Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 275 – 287

doi:10.3906/elk-1906-151

On the automorphisms and isomorphisms of MDS matrices and their efficient implementations

Muharrem Tolga SAKALLI[1,*], **Sedat AKLEYLEK**[2], **Kemal AKKANAT**[1], **Vincent RIJMEN**[3]
[1]Department of Computer Engineering, Faculty of Engineering, Trakya University, Edirne, Turkey
[2]Department of Computer Engineering, Faculty of Engineering, Ondokuz Mayıs University, Samsun, Turkey
[3]Department of ESAT/COSIC, KU Leuven and imec, Leuven, Belgium

Received: 24.06.2019 • **Accepted/Published Online:** 23.09.2019 • **Final Version:** 27.01.2020

Abstract: In this paper, we explicitly define the automorphisms of MDS matrices over the same binary extension field. By extending this idea, we present the isomorphisms between MDS matrices over $\mathbb{F}_{2^m}$ and MDS matrices over $\mathbb{F}_{2^{mt}}$, where $t \geq 1$ and $m > 1$, which preserves the software implementation properties in view of XOR operations and table lookups of any given MDS matrix over $\mathbb{F}_{2^m}$. Then we propose a novel method to obtain distinct functions related to these automorphisms and isomorphisms to be used in generating isomorphic MDS matrices (new MDS matrices in view of implementation properties) using the existing ones. The comparison with the MDS matrices used in AES, ANUBIS, and subfield-Hadamard construction shows that we generate an involutory 4×4 MDS matrix over $\mathbb{F}_{2^8}$ (from an involutory 4×4 MDS matrix over $\mathbb{F}_{2^4}$) whose required number of XOR operations is the same as that of ANUBIS and the subfield-Hadamard construction, and better than that of AES. The proposed method, due to its ground field structure, is intended to be a complementary method for the current construction methods in the literature.

Key words: MDS matrix, branch number, block cipher

1. Introduction

Confusion and diffusion as defined by Claude Shannon [1] are two important properties required for the design of block ciphers. Diffusion is provided by a linear transformation, which improves the avalanche characteristics of a block cipher. Maximum distance separable (MDS) matrices derived from MDS codes are used as diffusion layers in most of the block ciphers like the Advanced Encryption Standard (AES) [2] and hash functions like Whirlpool [3], the PHOTON family [4], and Whirlwind [5]. MDS matrices also provide security against differential cryptanalysis [6] and linear cryptanalysis [7] along with the use of a nonlinear layer (e.g., S-boxes) in a round function of a block cipher. Thus, it is important to find MDS matrices having nice implementation properties.

The methods to generate MDS matrices can be divided into two groups: direct construction methods and search-based methods. The former group contains the methods based on Cauchy matrices [8], companion matrices [4, 9], Vandermonde matrices [10, 11], shortened BCH codes [12, 13], and skewed recursive structures [14]. The latter group consists of several interesting ideas. These are to use recursive structures [15, 16], hybrid structures [17], and special matrix forms [9, 18, 19]. One of the easiest construction methods yielding efficient

*Correspondence: tolga@trakya.edu.tr

implementation is to use special matrix forms: circulant and finite field Hadamard (FFHadamard or simply Hadamard) matrices. Circulant matrices, a typical example of which is the AES MixColumns transformation, are preferred in MDS matrix constructions since each row of a circulant matrix differs from the previous row by a right shift, which provides a nice property to be implemented especially in hardware. On the other hand, Hadamard matrices are very useful in constructing involutory (self-inverse) MDS matrices. Involutory diffusion layers have an important effect on the performance of the block ciphers, since they provide the same implementation properties in encryption and decryption phases. In this respect, in [20], the authors proposed a generalization of the Hadamard matrix to generate new types of involutory/noninvolutory MDS matrices easily and they show that the idea used to generalize a Hadamard matrix can also be applied to $k \times k$ (involutory) MDS matrices, where k is not necessarily a power of 2.

In addition to construction methods described for MDS matrices, recently, MDS construction methods have evolved to find MDS matrices with minimal XOR counts [21], which is a metric used in the estimation of hardware implementation cost. In the literature, some studies focusing on generating MDS matrices with low/minimum XOR counts are given in [20, 22–24]. In this paper, we focus on a complementary method to generate isomorphic MDS matrices from existing ones to be applied to any MDS matrix generated by any construction method, which makes it generic over all construction methods. Also, implementation issues of MDS matrices are limited to software implementation properties in view of XOR operations, table lookups, or multiplication of byte a by 2_h (denoted by xtime (a), and 2_h corresponds to x in polynomial form).

1.1. Motivation

Generating MDS matrices over binary extension fields yielding efficient implementations is a challenging issue. We focus on the following question: Can we generate new MDS matrices via known MDS matrices by using algebraic construction methods? In [25], a method called subfield construction was proposed to generate an $m \times m$ MDS transform over $\mathbb{F}_{2^{sn}}$ from m $n \times n$ MDS transforms over $\mathbb{F}_{2^s}$. The method is mainly based on a divide-and-conquer idea and one MDS matrix from the known one can be generated by using this method. In [26] and [27], the transactions between MDS matrices are defined, but the explicit algorithms running in polynomial time are not given. There is a lack of a generic method generating isomorphic MDS matrices over different field representations, which may have better implementation properties than the existing ones, hence improving the efficiency of other construction methods. Moreover, many known methods such as those based on circulant and Hadamard matrices (except for Cauchy and Vandermonde matrix-based methods) to generate MDS matrices over large binary extension fields with efficient implementations involve some kind of time-consuming exhaustive search increasing computational complexity. Therefore, there is a need for a method to build MDS matrices over the large binary extension fields used in [5, 28] and at the same time provide MDS matrices with good software implementation properties in view of the required operations such as XOR operations, table lookups, or multiplication of byte a by 2_h (denoted by xtime (a), and 2_h corresponds to x in polynomial form).

1.2. Contribution

Our main contribution is to present a novel method that can be used to generate isomorphic MDS matrices (new MDS matrices in view of implementation properties over $\mathbb{F}_{2^{mt}}$, where $m > 1$ and $t \geq 1$) from MDS matrices over $\mathbb{F}_{2^m}$. The main idea is to obtain distinct functions related to the automorphisms and isomorphisms, which allow us to preserve the MDS property. Then these functions are applied to MDS matrices over $\mathbb{F}_{2^m}$ in order to generate new MDS matrices over $\mathbb{F}_{2^{mt}}$. Moreover, we generate an involutory 4×4 MDS matrix over $\mathbb{F}_{2^8}$ with

the same implementation cost as the involutory MDS matrix used in ANUBIS [29] and the subfield-Hadamard construction [17], and better implementation cost than the MDS matrix used in AES in view of the required number of XOR operations.

1.3. Organization

This paper is organized as follows: Section 2 describes how to find automorphisms and isomorphisms of MDS matrices over binary extension fields by using finite field properties. Then a generalization of this idea is studied and a novel method is presented, which defines distinct functions related to the isomorphisms to be used in generating new MDS matrices over $\mathbb{F}_{2^{mt}}$ (where $m > 1$ and $t \geq 1$) from MDS matrices over $\mathbb{F}_{2^m}$. In Section 3, some important properties of the method are given and implementation details of MDS matrices generated by the method are discussed. Section 4 concludes the paper.

2. The proposed method to generate MDS matrices over $\mathbb{F}_{2^{mt}}$ from MDS matrices over $\mathbb{F}_{2^m}$

In this section, we provide a detailed explanation to find distinct functions related to the automorphisms and isomorphisms of MDS matrices over binary extension fields. In Section 2.1, we investigate the automorphisms and distinct functions related to these automorphisms, which can be used in generating new MDS matrices from an MDS matrix over the same binary extension field. In Section 2.2, we generalize the idea given in Section 2.1 by defining the isomorphisms of MDS matrices over the same extension degree. In Section 2.3, we propose a novel method to obtain the isomorphisms and distinct functions related to these isomorphisms to be used in generating new MDS matrices over $\mathbb{F}_{2^{mt}}$, where $t \geq 1$, from an MDS matrix over $\mathbb{F}_{2^m}$.

In this paper, we focus on MDS matrices over $\mathbb{F}_{2^m}$. Let β be the primitive element used to construct the finite field $\mathbb{F}_{2^m}$, and then any finite field element is in the form $a_{m-1}\beta^{m-1} + a_{m-2}\beta^{m-2} + \ldots + a_1\beta + a_0$ with $a_i \in \{0, 1\}$. It can also be represented by the hexadecimal form of bits $(a_{m-1}a_{m-2}\ldots a_1a_0)$. Throughout this paper, we use the hexadecimal notation or powers of a primitive element (for nonzero elements) to represent the finite field elements.

Now we recall some facts about MDS matrices. Let C be an $[n, k, d]$ code; then C is called MDS if the equality $d = n - k + 1$ holds. MDS matrices are derived from MDS codes. They are used to provide diffusion in block ciphers and hash functions.

Definition 1 *[2] The differential and linear branch number of a $k \times k$ matrix A: $\mathbb{F}_{2^m}^k \rightarrow \mathbb{F}_{2^m}^k$ are defined by $B_d(A) = min\{wt(x) + wt(A \cdot x^T) | x \in \mathbb{F}_{2^m}^k - \{0\}\}$ and $B_l(A) = min\{wt(x) + wt(A^T \cdot x^T) | x \in \mathbb{F}_{2^m}^k - \{0\}\}$, where $wt(x)$ is the number of nonzero components in x, respectively.*

MDS matrices have the maximum differential and linear branch number ($k+1$ for $k \times k$ MDS matrices). Some important properties of MDS matrices can be given as follows:

1. A square matrix A is MDS if and only if every square submatrix of A is nonsingular.
2. The MDS property of a matrix is preserved upon permutations of rows/columns. Similarly, multiplication of a row/column of a matrix by a nonzero constant $c \in \mathbb{F}_{2^m}$ does not affect its MDS property. In general, the minimum distance d of an $[n, k, d]$ code C with generator matrix $G = [I|A]$, where A is a $k \times (n-k)$ matrix, is preserved after applying the above operations to A [30].
3. The MDS property of a matrix is preserved under the transpose operation [30].

2.1. MDS automorphisms

In this section, we investigate the automorphisms of MDS matrices over the same binary extension field and distinct functions related to these automorphisms with the help of properties 1 and 2. One can generate new MDS matrices over the same binary extension field by applying these automorphisms and distinct functions to any MDS matrix. In Proposition 1, we discuss the nonsingularity of these automorphisms.

Proposition 1 *Let A be a $k \times k$ matrix over the finite field $\mathbb{F}_{2^m}$. Let A' be generated by applying any distinct automorphism $f_i : b \mapsto b^{2^i}$ to the elements of A with $0 \leq i \leq m-1$ and $b \in \mathbb{F}_{2^m}^*$. Then the determinant of A' is equal to 0 if and only if the determinant of A is equal to 0.*

By Theorem 2.21 in [31], the automorphisms of $\mathbb{F}_{2^m}$ over $\mathbb{F}_2$ are given as b^{2^i} for all nonzero $b \in \mathbb{F}_{2^m}$ and $0 \leq i \leq m-1$. These mappings are one-to-one because each element in $\mathbb{F}_2$ maps to itself. Since the mappings are distinct, the determinant is related to the automorphism. Then the determinant of any matrix generated by applying any distinct automorphism to A remains unchanged, being either zero or nonzero, i.e. if $\det(A) \neq 0$ or $\det(A) = 0$, then $det(A') \neq 0$ or $\det(A') = 0$, respectively.

Proposition 1 is related to property 1 satisfied after applying the automorphism. The number of distinct automorphisms of $\mathbb{F}_{2^m}$ over $\mathbb{F}_2$ is m. On the other hand, one can obtain $m \cdot (2^m - 1)$ distinct and bijective functions with a constant multiplication after applying the automorphisms, i.e. $f_{i,c} : b \mapsto b^{2^i} \cdot c$ for any nonzero element $b \in \mathbb{F}_{2^m}$, $0 \leq i \leq m-1$, and $c \in \mathbb{F}_{2^m}^*$. In Theorem 1, we define distinct and bijective functions related to the automorphisms to be used in generating new MDS matrices.

Theorem 1 *There exist $m \cdot (2^m - 1)$ distinct and bijective functions related to the automorphisms in the form of $f_{i,c} : \beta \mapsto (\beta^{2^i}) \cdot c$, where β is any primitive element of $\mathbb{F}_{2^m}$, $c \in \mathbb{F}_{2^m}^*$, and $0 \leq i \leq m-1$. These functions preserve the MDS property of a square matrix over the same binary extension field, i.e. new MDS matrices are generated from the existing ones.*

Proof Here we need to show that the properties of being an MDS matrix are satisfied after applying distinct functions. The main idea depends on the fact that every square submatrix of an MDS matrix is nonsingular. We divide the proof into three parts. Note that all elements of an MDS matrix must be nonzero. Let $p(x)$ be an irreducible polynomial of degree m over $\mathbb{F}_2$ and $\beta \in \mathbb{F}_{2^m}$ be a primitive element. We divide the proof into three parts.

- Letting $f_i : x \mapsto x^{2^i}$, we have $\det A' = f_i(\det A)$. If $\det A \neq 0$, then $f_i(\det A) \neq 0$ since f_i is an automorphism.
- Let $g_c(x) \mapsto c \cdot x$, where $c \in \mathbb{F}_{2^m}^*$, and then $\det A' = c \cdot \det A$ from elementary linear algebra. Since $c \neq 0$ and $\det A \neq 0$, $\det A' \neq 0$.
- Now let $f_{i,c} = g_c(\beta) \circ f_i(\beta) = g_c(f_i(\beta)) = c \cdot \beta^{2^i}$. Then $\det A' = c \cdot f_i(\det A)$. Since $\det A \neq 0$, $\det A' \neq 0$.

Note that if $\det A' \neq 0$, then we obtain $\det A \neq 0$ by considering $\det A = f_{m-i}(\frac{1}{c} \det A')$. Since every square submatrix of A is invertible and each row or column of A is linearly independent, the MDS property is preserved. In conclusion, $\det A' \neq 0$ if and only if $\det A \neq 0$. □

In Example 1, we provide an automorphic involutory MDS matrix over $\mathbb{F}_{2^4}$ in Hadamard matrix form. Recall that a $k \times k$ Hadamard matrix A is constructed with $\alpha_0, \alpha_1, \ldots, \alpha_{k-1}$ such that $A_{i,j} = \alpha_{i \oplus j}$ for $0 \leq i, j \leq k-1$, and then $A = had(\alpha_0, \alpha_1, \ldots, \alpha_{k-1})$.

Example 1 *Let $\mathbb{F}_{2^4}$ be defined by the primitive polynomial $p(x) = x^4 + x + 1$. Let α be a root of $p(x)$. Then $M_1 = had(1_h, 2_h, 4_h, 6_h) = had(1, \alpha, \alpha^2, \alpha^5)$ is an involutory 4×4 MDS matrix. By Theorem 1, consider $f_{2,1} : \alpha \mapsto \alpha^4$ automorphism. Then the new involutory 4×4 MDS matrix generated from M_1 by $f_{2,1}$ is as follows: $M_1^{'} = had(1_h, 3_h, 5_h, 6_h) = had(1, \alpha^4, \alpha^8, \alpha^5)$. $M_1^{'}$ is called an automorphism of M_1 under $f_{2,1} : \alpha \mapsto \alpha^4$. Note that by Theorem 1 one can generate 59 more MDS matrices by using the MDS matrix M_1.*

2.2. MDS isomorphisms

In this section, we clearly define the isomorphisms and distinct functions related to these isomorphisms. Then we give a novel method to obtain these isomorphisms and distinct functions between MDS matrices over $\mathbb{F}_{2^m}$ and MDS matrices over $\mathbb{F}_{2^{mt}}$, where $m > 1$ and $t \geq 1$. The proposed idea is based on substituting the elements over the same binary extension field defined by different irreducible polynomials.

Proposition 2 *Let A be a $k \times k$ matrix over the finite field $\mathbb{F}_{2^m}/p_1(x)$ and β_1 be any primitive element of $\mathbb{F}_{2^m}/p_1(x)$. Let A' be a $k \times k$ matrix over the finite field $\mathbb{F}_{2^m}/p_2(x)$ generated by applying the isomorphism $f_{s_u} : \beta_1 \mapsto \beta_2^{s_u}$ to the elements of A (which can also be represented as β_1^d for $0 \leq d \leq 2^m - 2$), where β_2 is any primitive element of $\mathbb{F}_{2^m}/p_2(x)$, $s_u = e \cdot 2^i$ for $1 \leq e \leq 2^m - 2$, $gcd(e, 2^m - 1) = 1$, $p_1(\beta_2^{s_u}) = 0$, and $0 \leq u, i \leq m-1$. Then the determinant of A' is equal to 0 if and only if the determinant of A is equal to 0.*

Proof The proof is similar to Proposition 1 since we have the same mapping up to the isomorphism and all entries of an MDS matrix remain nonzero after applying the isomorphism. Note that each f_{s_u} maps each element in $\mathbb{F}_2$ to itself. The isomorphism f_{s_u} is related to automorphism as defined in Proposition 1 due to the structure of s_u. □

Theorem 2 *There exist $m \cdot (2^m - 1)$ distinct functions obtained by using isomorphisms in the form of $f_{s_u,c} : \beta_1 \mapsto (\beta_2^{s_u}) \cdot c$, where β_1 and β_2 are respectively any primitive element of $\mathbb{F}_{2^m}/p_1(x)$ and $\mathbb{F}_{2^m}/p_2(x)$, $c \in \mathbb{F}_{2^m}^*$, $s_u = e \cdot 2^i$ for $1 \leq e \leq 2^m - 2$, $gcd(e, 2^m - 1) = 1$, $p_1(\beta_2^{s_u}) = 0$, and $0 \leq u, i \leq m-1$. These functions can be used in generating new MDS matrices over $\mathbb{F}_{2^m}/p_2(x)$ from an MDS matrix over $\mathbb{F}_{2^m}/p_1(x)$, which preserve the MDS property of a square matrix.*

Proof Let $\beta \in \mathbb{F}_{2^m}$ be a primitive element. Recall that the minimal polynomial of the set $\beta, \beta^2, \ldots, \beta^{2^{m-1}}$, where m is the smallest integer such that $\beta^{2^m} = \beta$, is the same. Since the proof is similar to Theorem 1, we omit it. □

Algorithm 1 presents how to compute s_u values to define the isomorphisms in Theorem 2. Algorithm 1 only receives the primitive polynomial as an input. The main idea in Algorithm 1 is to substitute the elements with the powers of primitive elements. This helps us to define the isomorphisms between primitive polynomials for the same binary extension, i.e. the powers of a given primitive element are checked for whether it is a root of $p_1(x)$ modulo $p_2(x)$.

Algorithm 1 Computing s_u values to define the isomorphisms in Proposition 2.

Input: $p_1(\beta_1)$, β_2, and $p_2(x)$
Output: s_u, where $0 \leq u \leq m-1$

```
for s_u = 1 to 2^m − 2 do
    y_1 ← p_1(β_2^{s_u}) (mod p_2(x))
    if y_1 = 0 then
        Return (s_u)
    end if
end for
```

Example 2 *Let $\mathbb{F}_{2^4}$ be defined by the irreducible polynomial $p_1(x) = x^4+x^3+x^2+x+1$. Then β_1, defined by $\beta_1 = \alpha+1$, is a primitive element, where α is a root of $p_1(x)$ and $M_2 = had(1_h, 2_h, 4_h, 6_h) = had(1, \beta_1^{12}, \beta_1^9, \beta_1^{13})$ is an involutory 4×4 MDS matrix over $\mathbb{F}_{2^4}/p_1(x)$. We can rewrite $p_1(\alpha) = \alpha^4+\alpha^3+\alpha^2+\alpha+1$ in terms of β_1 as $p_1(\beta_1) = \beta_1^4+\beta_1^3+1$. Consider the finite field $\mathbb{F}_{2^4}/p_2(x)$, where $p_2(x) = x^4+x+1$. Let the primitive element β_2 of $\mathbb{F}_{2^4}/p_2(x)$ be α_1, which is also a root of $p_2(x)$. Then we can obtain 4 distinct isomorphisms from $\mathbb{F}_{2^4}/p_1(x)$ to $\mathbb{F}_{2^4}/p_2(x)$ by computing s_u values (which are $s_0 = 7$, $s_1 = 11$, $s_2 = 13$, and $s_3 = 14$) in Algorithm 1. These isomorphisms are $f_{7,1}: \beta_1 \mapsto \alpha_1^7$, $f_{11,1}: \beta_1 \mapsto \alpha_1^{11}$, $f_{13,1}: \beta_1 \mapsto \alpha_1^{13}$, and $f_{14,1}: \beta_1 \mapsto \alpha_1^{14}$. For example, by using the isomorphism $f_{7,1}: \beta_1 \mapsto \alpha_1^7$, we can generate the involutory 4×4 MDS matrix $M_2^{'}$ over $\mathbb{F}_{2^4}/p_2(x)$ from M_2 over $\mathbb{F}_{2^4}/p_1(x)$ as follows: $M_2^{'} = had(1_h, A_h, 8_h, 2_h) = had(1, \alpha_1^9, \alpha_1^3, \alpha_1)$.*

Note that by Theorem 2 one can generate 59 more MDS matrices over $\mathbb{F}_{2^4}/p_2(x)$ by using the MDS matrix M_2 over $\mathbb{F}_{2^4}/p_1(x)$. In Example 2, each s_u is a representative of the same cyclotomic coset C_s modulo $\mathbb{F}_{2^4}-1$ with $gcd(s, 2^4-1) = 1$ ($C_1 = \{1,2,4,8\}$ or $C_7 = \{7,11,13,14\}$). Thus, the computations to identify s_u values in Algorithm 1 can be performed by only using two coset leaders ($s = 1$ or $s = 7$) of these cyclotomic cosets.

2.3. Generalization of MDS isomorphisms

In this section, we give a generalization of the proposed idea for large dimensions to be used in generating new MDS matrices over $\mathbb{F}_{2^{mt}}$, where $t \geq 1$, from an MDS matrix over $\mathbb{F}_{2^m}$. By modifying the idea given in Proposition 2, Theorem 2, and Algorithm 1, a general method to obtain the isomorphisms and distinct functions related to these isomorphisms can be given as follows:

Step 1. Choose a primitive polynomial $p_1(x)$ of degree m and the primitive elements β_1 and β_2 for the finite fields $\mathbb{F}_{2^m}/p_1(x)$ and $\mathbb{F}_{2^{mt}}/p_2(x)$, respectively.

Step 2. Generate m isomorphisms, i.e. compute s_u values by using Algorithm 2.

Step 3. Compute $m\cdot(2^{mt}-1)$ distinct functions related to the isomorphisms by multiplying the isomorphisms generated in Step 2 with all nonzero constants $c \in \mathbb{F}_{2^{mt}}$.

Remark 1 *Let $p_1(x)$ be an irreducible polynomial but not primitive in Step 1. Then a primitive polynomial is constructed by evaluating β_1 in $p_1(x)$, i.e. $p_1(\beta_1)$. This polynomial is used as an input to Algorithm 2.*

Algorithm 2 Computing s_u values to define the isomorphisms between MDS matrices over $\mathbb{F}_{2^m}$ and MDS matrices over $\mathbb{F}_{2^{mt}}$.

Input: $p_1(\beta_1)$, β_2, and $p_2(x)$
Output: s_u, where $0 \leq u \leq m-1$
for $s_u = 1$ to $2^{mt}-2$ **do**
  $y_1 \leftarrow p_1(\beta_2^{s_u}) \pmod{p_2(x)}$
  if $y_1 = 0$ **then**
    Return s_u
  end if
end for

Remark 2 *In Algorithm 2, each s_u is a representative of the same cyclotomic coset C_s modulo $2^{mt}-1$ with $gcd(s, 2^{mt}-1) = \frac{2^{mt}-1}{2^m-1} = (2^m)^{t-1} + (2^m)^{t-2} + \cdots + 1$ for $t \geq 1$ since $2^m - 1$ elements of $\mathbb{F}_{2^m}$ are mapped to $2^m - 1$ elements of $\mathbb{F}_{2^{mt}}$.*

Example 3 *Let $\mathbb{F}_{2^4}$ be defined by the primitive polynomial $p_1(x) = x^4 + x + 1$. Let α be a root of $p_1(x)$. Then* $M_3 = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^4 \\ 1 & 1 & \alpha^3 & \alpha^2 \\ 1 & \alpha^2 & 1 & \alpha \\ \alpha & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1_h & 2_h & 4_h & 3_h \\ 1_h & 1_h & 8_h & 4_h \\ 1_h & 4_h & 1_h & 2_h \\ 2_h & 1_h & 1_h & 1_h \end{bmatrix}$ *is an involutory 4×4 MDS matrix over $\mathbb{F}_{2^4}/p_1(x)$ generated by the generalized Hadamard construction method given in [20]. Consider the finite field $\mathbb{F}_{2^8}/p_2(x)$, where $p_2(x) = x^8 + x^4 + x^3 + x + 1$. Let the primitive element β_2 of $\mathbb{F}_{2^8}/p_2(x)$ be $\alpha_1 + 1$, where α_1 is a root of $p_2(x)$. Then we can obtain 4 distinct isomorphisms from $\mathbb{F}_{2^4}/p_1(x)$ to $\mathbb{F}_{2^8}/p_2(x)$ by computing s_u values in Algorithm 2. These isomorphisms are $f_{17,1} : \alpha \mapsto \beta_2^{17}$, $f_{34,1} : \alpha \mapsto \beta_2^{34}$, $f_{68,1} : \alpha \mapsto \beta_2^{68}$, and $f_{136,1} : \alpha \mapsto \beta_2^{136}$. For example, by using the isomorphism $f_{17,1} : \alpha \mapsto \beta_2^{17}$, we can generate the involutory 4×4 MDS matrix $M_3^{'}$ over $\mathbb{F}_{2^8}/p_2(x)$ from M_3 over $\mathbb{F}_{2^4}/p_1(x)$ as follows:*

$$M_3^{'} = \begin{bmatrix} 1 & \beta_2^{17} & \beta_2^{34} & \beta_2^{68} \\ 1 & 1 & \beta_2^{51} & \beta_2^{34} \\ 1 & \beta_2^{34} & 1 & \beta_2^{17} \\ \beta_2^{17} & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 01_h & E1_h & 5C_h & E0_h \\ 01_h & 01_h & 0C_h & 5C_h \\ 01_h & 5C_h & 01_h & E1_h \\ E1_h & 01_h & 01_h & 01_h \end{bmatrix}.$$

Note that one can generate 1019 ($4 \cdot (2^8 - 1) - 1$) more MDS matrices over $\mathbb{F}_{2^8}/p_2(x)$ by using the MDS matrix M_3 over $\mathbb{F}_{2^4}/p_1(x)$.

Remark 3 *The 4×4 involutory MDS matrix M_3 over $\mathbb{F}_{2^4}$ was generated by GHadamard matrix type $Ghad(1, \alpha^8; \alpha^8, \alpha; \alpha, \alpha^{10}; \alpha^9)$ given in [20], where α is a root of the primitive polynomial $x^4 + x + 1$.*

Remark 4 *By Remark 2, each s_u in Example 3 can belong the same cyclotomic coset C_s with $gcd(s, 2^8 - 1) = \frac{2^8-1}{2^4-1} = 17$ ($C_{17} = \{17, 34, 68, 136\}$ or $C_{119} = \{119, 187, 221, 238\}$). Therefore, one can only try two coset leaders of these cyclotomic cosets ($s = 17$ or $s = 119$) to identify s_u values in Algorithm 2.*

Recall that a $k \times k$ circulant matrix $A = circ(a_0, a_1, \ldots, a_{k-1})$ over $\mathbb{F}_{2^m}$ can be given as $A_{i,j} = a_{j-i \pmod k}$, where $1 \leq i, j \leq k$. In Example 4, we generate a 4×4 MDS matrix over $\mathbb{F}_{2^8}$ from the circulant MDS matrix

$circ(2_h, 3_h, 1_h, 1_h) = circ(\alpha, \alpha^4, 1, 1)$ (the same MDS matrix used as the AES MixColumns transformation but not over $\mathbb{F}_{2^8}$) over $\mathbb{F}_{2^4}$ defined by the primitive polynomial $x^4 + x + 1$.

Example 4 *Consider the AES MixColumns transformation* $M_4 = circ(2_h, 3_h, 1_h, 1_h) = circ(\alpha, \alpha^4, 1, 1)$ *over* $\mathbb{F}_{2^4}/p_1(x)$, *where* $p_1(x) = x^4 + x + 1$ *and* α *is a root of* $p_1(x)$. *Then, by using the isomorphism* $f_{17,1} : \alpha \mapsto \beta_2^{17}$ *given in Example 3, we can generate* $M_4^{'}$ *over* $\mathbb{F}_{2^8}/p_2(x)$, *where* $p_2(x) = x^8 + x^4 + x^3 + x + 1$, *as follows:* $M_4^{'} = circ(E1_h, E0_h, 01_h, 01_h) = circ(\beta_2^{17}, \beta_2^{68}, 1, 1)$. *Since the inverse of the matrix* M_4 *can be obtained as* $M_4^{-1} = circ(E_h, B_h, D_h, 9_h) = circ(\alpha^{11}, \alpha^7, \alpha^{13}, \alpha^{14})$ *over* $\mathbb{F}_{2^4}/p_1(x)$, *we can generate* $(M_4^{'})^{-1}$ *over* $\mathbb{F}_{2^8}/p_2(x)$ *as follows:* $(M_4^{'})^{-1} = circ(B1_h, EC_h, 51_h, 0D_h) = circ((\beta_2^{17})^{11}, (\beta_2^{17})^7, (\beta_2^{17})^{13}, (\beta_2^{17})^{14}) = circ(\beta_2^{187}, \beta_2^{119}, \beta_2^{221}, \beta_2^{238})$.

In Example 5, we generate an involutory 8×8 MDS matrix over $\mathbb{F}_{2^{16}}$ from the KHAZAD diffusion matrix [32], which is an involutory 8×8 MDS matrix over $\mathbb{F}_{2^8}$.

Example 5 *Consider the KHAZAD diffusion matrix* $M_5 = had(1_h, 3_h, 4_h, 5_h, 6_h, 8_h, B_h, 7_h) = had(1, \alpha^{25}, \alpha^2, \alpha^{50}, \alpha^{26}, \alpha^3, \alpha^{238}, \alpha^{198})$ *over* $\mathbb{F}_{2^8}/p_1(x)$, *where the primitive polynomial* $p_1(x) = x^8 + x^4 + x^3 + x^2 + 1$ *and* α *is a root of* $p_1(x)$. *Consider the finite field* $\mathbb{F}_{2^{16}}/p_2(x)$, *where the primitive polynomial* $p_2(x) = x^{16} + x^{14} + x^{13} + x^9 + x^5 + x^4 + 1$, *and let the primitive element* β_2 *of* $\mathbb{F}_{2^{16}}/p_2(x)$ *be* α_1, *which is also a root of* $p_2(x)$. *Then we can obtain 8 distinct isomorphisms from* $\mathbb{F}_{2^8}/p_1(x)$ *to* $\mathbb{F}_{2^{16}}/p_2(x)$ *by computing* s_u *values in Algorithm 2. These isomorphisms are* $f_{11051,1} : \alpha \mapsto \beta_2^{11051}$, $f_{22102,1} : \alpha \mapsto \beta_2^{22102}$, $f_{22873,1} : \alpha \mapsto \beta_2^{22873}$, $f_{25957,1} : \alpha \mapsto \beta_2^{25957}$, $f_{38293,1} : \alpha \mapsto \beta_2^{38293}$, $f_{44204,1} : \alpha \mapsto \beta_2^{44204}$, $f_{45746,1} : \alpha \mapsto \beta_2^{45746}$, *and* $f_{51914,1} : \alpha \mapsto \beta_2^{51914}$. *For example, by using the isomorphism* $f_{11051,1} : \alpha \mapsto \beta_2^{11051}$, *we can generate the involutory* 8×8 *MDS matrix* $M_5^{'}$ *over* $\mathbb{F}_{2^{16}}/p_2(x)$ *from* M_5 *over* $\mathbb{F}_{2^8}/p_1(x)$ *as follows:*

$$\begin{aligned} M_5^{'} &= had(1, \beta_2^{14135}, \beta_2^{22102}, \beta_2^{28270}, \beta_2^{25186}, \beta_2^{33153}, \beta_2^{8738}, \beta_2^{25443}) \\ &= had(0001_h, 3D2A_h, 420C_h, 420D_h, 7F27_h, DF69_h, E243_h, 7F26_h). \end{aligned}$$

3. Properties of the proposed method and implementation details

In this section, we discuss the implementation details along with some important properties of the proposed method. In Sections 3.1 and 3.2, we respectively present some important properties of the proposed method and implementation properties of the generated MDS matrices.

3.1. Some important properties of the proposed method

The proposed method introduced in Section 2 takes an MDS matrix as input to generate new MDS matrices. In this respect, it may help other construction methods to generate isomorphic MDS matrices, which may have better implementation properties than the ones constructed by these methods (Example 6). Also, it presents the relationship between MDS matrices over $\mathbb{F}_{2^m}$ and MDS matrices over $\mathbb{F}_{2^{mt}}$, and it provides a way to generate isomorphic MDS matrices over $\mathbb{F}_{2^{mt}}$ having the same implementation properties (in view of XOR operations and table lookups) as those of MDS matrices constructed over $\mathbb{F}_{2^m}$. This is because of the fact that the proposed method moves the finite field $\mathbb{F}_{2^m}$ into the finite field $\mathbb{F}_{2^{mt}}$ by the help of isomorphisms obtained and therefore $2^m - 1$ elements of $\mathbb{F}_{2^{mt}}$ corresponding to $2^m - 1$ elements of $\mathbb{F}_{2^m}$ are used to construct and implement any MDS matrix over $\mathbb{F}_{2^{mt}}$. Some important properties of the proposed method can be given as follows:

- The method is intended to be applied to other construction methods in the literature to generate new MDS matrices from the existing ones.
- The method can be considered as a complementary method for the current construction methods, allowing them to look for MDS matrices having better implementation properties by mapping them to different field representations (see Example 6).
- The method helps to map any $k \times k$ MDS matrix over $\mathbb{F}_{2^m}$ to its isomorphic $k \times k$ MDS matrix over $\mathbb{F}_{2^{mt}}$.
- An MDS matrix generated over $\mathbb{F}_{2^{mt}}$ from an existing MDS matrix over $\mathbb{F}_{2^m}$ can take advantage of the small number of table lookups in the implementation, which can only be used with XOR operations. By the help of isomorphisms, it can also be implemented by the same number XOR operations and table lookups as that of an existing MDS matrix over $\mathbb{F}_{2^m}$ (see Section 3.2). In this respect, the method helps to generate MDS matrices over $\mathbb{F}_{2^{mt}}$ with efficient software implementations when mt is large (see Example 5, where an 8×8 involutory MDS matrix over $\mathbb{F}_{2^{16}}$ generated by the KHAZAD diffusion matrix, which is an 8×8 involutory MDS matrix over $\mathbb{F}_{2^8}$).
- Assume that an $[n, k, d]$ code C with generator matrix $G = [I|A]$, where A is a $k \times (n-k)$ matrix, is given. Then the minimum distance d of C is preserved after the application of any distinct function generated in Section 2.3 to A.

3.2. Implementation properties of the generated MDS matrices by the proposed method

In this section, we present important properties of the proposed method in generating MDS matrices with efficient software implementation on 8-bit platforms. We also compare the implementation properties of the generated MDS matrices with the MDS matrix constructed in [11] and the MDS matrices used in the well-known block cipher AES, namely AES MixColumns transformation, and the ANUBIS block cipher.

In the literature, the two types of implementation of an MDS matrix multiplication differ according to the use of two different operations, which are table lookups and xtime operations. The proposed method presents an advantage in the implementation in light of the small number of table lookups since an MDS matrix over $\mathbb{F}_{2^{mt}}$ can be generated from an MDS matrix over $\mathbb{F}_{2^m}$ by using the proposed method. The involutory MDS matrix $M_3^{'}$ over $\mathbb{F}_{2^8}$ given in Example 3 can take advantage of the small number of table lookups in the implementation since it is generated from M_3 over $\mathbb{F}_{2^4}$. A multiplication by $M_3^{'}$ can be implemented by using 12 XORs and 10 table lookups as follows:

$$\begin{aligned} y[0] &= x[0] \oplus table[x[1] \oplus table[x[2] \oplus table[table[x[3]]]]], \\ y[1] &= x[0] \oplus x[1] \oplus table[table[table[x[2]] \oplus x[3]]], \\ y[2] &= x[0] \oplus x[2] \oplus table[table[x[1]] \oplus x[3]], \\ y[3] &= table[x[0]] \oplus x[1] \oplus x[2] \oplus x[3], \end{aligned}$$

where the input is in $x[0..3]$ and the output in $y[0..3]$. Also, the multiplication by β_2^{17} is performed by one table lookup, namely $table$.

In Example 4, the MDS matrix $M_4^{'}$ generated over $\mathbb{F}_{2^8}$ can be implemented by 15 XORs and 4 table lookups and in a similar way to the AES MixColumns transformation as given in [2]. On the other hand, the

matrix M_4^{-1} over $\mathbb{F}_{2^4}$ in Example 4 has elements for which powers of the primitive element belong to the same cyclotomic coset. Then the inverse matrix of $M_4^{'}$ over $\mathbb{F}_{2^8}$, $(M_4^{'})^{-1}$, can also be expressed as follows: $(M_4^{'})^{-1} = circ((\beta_2^{119})^8, (\beta_2^{119}), (\beta_2^{119})^4, (\beta_2^{119})^2)$. Hence, the matrix $(M_4^{'})^{-1}$ can easily be implemented by 12 XORs and 16 table lookups (using two tables) as follows:

$$\begin{aligned} y[0] &= table2[table2[x[0]] \oplus x[2]] \oplus table1[table1[x[3]] \oplus x[1]], \\ y[1] &= table2[table2[x[1]] \oplus x[3]] \oplus table1[table1[x[0]] \oplus x[2]], \\ y[2] &= table2[table2[x[2]] \oplus x[0]] \oplus table1[table1[x[1]] \oplus x[3]], \\ y[3] &= table2[table2[x[3]] \oplus x[1]] \oplus table1[table1[x[2]] \oplus x[0]], \end{aligned}$$

where the input is in $x[0..3]$ and the output is in $y[0..3]$. Also, the multiplications by β_2^{119} and $(\beta_2^{119})^4$ $(= \beta_2^{221})$ are performed by two different table lookups, namely $table1$ and $table2$, respectively.

In Example 5, the involutory MDS matrix $M_5^{'}$ over $\mathbb{F}_{2^{16}}$ can be implemented in a similar way to the KHAZAD diffusion matrix and by the same number of XOR operations and table lookups of the KHAZAD diffusion matrix (56 XOR operations and 24 table lookups as given in [32]).

In Example 6, the MDS matrix $M_6^{'}$ can be implemented with 12 xtimes and 14 XOR operations and is generated from the MDS matrix M_6 given in [11], which needs 12 xtimes and 16 XOR operations. We compare 4×4 MDS matrices over $\mathbb{F}_{2^8}$ obtained by the proposed method with some of the known 4×4 MDS matrices in the Table, where temp stands for the temporary variables.

Table. Comparison of 4×4 MDS matrices over $\mathbb{F}_{2^8}$.

MDS matrix	# XOR	# table lookups or # xtimes	# temp	involutory
M_3'	12	10	-	Yes
M_4'	15	4	3	No
$(M_4^{'})^{-1}$	12	16	-	No
M_6'	14	12	-	Yes
M_6 [11]	16	12	4	Yes
M_7 [17]	12	12	-	Yes
M_8 [17]	12	4	4	Yes
ANUBIS [29]	12	6	4	Yes
AES [2, 33]	15	4	3	No

Example 6 *$M_6 = had(1, \alpha^{50}, \alpha^{224}, \alpha^{129}) = (01_h, 05_h, 12_h, 17_h)$ is an involutory 4×4 MDS matrix over $\mathbb{F}_{2^8}/p_1(x)$ given in [11], where the primitive polynomial $p_1(x) = x^8 + x^4 + x^3 + x^2 + 1$ and α is a root of $p_1(x)$. Consider the finite field $\mathbb{F}_{2^8}/p_2(x)$, where $p_2(x) = x^8 + x^4 + x^3 + x + 1$. Let the primitive element β_2 of $\mathbb{F}_{2^8}/p_2(x)$ be $\alpha_1 + 1$, where α_1 is a root of $p_2(x)$. Then we can obtain 8 distinct isomorphisms from $\mathbb{F}_{2^8}/p_1(x)$ to $\mathbb{F}_{2^8}/p_2(x)$ by computing s_u values in Algorithm 2. These isomorphisms are $f_{1,1} : \alpha \mapsto \beta_2^1$, $f_{2,1} : \alpha \mapsto \beta_2^2$,*

$f_{4,1} : \alpha \mapsto \beta_2^4$, $f_{8,1} : \alpha \mapsto \beta_2^8$, $f_{16,1} : \alpha \mapsto \beta_2^{16}$, $f_{32,1} : \alpha \mapsto \beta_2^{32}$, $f_{64,1} : \alpha \mapsto \beta_2^{64}$, $f_{128,1} : \alpha \mapsto \beta_2^{128}$. *For example, by using the isomorphism* $f_{1,1} : \alpha \mapsto \beta_2^1$, *we can generate the involutory* 4×4 *MDS matrix* $M_6^{'}$ *over* $\mathbb{F}_{2^8}/p_2(x)$ *from* M_6 *over* $\mathbb{F}_{2^8}/p_1(x)$ *as follows:* $M_6^{'} = had(1_h, 04_h, 12_h, 16_h) = had(1, \beta_2^{50}, \beta_2^{224}, \beta_2^{129})$.

Remark 5 *By using the same isomorphism (*$f_{17,1} : \alpha \mapsto \beta_2^{17}$*) given in Example 3, one can generate* 4×4 *isomorphic involutory MDS matrix* $M_7^{'} = had(1, \beta^{34}, \beta^{238}, \beta^{221}) = had(01_h, 5C_h, 0C_h, 51_h)$ *over* $\mathbb{F}_{2^8}$ *defined by the irreducible polynomial* $x^8+x^4+x^3+x+1$ *from the* 4×4 *involutory MDS matrix* $M_7 = (1_h, 4_h, 9_h, D_h)$ *over* $\mathbb{F}_{2^4}$ *defined by the primitive polynomial* $x^4 + x + 1$, *which is also used to construct the* 4×4 *involutory MDS matrix with XOR count 144 by using subfield-Hadamard construction in [17]. The* 4×4 *isomorphic involutory MDS matrix generated can be implemented by 12 XOR operations and 12 table lookups (and using two different tables). On the other hand, the* 4×4 *involutory MDS matrix* $M_8 = had(01_h, 02_h, B0_h, B2_h)$ *over* $\mathbb{F}_{2^8}$ *defined by the irreducible polynomial* $x^8 + x^6 + x^5 + x^2 + 1$ *with XOR count 160 given in [17] can be implemented by 12 XOR operations, 4 table lookups, and 4 temporary variables (and using two different tables).*

4. Conclusion

In this study, we present a novel method to generate distinct functions related to the automorphisms and isomorphisms of MDS matrices. These functions take any MDS matrix generated by any construction method as input to generate new MDS matrices from the existing ones, which makes the proposed method generic over all construction methods in the literature. Using the isomorphisms obtained, the method helps to generate isomorphic MDS matrices over different binary extension fields, which may have better implementation properties than an existing MDS matrix. Moreover, the proposed method can be used to generate MDS matrices over large binary extension fields (with good software implementation properties in light of the number of XOR operations and table lookups) for which many known methods involve some kind of exhaustive search. Finally, we compare the MDS matrices generated by the proposed method with the previous ones used in block ciphers in light of the number of XOR operations and table lookups.

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. Sedat Akleylek was partially supported by TÜBİTAK under Grant No. EEEAG-116E279.

References

[1] Shannon CE. Communication theory of secrecy systems. Bell System Technical Journal 1949; 28: 656-715. doi: 10.1002/j.1538-7305.1949.tb00928.x

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

[3] Barreto PSLM, Rijmen V. Whirlpool. In: van Tilborg HCA, Jajodia S. (editors). Encyclopedia of Cryptography and Security. 2nd ed. Boston, MA, USA: Springer, 2011, pp. 1384-1385.

[4] Guo J, Peyrin T, Poschmann A. The PHOTON family of lightweight hash functions. In: Proceedings of CRYPTO; Santa Barbara, CA, USA; 2011. pp. 222-239.

[5] Barreto PSLM, Nikov V, Nikova S, Rijmen V, Tischhauser E. Whirlwind: A new cryptographic hash function. Design, Codes and Cryptography 2010; 56: 141–162. doi: 10.1007/s10623-010-9391-y

[6] Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. In: Proceedings of CRYPTO; Santa Barbara, CA, USA; 1990. pp. 2-21.

[7] Matsui M. Linear cryptanalysis method for DES cipher. In: Proceedings of EUROCRYPT; Lofthus, Norway; 1993. pp. 386-397.

[8] Youssef AM, Mister S, Tavares SE. On the design of linear transformation for substitution permutation encryption networks. In: Proceedings of SAC; Ottowa, Canada; 1997. pp. 40-48.

[9] Gupta KC, Ray IG. On constructions of circulant MDS matrices for lightweight cryptography. In: Proceedings of ISPEC; Fuzhou, China; 2014. pp. 564-576.

[10] Lacan J, Fimes J. Systematic MDS erasure codes based on Vandermonde matrices. IEEE Transactions on Communications Letters 2004; 8 (9): 570-572. doi: 10.1109/LCOMM.2004.833807

[11] Sajadieh M, Dakhilalian M, Mala H, Omoomi B. On construction of involutory MDS matrices from Vandermonde matrices in GF(2^q). Design, Codes and Cryptography 2012; 64 (3): 287-308. doi: 10.1007/s10623-011-9578-x

[12] Augot D, Finiasz M. Direct construction of recursive MDS diffusion layers using shortened BCH codes. In: Proceedings of FSE; London, UK; 2014. pp. 3-17.

[13] Berger TP. Construction of recursive MDS diffusion layers from Gabidulin codes. In: Proceedings of INDOCRYPT; Mumbai, India; 2013. pp. 274-285.

[14] Cauchois V, Loidreau P, Merkiche N. Direct construction of quasi-involutory recursive-like MDS matrices from 2-cyclic codes. IACR Transactions on Symmetric Cryptology 2016; 2016 (2): 80-98. doi: 10.13154/tosc.v2016.i2.80-98

[15] Sajadieh M, Dakhilalian M, Mala H, Sepehrdad P. Recursive diffusion layers for block ciphers and hash functions. In: Proceedings of FSE; Washington, DC, USA; 2012. pp. 385-401.

[16] Wu S, Wang M, Wu W. Recursive diffusion layers for (lightweight) block ciphers and hash functions. In: Proceedings of SAC; Windsor, Canada; 2012. pp. 355-371.

[17] Sim SM, Khoo K, Oggier F, Peyrin T. Lightweight MDS involution matrices. In: Proceedings of FSE; İstanbul, Turkey; 2015. pp. 471-493.

[18] Li Y, Wang M. On the construction of lightweight circulant involutory MDS matrices. In: Proceedings of FSE; Bochum, Germany; 2016. pp. 121-139.

[19] Liu M, Siu SM. Lightweight MDS generalized circulant matrices. In: Proceedings of FSE; Bochum, Germany; 2016. pp. 101-120.

[20] Pehlivanoğlu MK, Sakallı MT, Akleylek S, Duru N, Rijmen V. Generalisation of Hadamard matrix to generate involutory MDS matrices for lightweight cryptography. IET Information Security 2018; 12 (4): 348-355. doi: 10.1049/iet-ifs.2017.0156

[21] Jean J, Peyrin T, Sim SM, Tourteaux J. Optimizing implementations of lightweight building blocks. IACR Transactions on Symmetric Cryptology 2017; 2017 (4): 130–168. doi: 10.13154/tosc.v2017.i4.130-168

[22] Güzel GG, Sakallı MT, Akleylek S, Rijmen V, Çengellenmiş Y. A new matrix form to generate all 3×3 involutory MDS matrices over $\mathbb{F}_{2^m}$. Information Processing Letters 2019; 147: 61-68. doi: 10.1016/j.ipl.2019.02.013

[23] Kranz H, Leander G, Stoffelen K, Wiemer F. Shorter linear straight-line programs for MDS matrices. IACR Transactions on Symmetric Cryptology 2017; 2017 (4): 188-211. doi: 10.13154/tosc.v2017.i4.188-211

[24] Sarkar S, Syed H. Lightweight diffusion layer: importance of Toeplitz matrices. IACR Transactions on Symmetric Cryptology 2016; 2016 (1): 95-113. doi: 10.13154/tosc.v2016.i1.95-113

[25] Choy J, Yap H, Khoo K, Guo J, Peyrin T et al. SPN-Hash: improving the provable resistance against differential collision attacks. In: Proceedings of AFRICACRYPT; Ifrane, Morocco; 2012. pp. 270-286.

[26] MacWilliams FJ. Combinatorial properties of elementary abelian groups. PhD, Radcliffe College, Cambridge, UK, 1962.

[27] Bonneau PGA. Codes et combinatoire. PhD, Université Pierre et Marie Curie, Paris, France, 1984 (in French).

[28] Buchmann J, Pyshkin A, Weinmann RP. Block ciphers sensitive to Gröbner basis attacks. In: Proceedings of CT-RSA; San Jose, CA, USA; 2006. pp. 313-331.

[29] Barreto PSLM, Rijmen V. The Anubis block cipher. In: First Open NESSIE Workshop, KU-Leuven, Belgium; 2000.

[30] MacWilliams FJ, Sloane NJA. The Theory of Error Correcting Codes. Amsterdam, the Netherlands: North Holland, 1986.

[31] Lidl R, Niederreiter H. Introduction to Finite Fields and Their Applications. Cambridge, UK: Cambridge University Press, 1986.

[32] Barreto PSLM, Rijmen V. The Khazad legacy-level block cipher. In: First Open NESSIE Workshop, KU-Leuven, Belgium; 2000.

[33] Junod P, Vaudenay S. Perfect diffusion primitives for block ciphers. In: Proceedings of SAC; Waterloo, Canada; 2004. pp. 84-99.