Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

# On the algebraic construction of cryptographically good 32 × 32 binary linear transformations

Muharrem Tolga Sakallı [a,*], Bora Aslan [b]

[a] Computer Engineering Department, Trakya University, Edirne, Turkey
[b] Computer Programming Department, Kırklareli University, Kırklareli, Turkey

## HIGHLIGHTS

- A new algebraic method to construct cryptographically good 32 × 32 binary matrices.
- How to construct 32 × 32 involutory binary matrices of branch number 12.
- To construct non-involutory binary matrices of branch number 11 with a fixed point.

## ARTICLE INFO

*Article history:*
Received 25 November 2012
Received in revised form 5 May 2013

*Keywords:*
Cryptography
Block cipher
Binary linear transformation
Branch number
Fixed points
Finite fields

## ABSTRACT

Binary linear transformations (also called binary matrices) have matrix representations over $GF(2)$. Binary matrices are used as diffusion layers in block ciphers such as Camellia and ARIA. Also, the 8 × 8 and 16 × 16 binary matrices used in Camellia and ARIA, respectively, have the maximum branch number and therefore are called Maximum Distance Binary Linear (MDBL) codes. In the present study, a new algebraic method to construct cryptographically good 32 × 32 binary linear transformations, which can be used to transform a 256-bit input block to a 256-bit output block, is proposed. When constructing these binary matrices, the two cryptographic properties; the branch number and the number of fixed points are considered. The method proposed is based on 8 × 8 involutory and non-involutory Finite Field Hadamard (FFHadamard) matrices with the elements of $GF(2^4)$. How to construct 32 × 32 involutory binary matrices of branch number 12, and non-involutory binary matrices of branch number 11 with one fixed point, are described.

## 1. Introduction

The two important structures in block cipher design are Feistel Networks and Substitution Permutation Networks (SPNs). An SPN structure consists of a substitution layer followed by a linear transformation, which is also called the diffusion layer. The diffusion layer is to ensure that all the output bits depend on all the input bits after a few rounds, while the substitution layer or the nonlinear layer ensures that this dependency is complex and nonlinear in nature [1]. Most diffusion layers are linear transformations having matrix representations over $GF(2^m)$ or $GF(2)$. Binary matrices, which have matrix representations over $GF(2)$, are also called binary linear transformations.

Binary matrices are used as diffusion layers of block ciphers such as Camellia [2] and ARIA [3]. Also, the binary matrices used in Camellia and ARIA have the maximum branch number and therefore are called Maximum Distance Binary Linear (MDBL) codes [1]. The maximum branch number of 8 × 8 and 16 × 16 binary matrices is 5 and 8 respectively, i.e., the input difference and the corresponding output difference across these matrices have total weight 5 and 8, respectively. However,

[*] Corresponding author. Tel.: +90 5052504681; fax: +90 2842261225.
*E-mail addresses:* tolga@trakya.edu.tr, mtolgasakalli@hotmail.com (M.T. Sakallı), bora.aslan@kirklareli.edu.tr (B. Aslan).

0377-0427/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cam.2013.05.008

the maximum branch number of 32 × 32 binary matrices is regarded as 12, which has not been proven yet [4]. Moreover, an advantage of using such binary matrices in the design of block ciphers compared with Maximum Distance Separable (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 [5].

The two important techniques of measuring diffusion are the branch number [6] and the number of fixed points [7], respectively. The branch number of a diffusion layer, which represents diffusion rate and measures security against linear [8] and differential cryptanalysis [9], denotes the minimum number of active S-boxes for any two consecutive rounds. The second measure, the number of fixed points, provides an indication of how well the linear transformation effectively changes the value of the input block when producing the output block. The basis of the idea is that there is no diffusion at these points since the input blocks are left intact by the linear transformation.

Koo et al. [4] proposed a method to construct a 32×32 binary matrix of branch number 10, which can be used to transform a 256-bit input to a 256-bit output. It seems that they aimed to use arbitrary 32 parallel S-boxes before this binary matrix, which is not involutory. In a previous study [10], we presented a new algebraic construction method to generate 8 × 8 and 16 × 16 binary matrices of maximum branch number. The present study concentrates on the two cryptographic properties; the branch number and the number of fixed points. A new algebraic construction method based on 8 × 8 involutory and non-involutory Finite Field Hadamard (FFHadamard) matrices with the elements of $GF(2^4)$ to generate 32 × 32 involutory binary matrices of branch number 12, and non-involutory binary matrices of branch number 11 with one fixed point is presented. Also, the constructed matrices have suitable implementation properties on 8-bit, 32-bit and 64-bit processors.

## 2. Mathematical background

A finite field is a commutative ring (with unity) in which all nonzero elements have a multiplicative inverse [5]. The finite field $GF(2^m)$ is the extension field of $GF(2)$ and has $2^m$ elements where $m$ is a nonzero positive integer. Also, all nonzero elements of $GF(2^m)$ can be uniquely represented with a polynomial degree up to $m-1$ with coefficients in $GF(2)$. Hence, polynomial or standard basis representation of an element in $GF(2^m)$ can be written as

$$u_{m-1}x^{m-1} + u_{m-2}x^{m-2} + \cdots + u_1x + u_0 \tag{1}$$

where $u_i \in GF(2)$ and $x$ denotes the primitive element used to construct the finite field $GF(2^m)$. In the finite field $GF(2^m)$, the addition (and subtraction) of two field elements is defined as polynomial addition. Thus, it is executed by modulo 2 addition (XOR operation) for every coefficient. On the other hand, multiplication of two field elements in $GF(2^m)$ is defined as a polynomial multiplication modulo $p(x)$, which is an irreducible polynomial of degree $m$. The present study concentrates on the finite field $GF(2^4)$, where the irreducible polynomial over $GF(2)$ is $x^4+x+1$. A compact representation of an element $u \in GF(2^4)$ uses hexadecimal digits (denoted with subscript $h$), expressing the coefficients of the polynomial representation. For example, $x^3 + x = A_h$ is in the finite field $GF(2^4)$. Information on finite fields is described elsewhere [11,12].

The following example denotes how a 4 × 4 binary matrix corresponds to a given element of $GF(2^4)$.

**Example 1.** Let $GF(2^4)$ be defined by the primitive polynomial $p(x) = x^4 + x + 1$. Let $x$ be a root of $p(x)$. Then, for any $u \in GF(2^4)$, we can write $u = u_3x^3 + u_2x^2 + u_1x + u_0$, where $u_i \in GF(2)$ and $\{x_3, x_2, x_1, x_0\} = \left\{x^3, x^2, x^1, 1\right\}$ is a polynomial basis of $GF(2^4)$ over $GF(2)$. A finite field multiplication (denoted with symbol $\otimes$) of $2_h$ or $x$ by any $u \in GF(2^4)$ can be given as

$$\begin{aligned}(2_h \otimes u) \bmod p(x) &= (x \otimes u) \bmod p(x)\\ &= (u_3x^4 + u_2x^3 + u_1x^2 + u_0x) \bmod p(x)\\ &= u_2x^3 + u_1x^2 + (u_3 + u_0)x + u_3,\end{aligned}$$

which corresponds to the 4 × 4 binary matrix

$$\begin{bmatrix}u'_0\\u'_1\\u'_2\\u'_3\end{bmatrix} = \begin{bmatrix}0&0&0&1\\1&0&0&1\\0&1&0&0\\0&0&1&0\end{bmatrix} \cdot \begin{bmatrix}u_0\\u_1\\u_2\\u_3\end{bmatrix}.$$

8 × 8 involutory or non-involutory matrices with the elements of $GF(2^4)$ can be transformed to 32 × 32 binary matrices by substituting the elements of $GF(2^4)$ with their corresponding 4 × 4 binary matrices. Generally, in the literature, MDS matrices used as diffusion layers are constructed by two types of matrices: circulant [6,13] and FFHadamard matrices [14]. In the present study, 8 × 8 FFHadamard matrices with distinct elements of $GF(2^4)$, which may have high branch numbers 7 or 8 (not MDS matrices), are used to construct cryptographically good 32 × 32 binary matrices since involutory binary matrices can easily be obtained by FFHadamard matrices as given in Lemma 2.

**Definition 1.** An 8 × 8 Finite Field Hadamard matrix with elements of $GF(2^m)$ can be represented as follows:

$$H = \text{had}(a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7) = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ a_1 & a_0 & a_3 & a_2 & a_5 & a_4 & a_7 & a_6 \\ a_2 & a_3 & a_0 & a_1 & a_6 & a_7 & a_4 & a_5 \\ a_3 & a_2 & a_1 & a_0 & a_7 & a_6 & a_5 & a_4 \\ a_4 & a_5 & a_6 & a_7 & a_0 & a_1 & a_2 & a_3 \\ a_5 & a_4 & a_7 & a_6 & a_1 & a_0 & a_3 & a_2 \\ a_6 & a_7 & a_4 & a_5 & a_2 & a_3 & a_0 & a_1 \\ a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \end{bmatrix}.$$

The following lemma is well-known and stated for convenience.

**Lemma 1.** *Let $a_0, a_1, \ldots, a_t$ be elements of $GF(2^m)$. Then*

$$(a_0 + a_1 + \cdots + a_t)^{2^k} = a_0^{2^k} + a_1^{2^k} + \cdots + a_t^{2^k} \tag{2}$$

*for $k = 1, 2, 3, \ldots$.*

**Lemma 2.** *Let $A = \text{had}(a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$ be an 8 × 8 FFHadamard matrix with the elements of $GF(2^m)$. Then, $A$ is the involutory matrix if and only if $\sum_{i=0}^{7} a_i = 1$, where the summation involves elements of $GF(2^m)$ and the addition is the addition of polynomials with binary coefficients. Binary coefficients are added modulo 2.*

**Proof.** As shown in Eq. (3), the identity matrix can be obtained if $\sum_{i=0}^{7} a_i^2 = 1$:

$$A^2 = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ a_1 & a_0 & a_3 & a_2 & a_5 & a_4 & a_7 & a_6 \\ a_2 & a_3 & a_0 & a_1 & a_6 & a_7 & a_4 & a_5 \\ a_3 & a_2 & a_1 & a_0 & a_7 & a_6 & a_5 & a_4 \\ a_4 & a_5 & a_6 & a_7 & a_0 & a_1 & a_2 & a_3 \\ a_5 & a_4 & a_7 & a_6 & a_1 & a_0 & a_3 & a_2 \\ a_6 & a_7 & a_4 & a_5 & a_2 & a_3 & a_0 & a_1 \\ a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \end{bmatrix} \cdot \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ a_1 & a_0 & a_3 & a_2 & a_5 & a_4 & a_7 & a_6 \\ a_2 & a_3 & a_0 & a_1 & a_6 & a_7 & a_4 & a_5 \\ a_3 & a_2 & a_1 & a_0 & a_7 & a_6 & a_5 & a_4 \\ a_4 & a_5 & a_6 & a_7 & a_0 & a_1 & a_2 & a_3 \\ a_5 & a_4 & a_7 & a_6 & a_1 & a_0 & a_3 & a_2 \\ a_6 & a_7 & a_4 & a_5 & a_2 & a_3 & a_0 & a_1 \\ a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \end{bmatrix}$$

$$= \begin{bmatrix} \sum\limits_{i=0}^{7} a_i^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sum\limits_{i=0}^{7} a_i^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sum\limits_{i=0}^{7} a_i^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sum\limits_{i=0}^{7} a_i^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sum\limits_{i=0}^{7} a_i^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sum\limits_{i=0}^{7} a_i^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sum\limits_{i=0}^{7} a_i^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sum\limits_{i=0}^{7} a_i^2 \end{bmatrix}. \tag{3}$$

Using Lemma 1, if $\sum_{i=0}^{7} a_i^2 = 1$, then $\sum_{i=0}^{7} a_i = 1$. Since $A$ is unitary ($A^{-1} = A$) and symmetric ($A = A^T$), matrix $A$ is involutory. □

**Definition 2** (*[15]*)**.** Two $n \times n$ binary matrices $A, B$ are permutation homomorphic to each other if there exists a row permutation $\rho$ and a column permutation $\gamma$ satisfying

$$\rho(\gamma(A)) = \gamma(\rho(A)) = B. \tag{4}$$

Please cite this article in press as: M.T. Sakallı, B. Aslan, On the algebraic construction of cryptographically good 32 × 32 binary linear transformations, Journal of Computational and Applied Mathematics (2013), http://dx.doi.org/10.1016/j.cam.2013.05.008

**Lemma 3** (*[15]*)**.** *If two binary matrices $A, B$ are permutation homomorphic to each other, then $A, B$ are of the same branch number.*

By Lemma 3, the branch number is the same for any row or column permutation, thus many matrices can be generated by using a binary matrix of maximum branch number. In the present study, two special permutations are used:

(1-) to rotate cyclically $l$ bits, where $l \in \{1, \dots, n-1\}$, to the right of all rows of an $n \times n$ binary matrix,
(2-) to rotate cyclically $l$ bits, where $l \in \{1, \dots, n-1\}$, to the downwards of all columns of an $n \times n$ binary matrix.

Note that these special permutations satisfy commutativity requirement in Eq. (4). More binary matrices of the same branch number but with different number of fixed points can be generated by applying special permutations 1 or 2 to any binary matrix. On the other hand, more binary matrices of the same branch number with the same number of fixed points can be generated when simultaneous cyclic rotations of 1 and 2 with the same number of $l$ bits are applied to any binary matrix. Note that these two characteristics related with the number of fixed points were observed experimentally. Suppose a 32 × 32 binary matrix of branch number 11 with one fixed point. Then, 31 more binary matrices of branch number 11 with one fixed point can be generated by applying special permutations 1 and 2 together with the same number of $l$ bits, where $l \in \{1, \dots, 31\}$.

It is stated that if the number of fixed points in a linear transformation greatly exceed the expected number for a random linear transformation, then this is an indication of poor diffusion of the linear transformation since the bits in these blocks are left intact when producing the output blocks [7]. Note also that the expected number of fixed points in a random permutation is one [7]. Consider an input block to a linear transformation $A$ formed by $m$-bit values in the field $GF(2^m)$. Let $A$ be an $n \times n$ matrix $A = (a_{ij})_{n\times n}$ where $a_{i,j} \in GF(2)$ or $a_{i,j} \in GF(2^m)$ and let $I$ denote the $n \times n$ identity matrix. Then, the set of all fixed points for that linear transformation can be obtained by solving the following equation

$$(A + I) \cdot x^T = 0 \tag{5}$$

where 0 is the all-zero vector of length $n$. Hence, the number of fixed points can be given as [7]

$$F_A = 2^{m(n-\mathrm{rank}(A+I))}. \tag{6}$$

Therefore, it is clear that the linear transformation $A$ has the lower number of fixed points if the matrix $(A + I)$ has bigger rank (Eq. (6)). For instance, the 16 × 16 involutory binary matrix of the ARIA has $2^{72}$ fixed points since the rank of the matrix $(A_{ARIA} + I)$ is 7.

In the present study, an SPN structure consisting of a number of rounds of the same 32 8-bit S-box connected by a 32 × 32 binary matrix is considered. Fig. 1 shows one round function of an assumed block cipher. The S-box considered in this structure has the same cryptographic properties with that of the AES (Advanced Encryption Standard) [6,13]. The definitions below are given to show the resistance against differential cryptanalysis and linear cryptanalysis of an assumed block cipher with a 256-bit block and a 256-bit key size. It should be noted that 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. In the present study, S-boxes are assumed to be bijective mappings defined on $\mathbb{Z}_2^m$.

**Definition 3** (*[16]*)**.** 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.

**Definition 4** (*[16]*)**.** The branch number of a diffusion layer is the minimum number of active S-boxes in the 2-round SPN.

**Definition 5.** The Hamming weight of a code word $c$ is the number of nonzero components in $c$ and denoted by $wt(c)$.

**Definition 6** (*[6]*)**.** The differential branch number of an $n \times n$ matrix $A : (\{0, 1\}^m)^n \to (\{0, 1\}^m)^n$ is defined by

$$\beta_d(A) = \min\{wt(x) + wt(A \cdot x^T) | x \in (\{0, 1\}^m)^n, x \neq 0\} \tag{7}$$

In applications, $n$ represents the number of S-boxes in a diffusion layer $A$ in the form of matrix $A$ and the size of each input and output of each S-box is $m$-bit. On the other hand, the linear branch number of an $n \times n$ matrix $A$ is related with the transposition of the matrix $A$ $(A^T)$ as given in Definition 7 [6]. In the present study, 32 × 32 binary matrices having the same differential and linear branch numbers are considered. That means branch numbers of 32 × 32 binary matrices and the transpose of them are equal.

**Definition 7** (*[6]*)**.** The linear branch number of an $n \times n$ matrix $A$ is defined by

$$\beta_l(A) = \min\{wt(x) + wt(A^T \cdot x^T) | x \in (\{0, 1\}^m)^n, x \neq 0\} \tag{8}$$

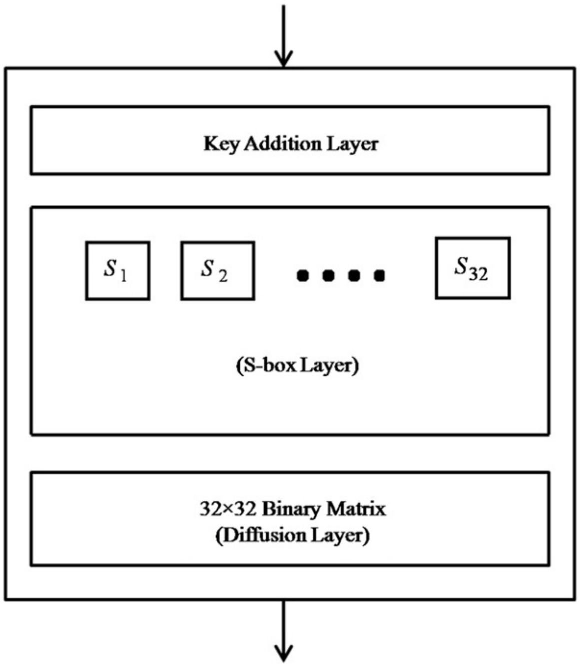

**Fig. 1.** One round function of an assumed block cipher.

**Definition 8** ([16]). Let $S : \mathbb{Z}_2^m \to \mathbb{Z}_2^m$ be an S-box. For any given $a, b, \Gamma_a, \Gamma_b \in \mathbb{Z}_2^m$, the differential and linear probability for the S-box are defined as

$$DP_S(a,b) = \frac{\#\{x \in \mathbb{Z}_2^m | S(x) \oplus S(x \oplus a) = b\}}{2^m} \tag{9}$$

$$LP_S(\Gamma_a, \Gamma_b) = \left(\frac{\#\{x \in \mathbb{Z}_2^m | \Gamma_a \bullet x = \Gamma_b \bullet S(x)\}}{2^{m-1}} - 1\right)^2 \tag{10}$$

where $x \bullet y$ denotes the parity (0 or 1) of the bitwise product of $x$ and $y$.

The $a$ and $b$ are called the input and output difference, respectively, for the S-box. In addition, the $\Gamma_a$ and $\Gamma_b$ are called the input mask and output mask, respectively, for the S-box.

**Definition 9** ([16]). The maximum differential and linear probability of an S-box are defined as

$$p = \max_{a \neq 0, b} DP_S(a,b) \tag{11}$$

$$q = \max_{\Gamma_a, \Gamma_b \neq 0} LP_S(\Gamma_a, \Gamma_b) \tag{12}$$

Note that the maximum differential probability ($p$) and the maximum linear probability ($q$) of the AES S-box can be obtained as $2^{-6}$ [16].

## 3. Algebraic construction of cryptographically good 32 × 32 binary linear transformations

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 since the upper and lower bounds are equal. For example, the maximum branch number for $8 \times 8$ matrices is 5 since both the upper and the lower bound for these matrices are 5. However, the lower and upper bounds may or may not be equal for $n \times n$ ($n > 18$) matrices [1]. For example, the lower bound for $32 \times 32$ binary matrices is 12 while the upper bound is 16. The method presented herein is successful for generating $32 \times 32$ binary matrices of branch number 12 (lower bound) and $32 \times 32$ binary matrices of branch number 11 (not lower bound) with one fixed point.

FFHadamard matrices are useful for constructing involutory diffusion transformations in the design of block ciphers. 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 [17]. FFHadamard matrices can also be implemented on 32-bit and 64-bit processors very efficiently. On the other hand, if these matrices are used with distinct elements in each row, matrices with high branch numbers may be obtained.

In the present study, $8 \times 8$ FFHadamard matrices with distinct elements of $GF(2^4)$ except for 0, which are also of branch numbers 7 or 8, are used in order to construct $32 \times 32$ involutory binary matrices of branch number 12 and non-involutory binary matrices of branch number 11. In this way, it is aimed to generate $32 \times 32$ binary matrices of high branch numbers and to use $4 \times 4$ binary matrices as the basis in the implementation on 32-bit and 64-bit processors. Since $8 \times 8$ FFHadamard matrices with the distinct elements of $GF(2^4)$ except for 0 are used, $\binom{15}{8} = 6435$ (400 of these classes are involutory) different classes are achieved where each class includes 40 320 (8!) members corresponding to the permutations of the elements of a class. Note also that each class with 8 elements is represented with hexadecimal values in ascending order. In fact, there are two reasons for this classification. The first one is that the Hamming weights of binary matrices constructed from distinct class elements affecting the number of XOR operations used to implement the binary matrix may be different from each other. The second one is that the branch number values of binary matrices constructed from the members of a class may again be different from each other. For example, the binary matrix constructed from the class representative $\text{had}(1_h, 2_h, 5_h, 7_h, A_h, B_h, E_h, F_h)$ is of branch number 8 while the binary matrix constructed from $\text{had}(1_h, 2_h, 5_h, B_h, 7_h, F_h, A_h, E_h)$, a member of this class, is of branch number 12.

In Example 2, a $32 \times 32$ involutory binary matrix is constructed from an $8 \times 8$ involutory matrix $A$ that satisfies four restrictions simultaneously such that:

(i) The $8 \times 8$ matrix $A$ should be involutory as given in Lemma 2.
(ii) The $32 \times 32$ binary matrix, $A_{\text{Binary}}$, transformed from the $8 \times 8$ involutory matrix $A$ should be of branch number 12.
(iii) The $8 \times 8$ involutory matrix $A$ should be chosen such that the rank of the $(A + I)$ matrix should be 4, which is in fact the highest achievable rank ($n/2$ for an $n \times n$ involutory matrix). Since the elements of $GF(2^4)$ are used to construct the $32 \times 32$ binary matrix, the rank of the matrix $(A_{\text{Binary}} + I)$ becomes 16. Therefore, if it is used as a 256-bit to a 256-bit linear transformation, where each input element is in $GF(2^8)$, the binary linear transformation has $2^{128}$ fixed points.
(iv) The elements of $8 \times 8$ matrix $A$ in $GF(2^4)$ should be chosen such that each row and column of the transformed binary matrix should have the Hamming weight equal to 15.

**Example 2.** Let $A = \text{had}(1_h, 2_h, 4_h, B_h, 6_h, C_h, 8_h, F_h)$ be an involutory $8 \times 8$ FFHadamard matrix, which is of branch number 7. It can be transformed into the $32 \times 32$ binary linear transformation satisfying the restrictions above as follows:

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

Please cite this article in press as: M.T. Sakallı, B. Aslan, On the algebraic construction of cryptographically good 32 × 32 binary linear transformations, Journal of Computational and Applied Mathematics (2013), http://dx.doi.org/10.1016/j.cam.2013.05.008

In Example 3, a $32 \times 32$ binary matrix from an $8 \times 8$ non-involutory matrix $B$ that satisfies four restrictions simultaneously is constructed such that:

(i) The $8 \times 8$ matrix $B$ should be nonsingular.

(ii) The $32 \times 32$ binary matrix, $B_{\text{Binary}}$, transformed from the $8 \times 8$ matrix $B$ should be of branch number 11.

(iii) The $8 \times 8$ matrix $B$ should be chosen such that the rank of the $(B + I)$ matrix should be 8. Since the elements of $GF(2^4)$ are used to construct the $32 \times 32$ binary matrix, the rank of the matrix $(B_{\text{Binary}} + I)$ becomes 32. Therefore, the binary matrix has one fixed point, which is all-zero input $(0, 0, 0, \ldots, 0)$.

(iv) The Hamming weight of the binary matrix, which impacts the needed number of XOR operations in the implementation, should be the lowest one.

**Example 3.** Let $B = \text{had}(1_h, 2_h, 4_h, 8_h, 9_h, B_h, C_h, D_h)$ be a non-involutory $8 \times 8$ FFHadamard matrix, which is of branch number 7. It can be transformed into the $32 \times 32$ binary linear transformation satisfying the restrictions above as follows:

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

In a straight coding on an 8-bit processor, the binary matrices given in Examples 2 and 3 require 448 and 392 byte XORs, respectively, for the implementation when input elements to the binary matrices are considered as byte values. The total number of byte XORs can be reduced to 328 (Appendix A) and 320, respectively, by adding 8 more variables to the implementation for both. It should also be noted that the inverse of the $32 \times 32$ binary matrix in Example 3 is constructed from the $8 \times 8$ matrix $\text{had}(8_h, 3_h, 6_h, C_h, 4_h, 7_h, A_h, 2_h)$ and is also of branch number 11. It requires 528 byte XORs in a straight coding on an 8-bit processor. But, the total number of byte XORs can be reduced to 363 by adding 12 more variables to the implementation. On the other hand, the advantage of the FFHadamard form of the $4 \times 4$ binary matrices can be used to implement these matrices on a 32-bit and a 64-bit processor.

## 4. Security assessment of an assumed block cipher with a 256-bit block and key size against differential and linear cryptanalysis

Consider a $2r$-round SPN where a round consists of an S-box layer followed by the $32 \times 32$ involutory binary matrix in Example 2 as a diffusion layer (Fig. 1). Also, consider the S-box layer consisting of the same 32 8-bit S-box which has the maximum differential probability and the maximum linear probability $2^{-6}$, which is the same value with the AES S-box. Then, the maximum probabilities of the differential ($p_D$) and linear characteristic ($q_L$) for $2r$-round SPN are as follows:

$$p_D^{2r} \leq (2^{-6})^{(r\times\beta_A)}, \qquad q_L^{2r} \leq (2^{-6})^{(r\times\beta_A)}$$

where $\beta_A$ denotes the branch number of a diffusion layer $A$. The maximum differential and linear probabilities of 2-round SPN is bounded by $(2^{-6})^{(1\times 12)} = 2^{-72}$ because the branch number of the involutory binary matrix is 12 and therefore the number of minimum active S-box is 12 in the 2-round SPN. In this context, the minimum number of rounds needed for the block cipher with 256-bit key size to be secure against differential and linear cryptanalysis is 8, because the maximum differential and linear probabilities of 8-round SPN is bounded by $(2^{-6})^{(4\times 12)} = 2^{-288} \leq 2^{-256}$. Notice that if the 512-bit key is used, then the minimum number of rounds to be secure against differential and linear cryptanalysis will be 16, because the maximum differential and linear probabilities of 16-round SPN will be bounded by $(2^{-6})^{(8\times 12)} = 2^{-576} \leq 2^{-512}$.

In the same manner, if the 32 × 32 non-involutory binary matrix in Example 3 is used as a diffusion layer in the same structure, the minimum number of rounds needed for the block cipher with 256-bit key size to be secure against differential and linear cryptanalysis will again be 8, because the maximum differential and linear probabilities of 8-round SPN will be bounded by $(2^{-6})^{(4\times 11)} = 2^{-264} \leq 2^{-256}$. The maximum differential and linear probabilities of 12-round SPN with the same 32 4-bit S-box are given in Remark 1.

**Remark 1.** If the block cipher with the 32 × 32 involutory binary matrix is assumed with the same 32 4-bit S-box for which the maximum differential probability and the maximum linear probability are $2^{-2}$, then the minimum number of rounds needed for the block cipher with the 128-bit block and the 128-bit key to be secure against differential and linear cryptanalysis will be 12. This is because the maximum differential and linear probabilities of 12-round SPN will be bounded by $(2^{-2})^{(6\times 12)} = 2^{-144} \leq 2^{-128}$.

## 5. Conclusions

In the present study, a new algebraic construction method for generating 32 × 32 matrices of branch number 12 and branch number 11, which can be used to transform a 256-bit input block to a 256-bit output block, is presented. In Example 2, an involutory binary matrix of branch number 12 with suitable implementation properties is constructed from a member of the class had$(1_h, 2_h, 4_h, 6_h, 8_h, B_h, C_h, F_h)$. After searching for all members of this class, a total of 2688 binary matrices (members) of branch number 12 were found. A total of 86 016 (32 × 2688) involutory binary matrices of branch number 12 can be determined by simultaneous application of special permutations 1 and 2 (Section 2) to these involutory binary matrices. It should be noted that a non-involutory binary matrix of branch number 12 with reduced fixed points but not with one fixed point can be determined by using special permutations 1 or 2 and any involutory binary matrix of branch number 12. Moreover, 8 more classes were determined by searching all classes satisfying the criteria in Example 2. There were some classes, which did not satisfy criterion (iv) in Example 2. These classes include members, which can be transformed into the involutory binary matrices of branch number 12. But, these classes have bigger Hamming weights and therefore do not possess suitable implementation properties.

9 classes satisfying the criteria given in Example 2 were found again after searching for the 8 × 8 matrices with the elements of $GF(2^4)$ defined by the irreducible polynomial $x^4 + x^3 + 1$. Therefore, these classes include members which can be transformed into the involutory binary matrices of branch number 12. 8 × 8 matrices with the elements defined by the irreducible polynomial $x^4+x^3+x^2+x+1$ to transform into the 32 × 32 binary matrices with good cryptographic properties were not searched in depth.

Finally, it was shown that (Section 4) the given binary matrices (with an 8-bit S-box having the same cryptographic properties with that of AES) for a 256-bit block cipher are resistant against linear and differential cryptanalysis when applied in a reasonable number of rounds. A further security analysis should be performed to analyze the resistance of the given binary matrices against other important attacks such as truncated differential cryptanalysis and impossible differential cryptanalysis. In this context, the existence of an additional linear transformation similar to the ShiftRows transformation in the AES block cipher can be questioned, if so, it may be used to further improve the assumed block cipher.

## Acknowledgments

The authors would like to express their gratitude to the anonymous reviewers for their invaluable suggestions in putting the present study into its final form. Thanks are also due to Prof. Dr. Vincent Rijmen, Dr. Nicky Mouha and Dr. Orhun Kara for their invaluable input.

## Appendix. 8-bit implementation of the 32 × 32 binary matrix given in Example 2

If the 32 × 32 binary matrix given in Example 2 is implemented into an 8-bit processor, then $A$ is represented by byte XORs of binary vectors as follows:

$$A \cdot x = y,$$

where $x = (x_0, x_1, \ldots, x_{31})^T$, $y = (y_0, y_1, \ldots, y_{31})^T$ with $x_i, y_i \in GF(2^8)$, $i = 0, 1, \ldots, 31$. Note also that $T_0, T_1, \ldots, T_7$ are additional variables used to reduce the number of operations to 328 XORs. Then,

$$\begin{aligned}
T_0 &= x_7 \oplus x_{10} \oplus x_{13} \oplus x_{19} \oplus x_{25} \oplus x_{30},\\
T_1 &= x_1 \oplus x_4 \oplus x_{14} \oplus x_{16} \oplus x_{21} \oplus x_{26},\\
T_2 &= x_2 \oplus x_5 \oplus x_{15} \oplus x_{17} \oplus x_{22} \oplus x_{27},\\
T_3 &= x_6 \oplus x_9 \oplus x_{12} \oplus x_{18} \oplus x_{24} \oplus x_{29},\\
T_4 &= x_3 \oplus x_9 \oplus x_{14} \oplus x_{23} \oplus x_{26} \oplus x_{29},\\
T_5 &= x_0 \oplus x_5 \oplus x_{10} \oplus x_{17} \oplus x_{20} \oplus x_{30},\\
T_6 &= x_1 \oplus x_6 \oplus x_{11} \oplus x_{18} \oplus x_{21} \oplus x_{31},\\
T_7 &= x_2 \oplus x_8 \oplus x_{13} \oplus x_{22} \oplus x_{25} \oplus x_{28},\\
y_0 &= T_0 \oplus x_0 \oplus x_{12} \oplus x_{15} \oplus x_{18} \oplus x_{21} \oplus x_{22} \oplus x_{28} \oplus x_{29} \oplus x_{31},\\
y_1 &= T_1 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{15} \oplus x_{18} \oplus x_{23} \oplus x_{25} \oplus x_{28},\\
y_2 &= T_2 \oplus x_8 \oplus x_{11} \oplus x_{13} \oplus x_{16} \oplus x_{19} \oplus x_{20} \oplus x_{26} \oplus x_{28} \oplus x_{29},\\
y_3 &= T_3 \oplus x_3 \oplus x_{14} \oplus x_{17} \oplus x_{20} \oplus x_{21} \oplus x_{23} \oplus x_{27} \oplus x_{28} \oplus x_{30},\\
y_4 &= T_4 \oplus x_4 \oplus x_8 \oplus x_{11} \oplus x_{17} \oplus x_{18} \oplus x_{22} \oplus x_{24} \oplus x_{25} \oplus x_{27},\\
y_5 &= T_5 \oplus x_3 \oplus x_8 \oplus x_{11} \oplus x_{14} \oplus x_{15} \oplus x_{19} \oplus x_{22} \oplus x_{24} \oplus x_{29},\\
y_6 &= T_6 \oplus x_9 \oplus x_{12} \oplus x_{15} \oplus x_{16} \oplus x_{20} \oplus x_{23} \oplus x_{24} \oplus x_{25} \oplus x_{30},\\
y_7 &= T_7 \oplus x_7 \oplus x_{10} \oplus x_{16} \oplus x_{17} \oplus x_{19} \oplus x_{21} \oplus x_{24} \oplus x_{26} \oplus x_{31},\\
y_8 &= T_2 \oplus x_4 \oplus x_7 \oplus x_8 \oplus x_{20} \oplus x_{21} \oplus x_{23} \oplus x_{26} \oplus x_{29} \oplus x_{30},\\
y_9 &= T_3 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_{15} \oplus x_{17} \oplus x_{20} \oplus x_{26} \oplus x_{31},\\
y_{10} &= T_0 \oplus x_0 \oplus x_3 \oplus x_5 \oplus x_{18} \oplus x_{20} \oplus x_{21} \oplus x_{24} \oplus x_{27} \oplus x_{28},\\
y_{11} &= T_1 \oplus x_6 \oplus x_{11} \oplus x_{19} \oplus x_{20} \oplus x_{22} \oplus x_{25} \oplus x_{28} \oplus x_{29} \oplus x_{31},\\
y_{12} &= T_6 \oplus x_0 \oplus x_3 \oplus x_{12} \oplus x_{16} \oplus x_{17} \oplus x_{19} \oplus x_{25} \oplus x_{26} \oplus x_{30},\\
y_{13} &= T_7 \oplus x_0 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_{11} \oplus x_{16} \oplus x_{21} \oplus x_{27} \oplus x_{30},\\
y_{14} &= T_4 \oplus x_1 \oplus x_4 \oplus x_7 \oplus x_{16} \oplus x_{17} \oplus x_{22} \oplus x_{24} \oplus x_{28} \oplus x_{31},\\
y_{15} &= T_5 \oplus x_2 \oplus x_{15} \oplus x_{16} \oplus x_{18} \oplus x_{23} \oplus x_{24} \oplus x_{25} \oplus x_{27} \oplus x_{29},\\
y_{16} &= T_4 \oplus x_2 \oplus x_5 \oplus x_6 \oplus x_{12} \oplus x_{13} \oplus x_{15} \oplus x_{16} \oplus x_{28} \oplus x_{31},\\
y_{17} &= T_5 \oplus x_2 \oplus x_7 \oplus x_9 \oplus x_{12} \oplus x_{23} \oplus x_{26} \oplus x_{27} \oplus x_{28} \oplus x_{31},\\
y_{18} &= T_6 \oplus x_0 \oplus x_3 \oplus x_4 \oplus x_{10} \oplus x_{12} \oplus x_{13} \oplus x_{24} \oplus x_{27} \oplus x_{29},\\
y_{19} &= T_7 \oplus x_1 \oplus x_4 \oplus x_5 \oplus x_7 \oplus x_{11} \oplus x_{12} \oplus x_{14} \oplus x_{19} \oplus x_{30},\\
y_{20} &= T_0 \oplus x_1 \oplus x_2 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{11} \oplus x_{20} \oplus x_{24} \oplus x_{27},\\
y_{21} &= T_1 \oplus x_3 \oplus x_6 \oplus x_8 \oplus x_{13} \oplus x_{19} \oplus x_{24} \oplus x_{27} \oplus x_{30} \oplus x_{31},\\
y_{22} &= T_2 \oplus x_0 \oplus x_4 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{14} \oplus x_{25} \oplus x_{28} \oplus x_{31},\\
y_{23} &= T_3 \oplus x_0 \oplus x_1 \oplus x_3 \oplus x_5 \oplus x_8 \oplus x_{10} \oplus x_{15} \oplus x_{23} \oplus x_{26},\\
y_{24} &= T_6 \oplus x_4 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{13} \oplus x_{14} \oplus x_{20} \oplus x_{23} \oplus x_{24},\\
y_{25} &= T_7 \oplus x_1 \oplus x_4 \oplus x_{10} \oplus x_{15} \oplus x_{18} \oplus x_{19} \oplus x_{20} \oplus x_{23} \oplus x_{31},\\
y_{26} &= T_4 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_8 \oplus x_{11} \oplus x_{12} \oplus x_{16} \oplus x_{19} \oplus x_{21},\\
y_{27} &= T_5 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_9 \oplus x_{12} \oplus x_{13} \oplus x_{15} \oplus x_{22} \oplus x_{27},\\
y_{28} &= T_2 \oplus x_0 \oplus x_1 \oplus x_3 \oplus x_9 \oplus x_{10} \oplus x_{14} \oplus x_{16} \oplus x_{19} \oplus x_{28},\\
y_{29} &= T_3 \oplus x_0 \oplus x_5 \oplus x_{11} \oplus x_{14} \oplus x_{16} \oplus x_{19} \oplus x_{22} \oplus x_{23} \oplus x_{27},\\
y_{30} &= T_0 \oplus x_0 \oplus x_1 \oplus x_6 \oplus x_8 \oplus x_{12} \oplus x_{15} \oplus x_{17} \oplus x_{20} \oplus x_{23},\\
y_{31} &= T_1 \oplus x_0 \oplus x_2 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{11} \oplus x_{13} \oplus x_{18} \oplus x_{31}.
\end{aligned}$$

## References

[1] D. Kwon, S.H. Sung, J.H. Song, S. Park, Design of block ciphers and coding theory, Trends in Mathematics 8 (1) (2005) 13–20.

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

[3] D. Kwon, J. Kim, S. Park, S.H. Sung, Y. Sohn, J.H. Song, Y. Yeom, E.-J. Yoon, S. Lee, J. Lee, S. Chee, D. Han, J. Hong, New block cipher: ARIA, in: Proceedings of International Conference on Information Security and Cryptology, in: Lecture Notes in Computer Science, vol. 2971, Springer, 2004, pp. 432–445.

[4] B.W. Koo, H.S. Jang, J.H. Song, 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, in: Lecture Notes in Computer Science, vol. 4296, Springer, 2006, pp. 51–64.

[5] J. Nakahara Jr., E. Abrahão, A new involutory MDS matrix for the AES, International Journal of Network Security 9 (2) (2009) 109–116.

[6] J. Daemen, V. Rijmen, The Design of Rijndael, AES—The Advanced Encryption Standard, Springer, 2002.

[7] M.R. Z'aba, Analysis of linear relationships in block ciphers, Ph.D. Thesis, Queensland University of Technology, Brisbane, Australia, 2010.

[8] M. Matsui, Linear cryptanalysis method for DES cipher, in: Proceedings of EUROCRYPT 93, in: Lecture Notes in Computer Science, vol. 765, Springer, 1994, pp. 386–397.

[9] E. Biham, A. Shamir, Differential cryptanalysis of DES-like cryptosystems, in: Proceedings of CRYPTO'90, in: Lecture Notes in Computer Science, vol. 537, Springer, 1990, pp. 2–21.

[10] B. Aslan, M.T. Sakallı, Algebraic construction of cryptographically good binary linear transformations, Security and Communication Networks (2012). http://dx.doi.org/10.1002/sec.556.

[11] R.J. McEliece, Finite Fields for Computer Scientists and Engineers, Kluwer Academic Publishers, Dordrecht, 1987.

[12] R. Lidl, H. Niederreiter, Finite Fields, in: Encyclopedia of Mathematics and its Applications, Addison-Wesley, Reading, Massachusetts, 1983.

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

[14] P.S.L.M. Barreto, V. Rijmen, The Khazad legacy-level block cipher, in: Proceedings of First open NESSIE Workshop, 2000.

[15] B.W. Koo, H.S. Jang, J.H. Song, Constructing and cryptanalysis of a 16 × 16 binary matrix as a diffusion layer, in: Proceedings of Information Security Applications: 4th International Workshop, WISA 2003, in: Lecture Notes in Computer Science, vol. 2908, Springer, 2003, pp. 489–503.

[16] K. Chun, S. Kim, S. Lee, S.H. Sung, S. Yoon, Differential and linear cryptanalysis for 2-round SPNs, Information Processing Letters 87 (2003) 277–282.

[17] M. Sajadieh, M. Dakhilalian, H. Mala, B. Omoomi, On construction of involutory MDS matrices from Vandermonde matrices in $GF(2^q)$, Designs, Codes and Cryptography 64 (3) (2012) 287–308.

Please cite this article in press as: M.T. Sakallı, B. Aslan, On the algebraic construction of cryptographically good 32 × 32 binary linear transformations, Journal of Computational and Applied Mathematics (2013), http://dx.doi.org/10.1016/j.cam.2013.05.008