The ISC Int'l Journal of Information Security

January 2019, Volume 11, Number 1 (pp. 35–46)

http://www.isecure-journal.org

Lightweight 4 x 4 MDS Matrices for Hardware-Oriented Cryptographic Primitives

Akbar Mahmoodi Rishakani [1], Mohammad Reza Mirzaee Shamsabad [2], Seyyed Mojtaba Dehnavi [3], Mohammad Amin Amiri [4], Hamid Reza Maimani [1], and Nasour Bagheri [5,6,*]

[1] *Department of Sciences, Shahid Rajaee Teacher Training University, Tehran, Iran*
[2] *Department of Mathematics and Computer Sciences, Shahid Beheshti University, Tehran, Iran*
[3] *Department of Mathematical and Computer Sciences, Kharazmi University, Tehran, Iran*
[4] *Department of Computer and Electronic Engineering, Malek Ashtar University of Technology, Tehran, Iran*
[5] *Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran 16788-15811, Iran*
[6] *School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran*

ARTICLE INFO.

Article history:
Received: 2 July 2018
Revised: 10 November 2018
Accepted: 1 December 2018
Published Online: 30 January 2019

Keywords:
Diffusion Layer, Branch Number, Lightweight Cryptographic Primitives, Companion Matrix, MDS Matrix.

ABSTRACT

Linear diffusion layer is an important part of lightweight block ciphers and hash functions. This paper presents an efficient class of lightweight 4×4 MDS matrices such that the implementation cost of them and their corresponding inverses are equal. The main target of the paper is hardware oriented cryptographic primitives and the implementation cost is measured in terms of the required number of XORs. Firstly, we mathematically characterize the MDS property of a class of matrices (derived from the product of binary matrices and companion matrices of σ-LFSRs aka recursive diffusion layers) whose implementation cost is $10m + 4$ XORs for $4 \leq m \leq 8$, where m is the bit length of inputs. Then, based on the mathematical investigation, we further extend the search space and propose new families of 4×4 MDS matrices with $8m + 4$ and $8m + 3$ XOR implementation cost. The lightest MDS matrices by our new approach have the same implementation cost as the lightest existent matrix.

© 2019 ISC. All rights reserved.

1 Introduction

Providing proper diffusion and confusion are two fundamental requirements of any secure cryptographic primitive. In general, in symmetric cryptography, e.g. block ciphers, stream ciphers and hash functions, to provide the desired confusion and diffusion, a combination of nonlinear components and linear diffusion layers are used iteratively through several rounds. Among various available linear layers, MDS and almost MDS matrices are of more interest for designing a secure cipher, especially in wide trails designing based approaches, because of their fast diffusion property which is also known as high branch number. However, for constrained applications, e.g. RFID and IoT, the implementation cost of these matrices is a bottleneck. In this paper, we aim to put one step forward to overcome this problem by presenting hardware-efficient classes of lightweight 4×4 MDS matrices M for which the implementation cost of M and M^{-1} are the same, where M^{-1} denotes the inverse of M.

* Corresponding author.
Email addresses: am.rishakani@sru.ac.ir (A. Mahmoodi Rishakani), m_mirzaee@sbu.ac.ir (M.R. Mirzaee Shamsabad), dehnavism@ipm.ir (S.M. Dehnavi), maamiri@mut.ac.ir (M. Amin Amiri), maimani@ipm.ir (H. Reza Maimani), nbagheri@srttu.edu (N. Bagheri)
ISSN: 2008-2045 © 2019 ISC. All rights reserved.

1.1 Related Work

The design of lightweight MDS matrices has drawn the attention of many cryptographic researchers. For example, designers of lightweight hash function PHOTON [1] use a companion matrix of an LFSR M such that M^4 is an MDS matrix; in this case, applying M four times on inputs has less implementation cost than applying M^4 directly on inputs. This idea is extended by Sajadieh et al. [2] and Wu et al. [3], which is named recursive perfect diffusion layers. This approach uses the companion matrix of $\sigma-$LFSRs [4] instead of LFSRs.

In [5], Xu et al. present 4×4 recursive perfect diffusion layers over $m-$bit inputs for hardware implementations which require $12m + 12$ XORs for $5 \leq m \leq 7$ and $12m + 24$ XORs for $m = 8$. Beierle et al. [6] construct lightweight circulant MDS matrices with the aid of lightweight multiplication in $\mathbb{F}_{2^m}$ (the field with 2^m elements). Their 4×4 MDS matrices require $12m + 12$ XORs for $4 \leq m \leq 8$. In [7], lightweight 4×4 MDS matrices is constructed with 61 and 106 XOR implementation cost over 4-bit and 8-bit inputs respectively.

Bai and Wang [8] characterize lightweight 4×4 MDS matrices with 4-bit inputs which requires 58 XORs for implementation. After that, in [9] a class of 4×4 MDS matrices was produced with the help of Toeplitz matrices with 58 XORs for 4-bit and 123 XORs for 8-bit inputs by Sarkar et al. Then, Guo et al. [10] provided a large class of 4×4 MDS matrices for arbitrary m-bit ($m \geq 4$) inputs; in the case of m=4,8 the presented matrices need 64 and 128 XORs respectively. Later, in [11] Cauchois et al. constructed quasi-involutory recursive-like MDS matrices from 2-cyclic codes for which the implementation cost of 4×4 MDS matrices with 4-bit inputs is 72 XORs. In [12] Zhang et al. constructed circulant 4×4 MDS matrices over 4-bit inputs which requires 60 XORs for implementation. Zhou et al. [13] proposed two kinds of lightweight 4×4 MDS matrices over 4-bit and 8-bit inputs which require 58 and 106 XORs, respectively.

In previously discussed researches, the authors investigate 4×4 matrices with non-zero entries which are MDS and the implementation cost of their entries are as lightweight as possible. In this regard, the implementation cost of the additions through the action of the matrix over m-bit inputs takes $12m$ XORs which was considered as a lower bound for the implementation cost of MDS matrices. In this notion, finding low-cost MDS matrices boils down to reducing the implementation cost of the entries of matrices. Another procedure to efficiently implement 4×4 MDS matrices acting on m-bit inputs, is to consider them as $4m \times 4m$ binary matrices and then improve the implementation cost by reusing the resources, which is common in hardware implementations. Applying this method, the authors of [14] give lighter implementations than the claimed costs in the previously discussed papers, which shows that $12m$ is not a lower bound for the implementation cost of 4×4 MDS matrices over m-bit inputs. Most notably, a 4×4 MDS matrix with 4-bit inputs is presented which takes 36 XORs. The proposed method of [14] is not efficient for large values of m: for example the presented matrix for $m = 8$ takes 72 XORs which is derived through the parallel application of two 4×4 MDS matrices over 4-bit inputs, while the lightest ones take 67 XORs. Recently, two other papers on the construction of 4×4 MDS matrices over m-bit inputs are published which break the claimed $12m$ XORs lower bound [15, 16]. In [15], a new class of lightweight serial-type 4×4 MDS matrices are presented which need 4 clocks for implementation. Especially in the case of 4-bit inputs, each clock takes 10 XORs which requires 40 XORs for implementation in one clock. In [16], the authors follow the search idea of [14] with a different approach. In fact, their approach is somehow finding linear straight line programs by the limitation on the number of simultaneously available variables and only use operations on words rather than on bits. In fact, [15, 16] have a structural approach to optimize the implementation cost of MDS matrices. As a result, they give families of lightweight MDS matrices. Unlike the implementation concept of [14], the constructions presented in [15, 16] over m-bit inputs fit software implementations over m-bit processors. The papers [15, 16] use both global and local optimizations in their structures. The result is the construction of lightweight 4×4 MDS matrices over 4,8-bit inputs with 35 and 67 XORs implementation cost respectively, which are the lightest to the best of our knowledge.

1.2 Our Contribution

Our concern in this paper is to construct lightweight 4×4 MDS matrices with efficient implementation in hardware, measured by the number of XOR gates required. As stated in [17], the implementation cost of a given linear layer depends not only on its matrix but also on its implementation methods. So, we use a composition method to construct our 4×4 lightweight MDS matrices.

For more details, we characterize the MDS property of matrices in the form of $M = BC^4$ where C is a companion matrix of a 4-stage σ-LFSR and B is a 4×4 matrix with entries in $R = \{\mathbf{0}, I\}$. This method produces MDS matrices on m-bit inputs which require $10m + 4$ XORs for $4 \leq m \leq 8$. By alternating the positions of binary and companion matrices in the characterized MDS matrices, we search the matrices of

Table 1. A Comparison between the implementation cost of 4×4 MDS matrices, for $m = 4, 8$ as the bit length

m	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[15]	[14]	[16]	This paper
4	60	61	58	58	64	72	60	58	40	36	35	35
8	108	106	—	123	128	—	—	106	72	72	67	67

the form $M = B_1C_1B_2C_2B_3C_3B_4C_4B_5$, where B_i's and C_j's, $1 \leq i \leq 5, 1 \leq j \leq 4$, are binary and companion matrices, respectively. The result is the production of new families of 4×4 MDS matrices with $8m+4$ and $8m+3$ XOR implementation cost on m-bit input words. Our resulted MDS matrices over m-bit inputs fit software implementations over m-bit processors.

A comparison between the implementation cost of our proposed lightweight 4×4 MDS matrices and the best known related constructions is given in Table 1 for $m = 4, 8$ bit inputs (see Section 3 and Section 4 for details of our constructions).

1.3 Paper Organization

In Section 2, we give the preliminary notations and definitions. Section 3 characterizes new families of MDS matrices. In Section 4, we propose new families of the lightest MDS matrices. Section 5 concludes the paper.

2 Preliminaries

We use the following notations and definitions throughout this paper.

2.1 Notations

In this paper, n and m are natural numbers. By $|A|$ we mean the number of elements or cardinality of a finite set A. We denote the set of all $n \times n$ matrices with entries in R by $\mathcal{M}_n(R)$. The determinant of a matrix A in $\mathcal{M}_n(R)$ is denoted by $det_R(A)$. A^T represents transpose of a matrix A. The XOR of two binary vectors or matrices v and w is denoted by $v \oplus w$. Zero vectors or matrices are denoted by $\mathbf{0}$ and any identity matrix is denoted by I. We use $\mathbb{F}_2$ and $\mathbb{F}_2^m$ to represent a finite field with two elements and the set of all m-bit vectors, respectively.

For the sake of simplicity, to represent binary square matrices, only non-zero positions in each row will be listed; for example, $[4; (1, 3, 4); (2, 5); 1; 3]$ is the representation of the following matrix:

$$\begin{pmatrix} 0&0&0&1&0 \\ 1&0&1&1&0 \\ 0&1&0&0&1 \\ 1&0&0&0&0 \\ 0&0&1&0&0 \end{pmatrix}.$$

A square matrix M of order n is represented by $M = [M_1, M_2, \ldots, M_n]$ which M_i, $1 \leq i \leq n$, is the i-th row of M. The square submatrix of M including the rows $i_1, \ldots, i_t$ and columns $j_1, \ldots, j_t$, $1 \leq t \leq n$, is denoted by $M_{\{i_1,\ldots,i_t\}\{j_1,\ldots,j_t\}}$.

A cyclic matrix is a matrix whose rows (columns) are cyclic shifts of each other. By the notation $A = cycl(a_1, a_2, a_3, \ldots, a_n)$ we mean

$$A = \begin{pmatrix} a_1 & a_2 & a_3 & \ldots & a_n \\ a_n & a_1 & a_2 & \ldots & a_{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_2 & a_3 & a_4 & \ldots & a_1 \end{pmatrix}.$$

In this paper, we use invertible 4×4 matrices with entries in $R = \{\mathbf{0}, I\} \subset \mathcal{M}_m(\mathbb{F}_2)$. For a vector $v = (v_3, v_2, v_1, v_0) \in R^4$, we correspond a number $\bar{v} = \sum_{v_i \neq 0} 2^i$ in hexadecimal representation. Accordingly, a matrix $M = [M_1, M_2, M_3, M_4] \in \mathcal{M}_4(R)$ is denoted by $\bar{M_1}\bar{M_2}\bar{M_3}\bar{M_4}$. For instance, the following matrix is represented by $1bc9$:

$$M = \begin{pmatrix} \mathbf{0}&\mathbf{0}&\mathbf{0}&I \\ I&\mathbf{0}&I&I \\ I&I&\mathbf{0}&\mathbf{0} \\ I&\mathbf{0}&\mathbf{0}&I \end{pmatrix}.$$

An n-stage σ-LFSR over $\mathbb{F}_2^m$, generates a sequence of states $\mathcal{S}_i \in (\mathbb{F}_2^m)^n$. Each state is obtained by applying a matrix $\mathcal{A} \in \mathcal{M}_n(\mathcal{M}_m(\mathbb{F}_2))$ on the previous state as follows:

$$\mathcal{S}_{i+1} = \mathcal{S}_i\mathcal{A}, \quad i \geq 0,$$

$$\mathcal{A} = \begin{pmatrix} \mathbf{0} & I & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & I \\ A_1 & A_2 & A_3 & \dots & A_n \end{pmatrix}.$$

The matrix $\mathcal{A}$ is called the *companion matrix* of the σ-LFSR and we denote it by $\mathcal{A} = comp(A_1, A_2, A_3, \dots, A_n)$.

2.2 Definitions

A matrix $M \in \mathcal{M}_{nm}(\mathbb{F}_2)$ can be represented as (a block-wise matrix)

$$M = [A_{i,j}]_{n\times n}, \quad A_{i,j} \in \mathcal{M}_m(\mathbb{F}_2), \quad 1 \le i,j \le n. \tag{1}$$

In fact, the matrix M could be considered as a member of $\mathcal{M}_n(\mathcal{M}_m(\mathbb{F}_2))$. The i-th component of a vector $x \in (\mathbb{F}_2^m)^n$ is denoted by x_i, i.e. $x = (x_{n-1}, ..., x_0)$. The weight of a vector $x \in (\mathbb{F}_2^m)^n$ with respect to m-bit inputs is denoted by $wt_m(x)$ and is defined as

$$wt_m(x) = |\{\, x_i : x_i \neq \mathbf{0}, \quad 0 \le i < n\}|.$$

Definition 1. Let $M \in \mathcal{M}_{nm}(\mathbb{F}_2)$. The differential branch number of M with respect to m-bit inputs is defined as

$$\mathcal{B}_m^d(M) = \min_{x\neq\mathbf{0}}\{wt_m(x) + wt_m(xM) : x \in (\mathbb{F}_2^m)^n\};$$

and the linear branch number is defined as

$$\mathcal{B}_m^l(M) = \min_{x\neq\mathbf{0}}\{wt_m(x) + wt_m(xM^T) : x \in (\mathbb{F}_2^m)^n\}.$$

Definition 2. A matrix $M \in \mathcal{M}_{nm}(\mathbb{F}_2)$ is called MDS with respect to m-bit inputs if and only if

$$\mathcal{B}_m^d(M) = \mathcal{B}_m^l(M) = n + 1.$$

A sufficient condition for a matrix M to be MDS is that $\mathcal{B}_m^d(M) = n + 1$ [18].

Let M and N be two matrices such that the rows of M are a permutation of the rows of N (and vice versa). In this case, we say that M and N are equivalent matrices and we write $M \equiv N$.

3 Characterization of a New Family of 4 × 4 MDS Matrices

In this section, we characterize 4×4 MDS matrices of the form $M = BC^4$. Here, for $1 \le i \le 4$,

$$C = comp(A_1, A_2, A_3, A_4), \quad A_i \in \{0, I, A\} \subset \mathcal{M}_m(\mathbb{F}_2),$$

is a companion matrix of a 4-stage σ-LFSR and just two of A_i's are nonzero (to reduce the implementation cost) and B is a matrix with entries in $R = \{\mathbf{0}, I\} \subset \mathcal{M}_m(\mathbb{F}_2)$. According to [19, Theorem 4.1.15], we can state that there are $\prod_{i=0}^{3}(2^4 - 2^i) = 20160$ invertible matrices in $\mathcal{M}_4(R)$. In addition, we have 6 choices for C to be invertible. Hence, it is enough to verify the MDS property of $20160 \times 6 = 120960$ classes of matrices. For the mentioned characterization, we need the following theorems and lemma from related literature:

Theorem 1. *[20] For $M \in \mathcal{M}_n(\mathcal{M}_m(\mathbb{F}_2))$, M is MDS with respect to m-bit inputs if and only if every square submatrix of M of order t, $1 \le t \le n$, is invertible.*

Theorem 2. *[21] For $M \in \mathcal{M}_{nm}(\mathbb{F}_2)$, according to representation (1), if the entries of M in $R = \mathcal{M}_m(\mathbb{F}_2)$ are pairwise commuting, then*

$$det_{\mathbb{F}_2}(M) = det_{\mathbb{F}_2}(det_R(M)).$$

The following lemma is a straightforward result of [19, Theorem 4.5.6].

Lemma 1. *The differential branch numbers of equivalent matrices are equal.*

Given the above theorems and lemma, we characterize the target MDS matrices of the current section. To do this, let $R = \{\mathbf{0}, I\} \subset \mathcal{M}_m(\mathbb{F}_2)$, $A \in \mathcal{M}_m(\mathbb{F}_2)$, $B \in \mathcal{M}_4(R)$ and $C = comp(I, A, \mathbf{0}, \mathbf{0})$. We verify the MDS property of the matrices $M = BC^4$. Firstly, we analyze the different cases of matrix B. Each invertible matrix $B \in \mathcal{M}_4(R)$ is equivalent to 4! other different matrices; so, by Lemma 1, it is sufficient to verify $\frac{20160}{4!} = 840$ different equivalent classes for B. We have

$$C^4 = \begin{pmatrix} I & A & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I & A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & A \\ A & A^2 & \mathbf{0} & I \end{pmatrix}.$$

By our notations, let $B = [B_1, B_2, B_3, B_4]$ where each B_i, $1 \le i \le 4$, is the i-th row of B. We write $\mathbf{w}(B_i) = t$, $1 \le i \le 4$, if the number of non-zero entries of B_i equals to t. Based on the values of $\mathbf{w}(B_i)$, we distinguish three cases:

Case 1: $\mathbf{w}(B_i) = 1$ for some $1 \le i \le 4$.

Without loss of generality, let $\mathbf{w}(B_1) = 1$. Then, M_1 (the first row of M) would be a row of C^4. Since, each row of C^4 has at least a zero entry, so, by Theorem 1, M could not be an MDS matrix.

Case 2: $\mathbf{w}(B_i) = 2$ for some $1 \le i \le 4$ and $\mathbf{w}(B_j) \neq 1$ for each $1 \le j \le 4$.

Without loss of generality, let $\mathbf{w}(B_1) = 2$; so, we have 6 vectors $B_1 \in R^4$ as follows:

$$(I, I, \mathbf{0}, \mathbf{0}), (I, \mathbf{0}, I, \mathbf{0}), (I, \mathbf{0}, \mathbf{0}, I),$$
$$(\mathbf{0}, I, I, \mathbf{0}), (\mathbf{0}, I, \mathbf{0}, I), (\mathbf{0}, \mathbf{0}, I, I).$$

It turns out that the corresponding first rows of M are as follows, respectively:

$$(I, I \oplus A, A, \mathbf{0}), (I, A, A, A), (I \oplus A, A \oplus A^2, \mathbf{0}, I),$$
$$(\mathbf{0}, I, I \oplus A, A), (A, I \oplus A^2, A, I), (A, A^2, I, I \oplus A).$$

Hence, given that for $B_1 \in \{(I, I, \mathbf{0}, \mathbf{0}), (I, \mathbf{0}, \mathbf{0}, I), (\mathbf{0}, I, I, \mathbf{0})\}$ the corresponding matrix M has at least one zero entry, so, such matrix M could not be MDS. For the remaining B_1's, i.e. $B_1 \in \mathcal{S} = \{(I, \mathbf{0}, I, \mathbf{0}), (\mathbf{0}, I, \mathbf{0}, I), (\mathbf{0}, \mathbf{0}, I, I)\}$, it is enough to verify the following three subcases:

Case 2.1: $\mathbf{w}(B_i) \geq 3$, $i = 2, 3, 4$.

By programming, we obtained all of the invertible matrices B satisfying the conditions of this subcase. Each matrix B is equivalent to one of the following matrices:

$(3db7, M_{\{1,3\}\{3,4\}}), (3eb7, M_{\{1,4\}\{1,4\}}), (3fd7, M_{\{2,4\}\{3,4\}}),$
$(3fdb, M_{\{1,4\}\{3,4\}}), (3fe7, M_{\{2,4\}\{3,4\}}), (3feb, M_{\{1,4\}\{3,4\}}),$
$(5db7, M_{\{1,2\}\{3,4\}}), (5ed7, M_{\{1,3\}\{3,4\}}), (5fdb, M_{\{1,3\}\{3,4\}}),$
$(5fe7, M_{\{2,4\}\{3,4\}}), (5fed, M_{\{1,4\}\{3,4\}}), (5fb7, M_{\{2,4\}\{2,4\}})),$
$(aeb7, M_{\{1,3\}\{1,2\}}), (aedb, M_{\{1,4\}\{1,2\}}), (afb7, M_{\{2,4\}\{3,4\}}),$
$(afdb, M_{\{1,4\}\{1,2\}}), (afe7, M_{\{2,4\}\{3,4\}}), (afed, M_{\{2,4\}\{1,2\}}).$

Note that, the above list also contains a related non-invertible submatrix of the corresponding matrix $M = BC^4$.

Case 2.2: $B_2 \in \mathcal{S}$, $\mathbf{w}(B_i) \geq 3$, $i = 3, 4$.

In this subcase, any invertible matrix B is equivalent to one of the following matrices which are listed by a related non-invertible submatrix of the corresponding matrix M:

$(53b7, M_{\{2,3\}\{3,4\}}), (53d7, M_{\{1,3\}\{3,4\}}), (53e7, M_{\{2,4\}\{1,4\}}),$
$(53f7, M_{\{1,4\}\{1,2\}}), (53fb, M_{\{2,4\}\{3,4\}}), (53fd, M_{\{1,4\}\{3,4\}}),$
$(53fe, M_{\{1,3\}\{1,3\}}), (a3b7, M_{\{2,3\}\{3,4\}}), (a3db, M_{\{1,2\}\{1,2\}}),$
$(a3eb, M_{\{1,2\}\{1,2\}}), (a3f7, M_{\{1,2\}\{1,2\}}), (a3fb, M_{\{1,2\}\{1,2\}}),$
$(a3fd, M_{\{1,2\}\{1,2\}}), (a3fe, M_{\{1,2\}\{1,2\}}), (a5b7, M_{\{1,2\}\{1,3\}}),$
$(a5db, M_{\{1,2\}\{1,3\}}), (a5e7, M_{\{1,2\}\{1,3\}}), (a5ed, M_{\{2,4\}\{3,4\}}).$

Case 2.3: $B_i, B_j \in \mathcal{S}$, $i, j > 1$, $i \neq j$.

By the assumptions of this subcase, each invertible matrix B is equivalent to one of the following four matrices:

$$(a537, M_{\{1,3\}\{1,2\}}), (a53b, M_{\{1,3\}\{1,2\}}),$$
$$(a53d, M_{\{1,3\}\{1,2\}}), (a53e, M_{\{1,3\}\{1,2\}}).$$

Case 3: $\mathbf{w}(B_i) \geq 3$, $1 \leq i \leq 4$, and $B \not\equiv cycl(\mathbf{0}, I, I, I)$.

In this case, each invertible matrix B is equivalent to one of the following four matrices:

$$(fdb7, M_{\{1,4\}\{3,4\}}), (feb7, M_{\{1,4\}\{3,4\}}),$$

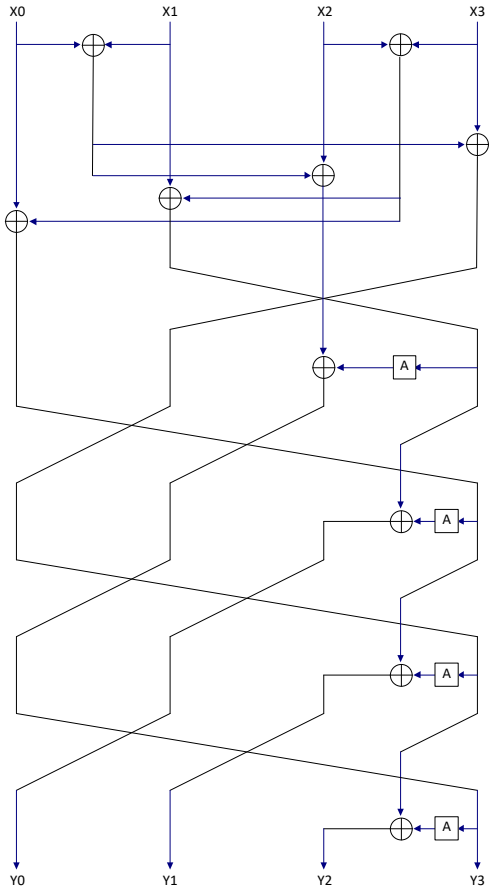

Figure 1. The corresponding diffusion layer of (2)

$$(fed7, M_{\{1,3\}\{1,2\}}), (fedb, M_{\{1,3\}\{1,2\}}).$$

Up to now, our analysis for different cases of B shows that the matrix $M = BC^4$ would not be MDS if $B \not\equiv cycl(\mathbf{0}, I, I, I)$. So, we verify the case $B \equiv cycl(0, I, I, I)$ separately in the next theorem.

Theorem 3. *Let* $R = \{\mathbf{0}, I\} \subset \mathcal{M}_m(\mathbb{F}_2)$, $A \in \mathcal{M}_m(\mathbb{F}_2)$, $B \equiv cycl(\mathbf{0}, I, I, I) \in \mathcal{M}_4(R)$ *and* $C = comp(I, A, \mathbf{0}, \mathbf{0})$. *The matrix* $M = BC^4$ *is MDS with respect to m-bit inputs if and only if* $A, I \oplus A^3$ *and* $I \oplus A^7$ *are invertible.*

Proof: Let $x = (x_3, x_2, x_1, x_0)$ and $y = (y_3, y_2, y_1, y_0)$ be the inputs and outputs of the linear mapping corresponding to the matrix M, respectively; i.e. $y = xM$. Without loss of generality, let $B = cycl(\mathbf{0}, I, I, I)$. The explicit relations between x and y are

$$M : \begin{cases} y_0 & = x_1 \oplus x_2 \oplus x_3 \oplus (x_0 \oplus x_2 \oplus x_3)A, \\ y_1 & = x_0 \oplus x_2 \oplus x_3 \oplus (x_0 \oplus x_1 \oplus x_3)A, \\ y_2 & = x_0 \oplus x_1 \oplus x_3 \oplus (x_0 \oplus x_1 \oplus x_2)A \\ & \quad \oplus (x_1 \oplus x_2 \oplus x_3)A^2, \\ y_3 & = x_0 \oplus x_1 \oplus x_2 \oplus (x_1 \oplus x_2 \oplus x_3)A. \end{cases} \quad (2)$$

The corresponding diffusion layer of M is illustrated

in Figure 1. We must show that for each $x \neq \mathbf{0}$, $wt_m(x)+wt_m(y) \geq 5$ is satisfied if and only if $A, I \oplus A^3$ and $I \oplus A^7$ are invertible. We have $det_R(M) = I$. By Theorem 2, $det_{\mathbb{F}_2}(M) = det_{\mathbb{F}_2}(det_R(M)) = 1$; so M is invertible. Thus, for every $x \neq \mathbf{0}$ we have $y \neq \mathbf{0}$. This means that if $wt_m(x) = 4$, then $wt_m(y) \geq 1$ for any choice of A. In the following, we show that if $wt_m(x) = i, i = 1, 2, 3$, then $wt_m(xM) \geq 5 - i$ provided that $A, I \oplus A^3$ and $I \oplus A^7$ are invertible and vice versa.

Case 1: $wt_m(x) = 1$.

In this case, we should verify four subcases in which, exactly one of x_i's, $i = 0, 1, 2, 3$ are non-zero. Now, consider the subcase,

$$x_0 \neq \mathbf{0}, x_1 = x_2 = x_3 = \mathbf{0} \ \ or \ \ x = (\mathbf{0}, \mathbf{0}, \mathbf{0}, x_0 \neq \mathbf{0}).$$

In this subcase, (2) will be simplified to

$$M : \begin{cases} y_0 &= x_0 A, \\ y_1 &= x_0(I \oplus A), \\ y_2 &= x_0(I \oplus A), \\ y_3 &= x_0. \end{cases}$$

If M is an MDS matrix, then y_0, y_1, y_2, y_3 must be non-zero for any choice of $x_0 \neq \mathbf{0}$. Clearly, y_3 is nonzero. By the basic theorems of matrix theory y_0, y_1, y_2 are non-zero if and only if A and $I \oplus A$ are invertible. Similarly, the other three subcases adds the invertibility of $I \oplus A \oplus A^2, A \oplus A^2$ and $I \oplus A^2$ to the previous conditions. Since, $A \oplus A^2 = A(I \oplus A)$ and $I \oplus A^2 = (I \oplus A)^2$, the conditions derived from this subcase would be the invertibility of $A, I \oplus A$ and $I \oplus A \oplus A^2$.

Case 2 $wt_m(x) = 2$.

In this case, we have six different subcases. We analyze the subcase

$$x_0 \neq \mathbf{0}, x_1 \neq \mathbf{0}, x_2 = x_3 = \mathbf{0} \ \ or \ \ x = (\mathbf{0}, \mathbf{0}, x_1 \neq \mathbf{0}, x_0 \neq \mathbf{0}),$$

in details. By the assumptions, (2) will be reduced to

$$M : \begin{cases} y_0 &= x_1 \oplus x_0 A, \\ y_1 &= x_0(I \oplus A) \oplus x_1 A, \\ y_2 &= x_0(I \oplus A) \oplus x_1(I \oplus A \oplus A^2), \\ y_3 &= x_0 \oplus x_1(I \oplus A). \end{cases} \quad (3)$$

To guarantee $wt_m(x) + wt_m(y) \geq 5$ we should find conditions for which at most one of y_i's, $i = 0, 1, 2, 3$, is zero. Now, let $y_0 = x_1 \oplus x_0 A = \mathbf{0}$. Then we have $x_1 = x_0 A$. Replacing x_1 in (3) we obtain,

$$M : \begin{cases} y_1 &= x_0(I \oplus A \oplus A^2), \\ y_2 &= x_0(I \oplus A^2 \oplus A^3), \\ y_3 &= x_0(I \oplus A \oplus A^2). \end{cases}$$

These equations imply that y_1, y_2, y_3 are non-zero if and only if $I \oplus A \oplus A^2$ and $I \oplus A^2 \oplus A^3$ are invertible. Moreover, the invertibility of $I \oplus A \oplus A^2$ and $I \oplus A^2 \oplus A^3$ impose that y_0 and y_1 can not be zero simultaneously.

Now, let $y_1 = x_0(I \oplus A) \oplus x_1 A = \mathbf{0}$. We get $x_0(I \oplus A) = x_1 A$. As stated before, y_0 and y_1 can not be zero simultaneously. Thus, we have $y_0 \neq \mathbf{0}$. Replacing $x_0(I \oplus A)$ in the third equation of (3), we get

$$y_2 = x_1(I \oplus A^2).$$

For y_2 to be non-zero, $I \oplus A^2$ should be invertible, which we have already taken. Now, from the fourth equation of (3) we have

$$\begin{aligned} y_3(I \oplus A) &= x_0(I \oplus A) \oplus x_1(I \oplus A^2) = x_1 A \oplus x_1(I \oplus A^2) \\ &= x_1(I \oplus A \oplus A^2). \end{aligned}$$

This equation implies that $y_3 \neq \mathbf{0}$ if and only if $I \oplus A \oplus A^2$ is invertible. This condition has also been appeared in the previous subcases. The proof procedure implies that the set of $\{y_0, y_1, y_2\}$ could not have more than one zero element. So, it turns out that if $y_2 = \mathbf{0}$, then $y_0, y_1 \neq \mathbf{0}$.

Now let $y_2 = x_0(I \oplus A) \oplus x_1(I \oplus A \oplus A^2) = \mathbf{0}$. We get $x_0(I \oplus A) = x_1(I \oplus A \oplus A^2)$. As discussed before, y_0 and y_1 in this subcase would be non-zero. From the fourth equation of (3) we have

$$\begin{aligned} y_3(I \oplus A) &= x_0(I \oplus A) \oplus x_1(I \oplus A^2) \\ &= x_1(I \oplus A \oplus A^2) \oplus x_1(I \oplus A^2) = x_1 A. \end{aligned}$$

Here, y_3 is non-zero if and only if A is invertible, which we obtained before. Similarly, based upon the imposed conditions, the proof procedure implies that the set of $\{y_0, y_1, y_2, y_3\}$ could not have more than one zero element. Thus $y_3 = \mathbf{0}$ implies that $y_0, y_1, y_2 \neq \mathbf{0}$.

We verified the remaining five subcases (with $wt_m(x) = 2$) similar to the procedure of the proof in this case. Only the invertibility of $I \oplus A \oplus A^3$ would be added to the previous conditions.

Case 3 $wt_m(x) = 3$.

In this case, we verify the matrix M^{-1} in order to find conditions for which $wt_m(y) \geq 2$. By matrix calculations, the explicit relations between x and y such that $x = yM^{-1}$ is as follows (the corresponding diffusion layer of M^{-1} is illustrated in Figure 2):

$$M^{-1} : \begin{cases} x_0 &= y_1 \oplus y_2 \oplus y_3 \oplus (y_0 \oplus y_2 \oplus y_3)A \\ &\oplus (y_1 \oplus y_3)A^2 \oplus y_2 A^3 \oplus y_3 A^4, \\ x_1 &= y_0 \oplus y_2 \oplus y_3 \oplus (y_0 \oplus y_1 \oplus y_3)A \\ &\oplus (y_1 \oplus y_2)A^2 \oplus (y_2 \oplus y_3)A^3 \oplus y_3 A^4, \\ x_2 &= y_0 \oplus y_1 \oplus y_3 \oplus (y_0 \oplus y_1 \oplus y_2)A \\ &\oplus (y_1 \oplus y_2 \oplus y_3)A^2 \oplus (y_2 \oplus y_3)A^3 \oplus y_3 A^4, \\ x_3 &= y_0 \oplus y_1 \oplus y_2 \oplus (y_1 \oplus y_2 \oplus y_3)A \\ &\oplus (y_2 \oplus y_3)A^2 \oplus y_3 A^3. \end{cases} \quad (4)$$

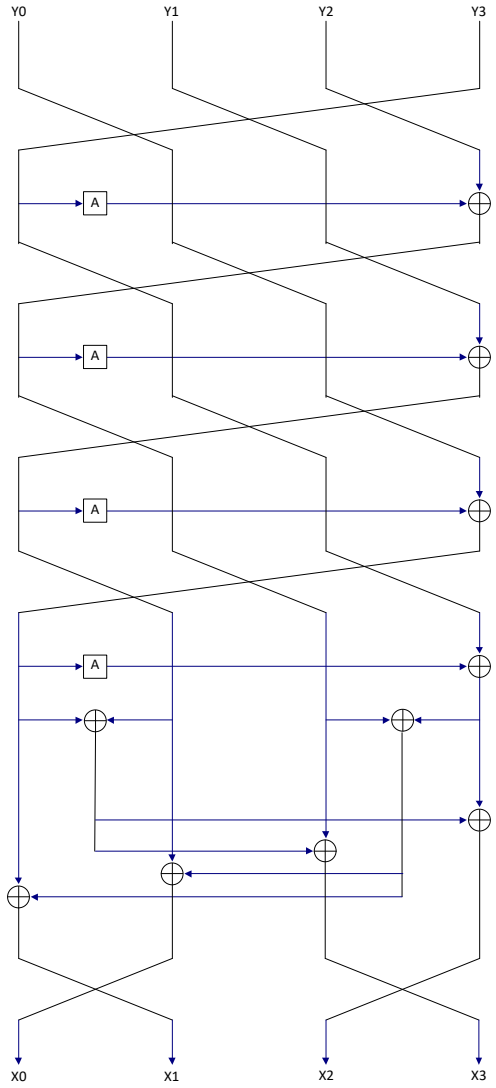

Figure 2. The corresponding diffusion layer of (4)

Now, let $y_3 \neq \mathbf{0}$ and $y_0 = y_1 = y_2 = \mathbf{0}$. By these assumptions, (4) will be simplified to

$$M^{-1}: \begin{cases} x_0 &= y_3(I \oplus A)(I \oplus A^2 \oplus A^3), \\ x_1 &= y_3(I \oplus A)^2(I \oplus A \oplus A^2), \\ x_2 &= y_3(I \oplus A)(I \oplus A \oplus A^3), \\ x_3 &= y_3 A(I \oplus A \oplus A^2). \end{cases}$$

According to the conditions derived from Case 1 and Case 2; i.e. $A, I \oplus A, I \oplus A \oplus A^2, I \oplus A^2 \oplus A^3$ and $I \oplus A \oplus A^3$ are invertible, we conclude that x_0, x_1, x_2 and x_3 are non-zero. The same argument shows that if $wt_m(y) = 1$, then $wt_m(x) = 4$. By contraposition, we obtain that if $wt_m(x) \neq 4$, then $wt_m(y) \neq 1$. So, if $wt_m(x) = 3$ we get $wt_m(y) \neq 1$ and since M is invertible, we have $wt_m(y) \neq 0$. Thus, $wt_m(y) \geq 2$. Therefore, in this case $wt_m(x) + wt_m(y) \geq 5$, adding no extra condition to the set of conditions in Case 1 and Case 2.

Summing up, M is MDS if and only if $A, I \oplus A, I \oplus A \oplus A^2, I \oplus A^2 \oplus A^3$ and $I \oplus A \oplus A^3$ are invertible. Given that $I \oplus A^3 = (I \oplus A)(I \oplus A \oplus A^2)$ and $I \oplus A^7 = (I \oplus A)(I \oplus A \oplus A^3)(I \oplus A^2 \oplus A^3)$, the proof completes. □

To complete the characterization of the matrices mentioned in the beginning of the current section, we applied programming for the other five choices of the matrix C, i.e.

$$C \in \{comp(I, \mathbf{0}, \mathbf{0}, A), comp(I, \mathbf{0}, A, \mathbf{0}), comp(A, I, \mathbf{0}, \mathbf{0}), comp(A, \mathbf{0}, I, \mathbf{0}), comp(A, \mathbf{0}, \mathbf{0}, I)\}.$$

Regarding Theorem 1 and Theorem 2, the results of the programming shows that in the case of

$$C \in \{comp(I, \mathbf{0}, A, \mathbf{0}), comp(A, I, \mathbf{0}, \mathbf{0}), comp(A, \mathbf{0}, I, \mathbf{0}), comp(A, \mathbf{0}, \mathbf{0}, I)\},$$

for each choice of $B \in \mathcal{M}_4(R)$, the matrix $M = BC^4$ could not be MDS. Further, the next corollary, which could also be proved in the same manner as Theorem 3, is reaffirmed by our programming.

Corollary 1. *For $R = \{\mathbf{0}, I\} \subset \mathcal{M}_m(\mathbb{F}_2)$, $A \in \mathcal{M}_m(\mathbb{F}_2)$, $B \in \mathcal{M}_4(R)$ and $C = comp(I, \mathbf{0}, \mathbf{0}, A)$, the matrix $M = BC^4$ is MDS if and only if $B \equiv cycl(\mathbf{0}, I, I, I)$ and $A, I \oplus A^3$ and $I \oplus A^7$ are invertible.*

4 The Lightest 4×4 MDS Matrices

In this section, we first give examples of the most efficient MDS matrices derived from Theorem 3 and Corollary 1. Then we propose families of lightweight 4×4 MDS matrices as well as an improved family, which has the lightest implementation cost up to now.

Since the structures of the matrices in Theorem 3 and Corollary 1 are similar, the implementation cost of the proposed matrices and their inverses require $10m+4a$ XORs for m-bit inputs, according to Figure 1 and Figure 2. Here, a is the number of XORs needed to implement the matrix A. As, A and $I \oplus A^3$ should be invertible, we have $a \geq 1$. The most applicable cases for hardware-oriented diffusion layers used in lightweight ciphers are $m = 4, 5, 6, 7, 8$. To construct the lightest 4×4 MDS matrices with respect to 4-bit entries, given in Theorem 3 and Corollary 1, the first step is to find invertible matrices $A \in \mathcal{M}_4(\mathbb{F}_2)$ with $a = 1$ such that $I \oplus A^3$ and $I \oplus A^7$ are invertible. We exhaustively searched among all the invertible matrices in $\mathcal{M}_4(\mathbb{F}_2)$ satisfying the desired conditions. In (5), we list the set of all 48 matrices with $a = 1$. So, choosing A from (5), the implementation cost of the resulted MDS matrices as well as their inverses is 44 XORs.

Table 2. List of the lightest 4×4 MDS matrices: $C = comp(I, \mathbf{0}, \mathbf{0}, A), C^* = comp(A, \mathbf{0}, \mathbf{0}, I)$.

Matrix	Corresponding conditions (matrices should be invertible)
$M_1 = 6419 \times C \times 8214 \times C \times 8241 \times C \times 8214 \times C \times 81a5$	$A, I \oplus A^3, I \oplus A^7, I \oplus A \oplus A^4$
$M_2 = C \times 2194 \times C \times 8214 \times C \times 42c1 \times C \times 183c$	$A, I \oplus A^3, I \oplus A^7, I \oplus A \oplus A^4$
$M_3 = 6419 \times C \times 2194 \times C \times 8214 \times C \times 42c1 \times C$	$A, I \oplus A^3, I \oplus A^7, I \oplus A \oplus A^4$
$M_4 = C \times 8214 \times C \times 2814 \times C \times 8214 \times C \times 5adb$	$A, I \oplus A^3, I \oplus A \oplus A^3, I \oplus A \oplus A^4$
$M_5 = 124c \times C^* \times 2814 \times C \times 8214 \times C \times 8241 \times C \times 15da$	$A, I \oplus A^3, I \oplus A^7$

$$\begin{aligned}
&1286, 1294, 1846, 18c2, 1942, 1a84, 214a, 2158,\\
&281c, 2854, 2948, 2a14, 3814, 3842, 418a, 41c2,\\
&421c, 4298, 4318, 4382, 5182, 5284, 6148, 6218.\\
&1285, 12a4, 1684, 1843, 1862, 1c42, 2149, 2168,\\
&2548, 2815, 2834, 2c14, 4183, 41a2, 4219, 4238,\\
&4582, 4618, 9284, 9842, a148, a814, c182, c218.
\end{aligned} \tag{5}$$

Similarly, for $5 \leq m \leq 8$, we have exhaustively searched invertible matrices $A \in \mathcal{M}_m(\mathbb{F}_2)$ with 1 XOR implementation cost, such that $I \oplus A^3$ and $I \oplus A^7$ are invertible. Our experimental results show that there are 240, 2160, 20160 and 93600 such matrices for $m = 5, 6, 7, 8$, respectively. We present some samples for each value of m in (6). Hence, employing any of those matrices in the constructions derived from Theorem 3 and Corollary 1, leads to MDS matrices over m-bit inputs with the implementation cost of $10m+4$ XORs.

$$\begin{aligned}
&[2; 5; 4; (1, 5); 3], [3; (3, 5); 2; 1; 4],\\
&[(2, 3); 1; 4; 5; 2], [(2, 5); 4; 5; 3; 1],\\
&[2; (2, 5); 1; 6; 4; 3], [2; (3, 4); 6; 3; 1; 5],\\
&[(1, 2); 3; 2; 5; 6; 1], [(1, 6); 1; 2; 3; 4; 5],\\
&[(4, 6); 1; 2; 3; 4; 7; 5], [2; 6; 4; (1, 7); 3; 7; 5],\\
&[7; (1, 4); 5; 1; 2; 3; 6], [7; 1; 2; 5; (1, 6); 3; 4],\\
&[8; 7; 5; 6; 4; 3; 1; (2, 6)], [8; 7; 5; (3, 4); 6; 4; 1; 2],\\
&[(2, 3); 1; 4; 5; 6; 7; 8; 2], [(2, 3); 6; 8; 1; 7; 4; 3; 5].
\end{aligned} \tag{6}$$

Now, we present the main experimental results of the paper. We use programming to find potential 4×4 MDS matrices with an implementation cost less than $10m + 4$ XORs on m-bit inputs based on the product of companion and binary matrices. The investigated matrices are of the form

$$M = B_1 C_1 B_2 C_2 B_3 C_3 B_4 C_4 B_5,$$

where, B_i's, $1 \leq i \leq 5$, are invertible binary matrices and for $1 \leq i \leq 4$,

$$\begin{aligned}
C_i \in \{&comp(I, A, \mathbf{0}, \mathbf{0}), comp(I, \mathbf{0}, A, \mathbf{0}),\\
&comp(I, \mathbf{0}, \mathbf{0}, A), comp(A, I, \mathbf{0}, \mathbf{0}),\\
&comp(A, \mathbf{0}, I, \mathbf{0}), comp(A, \mathbf{0}, \mathbf{0}, I)\}.
\end{aligned}$$

As stated before, there are 20160 4×4 invertible binary matrices. So, an exhaustive search in this space includes the investigation of $(20160)^5 6^4 \approx 2^{81.83}$ matrices which is infeasible. In this class of matrices we could take $C_i \in \{comp(I, \mathbf{0}, \mathbf{0}, A), comp(A, \mathbf{0}, \mathbf{0}, I)\}$, $1 \leq i \leq 4$, because,

$$\begin{aligned}
comp(I, A, \mathbf{0}, \mathbf{0}) &= 2481\, comp(I, \mathbf{0}, \mathbf{0}, A)\, 8124,\\
comp(I, \mathbf{0}, A, \mathbf{0}) &= 8241\, comp(I, \mathbf{0}, \mathbf{0}, A)\, 8412,\\
comp(A, I, \mathbf{0}, \mathbf{0}) &= 2481\, comp(A, \mathbf{0}, \mathbf{0}, I)\, 8124,\\
comp(A, \mathbf{0}, I, \mathbf{0}) &= 8241\, comp(A, \mathbf{0}, \mathbf{0}, I)\, 8412.
\end{aligned}$$

This reduces the space to $(20160)^5 2^4 \approx 2^{75.49}$ matrices which is yet infeasible. Using Theorem 1, Theorem 2 and limiting the variations of B_i's, $1 \leq i \leq 5$, we found five families of MDS matrices. The acquired matrices and their corresponding conditions to be MDS are presented in Table 2. For instance, we elaborate on M_2:

$$M_2 = \begin{pmatrix} I \oplus A & I & A & A \\ I & I \oplus A & A & I \oplus A \\ A^2 & A^2 & I & I \oplus A \\ I \oplus A^3 & A^3 & I \oplus A & I \oplus A \oplus A^2 \end{pmatrix}.$$

According to Theorem 2, the determinants of all square sub-matrices of M_2 are

$$\begin{aligned}
&I,\ A,\ I \oplus A,\ A^2,\ (I \oplus A)^2,\ I \oplus A^3,\\
&I \oplus A \oplus A^3,\ I \oplus A^2 \oplus A^3,\\
&A^2(I \oplus A \oplus A^2),\ (I \oplus A)^3,\\
&(I \oplus A)^4,\ (I \oplus A \oplus A^2)^2,\\
&(I \oplus A)(I \oplus A \oplus A^3),\ (I \oplus A)(I \oplus A^3),\ I \oplus A \oplus A^4.
\end{aligned} \tag{7}$$

Given that $I \oplus A^3 = (I \oplus A)(I \oplus A \oplus A^2)$, $I \oplus A^7 = (I \oplus A)(I \oplus A \oplus A^3)(I \oplus A^2 \oplus A^3)$ and regarding (7), all

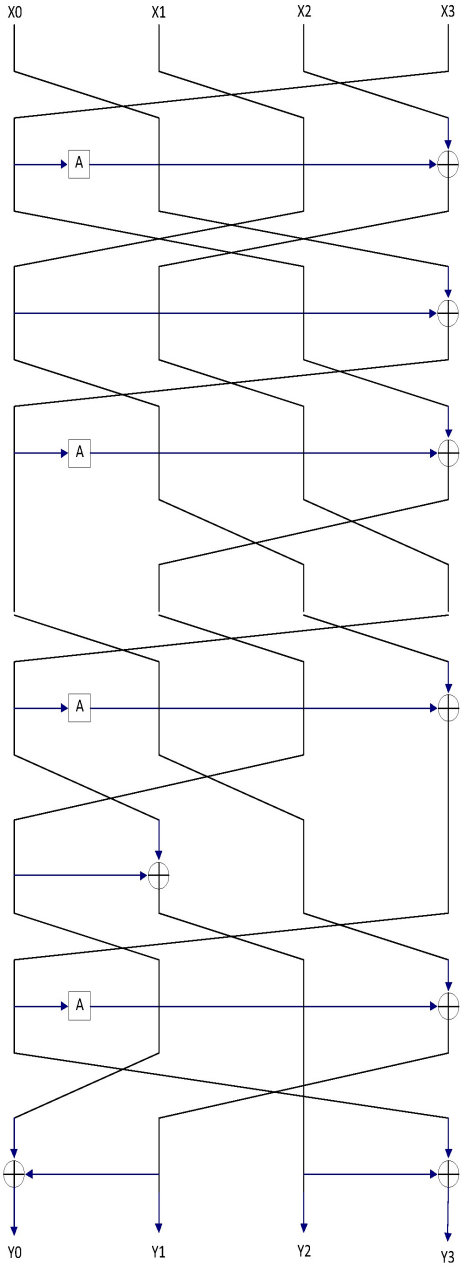

Figure 3. Implementation of the matrix M_2, in Table 2

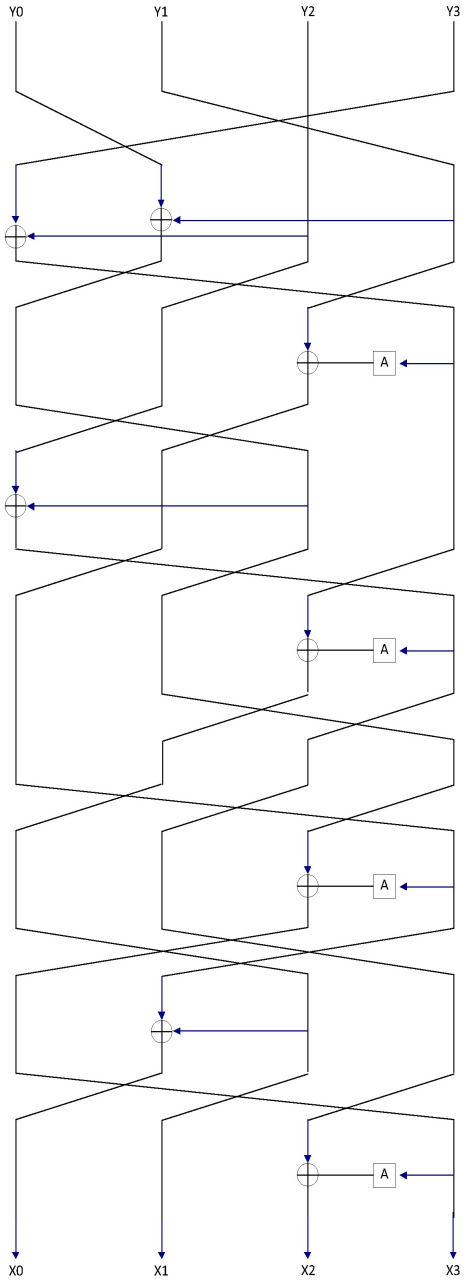

Figure 4. Implementation of the matrix M_2^{-1}, in Table 2

sub-matrices of M_2 are invertible (M_2 is MDS) if and only if $A, I \oplus A^3, I \oplus A^7$ and $I \oplus A \oplus A^4$ are invertible.

To calculate the precise implementation cost of the matrices presented in Table 2, the implementation of M_2 and M_2^{-1} are presented in Figure 3 and Figure 4, respectively. So, the implementation cost of M_2 and M_2^{-1} over m-bit inputs are $8m + 4a$ XORs, where a is the implementation cost of the matrix A. Similar calculations show that the implementation costs of all presented matrices in Table 2 and their corresponding inverses are equal (note that, $5adb = 183c \times a185$ and $15da = 4138 \times a185$). Thus, an exhaustive search over 1 XOR matrices $A \in \mathcal{M}_4(\mathbb{F}_2)$ for which $A, I \oplus A^3, I \oplus A^7, I \oplus A \oplus A^4$ are invertible, results in 24 matrices which are listed in (8). This list also presents all of 4×4, 1 XOR matrices A, for which $A, I \oplus A^3, I \oplus A \oplus A^3, I \oplus A \oplus A^4$ are invertible. This means that, for any choice of A from (8), all of the matrices in Table 2 would be MDS over 4-bit inputs with 36 XOR implementation cost.

$$\begin{aligned} &1285, 12a4, 1684, 1843, 1862, 1c42, 2149, 2168, \\ &2548, 2815, 2834, 2c14, 4183, 41a2, 4219, 4238, \\ &4582, 4618, 9284, 9842, a148, a814, c182, c218. \end{aligned} \tag{8}$$

We have exhaustively searched invertible matrices

$A \in \mathcal{M}_m(\mathbb{F}_2)$, $5 \leq m \leq 8$, with 1 XOR implementation cost, such that $I \oplus A^3, I \oplus A^7$ and $I \oplus A \oplus A^4$ are invertible. There are 240, 2160, 20160 and 40320 such matrices for $m = 5, 6, 7, 8$, respectively. Hence, the matrices in Table 2 produce MDS matrices with $8m + 4$ XOR implementation cost.

4.1 The Improved Implementation Cost of M_5 in Table 2

Here, we improve the implementation cost of the M_5 family of matrices presented in Table 2 which culminates in 4×4 MDS matrices over 4-bit inputs with 35 XOR implementation cost. Using some matrix calculations we have:

$$C^* \times 2814 \times C =$$

$$\begin{pmatrix} \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I \\ A & \mathbf{0} & \mathbf{0} & I \end{pmatrix} \times \begin{pmatrix} \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I \\ \mathbf{0} & I & \mathbf{0} & \mathbf{0} \end{pmatrix} \times \begin{pmatrix} \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I \\ I & \mathbf{0} & \mathbf{0} & A \end{pmatrix} =$$

$$\begin{pmatrix} \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I \\ I & \mathbf{0} & \mathbf{0} & I \end{pmatrix} \times \begin{pmatrix} \mathbf{0} & \mathbf{0} & A & \mathbf{0} \\ I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I \\ \mathbf{0} & I & \mathbf{0} & \mathbf{0} \end{pmatrix} \times \begin{pmatrix} \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I \\ I & \mathbf{0} & \mathbf{0} & A \end{pmatrix} =$$

$$\begin{pmatrix} \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I \\ I & \mathbf{0} & \mathbf{0} & I \end{pmatrix} \times \begin{pmatrix} \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I \\ \mathbf{0} & I & \mathbf{0} & \mathbf{0} \end{pmatrix} \times \begin{pmatrix} \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & A \\ I & \mathbf{0} & \mathbf{0} & A \end{pmatrix} =$$

$$\begin{pmatrix} I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I \\ \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I & I & \mathbf{0} \end{pmatrix} \times \begin{pmatrix} \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & A \\ I & \mathbf{0} & \mathbf{0} & A \end{pmatrix}. \tag{9}$$

Let $x = (x_3, x_2, x_1, x_0)$ and $y = (y_3, y_2, y_1, y_0)$ be the inputs and outputs of the following matrix; i.e. $y = xN$:

$$N = \begin{pmatrix} \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & A \\ I & \mathbf{0} & \mathbf{0} & A \end{pmatrix}.$$

The explicit relations between x and y are

$$N : \begin{cases} y_0 &= (x_0 \oplus x_1)A, \\ y_1 &= x_2, \\ y_2 &= x_3, \\ y_3 &= x_0. \end{cases} \tag{10}$$

By our proposed implementation method, the implementation cost of $C^* \times 2814 \times C$ is $2m + 2a$ XORs for m-bit inputs, where a is the implementation cost of A. By equations (9) and (10), the implementation cost of $C^* \times 2814 \times C$ reduces to $2m + a$ XORs. If we choose A from (5), then the MDS matrices produced by M_5 will take 35 XORs. As stated before, for $5 \leq m \leq 8$, there are 240, 2160, 20160 and 93600 invertible matrices $A \in \mathcal{M}_m(\mathbb{F}_2)$ with 1 XOR implementation cost, such that $I \oplus A^3$ and $I \oplus A^7$ are invertible. So, the implementation cost of M_5 for $m = 8$, takes 67 XORs. The implementation of the inverse of $C^* \times 2814 \times C$ is improved in the same manner as (9); the only difference is that we need A^{-1}. Fortunately, the implementation cost of any binary 1 XOR matrix equals to its inverse. Therefore, the implementation cost of M_5^{-1} is also reduced to 35 and 67 XORs for $m = 4, 8$, respectively.

Since the lightest 4×4 MDS matrices (up to now) are presented in the current paper and [16], we briefly explain the differences of the search strategies. In general, the authors of [16] follow the global optimization of MDS matrices rather than the optimization of coefficients. They use only three linear operations in the construction of MDS diffusion layers: XOR of two words, linear mapping on a word, and reusing a register. Starting from the identity mapping, a search algorithm (based on the Dijkstra algorithm) is applied to add one of the three mentioned linear mappings to the previous ones until it finds an MDS matrix with optimum implementation cost and some conditions on the coefficients. The conditions on the coefficients come from the way of checking the MDS property based on the fact: a matrix with coefficients in a commutative ring is MDS if and only if all of its minors are invertible. In contrast, our strategy to find lightweight 4×4 MDS matrices is a smart search based upon Theorem 3 and Corollary 1. Characterization of the MDS property of a family of 4×4 matrices in Theorem 3 and Corollary 1 guarantees that the product of sparse recursive and binary matrices leads to an optimization in implementation cost of the classical recursive MDS matrices. So, we extend the search space to verify the MDS property of matrices constructed from the production of sparse recursive and binary matrices, alternatively. It is obvious that investigating all matrices in this space is infeasible due to lack of enough time. However, a case study on the mentioned class of matrices leads to some efficient MDS matrices

which are reported in Table 2. The presented results in Table 2 and the time we spent to achieve them give a good indication that more investigation on this type of matrices may produce better results in lightweight MDS matrices than the existent ones.

5 Conclusion

In this paper, we propose new families of lightweight 4×4 MDS matrices with respect to m-bit inputs based on the product of binary and companion matrices. This method leads to the construction of lightweight MDS matrices such that the implementation cost of them and their corresponding inverses are equal. The lightest resultant MDS matrices and their inverses need $8m+3$ XORs for $m=4,5,6,7,8$.

In the case of $m=4$, which is more appropriate for constrained hardware-oriented platforms, the provided 4×4 MDS matrices need 35 XORs for implementation, which is the same as the lightest existent ones.

References

[1] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 222–239. Springer, 2011.

[2] Mahdi Sajadieh, Mohammad Dakhilalian, Hamid Mala, and Pouyan Sepehrdad. Efficient recursive diffusion layers for block ciphers and hash functions. *J. Cryptology*, 28(2):240–256, 2015.

[3] Shengbao Wu, Mingsheng Wang, and Wenling Wu. Recursive diffusion layers for (lightweight) block ciphers and hash functions. In *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, pages 355–371, 2012.

[4] Guang Zeng, Wenbao Han, and Kaicheng He. High efficiency feedback shift register: sigma-lfsr. *IACR Trans. Symmetric Cryptol.*, 2007:114, 2007.

[5] Hong Xu, Yonghui Zheng, and Xuejia Lai. Construction of perfect diffusion layers from linear feedback shift registers. *IET Information Security*, 9(2):127–135, 2015.

[6] Christof Beierle, Thorsten Kranz, and Gregor Leander. Lightweight multiplication in GF(2ˆn) with applications to MDS matrices. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 625–653. Springer, 2016.

[7] Yongqiang Li and Mingsheng Wang. On the construction of lightweight circulant involutory MDS matrices. In *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, pages 121–139, 2016.

[8] Jian Bai and Dingkang Wang. The lightest 4x4 MDS matrices over GL(4, $\mathbb{F}_2$). *IACR Trans. Symmetric Cryptol.*, 2016:686, 2016.

[9] Sumanta Sarkar and Habeeb Syed. Lightweight diffusion layer: Importance of toeplitz matrices. *IACR Trans. Symmetric Cryptol.*, 2016(1):95–113, 2016.

[10] Zhiyuan Guo, Renzhang Liu, Si Gao, Wenling Wu, and Dongdai Lin. Direct construction of optimal rotational-xor diffusion primitives. *IACR Trans. Symmetric Cryptol.*, 2017(4):169–187, 2017.

[11] Victor Cauchois, Pierre Loidreau, and Nabil Merkiche. Direct construction of quasi-involutory recursive-like MDS matrices from 2-cyclic codes. *IACR Trans. Symmetric Cryptol.*, 2016(2):80–98, 2016.

[12] Shiyi Zhang, Yongjuan Wang, Yang Gao, and Tao Wang. On the construction of the 4 x 4 lightest circulant MDS matrices. In *Proceedings of the 2017 International Conference on Cryptography, Security and Privacy, ICCSP 2017, Wuhan, China, March 17 - 19, 2017*, pages 1–6, 2017.

[13] Lijing Zhou, Licheng Wang, and Yiru Sun. On efficient constructions of lightweight MDS matrices. *IACR Trans. Symmetric Cryptol.*, 2018(1):180–200, 2018.

[14] Thorsten Kranz, Gregor Leander, Ko Stoffelen, and Friedrich Wiemer. Shorter linear straight-line programs for MDS matrices. *IACR Trans. Symmetric Cryptol.*, 2017(4):188–211, 2017.

[15] Dylan Toh, Jacob Teo, Khoongming Khoo, and Siang Meng Sim. Lightweight MDS serial-type matrices with minimal fixed XOR count. In *Progress in Cryptology - AFRICACRYPT 2018 - 10th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 7-9, 2018, Proceedings*, pages 51–71, 2018.

[16] Sébastien Duval and Gaëtan Leurent. MDS matrices with lightweight circuits. *IACR Trans. Symmetric Cryptol.*, 2018(2):48–78, 2018.

[17] Ruoxin Zhao, Baofeng Wu, Rui Zhang, and Qian Zhang. Designing optimal implementations of linear layers (full version). Cryptology ePrint Archive, Report 2016/1118, 2016.

[18] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[19] S. Ling and C. Xing. *Coding Theory: A First Course*. Cambridge University Press, 2004.

[20] Mario Blaum and Ron M. Roth. On lowest density MDS codes. *IEEE Trans. Information Theory*, 45(1):46–59, 1999.

[21] Daniel S. Silver Ivan Kovacs and Susan G. Williams. Determinants of commuting-block matrices. *The American Mathematical Monthly*, 106(10):950–952, 1999.

Akbar Mahmoodi Rishakani received his B.S. and M.S. degrees in pure mathematics from Shahid Beheshti University in 2005 and 2008, respectively. He is now Ph.D. student of mathematical cryptography in Shahid Rajaee Teacher Training University under the supervision of Prof. Hamid Reza Maimani and Prof. Nasour Bagheri. His current research interests include information security, cryptology and combinatorics.

Mohammad Reza Mirzaee Shamsabad was born in 1983 in Iran. He received his B.S. in applied mathematics in 2006 from Azad University, his M.S. in pure mathematics in 2010 from Shahid Bahonar University. He is now a candidate of Ph.D. in mathematical cryptography in Shahid Beheshti University under supervision of Prof. Hossein Hajiabolhassan.

Seyed Mojtaba Dehnavi was born in 1975 in Iran. He received his B.S. in applied mathematics and hardware engineering in 2001 from Iranian University of Science and Technology, his M.S. in pure mathematics in 2004 from Amir Kabir University of Technology, and his Ph.D. in mathematical cryptography in 2015 from Kharazmi University under supervision of Prof. Hamid Reza Maimani.

Mohammad Amin Amiri received the M.S. and Ph.D. degrees in electronics from Iran University of Science and Technology (IUST), Tehran, Iran, in 2004 and 2011 respectively. In 2013, he joined the Electrical Engineering Department as an assistant professor at Malek Ashtar University of Technology, Tehran, Iran. His current research interests include digital system design and implementation, fault tolerant design and secure system design.

Hamid Reza Maimani received the M.S. degree in mathematics in 1981 from the Teacher Training University of Tehran, Iran, and the Ph.D. degree in mathematics (combinatorics) in 1986 from Tehran University,Tehran, Iran. He joined Shahid Rajaee Teacher Training University in September 1986, where he is now a Professor in the department of mathematical sciences. His current research interests include graph theory, coding, cryptography and combinatorics.

Nasour Bagheri is an associate professor at electrical engineering department, Shahid Rajaee Teacher Training University, Tehran, Iran. He is the author of more than 60 articles on information security and cryptology. Homepage of the author is available at: https://www.srttu.edu/english-cv-dr-bagheri/.