed in Section 4. Conclusion is given in Section 5. In Appendices A, B, and C implementation details of given examples are discussed.

2. Preliminaries

In this section, we give the mathematical background and a view of the proposed method.

Let $GF(2^m) \cong GF(2)/\langle p(x)\rangle$, where $p(x) = a_m x^m + \cdots + a_1 x + a_0$ is an irreducible polynomial over GF(2) with degree m. Let C_m be the companion matrix for the irreducible polynomial over GF(2) with degree m. The powers of C_m can be considered as the nonzero elements of $GF(2^m)$ [18, 19]. Then, the matrix C_m can be viewed as a polynomial, that is, $M : x, M^2 : x^2, \ldots$. This is the core part of the proposed method. Note that this multiplication is modulo $p(x)$ and rank of these matrices is the extension degree m. The identity matrix can be obtained by $C_m^{2^m-1}$

$$C_m = \begin{pmatrix} 0 & 0 & \cdots & 0 & a_0 \\ 1 & 0 & \cdots & 0 & a_1 \\ 0 & 1 & \cdots & 0 & a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & a_{m-1} \end{pmatrix}. \tag{1}$$

In this study we focus on the finite fields $GF(2^4)$ and $GF(2^5)$, where the irreducible polynomials over GF(2) are, respectively, x^4+x+1 and x^5+x^2+1. Now we give an example on how to obtain the elements of $GF(2^4)$.

Example 1. Let $GF(2^4) \cong GF(2)/\langle p(x)\rangle$, where $p(x) = x^4 + x + 1$ is the irreducible polynomial over GF(2). Then,

$$C_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, C_4^2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \ldots, C_4^{15} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2)$$

6×6 matrices with the elements of C_4^i, where $1 \leq i \leq 2^4 - 1$, can be transformed to 24×24 binary matrices by substituting the powers of C_4 with their corresponding 4×4 binary matrices. Similarly, 4×4 matrices with the elements of C_5^i, where $1 \leq i \leq 2^5 - 1$, can be transformed to 20×20 binary matrices by substituting the powers of C_5 with their corresponding 5×5 binary matrices.

Now we recall some facts on the linear transformations. The linear transformations of diffusion layers used in most block ciphers are represented as matrices. Hence, a linear transformation $A : (\{0,1\}^m)^n \mapsto (\{0,1\}^m)^n$ can be defined as follows:

$$A(x) = A \cdot x^T = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad (3)$$

where $x = (x_1, x_2, \ldots, x_n)^T$ and $x_i \in \{0,1\}^m$, $i = 1, \ldots, n$. Also n represents the number of S-boxes in a diffusion layer A, where the size of each input and output is m-bit [4].

Definition 2 (see [6]). The differential and linear branch numbers of an $n \times n$ matrix $A : (\{0,1\}^m)^n \mapsto (\{0,1\}^m)^n$ are defined by

$$B_d(A) = \min\left\{wt(x) + wt\left(A \cdot x^T\right) \mid x \in \left(\{0,1\}^m\right)^n - \{0\}\right\},$$
$$B_l(A) = \min\left\{wt(x) + wt\left(A^T \cdot x^T\right) \mid x \in \left(\{0,1\}^m\right)^n - \{0\}\right\}, \quad (4)$$

where $wt(x)$ is the number of nonzero components in x, respectively.

Definition 3. Let n be a power of 2. An $n \times n$ finite field Hadamard (FFHadamard) matrix with the elements of $GF(2^m)$ can be given as follows:

$$\operatorname{had}(a_1, a_2, \ldots, a_n) = \begin{pmatrix} a_1 & a_2 & \cdots & a_{n-1} & a_n \\ a_2 & a_1 & \cdots & a_n & a_{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_n & a_{n-1} & \cdots & a_2 & a_1 \end{pmatrix}. \quad (5)$$

Remark 4. Note that one can also divide the FFHadamard matrix into the submatrices. For example, for 4×4 FFHadamard matrix, we have $\operatorname{had}(a_1, a_2, a_3, a_4) = \left(\begin{smallmatrix} A & B \\ B & A \end{smallmatrix}\right)$, where $A = \left(\begin{smallmatrix} a_1 & a_2 \\ a_2 & a_1 \end{smallmatrix}\right)$ and $B = \left(\begin{smallmatrix} a_3 & a_4 \\ a_4 & a_3 \end{smallmatrix}\right)$. The matrices A and B have Toeplitz matrix properties. We use this observation while constructing a diffusion layer.

Definition 5. An $n \times n$ circulant matrix with the elements of $GF(2^m)$ can be given as follows:

$$\operatorname{circ}(a_1, a_2, \ldots, a_n) = \begin{pmatrix} a_1 & a_2 & \cdots & a_{n-1} & a_n \\ a_n & a_1 & \cdots & a_{n-2} & a_{n-1} \\ a_{n-1} & a_n & \cdots & a_{n-3} & a_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_2 & a_3 & \cdots & a_n & a_1 \end{pmatrix}. \quad (6)$$

Note that Remark 4 is also applicable in this case. In Lemma 6, the construction of involutory 4×4 FFHadamard matrix is given.

Lemma 6. *Let A be a 4×4 FFHadamard matrix with distinct elements of $GF(2^m) - \{0\}$. Then A is involutory if and only if $\sum_{i=1}^{4} a_i = 1$.*

Proof. The identity matrix satisfies $\sum_{i=1}^{4} a_i^2 = 1$ and $\sum_{i=1}^{4} a_i = 1$. Since A is unitary ($A^{-1} = A$) and symmetric ($A = A^T$), the matrix A is involutory:

$$A^2 = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{pmatrix} \cdot \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{pmatrix}$$
$$= \begin{pmatrix} \sum_{i=1}^{4} a_i^2 & 0 & 0 & 0 \\ 0 & \sum_{i=1}^{4} a_i^2 & 0 & 0 \\ 0 & 0 & \sum_{i=1}^{4} a_i^2 & 0 \\ 0 & 0 & 0 & \sum_{i=1}^{4} a_i^2 \end{pmatrix}. \quad (7)$$

□

In this study, 20×20 binary matrices are constructed by using 4×4 FFHadamard matrices with the elements of $GF(2^5)$ and also 24×24 noninvolutory binary matrices are constructed by using 6×6 matrices with the elements of $GF(2^4)$. The 20×20 binary matrices constructed are both involutory and noninvolutory with minimum number of fixed points. Involutory transformations can make the decryption process the same as the encryption process. Thus the encryption and decryption can be implemented by the same module and with equal speeds. However, noninvolutory transformations constructed in this study are aimed at having close encryption and decryption speeds. An input block is a fixed point of a transformation if the input block equals its output block. Clearly, in this context, there is no diffusion at the fixed points since the input blocks at these points are left intact by the linear transformation. Therefore, if the number of fixed points in a linear transformation greatly exceeds the expected number for a random linear transformation, then this is an indication of poor diffusion of the linear transformation. Note that the expected number of fixed points in a random linear transformation is one [5]. Consider an input block to a linear transformation formed by m-bit values in the field $\mathrm{GF}(2^m)$ and let the linear transformation matrix be an $n \times n$ matrix $A = (a_{ij})_{n\times n}$, where $a_{ij} \in \mathrm{GF}(2)$ or $a_{ij} \in \mathrm{GF}(2^m)$ and I is 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: $(A+I)\cdot x^T = 0$, where 0 is the all zero vector of length n. Hence, the number of fixed points can be given as

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

It is obvious that if the matrix $(A + I)$ has bigger rank, the matrix A has lower number of fixed points.

Remark 7. The existence of fixed points in the round function of block ciphers is used as the basis for some cryptographic attacks and these attacks use fixed points that exist across one or more rounds [5]. The block ciphers DES, SAFER K, Blowfish, GOST, DEAL, and KeeLog were previously found vulnerable to attacks based on the existence of fixed points [20–23]. For SPN ciphers, the existence of fixed points in a linear transformation hints at the presence of 1-round self-iterating differential characteristic. It should be also noted that not all fixed points are useful in constructing a self-iterating characteristic. The usefulness of a fixed point, in this case, depends on its interaction with the subsequent nonlinear transformation. If the input difference is a fixed point, then the linear transformation will replicate this difference into the same S-boxes in the next round. In this context, when designing a block cipher, the linear transformation should be considered with the S-boxes and self-iterating characteristics should be searched. The designer should decide on the number of rounds of the block cipher according to some further investigations (e.g., the resistance of the linear transformation against other attacks like impossible differential cryptanalysis and truncated differential cryptanalysis). To ensure that the large number of fixed points does not trigger an attack to the cipher where the construction is used as a building block depends on the cipher. What we expect is that the cipher itself should behave like a random permutation. Therefore, if the cipher itself does not have many fixed points then it would be almost impossible to exploit the large number of fixed points of the matrix used in the cipher. Therefore, the other building blocks of the cipher should not leverage and extend the fixed points of the matrix to the high level structure of the cipher. Otherwise the cipher may be vulnerable to some self-similarity attack such as reflection attacks.

3. The Proposed Method

In this section, we explain our strategy by using the definitions given in Section 2. Then, we give algebraic construction of 20 × 20 and 24 × 24 binary matrices. The construction procedure has four main steps.

Step 1. Construct companion (state transform) matrix C_m for a given irreducible polynomial $p(x)$ of degree m. Note that C_m is an $m \times m$ matrix.

Step 2. Choose some integers s_i's with $1 \leq s_i \leq 2^m - 1$ and compute the corresponding $C_m^{s_i}$'s. Note that the selection of s_i's depends on the Hamming weight of each row of the big matrix D.

Step 3. Construct D by using $\mathrm{had}(C_m^{s_1}, C_m^{s_2}, \dots, C_m^{s_\ell})$ or $\mathrm{circ}(C_m^{s_1}, C_m^{s_2}, \dots, C_m^{s_\ell})$, where ℓ is a positive integer. Choose matrix D whose Hamming weight of the each row is as small as possible. This condition helps us to have low-cost (XOR friendly) hardware implementations.

Step 4. Check whether the branch and the number of fixed points are satisfactory.

This algorithm can be easily implemented on a computer. The results given in this study are obtained by using Magma Computational Algebra System [24]. With the help of Magma Computational Algebra System, one can evaluate hundreds of 20 × 20 or 24 × 24 binary matrices in a second.

Remark 8. Note that the diffusion layers proposed in this study can be implemented by only XOR operations whereas other diffusion layers like MDS (maximum distance separable) matrices may use table look-ups, xtime calls, and so forth [11]. Thus, performing the proposed diffusion layers gives us better implementation properties.

3.1. Algebraic Construction of Cryptographically Good 20 × 20 *Binary Matrices.*

The maximum branch number of $n \times n$ binary matrices is equal to the maximum distance of binary linear $[2n, n]$ codes. The exact maximum distance for $n \times n$ ($n \leq 18$) binary matrices is known. For example, the maximum branch number and also the upper bound for 8 × 8 matrices are 5 [4]. 20 × 20 binary matrix with a branch number 9 is known and the upper bound is 10 in theory. Note that there is no theoretical bound for the involutory binary matrices in view of branch number. The method presented herein is successful for generating 20 × 20 involutory and noninvolutory binary matrices of branch number 8. Also, 20 × 20 involutory and noninvolutory binary matrices are constructed such that the rank of $A + I$ matrix is the highest achievable rank, which is 10 for 20 × 20 involutory binary matrices and 20 for 20 × 20 noninvolutory binary matrices. In Example 11, a 20 × 20 involutory binary matrix ($A_{\mathrm{Binary}} = A_{\mathrm{Binary}}^{-1}$) is constructed from a 4 × 4 involutory FFHadamard matrix A that satisfies four restrictions simultaneously such that

(i) the 4 × 4 matrix A should be involutory as given in Lemma 6,

(ii) the 20×20 binary matrix A_{Binary} transformed from the 4×4 involutory matrix A should be of differential and linear branch number 8,

(iii) the 4×4 involutory matrix A should be chosen such that the rank of the $(A+I)$ matrix should be 2, which is in fact the highest achievable rank ($n/2$ for an $n \times n$ involutory matrix). Since the elements of $GF(2^5)$ are used to construct the 20×20 binary matrix, the rank of the matrix $(A_{\text{Binary}} + I)$ becomes 10. Thus, if it is used as 80-bit to 80-bit linear transformation, where each input element is in $GF(2^4)$, the binary linear transformation includes 2^{40} fixed points,

(iv) the elements 4×4 matrix A in $GF(2^5)$ should be chosen such that each row and column of the transformed binary matrix should have the Hamming weight equal to 7, which provides suitable implementation properties.

Remark 9. If we want to construct a 20×20 binary matrix of branch number 8 with minimum Hamming weight (in each row and column), then we need to focus on a binary matrix which has Hamming weight 7 in each row and column. That means in random search we should search $(C(20,7))^{20} \approx 2^{324}$ binary matrices whereas our search space in the proposed method is $C(31,4) = 31465$, where $2^5 - 1 = 31$ represents the number of 5×5 binary matrices (different elements) used in the construction and obtained by using the primitive polynomial $x^5 + x^2 + 1$ and 4 represents the first 4 elements in Hadamard matrix. Therefore, the main idea of the method is to reduce search space and construct binary matrices of high branch number.

Remark 10. If one wants to construct an involutory 20×20 binary matrix and uses it with 4-bit S-boxes, then the minimum number of fixed point is 2^{40} since the rank of $(D+I)$ matrix becomes at most 10 (or at most $n/2$ for an $n \times n$ involutory binary matrix). In this respect, this matrix has as possible the lowest number of fixed points. For example, the AES includes 2^{16} fixed points though the diffusion layer of the AES (shiftrows + mixcolumns) is not involutory [5]. Noninvolutory diffusion layers may provide less number of fixed points as shown in Example 12 (one fixed point).

Example 11. Let

$$A = \operatorname{had}\left(C_5^{31}, C_5^{18}, C_5^{3}, C_5^{27}\right) = \begin{pmatrix} C_5^{31} & C_5^{18} & C_5^{3} & C_5^{27} \\ C_5^{18} & C_5^{31} & C_5^{27} & C_5^{3} \\ C_5^{3} & C_5^{27} & C_5^{31} & C_5^{18} \\ C_5^{27} & C_5^{3} & C_5^{18} & C_5^{31} \end{pmatrix}. \tag{9}$$

be an involutory 4×4 FFHadamard matrix, which is also MDS matrix over the finite field $GF(2^5)$ defined by the primitive polynomial $p(x) = x^5 + x^2 + 1$; that is, the branch number of the 4×4 matrix is 5. It can be transformed into the 20×20 binary matrix satisfying the restrictions above as follows:

$$A_{\text{Binary}} = \begin{bmatrix} 1&0&0&0&0&1&0&0&0&1&0&0&1&0&0&1&0&1&0&1 \\ 0&1&0&0&0&1&1&0&0&0&0&0&0&1&0&1&1&0&1&0 \\ 0&0&1&0&0&0&1&1&0&1&0&0&1&0&1&0&1&0&0&0 \\ 0&0&0&1&0&0&0&1&1&0&1&0&0&1&0&1&0&1&0&0 \\ 0&0&0&0&1&0&0&0&1&1&0&1&0&0&1&0&1&0&1&0 \\ 1&0&0&0&1&1&0&0&0&0&1&0&1&0&1&0&0&1&0&0 \\ 1&1&0&0&0&0&1&0&0&0&1&1&0&1&0&0&0&0&1&0 \\ 0&1&1&0&1&0&0&1&0&0&0&1&0&0&0&0&0&1&0&1 \\ 0&0&1&1&0&0&0&0&1&0&1&0&1&0&0&1&0&0&1&0 \\ 0&0&0&1&1&0&0&0&0&1&0&1&0&1&0&0&1&0&0&1 \\ 0&0&1&0&0&1&0&1&0&1&1&0&0&0&0&1&0&0&0&1 \\ 0&0&0&1&0&1&1&0&1&0&0&1&0&0&0&1&1&0&0&0 \\ 0&0&1&0&1&0&1&0&0&0&0&0&1&0&0&0&1&1&0&1 \\ 1&0&0&1&0&1&0&1&0&0&0&0&0&1&0&0&0&1&1&0 \\ 0&1&0&0&1&0&1&0&1&0&0&0&0&0&1&0&0&0&1&1 \\ 1&0&1&0&1&0&0&1&0&0&1&0&0&0&1&1&0&0&0&0 \\ 1&1&0&1&0&0&0&0&1&0&1&1&0&0&0&0&1&0&0&0 \\ 0&1&0&0&0&0&0&1&0&1&0&1&1&0&1&0&0&1&0&0 \\ 1&0&1&0&0&1&0&0&1&0&0&0&1&1&0&0&0&0&1&0 \\ 0&1&0&1&0&0&1&0&0&1&0&0&0&1&1&0&0&0&0&1 \end{bmatrix}. \tag{10}$$

Note that 20×20 binary matrix, A_{Binary} given in Example 11, requires 120 XOR operations in the implementation for both encryption and decryption. In Example 12, a 20×20 noninvolutory binary matrix is constructed from 4×4 noninvolutory matrix B that satisfies three restrictions simultaneously such that

(i) the 20 × 20 binary matrix, B_{Binary}, transformed from the 4 × 4 noninvolutory matrix B should be of differential and linear branch number 8,

(ii) the rank of 4 × 4 noninvolutory matrix B should be 4, which is in fact the highest achievable rank (n for $n \times n$ matrix). Since the elements of GF(2^5) are used to construct the 20 × 20 binary matrix, the rank of the matrix ($B_{\text{Binary}} + I$) becomes 20. Therefore, if it is used as 80-bit to 80-bit linear transformation, where each input element is in GF(2^4), the binary linear transformation includes only one fixed point,

(iii) the elements of 4 × 4 matrix B in GF(2^5) should be chosen such that the constructed binary matrix should have suitable implementation properties.

Example 12. Let

$$B = \text{had}\left(C_5^{31}, C_5, C_5^{27}, C_5^{9}\right) = \begin{pmatrix} C_5^{31} & C_5 & C_5^{27} & C_5^{9} \\ C_5 & C_5^{31} & C_5^{9} & C_5^{27} \\ C_5^{27} & C_5^{9} & C_5^{31} & C_5 \\ C_5^{9} & C_5^{27} & C_5 & C_5^{31} \end{pmatrix}. \tag{11}$$

be a noninvolutory 4 × 4 FFHadamard matrix, which is also MDS matrix over the finite field GF(2^5) defined by the primitive polynomial $p(x) = x^5 + x^2 + 1$; that is, the branch number of the 4 × 4 matrix is 5. It can be transformed into the 20 × 20 binary matrix satisfying the restrictions above as follows:

$$B_{\text{Binary}} = \begin{bmatrix}
1&0&0&0&0&0&0&0&0&1&1&0&1&0&1&0&1&1&0&0\\
0&1&0&0&0&1&0&0&0&0&1&1&0&1&0&1&0&1&1&0\\
0&0&1&0&0&0&1&0&0&1&0&1&0&0&0&0&0&1&1&1\\
0&0&0&1&0&0&0&1&0&0&1&0&1&0&0&1&0&0&1&1\\
0&0&0&0&1&0&0&0&1&0&0&1&0&1&0&1&1&0&0&1\\
0&0&0&0&1&1&0&0&0&0&0&1&1&0&0&1&0&1&0&1\\
1&0&0&0&0&0&1&0&0&0&1&0&1&1&0&1&1&0&1&0\\
0&1&0&0&1&0&0&1&0&0&0&0&1&1&1&0&1&0&0&0\\
0&0&1&0&0&0&0&0&1&0&1&0&0&1&1&1&0&1&0&0\\
0&0&0&1&0&0&0&0&0&1&1&1&0&0&1&0&1&0&1&0\\
1&0&1&0&1&0&1&1&0&0&1&0&0&0&0&0&0&0&0&1\\
1&1&0&1&0&1&0&1&1&0&0&1&0&0&0&1&0&0&0&0\\
0&1&0&0&0&0&0&1&1&1&0&0&1&0&0&0&1&0&0&1\\
1&0&1&0&0&1&0&0&1&1&0&0&0&1&0&0&0&1&0&0\\
0&1&0&1&0&1&1&0&0&1&0&0&0&0&1&0&0&0&1&0\\
0&1&1&0&0&1&0&1&0&1&0&0&0&0&1&1&0&0&0&0\\
1&0&1&1&0&1&1&0&1&0&1&0&0&0&0&0&1&0&0&0\\
0&0&1&1&1&0&1&0&0&0&0&1&0&0&1&0&0&1&0&0\\
1&0&0&1&1&1&0&1&0&0&0&0&1&0&0&0&0&0&1&0\\
1&1&0&0&1&0&1&0&1&0&0&0&0&1&0&0&0&0&0&1
\end{bmatrix}. \tag{12}$$

Note that the 20×20 binary matrix B_{Binary} given in Example 12 and the inverse of 20 × 20 binary matrix (Appendix A) require 124 XOR operations and 140 XOR operations in the implementation for encryption and decryption, respectively.

3.2. Algebraic Construction of Cryptographically Good 24 × 24 *Binary Matrices.*

The exact maximum distance (upper bound) and therefore maximum branch number for 24 × 24 binary matrices are 12 [4]. The method presented herein is successful for generating 24 × 24 noninvolutory binary matrices of branch number 10. Note that there is no known 24×24 binary matrices of branch number 10 or more. 24×24 noninvolutory binary matrices are constructed such that the rank of ($D + I$) matrix is as possible as high rank. Also, when constructing 24 × 24 binary matrices, 6 × 6 circulant matrices with the elements of GF(2^4) are used. In Example 13, a 24 × 4 noninvolutory binary matrix is constructed from a 6 × 6 circulant matrix D that satisfies three restrictions simultaneously such that

(i) the 24 × 24 binary matrix, D_{Binary}, transformed from the 6 × 6 circulant matrix C should be of differential and linear branch number 10,

(ii) the 6 × 6 circulant matrix C should be chosen such that the rank of the ($D + I$) matrix should be 5, which is in fact the highest achievable rank satisfying the previous restriction. Since the elements of $GF(2^4)$ are used to construct the 24×24 binary matrix, the rank of the matrix $(D_{\text{Binary}} + I)$ becomes 20. Thus, if it is used as 96-bit to 96-bit linear transformation, where each input element is in $GF(2^4)$, the binary linear transformation includes 2^{16} fixed points.

(iii) The elements of 6×6 matrix D in $GF(2^4)$ should be chosen such that the constructed binary matrix should have suitable implementation properties.

Example 13. Let

$$D = \operatorname{circ}\left(C_4^7, C_4^{15}, C_4^3, C_4^{14}, C_4^{11}, C_4^2\right)$$

$$= \begin{pmatrix} C_4^7 & C_4^{15} & C_4^3 & C_4^{14} & C_4^{11} & C_4^2 \\ C_4^2 & C_4^7 & C_4^{15} & C_4^3 & C_4^{14} & C_4^{11} \\ C_4^{11} & C_4^2 & C_4^7 & C_4^{15} & C_4^3 & C_4^{14} \\ C_4^{14} & C_4^{11} & C_4^2 & C_4^7 & C_4^{15} & C_4^3 \\ C_4^3 & C_4^{14} & C_4^{11} & C_4^2 & C_4^7 & C_4^{15} \\ C_4^{15} & C_4^3 & C_4^{14} & C_4^{11} & C_4^2 & C_4^7 \end{pmatrix} \tag{13}$$

be a 6×6 circulant matrix, which is of branch number 6 over finite field $GF(2^4)$ defined by the primitive polynomial $p(x) = x^4 + x + 1$. It can be transformed into the 24×24 binary matrix satisfying the restrictions above as follows:

$$D_{\text{Binary}} = \left[\begin{array}{cccccccccccccccccccccccc}
1&1&0&1&1&0&0&0&0&1&0&0&1&1&0&0&0&1&1&1&0&0&1&0\\
1&0&1&1&0&1&0&0&0&1&1&0&0&0&1&0&1&1&0&0&0&0&1&1\\
0&1&0&1&0&0&1&0&0&0&1&1&0&0&0&1&1&1&1&0&1&0&0&1\\
1&0&1&0&0&0&0&1&1&0&0&1&1&0&0&0&1&1&1&1&0&1&0&0\\
0&0&1&0&1&1&0&1&1&0&0&0&0&1&0&0&1&1&0&0&0&1&1&1\\
0&0&1&1&1&0&1&1&0&1&0&0&0&1&1&0&0&0&1&0&1&1&0&0\\
1&0&0&1&0&1&0&1&0&0&1&0&0&0&1&1&0&0&0&1&1&1&1&0\\
0&1&0&0&1&0&1&0&0&0&0&1&1&0&0&1&1&0&0&0&1&1&1&1\\
0&1&1&1&0&0&1&0&1&1&0&1&1&0&0&0&0&1&0&0&1&1&0&0\\
1&1&0&0&0&0&1&1&1&0&1&1&0&1&0&0&0&1&1&0&0&0&1&0\\
1&1&1&0&1&0&0&1&0&1&0&1&0&0&1&0&0&0&1&1&0&0&0&1\\
1&1&1&1&0&1&0&0&1&0&1&0&0&0&0&1&1&0&0&1&1&0&0&0\\
1&1&0&0&0&1&1&1&0&0&1&0&1&1&0&1&1&0&0&0&0&1&0&0\\
0&0&1&0&1&1&0&0&0&0&1&1&1&0&1&1&0&1&0&0&0&1&1&0\\
0&0&0&1&1&1&1&0&1&0&0&1&0&1&0&1&0&0&1&0&0&0&1&1\\
1&0&0&0&1&1&1&1&0&1&0&0&1&0&1&0&0&0&0&1&1&0&0&1\\
0&1&0&0&1&1&0&0&0&1&1&1&0&0&1&0&1&1&0&1&1&0&0&0\\
0&1&1&0&0&0&1&0&1&1&0&0&0&0&1&1&1&0&1&1&0&1&0&0\\
0&0&1&1&0&0&0&1&1&1&1&0&1&0&0&1&0&1&0&1&0&0&1&0\\
1&0&0&1&1&0&0&0&1&1&1&1&0&1&0&0&1&0&1&0&0&0&0&1\\
1&0&0&0&0&1&0&0&1&1&0&0&0&1&1&1&0&0&1&0&1&1&0&1\\
0&1&0&0&0&1&1&0&0&0&1&0&1&1&0&0&0&0&1&1&1&0&1&1\\
0&0&1&0&0&0&1&1&0&0&0&1&1&1&1&0&1&0&0&1&0&1&0&1\\
0&0&0&1&1&0&0&1&1&0&0&0&1&1&1&1&0&1&0&0&1&0&1&0
\end{array}\right]. \tag{14}$$

Note that in a straight coding the 24×24 binary matrix, D_{Binary} given in Example 13, requires 240 XOR operations in the implementation for both encryption and decryption. The required number of XOR operations can be reduced to 186 by adding 6 temporary variables to the implementation for both encryption and decryption (Appendices B and C).

4. Security Assessment of an Assumed Lightweight Block Cipher with 80-Bit or 96-Bit Block Size

In this section, we focus on the security analysis of the assumed block cipher using the proposed linear transformation. A differentially active S-box is defined as an S-box given a nonzero input difference, and a linearly active S-box is defined as an S-box given a nonzero output mask. In this study, S-boxes are assumed to be bijective mappings defined on $GF(2^m)$ and round keys are assumed to be independent and random uniform. Thus the number of active S-boxes is not affected by the key addition layer. The branch number of a diffusion layer is the minimum number of active S-boxes in the 2-round SPN (substitution permutation network). We follow the method defined in [25, 26]. Let p_D and q_L be the the maximum probabilities of the differential and linear characteristic for $2r$-round SPN, respectively. Let $p_D^{2r} \leq p^{r \cdot \beta}$ and $q_L^{2r} \leq q^{r \cdot \beta}$, where p, q, and β denote the maximum differential probability for the S-box, the maximum linear probability for the S-box, and branch number for the diffusion layer used in a block cipher, respectively. In this study, an SPN structure consisting of a number of rounds of the same 20 4-bit S-boxes connected by a 20×20 binary matrix is considered for 80-bit block size. Figure 1 shows one round function of an assumed block cipher. Note that the maximum differential and linear probabilities of the S-box are assumed to be 2^{-2} which is the best value for 4×4 S-boxes [27]. Then, the maximum probabilities of the differential, p_D, and linear characteristic, q_L, for $2r$-round SPN are as follows:

$$p_D^{2r} \le \left(2^{-2}\right)^{r\cdot\beta_A}, \qquad q_L^{2r} \le \left(2^{-2}\right)^{r\cdot\beta_A}, \tag{15}$$

where β_A denotes the branch number of 20×20 binary matrix assumed for the lightweight block cipher demonstrated in Figure 1. The maximum differential and linear probabilities of 2-round SPN are bounded by $(2^{-2})^{1\cdot 8} = 2^{-16}$ since the branch number of the binary matrix is assumed to be 8 and thus the number of minimum active S-box is 8 in the 2-round SPN. In Table 1, the lower bounds for the number of active S-boxes and the upper bounds for the probabilities for linear and differential probabilities in each round size are computed for the assumed block cipher of 80-bit and 96-bit block sizes.

In this context, the minimum number of rounds needed for the lightweight block cipher with 80-bit block size to be secure against differential and linear cryptanalysis is 10 because the maximum differential and linear probabilities of 10-round SPN are bounded by $(2^{-2})^{5\cdot 8} = 2^{-80} \le 2^{-80}$. Similarly, if an SPN structure consisting of a number of rounds of the same 24 4-bit S-boxes connected by a 24×24 binary matrix is considered for 96-bit block size, then the minimum number of rounds needed for the lightweight block cipher to be secure against differential and linear cryptanalysis is again obtained as 10 because the maximum differential and linear probabilities of 10-round SPN are bounded by $(2^{-2})^{5\cdot 10} = 2^{-100} \le 2^{-96}$.

5. Conclusion

In this study, an algebraic method based on state transform matrix (companion matrix) to construct $n \times n$ binary matrices with good implementation properties for lightweight block ciphers and hash functions is given. The proposed method can also be considered as a generalization and different interpretation of the methods given in [12, 13] since it works for any n. For 20×20 and 24×24 binary matrices, examples are provided with good implementation properties. The binary matrices are also constructed to have suitable software and hardware implementation properties for lightweight block ciphers. In other words, by using the proposed method, the matrices have smaller hardware implementations in view of the required number of XOR gates. Note that the binary matrices with these sizes have not been studied in the literature well enough, which may allow us to design a lightweight block cipher with 80-bit and 96-bit block sizes if these matrices are used with 4-bit S-boxes. 20×20 binary matrix given in Example 12 has only one fixed point and branch number 8.

Appendices

A. Inverse of the 20×20 Binary Matrix Given in Example 12

The inverse of the 20×20 binary matrix (B_{Binary}^{-1}) given in Example 12 can be constructed by transforming 4×4 FFHadamard matrix $\text{had}(C_5^2, C_5^3, C_5^{29}, C_5^{11})$ into the binary form as given below:

$$B_{\text{Binary}}^{-1} = \begin{bmatrix}
0&0&0&1&0&0&0&1&0&0&1&0&1&0&0&1&0&0&1&1\\
0&0&0&0&1&0&0&0&1&0&0&1&0&1&0&1&1&0&0&1\\
1&0&0&1&0&0&0&1&0&1&0&0&0&0&1&1&1&1&1&1\\
0&1&0&0&1&1&0&0&1&0&1&0&0&0&0&0&1&1&1&1\\
0&0&1&0&0&0&1&0&0&1&0&1&0&0&0&0&0&1&1&1\\
0&0&1&0&0&0&0&0&1&0&1&0&0&1&1&1&0&1&0&0\\
0&0&0&1&0&0&0&0&0&1&1&1&0&0&1&0&1&0&1&0\\
0&0&1&0&1&1&0&0&1&0&1&1&1&1&1&0&0&0&0&1\\
1&0&0&1&0&0&1&0&0&1&0&1&1&1&1&1&0&0&0&0\\
0&1&0&0&1&0&0&1&0&0&0&0&1&1&1&0&1&0&0&0\\
1&0&1&0&0&1&0&0&1&1&0&0&0&1&0&0&0&1&0&0\\
0&1&0&1&0&1&1&0&0&1&0&0&0&0&1&0&0&0&1&0\\
0&0&0&0&1&1&1&1&1&1&1&0&0&1&0&0&0&1&0&1\\
1&0&0&0&0&0&1&1&1&1&0&1&0&0&1&1&0&0&1&0\\
0&1&0&0&0&0&0&1&1&1&0&0&1&0&0&0&1&0&0&1\\
1&0&0&1&1&1&0&1&0&0&0&0&1&0&0&0&0&0&1&0\\
1&1&0&0&1&0&1&0&1&0&0&0&0&1&0&0&0&0&0&1\\
1&1&1&1&1&0&0&0&0&1&0&0&1&0&1&1&0&0&1&0\\
0&1&1&1&1&1&0&0&0&0&1&0&0&1&0&0&1&0&0&1\\
0&0&1&1&1&0&1&0&0&0&0&1&0&0&1&0&0&1&0&0
\end{bmatrix}. \tag{A.1}$$

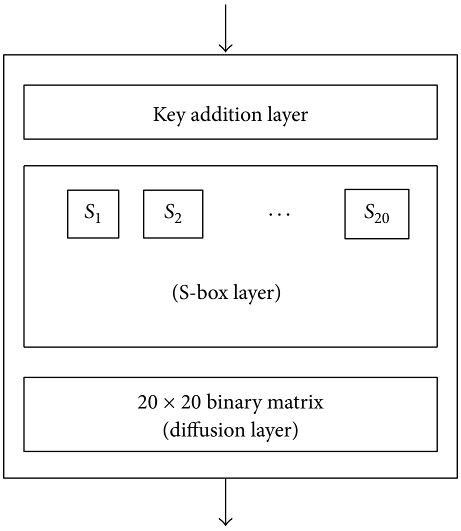


Figure 1: One round function of an assumed lightweight block cipher.

Table 1: The lower bounds for the number of active S-boxes and the upper bounds for the linear and differential probabilities for the assumed block cipher of 80-bit and 96-bit block size.

Round	The lower bounds for the number of active S-boxes		The upper bounds for the linear and differential probabilities	
	80-bit block cipher	96-bit block cipher	80-bit block cipher	96-bit block cipher
2	8	10	2^{-16}	2^{-20}
3	9	11	2^{-18}	2^{-22}
4	16	20	2^{-32}	2^{-40}
5	17	21	2^{-34}	2^{-42}
6	24	30	2^{-48}	2^{-60}
7	25	31	2^{-50}	2^{-62}
8	32	40	2^{-64}	2^{-80}
9	33	41	2^{-66}	2^{-82}
10	40	50	2^{-80}	2^{-100}
11	41	51	2^{-82}	2^{-102}
12	48	60	2^{-96}	2^{-120}

B. Implementation of the 24×24 Binary Matrix Given in Example 13

If the 24 × 24 binary matrix given in Example 13 is implemented with 4-bit XORs, then D_{Binary} is represented by 4-bit XORs of binary vectors as follows: $D_{\text{Binary}} \cdot x = y$, where $x = (x_0, x_1, \ldots, x_{31})^T$ and $y = (y_0, y_1, \ldots, y_{31})^T$ with $x_i, y_i \in \mathrm{GF}(2^4)$, $i = 0, 1, \ldots, 31$. Note also that $T_0, T_1, \ldots, T_5$ are temporary variables used to reduce the number of XOR operations from 240 XOR to 186 XOR. Then,

$$T_0 = x_3 \oplus x_4 \oplus x_{13} \oplus x_{18},$$

$$T_1 = x_0 \oplus x_9 \oplus x_{14} \oplus x_{23},$$

$$T_2 = x_1 \oplus x_6 \oplus x_{15} \oplus x_{16},$$

$$T_3 = x_2 \oplus x_{11} \oplus x_{12} \oplus x_{21},$$

$$T_4 = x_7 \oplus x_8 \oplus x_{17} \oplus x_{22},$$

$$T_5 = x_5 \oplus x_{10} \oplus x_{19} \oplus x_{20},$$

$$y_0 = T_0 \oplus x_0 \oplus x_1 \oplus x_9 \oplus x_{12} \oplus x_{17} \oplus x_{19} \oplus x_{22},$$

$$y_1 = T_1 \oplus x_2 \oplus x_3 \oplus x_5 \oplus x_{10} \oplus x_{16} \oplus x_{17} \oplus x_{22},$$

$$y_2 = T_2 \oplus x_3 \oplus x_{10} \oplus x_{11} \oplus x_{17} \oplus x_{18} \oplus x_{20} \oplus x_{23},$$

$$y_3 = T_3 \oplus x_0 \oplus x_7 \oplus x_8 \oplus x_{16} \oplus x_{17} \oplus x_{18} \oplus x_{19},$$

$$y_4 = T_4 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_{13} \oplus x_{16} \oplus x_{21} \oplus x_{23},$$

$$y_5 = T_0 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{14} \oplus x_{20} \oplus x_{21},$$

$$y_6 = T_5 \oplus x_0 \oplus x_3 \oplus x_7 \oplus x_{14} \oplus x_{15} \oplus x_{21} \oplus x_{22},$$

$$y_7 = T_2 \oplus x_4 \oplus x_{11} \oplus x_{12} \oplus x_{20} \oplus x_{21} \oplus x_{22} \oplus x_{23},$$

$$y_8 = T_3 \oplus x_1 \oplus x_3 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{17} \oplus x_{20},$$

$$y_9 = T_4 \oplus x_0 \oplus x_1 \oplus x_6 \oplus x_{10} \oplus x_{11} \oplus x_{13} \oplus x_{18},$$

$$
\begin{aligned}
y_{10} &= T_1 \oplus x_1 \oplus x_2 \oplus x_4 \oplus x_7 \oplus x_{11} \oplus x_{18} \oplus x_{19},\\
y_{11} &= T_5 \oplus x_0 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_8 \oplus x_{15} \oplus x_{16},\\
y_{12} &= T_2 \oplus x_0 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{12} \oplus x_{13} \oplus x_{21},\\
y_{13} &= T_3 \oplus x_4 \oplus x_5 \oplus x_{10} \oplus x_{14} \oplus x_{15} \oplus x_{17} \oplus x_{22},\\
y_{14} &= T_0 \oplus x_5 \oplus x_6 \oplus x_8 \oplus x_{11} \oplus x_{15} \oplus x_{22} \oplus x_{23},\\
y_{15} &= T_1 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_{12} \oplus x_{19} \oplus x_{20},\\
y_{16} &= T_5 \oplus x_1 \oplus x_4 \oplus x_9 \oplus x_{11} \oplus x_{14} \oplus x_{16} \oplus x_{17},\\
y_{17} &= T_2 \oplus x_2 \oplus x_8 \oplus x_9 \oplus x_{14} \oplus x_{18} \oplus x_{19} \oplus x_{21},\\
y_{18} &= T_4 \oplus x_2 \oplus x_3 \oplus x_9 \oplus x_{10} \oplus x_{12} \oplus x_{15} \oplus x_{19},\\
y_{19} &= T_0 \oplus x_0 \oplus x_8 \oplus x_9 \oplus x_{10} \oplus x_{11} \oplus x_{16} \oplus x_{23},\\
y_{20} &= T_1 \oplus x_5 \oplus x_8 \oplus x_{13} \oplus x_{15} \oplus x_{18} \oplus x_{20} \oplus x_{21},\\
y_{21} &= T_5 \oplus x_1 \oplus x_6 \oplus x_{12} \oplus x_{13} \oplus x_{18} \oplus x_{22} \oplus x_{23},\\
y_{22} &= T_3 \oplus x_6 \oplus x_7 \oplus x_{13} \oplus x_{14} \oplus x_{16} \oplus x_{19} \oplus x_{23},\\
y_{23} &= T_4 \oplus x_3 \oplus x_4 \oplus x_{12} \oplus x_{13} \oplus x_{14} \oplus x_{15} \oplus x_{20}.
\end{aligned}
\tag{B.1}
$$

C. Inverse of 24 × 24 Binary Matrix Given in Example 13 and Implementation Details

The inverse of the 24 × 24 binary matrix given in Example 13 is constructed from the 6 × 6 circulant matrix $D^{-1} = \mathrm{circ}(C_4^{11}, C_4^{14}, C_4^{14}, C_4^{2}, C_4^{9}, C_4^{15})$ as follows:

$$
D^{-1}_{\text{Binary}} = \begin{bmatrix}
0&1&1&1&1&1&0&0&1&1&0&0&0&0&1&0&0&1&0&1&1&0&0&0\\
1&1&0&0&0&0&1&0&0&0&1&0&0&0&1&1&1&1&1&1&0&1&0&0\\
1&1&1&0&0&0&0&1&0&0&0&1&1&0&0&1&0&1&1&1&0&0&1&0\\
1&1&1&1&1&0&0&0&1&0&0&0&0&1&0&0&1&0&1&1&0&0&0&1\\
1&0&0&0&0&1&1&1&1&1&0&0&1&1&0&0&0&0&1&0&0&1&0&1\\
0&1&0&0&1&1&0&0&0&0&1&0&0&0&1&0&0&0&1&1&1&1&1&1\\
0&0&1&0&1&1&1&0&0&0&0&1&0&0&0&1&1&0&0&1&0&1&1&1\\
0&0&0&1&1&1&1&1&1&0&0&0&1&0&0&0&0&1&0&0&1&0&1&1\\
0&1&0&1&1&0&0&0&0&1&1&1&1&1&0&0&1&1&0&0&0&0&1&0\\
1&1&1&1&0&1&0&0&1&1&0&0&0&0&1&0&0&0&1&0&0&0&1&1\\
0&1&1&1&0&0&1&0&1&1&1&0&0&0&0&1&0&0&0&1&1&0&0&1\\
1&0&1&1&0&0&0&1&1&1&1&1&1&0&0&0&1&0&0&0&0&1&0&0\\
0&0&1&0&0&1&0&1&1&0&0&0&0&1&1&1&1&1&0&0&1&1&0&0\\
0&0&1&1&1&1&1&1&0&1&0&0&1&1&0&0&0&0&1&0&0&0&1&0\\
1&0&0&1&0&1&1&1&0&0&1&0&1&1&1&0&0&0&0&1&0&0&0&1\\
0&1&0&0&1&0&1&1&0&0&0&1&1&1&1&1&1&0&0&0&1&0&0&0\\
1&1&0&0&0&0&1&0&0&1&0&1&1&0&0&0&0&1&1&1&1&1&0&0\\
0&0&1&0&0&0&1&1&1&1&1&1&0&1&0&0&1&1&0&0&0&0&1&0\\
0&0&0&1&1&0&0&1&0&1&1&1&0&0&1&0&1&1&1&0&0&0&0&1\\
1&0&0&0&0&1&0&0&1&0&1&1&0&0&0&1&1&1&1&1&1&0&0&0\\
1&1&0&0&1&1&0&0&0&0&1&0&0&1&0&1&1&0&0&0&0&1&1&1\\
0&0&1&0&0&0&1&0&0&0&1&1&1&1&1&1&0&1&0&0&1&1&0&0\\
0&0&0&1&0&0&0&1&1&0&0&1&0&1&1&1&0&0&1&0&1&1&1&0\\
1&0&0&0&1&0&0&0&0&1&0&0&1&0&1&1&0&0&0&1&1&1&1&1
\end{bmatrix}. \tag{C.1}
$$

Let $D^{-1}_{\text{Binary}} \cdot x = y$, where $x = (x_0, x_1, \ldots, x_{31})^T$ and $y = (y_0, y_1, \ldots, y_{31})^T$ with $x_i, y_i \in \mathrm{GF}(2^4)$, $i = 0, 1, \ldots, 31$. Note also that $T_0, T_1, \ldots, T_5$ are temporary variables used to reduce the number of XOR operations from 240 XORs to 186 XORs. Then,

$$
\begin{aligned}
T_0 &= x_2 \oplus x_3 \oplus x_8 \oplus x_9,\\
T_1 &= x_0 \oplus x_1 \oplus x_{18} \oplus x_{19},\\
T_2 &= x_5 \oplus x_6 \oplus x_7 \oplus x_{12},\\
T_3 &= x_4 \oplus x_{21} \oplus x_{22} \oplus x_{23},\\
T_4 &= x_{10} \oplus x_{11} \oplus x_{16} \oplus x_{17},\\
T_5 &= x_{13} \oplus x_{14} \oplus x_{15} \oplus x_{20},\\
y_0 &= T_0 \oplus x_1 \oplus x_4 \oplus x_5 \oplus x_{14} \oplus x_{17} \oplus x_{19} \oplus x_{20},\\
y_1 &= T_1 \oplus x_6 \oplus x_{10} \oplus x_{14} \oplus x_{15} \oplus x_{16} \oplus x_{17} \oplus x_{21},\\
y_2 &= T_1 \oplus x_2 \oplus x_7 \oplus x_{11} \oplus x_{12} \oplus x_{15} \oplus x_{17} \oplus x_{22},\\
y_3 &= T_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_8 \oplus x_{13} \oplus x_{16} \oplus x_{23},\\
y_4 &= T_2 \oplus x_0 \oplus x_8 \oplus x_9 \oplus x_{13} \oplus x_{18} \oplus x_{21} \oplus x_{23},\\
y_5 &= T_3 \oplus x_1 \oplus x_5 \oplus x_{10} \oplus x_{14} \oplus x_{18} \oplus x_{19} \oplus x_{20},\\
y_6 &= T_3 \oplus x_2 \oplus x_5 \oplus x_6 \oplus x_{11} \oplus x_{15} \oplus x_{16} \oplus x_{19},
\end{aligned}
$$

$$\begin{aligned}
y_7 &= T_2 \oplus x_3 \oplus x_4 \oplus x_8 \oplus x_{17} \oplus x_{20} \oplus x_{22} \oplus x_{23},\\
y_8 &= T_4 \oplus x_1 \oplus x_3 \oplus x_4 \oplus x_9 \oplus x_{12} \oplus x_{13} \oplus x_{22},\\
y_9 &= T_0 \oplus x_0 \oplus x_1 \oplus x_5 \oplus x_{14} \oplus x_{18} \oplus x_{22} \oplus x_{23},\\
y_{10} &= T_0 \oplus x_1 \oplus x_6 \oplus x_{10} \oplus x_{15} \oplus x_{19} \oplus x_{20} \oplus x_{23},\\
y_{11} &= T_0 \oplus x_0 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{16} \oplus x_{21},\\
y_{12} &= T_5 \oplus x_2 \oplus x_5 \oplus x_7 \oplus x_8 \oplus x_{16} \oplus x_{17} \oplus x_{21},\\
y_{13} &= T_2 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_9 \oplus x_{13} \oplus x_{18} \oplus x_{22},\\
y_{14} &= T_2 \oplus x_0 \oplus x_3 \oplus x_{10} \oplus x_{13} \oplus x_{14} \oplus x_{19} \oplus x_{23},\\
y_{15} &= T_5 \oplus x_1 \oplus x_4 \oplus x_6 \oplus x_7 \oplus x_{11} \oplus x_{12} \oplus x_{16},\\
y_{16} &= T_1 \oplus x_6 \oplus x_9 \oplus x_{11} \oplus x_{12} \oplus x_{17} \oplus x_{20} \oplus x_{21},\\
y_{17} &= T_4 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{13} \oplus x_{22},\\
y_{18} &= T_4 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_9 \oplus x_{14} \oplus x_{18} \oplus x_{23},\\
y_{19} &= T_4 \oplus x_0 \oplus x_5 \oplus x_8 \oplus x_{15} \oplus x_{18} \oplus x_{19} \oplus x_{20},\\
y_{20} &= T_3 \oplus x_0 \oplus x_1 \oplus x_5 \oplus x_{10} \oplus x_{13} \oplus x_{15} \oplus x_{16},\\
y_{21} &= T_5 \oplus x_2 \oplus x_6 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{17} \oplus x_{21},\\
y_{22} &= T_5 \oplus x_3 \oplus x_7 \oplus x_8 \oplus x_{11} \oplus x_{18} \oplus x_{21} \oplus x_{22},\\
y_{23} &= T_3 \oplus x_0 \oplus x_9 \oplus x_{12} \oplus x_{14} \oplus x_{15} \oplus x_{19} \oplus x_{20}.
\end{aligned} \tag{C.2}$$

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Sedat Akleylek is partially supported by OMÜ under the Grant no. PYO.MUH.1904.12.014. The authors thank the anonymous referees for their detailed and very helpful comments and for bringing reference [26] to our attention. The authors also thank Orhun Kara for his valuable comments on the discussion of Remark 7.

References

[1] C. E. Shannon, “Communication theory of secrecy systems,” *The Bell System Technical Journal*, vol. 28, pp. 656–715, 1949.

[2] O. Karaahmetoğlu, M. T. Sakallı, E. Buluş, and I. Tutănescu, “A new method to determine algebraic expression of power mapping based S-boxes,” *Information Processing Letters*, vol. 113, no. 7, pp. 229–235, 2013.

[3] A. M. Youssef and S. E. Tavares, “Affine equivalence in the AES round function,” *Discrete Applied Mathematics*, vol. 148, no. 2, pp. 161–170, 2005.

[4] D. Kwon, S. H. Sung, J. H. Song, and S. Park, “Design of block ciphers and coding theory,” *Trends in Mathematics*, vol. 8, no. 1, pp. 13–20, 2005.

[5] M. R. Z’aba, *Analysis of linear relationships in block ciphers [Ph.D. thesis]*, Queensland University of Technology, Brisbane, Australia, 2010.

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

[7] FIPS 197, *Advanced Encryption Standard*, US National Institute of Standards and Technology, 2001.

[8] P. S. L. M. Barreto and V. Rijmen, “The Khazad legacy-level block cipher,” in *Proceedings of the 1st Open NESSIE Workshop*, 2000.

[9] K. Aoki, T. Ichikawa, M. Kanda et al., “Camellia: a 128-bit block cipher suitable for multiple platforms—design and analysis,” in *Proceedings of the 7th Annual International Workshop on Selected Areas in Cryptography (SAC ’00)*, vol. 2012 of *Lecture Notes in Computer Science*, pp. 39–56, 2000.

[10] D. Kwon, J. Kim, S. Park et al., “New block cipher: ARIA,” in *Information Security and Cryptology—ICISC 2003*, vol. 2971 of *Lecture Notes in Computer Science*, pp. 432–445, Springer, Berlin, Germany, 2004.

[11] J. Nakahara Jr. and É. Abrahão, “A new involutory MDS matrix for the AES,” *International Journal of Network Security*, vol. 9, no. 2, pp. 109–116, 2009.

[12] B. Aslan and M. T. Sakallı, “Algebraic construction of cryptographically good binary linear transformations,” *Security and Communication Networks*, vol. 7, no. 1, pp. 53–63, 2014.

[13] M. T. Sakallı and B. Aslan, “On the algebraic construction of cryptographically good 32 × 32 binary linear transformations,” *Journal of Computational and Applied Mathematics*, vol. 259, pp. 485–494, 2014.

[14] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, “The simon and speck families of lightweight block ciphers,” Cryptology ePrint Archive, Report 2013/404, 2013.

[15] H. Yap, K. Khoo, A. Poschmann, and M. Henricksen, “EPCBC—a block cipher suitable for electronic product code encryption,” in *Cryptology and Network Security: Proceedings of the 10th International Conference, CANS 2011, Sanya, China, December 10–12, 2011*, vol. 7092 of *Lecture Notes in Computer Science*, pp. 76–97, Springer, Berlin, Germany, 2011.

[16] F. Karakoc, H. Demirci, and A. E. Harmanci, “ITUbee: a software oriented lightweight block cipher,” in *Lightweight Cryptography for Security and Privacy: 2nd International Workshop, LightSec 2013, Gebze, Turkey, May 6-7, 2013, Revised Selected Papers*, vol. 8162, pp. 16–27, Springer, Berlin, Germany, 2013.

[17] F. X. Standaert, G. Piret, N. Gershenfeld, and J.-J. Quisquater, “SEA: a scalable encryption algorithm for small embedded applications,” in *Smart Card Research and Advanced Applications*, vol. 3928 of *Lecture Notes in Computer Science*, pp. 222–236, Springer, Berlin, Germany, 2006.

[18] R. J. McEliece, *Finite Fields for Computer Scientists and Engineers*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1987.

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

[20] N. T. Courtois, G. V. Bard, and D. Wagner, “Algebraic and slide attacks on KeeLoq,” in *Fast Software Encryption*, vol. 5086 of *Lecture Notes in Computer Science*, pp. 97–115, Springer, Berlin, Germany, 2008.

[21] S. Vaudenay, “Related-key attack against triple encryption based on fixed points,” in *Proceedings of the International Conference on Security and Cryptography (SECRYPT ’11)*, pp. 59–67, July 2011.

[22] A. Bay, A. Mashatan, and S. Vaudenay, “A related-key attack against multiple encryption based on fixed points,” in *E-Business and Telecommunications: International Joint Conference, ICETE 2011, Seville, Spain, July 18–21, 2011, Revised Selected Papers*, vol. 314 of *Communications in Computer and Information Science*, pp. 264–280, Springer, Berlin, Germany, 2012.

[23] I. Dinur, O. Dunkelmann, and A. Shamir, “Improved attacks on full GOST,” in *Fast Software Encryption: 19th International Workshop, FSE 2012, Washington, DC, USA, March 19–21, 2012. Revised Selected Papers*, vol. 7549 of *Lecture Notes in Computer Science*, pp. 9–28, Springer, Berlin, Germany, 2012.

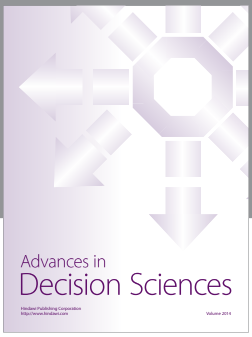
[24] W. Bosma, J. Cannon, and C. Playoust, “The Magma algebra system I: the user language,” *Journal of Symbolic Computation*, vol. 24, no. 3-4, pp. 235–265, 1997.

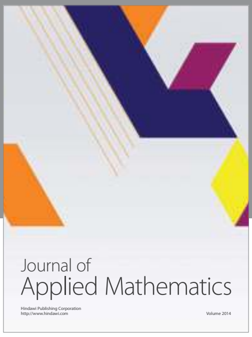
[25] B. W. Koo, H. S. Jang, and J. H. Song, “On constructing of a 32×32 binary matrix as a diffusion layer for a 256-bit block cipher,” in *Information Security and Cryptology—ICISC 2006*, vol. 4296 of *Lecture Notes in Computer Science*, pp. 51–64, Springer, Berlin, Germany, 2006.

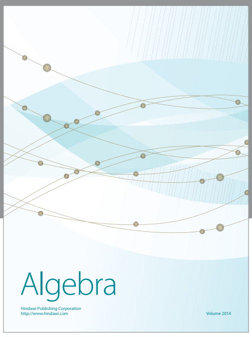
[26] S. Hong, S. Lee, J. Lim, J. Sung, D. Cheon, and I. Cho, “Provable security against differential and linear cryptanalysis for the SPN structure,” in *Fast Software Encryption*, vol. 1978 of *Lecture Notes in Computer Science*, pp. 273–283, Springer, Berlin, Germany, 2001.

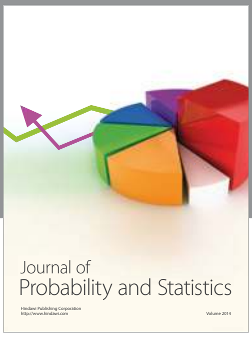
[27] M.-J. O. Saarinen, “Cryptographic analysis of all 4 × 4-bit S-boxes,” in *Selected Areas in Cryptography*, vol. 7118 of *Lecture Notes in Computer Science*, pp. 118–133, Springer, Berlin, Germany, 2012.

Advances in
Operations Research

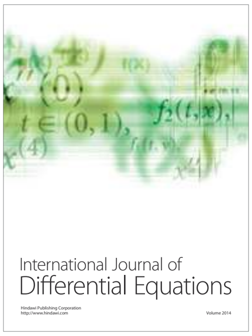

Advances in
Decision Sciences


Journal of
Applied Mathematics

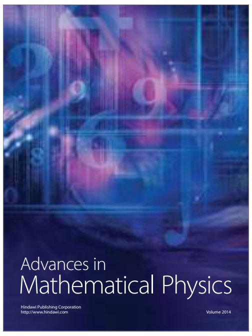

Algebra

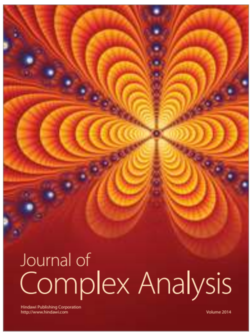

Journal of
Probability and Statistics

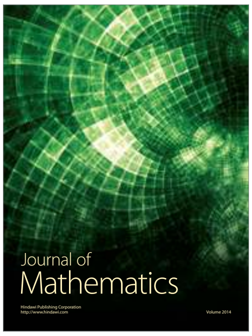
The Scientific
World Journal

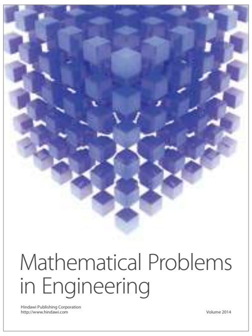

International Journal of
Differential Equations

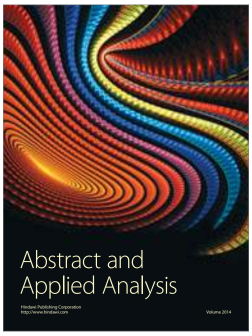
International Journal of
Combinatorics

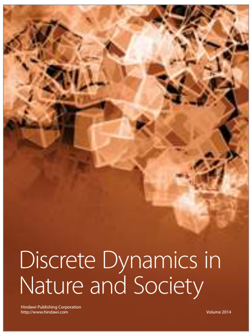

Advances in
Mathematical Physics

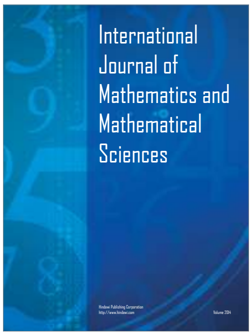

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

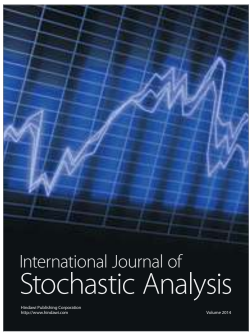

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization