Construction of MDS Matrices from Generalized Feistel Structures

Mahdi Sajadieh[1] and Mohsen Mousavi[2]

[1] Department of Electrical Engineering, Islamic Azad University, Isfahan (Khorasgan) Branch, Isfahan, Iran, m.sajadieh@khuisf.ac.ir
[2] Department of Mathematics, Malek Ashtar University of Technology, Isfahan, Iran, m.mousavi@mut-es.ac.ir

Abstract. This paper investigates the construction of MDS matrices with generalized Feistel structures (GFS). The approach developed by this paper consists in deriving MDS matrices from the product of several sparser ones. This can be seen as a generalization to several matrices of the recursive construction which derives MDS matrices as the powers of a single companion matrix.
The first part of this paper gives some theoretical results on the iteration of GFS. In second part, using GFS and primitive matrices, we propose some types of sparse matrices that are called extended primitive GFS (EGFS) matrices. Then, by applying binary linear functions to several round of EGFS matrices, lightweight 4×4, 6×6 and 8×8 MDS matrices are proposed which are implemented with 67, 156 and 260 XOR for 8-bit input, respectively. The results match the best known lightweight 4×4 MDS matrix and improve the best known 6×6 and 8×8 MDS matrices.
Moreover, we propose 8×8 Near-MDS matrices such that the implementation cost of the proposed matrices are 108 and 204 XOR for 4 and 8-bit input, respectively. Although none of the presented matrices are involutions, the implementation cost of the inverses of the proposed matrices is equal to the implementation cost of the given matrices. Furthermore, the construction presented in this paper is relatively general and can be applied for other matrix dimensions and finite fields as well.

Keywords: MDS matrix · Generalized Feistel Structures · XOR counts · Circuit

1 Introduction

There are several approaches to construct MDS matrices which can be applied as diffusion layers for block ciphers and hash functions. The first method is based on the algebraic structures such as Cauchy matrix [YMT97, GR13]. The next efficient method to be used in construction of MDS matrices is based on the recursive matrices [GPP11, SDM+12, Ber13, AF14]. The first two approaches are called local optimization, since these methods focus on the implementation cost of entries of an MDS matrix [SKO+15, SS16]. For instance, completes the search for lightweight circulant matrices is provided in [LS16].

First of all, diffusion matrices are quantified with XOR gates in [KPP+14]. Then, binary linear functions are applied to diffusion matrices in [LW16]. Next, two new approaches are proposed which are called global optimization [KLS+17, DL18, LSL+19]. In fact, the first work on local optimization which was popularised by [BKL16] and later leads to global optimization is [JPS+17]. The proposed method in [KLS+17, LSL+19] is an application of heuristics algorithms to obtain a suitable implementation of previously known MDS matrices. The introduced technique in [BKL16, DL18] is a type of search over formal matrices independently of binary linear functions ($\mathbf{L}$) and to instantiate $\mathbf{L}$ later. In fact, the work of [BKL16, DL18] is not an application of heuristics algorithms to obtain an efficient implementation of previously known MDS matrices, but a search for new lightweight MDS matrices starting from the implementation.

The space of matrix explored in this paper is a subspace of the space explored in [DL18]. Although, the presented technique in this paper is quite similar to [DL18], the class of construction considered is different. In fact, the advantage of this work compared to [DL18] is that the smaller search space can be used with larger dimensions.

In recursive or LFSR-based approach, we consider an $n \times n$ companion matrix $\mathbf{A}$ such that the entries of the last row of $\mathbf{A}$ are no-zero elements over a field $\mathbb{F}$. Then we check whether the n-th power of $\mathbf{A}$, denoted with $\mathbf{A}^n$, is an MDS matrix over $\mathbb{F}$. The limitation to the recursive approach is that all entries of the last row of $\mathbf{A}$ must be non-zero. In other words, the number of non-zero entries of $\mathbf{A}$ must be at least $2n-1$. Applying sparse matrices based on Feistel structures is one of the best solutions for the mentioned limitation [WWW12]. Actually, by applying Feistel structures, we can construct an $n \times n$ sparse matrix $\mathbf{B}$ so that $\mathbf{B}^n$ is an MDS matrix and also the number of non-zero entries of $\mathbf{B}$ is less than $2n-1$. The next limitation on the construction of an $n \times n$ MDS matrix, based on the recursive approach or Feistel structures, is that the n-th power of a matrix is used to obtain an MDS matrix. In fact, it is not possible to select an $n \times n$ companion matrix $\mathbf{A}$ such that all entries of $\mathbf{A}^k$ are non-zero provided that $k < n$. In addition, all known lightweight $n \times n$ MDS matrices which are derived from Feistel structures, are constructed from the n-th power of $n \times n$ sparse matrices [WWW12, TTK+18].

Contribution of this work This paper follows a list of recent papers to design new MDS matrices with low implementation costs. Concerning the standard XOR count metric, it yields several new matrices having lower XOR cost than previous results. While for 4×4 matrices, the results match the best known lightweight 4×4 MDS matrix [DL18] the results for 6×6 are slightly better and for 8×8 are substantially better. Compared with the recursive constructions, the matrices considered in this paper are generally sparse.

It is stated in [DL18] that the XOR count may not be the best metric to estimate the true hardware cost of an implementation and because of this we consider the depth of the circuit of the proposed MDS matrices. In other words, depth corresponds to time which implies that the lower depth results in faster operation in parallel implementation. Therefore, we propose MDS matrices that are not only with low implementation cost, but also their circuits have low depth by terminology of the maximum number of gates in each path from all source to the sink [DL18]. On the other hand, for software implementation, we propose MDS matrices with low implementation cost such that the proposed matrices are constructed from the minimum number of binary linear functions.

The structure of the proposed MDS matrices is based on GFS [Shi11]. In addition, the proposed matrices are not only constructed from GFS, but also should be a primitive matrix over $\mathbb{R}$, the field of real numbers. Moreover, binary linear functions are applied to the construction of the proposed matrices. First of all, we propose an 6×6 MDS matrix which is implemented with 156 XOR for 8-bit input such that the proposed matrix is constructed from six binary linear functions. Also, the proposed 6×6 MDS matrix is implemented with 114 XOR for 6-bit input. In addition, an 6×6 MDS matrix for 4-bit input is proposed such that its implementation cost is 90 XOR.

The next new result is the construction of two 8×8 lightweight MDS matrices for 8-bit input. The first 8×8 MDS matrix is implemented with 272 XOR such that the number of binary linear functions which are used in the construction of the proposed matrix is 16. The first proposed 8×8 matrix can be used in software implementation. The implementation cost of the second 8×8 MDS matrix is 260 XOR such that the depth of its circuit is 8. The second proposed 8×8 matrix can be applied in hardware implementation.

By applying GFS and random searches, we could not get any 8×8 MDS matrices for 4-bit input. Therefore, we tried to construct 8×8 lightweight Near-MDS matrices [LW17] for 4-bit input with the following conditions. First, the proposed matrices have low implementation cost. Second, the inverse of the proposed matrices can be implemented such as the proposed matrices. Moreover, the depth of their circuits are less than eight. Finally, in the construction of the proposed matrices the minimum number of binary linear functions are used. Table 1 is provided for comparison our results with the known results.

Table 1: Comparison our results with the best known results.

Type	Input	XOR	Method	Depth	Involutory	Reference
4×4 Matrices						
$\mathbb{F}_{2^4}$	4-bit	64	Hadamard	—	✓	[PSA+18]
$\mathbb{F}_{2^4}$	4-bit	42	Heuristics	—	✓	[KLS+17]
$\mathbb{F}_{2^4}$	4-bit	36	Heuristics	—	×	[KLS+17]
$\mathbb{F}_{2^4}$/0x19	4-bit	36	GFS St.	6	×	Subsection 5.1
$\mathbb{F}_{2^4}$	4-bit	35	GFS St.	5	×	[DL18]
$\mathbb{F}_{2^4}$	4-bit	35	GFS St.	5	×	Subsection 6.1
$\mathbb{F}_{2^8}$	8-bit	158	Hadamard	—	✓	[PSA+18]
$\mathbb{F}_2[\mathbf{L}]$	8-bit	78	Heuristics	4	✓	[LSL+19]
$\mathbb{F}_{2^8}$	8-bit	72	Subfield	—	×	[KLS+17]
$\mathbb{F}_{2^8}$	8-bit	70	GFS St.	5	×	Subsection 6.1
$\mathbb{F}_2[\mathbf{L}]$	8-bit	69	GFS St.	4	×	[DL18]
$\mathbb{F}_2[\mathbf{L}]$	8-bit	68	GFS St.	6	×	Subsection 5.1
$\mathbb{F}_2[\mathbf{L}]$	8-bit	67	GFS St.	5	×	[DL18]
$\mathbb{F}_2[\mathbf{L}]$	8-bit	67	GFS St.	5	×	Subsection 6.1
6×6 Matrices						
$\mathbb{F}_2[\mathbf{L}]$	4-bit	150	COM St.	—	×	[WWW12]
$\mathbb{F}_{2^4}$/0x13	4-bit	**90**	GFS St.	9	×	Subsection 6.2
$\mathbb{F}_{2^6}$/0x43	6-bit	**114**	GFS St.	8	×	Subsection 6.2
$\mathbb{F}_{2^8}$/0x1C3	8-bit	186	DSI St.	—	×	[TTK+18]
$\mathbb{F}_{2^8}$	8-bit	**156**	GFS St.	8	×	Subsection 6.2
8×8 Near-MDS Matrices						
$\mathbb{F}_{2^4}$/0x13	4-bit	216	Circulant	—	×	[LW17]
$\mathbb{F}_{2^4}$/0x13	4-bit	**116**	GFS St.	**6**	×	Subsection 6.3.1
$\mathbb{F}_{2^4}$/0x13	4-bit	**108**	GFS St.	**7**	×	Subsection 6.3.1
$\mathbb{F}_{2^8}$	8-bit	432	Circulant	—	×	[LW17]
$\mathbb{F}_2[\mathbf{L}]$	8-bit	**212**	GFS St.	**6**	×	Subsection 6.3.1
$\mathbb{F}_2[\mathbf{L}]$	8-bit	**204**	GFS St.	**7**	×	Subsection 6.3.1
8×8 MDS Matrices over 8-bit input						
$\mathbb{F}_{2^8}$	8-bit	392	Subfield	—	×	[KLS+17]
$\mathbb{F}_{2^8}$/0x1E7	8-bit	**272**	GFS St.	9	×	Subsection 6.3.2
$\mathbb{F}_{2^8}$/0x187	8-bit	**260**	GFS St.	**8**	×	Subsection 6.3.3

In general, using binary linear functions ($\mathbf{L}$) and applying GFS, we select $n \times n$ sparse matrices $\mathbf{A}_i$ with $1 \leq i \leq k$ such that $\mathbf{A}_i$'s satisfy the following conditions. First $\mathbf{A}_i$'s have the same structure with respect to the location of their zero entries. Second, $\mathbf{A}_i$'s are primitive matrices over $\mathbb{R}$. Then, $\mathbf{A}_i$'s are non-singular matrices over $\mathbb{F}_2[\mathbf{L}]$. Moreover, the multiplication of $\mathbf{A}_i$'s is an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$. The last and the main condition is that $\mathbf{A}_i$'s can be implemented with low implementation cost by terminology of XOR counts. In fact, we try to decompose an MDS matrix into sparse matrices provided that these sparse matrices, first have the same and simple structure and then are non-singular over the ground field and finally can be implemented with the minimal XOR cost.

Outline of this paper The rest of paper is organized as follows. Definitions and notations are given in Section 2. In Section 3, it is motivated why primitive matrices are used in this paper. Definition of primitive GFS matrices is provided in Section 4. Moreover, using primitive GFS matrices a probabilistic algorithm for the construction of MDS matrices is proposed in Section 4. In Section 5, by applying binary linear functions over primitive GFS matrices, 4×4 and 8×8 MDS matrices are proposed. An extension of primitive GFS matrices, called EGFS matrices, is given in Section 6. Furthermore, the best result of this paper by applying binary linear functions over EGFS matrices are provided in Section 6. Finally, a conclusion is given in Section 7.

2 Definitions and Notations

Let $\mathbf{A}$ be an $n \times n$ matrix over a field $\mathbb{F}_q$, the finite field with q elements. $\mathbf{A}$ is called MDS over $\mathbb{F}_q$ if all square submatrix of $\mathbf{A}$ is nonsingular over $\mathbb{F}_q$ [BR99]. Moreover, a finite field with characteristic 2 is denoted with $\mathbb{F}_{2^q}$ for some q. Furthermore, we present a finite field $\mathbb{F}_{2^q}$ by hexadecimal representation. For instance, $\mathbb{F}_{2^8}/\mathtt{0x18D}$ is the finite field $\mathbb{F}_{2^8}$ which is constructed from the primitive polynomial $\mathbf{f} = x^8 + x^7 + x^3 + x^2 + 1$. For simplicity, we use non-zero positions in each row of a binary matrix as a representation of the matrix. As an example, $[[1,2,4],[1,3],[2,4],[3,4]]$ is applied for $\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$.

We denote a matrix with $(a_{i,j})$ where $a_{i,j}$ is the (i,j)th entry of the matrix. Consider an $n \times n$ matrix $\mathbf{A} = (a_{i,j})$ with $1 \le i,j \le n$ over $\mathbb{R}$, the field of real numbers. Then, $\mathbf{A}$ is called a positive matrix over $\mathbb{R}$ provided that $a_{i,j} > 0$ for all $1 \le i,j \le n$.

Definition 1 ([HJ13])**.** Consider an $n \times n$ non-negative matrix $\mathbf{A}$ over $\mathbb{R}$. The matrix $\mathbf{A}$ is called a primitive matrix over $\mathbb{R}$ if $\mathbf{A}^k$ is a positive matrix, denoted $\mathbf{A}^k > 0$, for some integer $k \ge 1$. The primitive order $\mathbf{A}$ is the minimum number k which satisfies $\mathbf{A}^k > 0$ and the matrix $\mathbf{A}$ is called an k-primitive matrix over $\mathbb{R}$. Moreover, an $n \times n$ non-negative matrix $\mathbf{A}$ is called a non-primitive matrix over $\mathbb{R}$, if there is no an integer number $k \ge 1$ such that $\mathbf{A}^k > 0$.

In this paper, the symbol $\mathbb{F}_2[\mathbf{L}]$ is considered as a set of all finite polynomials in the following form $\sum_{i=1}^{n} b_i\, \mathbf{L}^{t_i}$ where $b_i \in \mathbb{F}_2$ and t_i's are integer numbers. Consider an $n \times n$ matrix $\mathbf{A}$ over $\mathbb{F}_2[\mathbf{L}]$. Then $\mathbf{A} = (a_{i,j})$ is called a positive matrix over $\mathbb{F}_2[\mathbf{L}]$ if $a_{i,j} \neq 0$ for all $1 \le i,j \le n$. Moreover, $\mathbf{A}$ is called an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$ if determinant of all square submatrices of $\mathbf{A}$ are non-zero over $\mathbb{F}_2[\mathbf{L}]$ [WWW12].

Assume that $\mathbf{r}$ is a set over $\mathbb{F}_2[\mathbf{L}]$. Then the set of all prime factors of $\mathbf{r}$ is called the base set of $\mathbf{r}$. For instance, consider the set $\mathbf{r} = \{\mathbf{L}, \mathbf{L}^2, \mathbf{L}+1, \mathbf{L}^2+1, (\mathbf{L}^2+\mathbf{L}+1)^2, \mathbf{L}^6+\mathbf{L}^2+1\}$. Then the base set of $\mathbf{r}$ is $\widetilde{\mathbf{r}} = \{\mathbf{L}, \mathbf{L}+1, \mathbf{L}^2+\mathbf{L}+1, \mathbf{L}^3+\mathbf{L}+1\}$, since we have $\mathbf{L}^2+1 = (\mathbf{L}+1)^2$ and $\mathbf{L}^6+\mathbf{L}^2+1 = (\mathbf{L}^3+\mathbf{L}+1)^2$ over $\mathbb{F}_2[\mathbf{L}]$. Now consider an $n \times n$ non-singular matrix $\mathbf{A}$ over $\mathbb{F}_2$. If $\mathbf{A}$, $\mathbf{A}+\mathbf{I}_n$, $\mathbf{A}^2+\mathbf{A}+\mathbf{I}_n$ and $\mathbf{A}^3+\mathbf{A}+\mathbf{I}_n$ are non-singular matrices over $\mathbb{F}_2$, then we say the elements of $\widetilde{\mathbf{r}}$ are non-singular matrices over $\mathbb{F}_2$ by applying $\mathbf{A}$. Notice that if by using an $n \times n$ matrix $\mathbf{A}$ the elements of $\widetilde{\mathbf{r}}$ are non-singular matrices over $\mathbb{F}_2$, then it can be verified that the elements of $\mathbf{r}$ are non-singular matrices over $\mathbb{F}_2$ using $\mathbf{A}$.

Definition 2 ([DR13])**.** For an $\mathbb{F}_{2^m}$-linear transformation L, the branch number is defined by $\mathcal{B}_L = \min_{a \neq 0}\{wt(a) + wt(L(a))\}$ where $wt(x)$ is the number of non-zero words in x. Moreover, an $n \times n$ matrix M is called a near-MDS matrix if $\mathcal{B}_L = n$.

Example 1. Let $\mathbf{A} = [[1,2],[3],[3,4],[1]]$ is a 4×4 binary matrix. Consider $\mathbf{B} = \mathbf{A}^3$ over $\mathbb{F}_2$. It can be checked that the matrix $\mathbf{B}$ is an involutory near-MDS over $\mathbb{F}_{2^m}$ which means its branch number is 4. Moreover, it is easy to verify that the depth of the circuit of $\mathbf{B}$ is 3. Furthermore, the implementation cost of the matrix $\mathbf{B}$ is equal to $6m$ for m-bit input. The matrix $\mathbf{A}$ is a 4×4 primitive GFS matrix that will be defined in Section 4.

3 Relation between Primitive Matrices and Search Space

In this section, the role of primitive matrices for the construction of proposed MDS matrices are explained. Let $\mathbf{A}$ be an $n \times n$ non-negative matrix over $\mathbb{R}$. It is easy to see that if $\mathbf{A}$ is a non-primitive matrix over $\mathbb{R}$ then $\mathbf{A}$ can not be a primitive matrix over $\mathbb{F}_2[\mathbf{L}]$. Next, we show that if $\mathbf{A}$ is an k-primitive matrix over $\mathbb{R}$, then $\mathbf{A}$ may be a non-primitive over $\mathbb{F}_2[\mathbf{L}]$ or $\mathbf{A}$ can be an k'-primitive matrix over $\mathbb{F}_2[\mathbf{L}]$ such that $k' \geq k$. In other words, the characteristic 2 in $\mathbb{F}_2[\mathbf{L}]$, puts limitation on the order of primitive matrices.

Example 2. Consider the following three matrices

$$\mathbf{A}_1 = \begin{pmatrix} \mathbf{L} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \mathbf{L} & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} \mathbf{L} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{A}_3 = \begin{pmatrix} \mathbf{L} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ \mathbf{L} & 0 & 0 & 0 \end{pmatrix}. \qquad (1)$$

Assume that $\mathbf{L}$ is a positive integer in $\mathbb{R}$. Then it can be checked that $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ are 4-primitive matrices over $\mathbb{R}$. For instance, $\mathbf{A}_1^4$ over $\mathbb{R}$ is in the following form.

$$\mathbf{A}_1^4 = \begin{pmatrix} \mathbf{L}^4+1 & \mathbf{L}^3 & 3\,\mathbf{L}^2 & 2\,\mathbf{L} \\ 2\,\mathbf{L} & 1 & \mathbf{L}^3 & \mathbf{L}^2 \\ 3\,\mathbf{L}^2 & 2\,\mathbf{L} & \mathbf{L}^4+1 & \mathbf{L}^3 \\ \mathbf{L}^3 & \mathbf{L}^2 & 2\,\mathbf{L} & 1 \end{pmatrix}.$$

In addition, there is no integer number $1 \leq k \leq 3$ such that $\mathbf{A}_1^k$ be a positive matrix over $\mathbb{R}$. Now consider $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ over $\mathbb{F}_2[\mathbf{L}]$.

First of all, we prove $\mathbf{A}_1$ is a non-primitive matrix over $\mathbb{F}_2[\mathbf{L}]$. The characteristic polynomial of $\mathbf{A}_1$ over $\mathbb{F}_2[\mathbf{L}]$ is $x^4 + \mathbf{L}^2x^2 + 1$. Consider the equation $x^k = (x^4 + \mathbf{L}^2x^2 + 1)h(x) + r(x)$ where $r(x)$ is a polynomial of degree less than 4 over $\mathbb{F}_2[\mathbf{L}]$. Therefore, we get $\mathbf{A}_1^k = r(\mathbf{A}_1)$, since $\mathbf{A}_1$ satisfies its own characteristic equation [HJ13]. It can be verified that $r(x) = a_1 + a_2x^2$ when k is an even number and $r(x) = b_1x + b_2x^3$ when k is an odd number where a_i and b_i with $1 \leq i \leq 2$, are in $\mathbb{F}_2[\mathbf{L}]$. Hence, $\mathbf{A}_1^k$ is a linear combination of ($\mathbf{I}_4$ and $\mathbf{A}_1^2$) or ($\mathbf{A}_1$ and $\mathbf{A}_1^3$) where $\mathbf{I}_4$ is the identity matrix of order 4. Moreover, $\mathbf{A}_1$, $\mathbf{A}_1^2$ and $\mathbf{A}_1^3$ are not positive matrices over $\mathbb{F}_2[\mathbf{L}]$. Furthermore, it can be checked that ($\mathbf{I}_4$ and $\mathbf{A}_1^2$) and ($\mathbf{A}_1$ and $\mathbf{A}_1^3$) have zero entries in the same positions. Therefore, for any positive integer k, $\mathbf{A}_1^k$ has at least one zero entry which results in $\mathbf{A}_1$ is a non-primitive matrix over $\mathbb{F}_2[\mathbf{L}]$. In addition, $\mathbf{A}_2$ and $\mathbf{A}_3$ are 7-primitive and 4-primitive matrices over $\mathbb{F}_2[\mathbf{L}]$.

Definition 3. Suppose that $\mathbf{A} = (a_{i,j})$ and $\mathbf{B} = (b_{i,j})$ with $1 \leq i, j \leq n$ are two $n \times n$ sparse matrices over $\mathbb{R}$. $\mathbf{A}$ and $\mathbf{B}$ are called with the same structure provided that if $a_{i,j} = 0$ then $b_{i,j} = 0$ and vice versa.

The 4×4 sparse matrices $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$, given in Example 2, are with the same structure. In the rest, by applying Definition 3 to primitive matrices, a systematic approach for the construction of MDS matrices are proposed. Assume $\mathbf{A}_i$ with $1 \leq i \leq k$ are $n \times n$ sparse matrices over $\mathbb{R}$. Now based on the primitivity and structures of $\mathbf{A}_i$'s the following five cases are considered.

Case 1: $\mathbf{A}_i$'s are non-primitive over $\mathbb{R}$ and are with the same structure.

Case 2: $\mathbf{A}_i$'s are non-primitive over $\mathbb{R}$ and are not with the same structure.

Case 3: $\mathbf{A}_i$'s are primitive over $\mathbb{R}$ and are with the same structure.

Case 4: $\mathbf{A}_i$'s are primitive over $\mathbb{R}$ and are not with the same structure.

Case 5: Let $\mathbf{I} = \{j_1, j_2, \cdots, j_m\}$ with $1 \leq m < k$ be a subset of the set $\{1, 2, \cdots, k\}$. Consider $\mathbf{A}_{j_t}$ with $1 \leq t \leq m$ are primitive matrices over $\mathbb{R}$ and $\mathbf{A}_i$ for $i \notin \mathbf{I}$ are not primitive matrices over $\mathbb{R}$ (the structures of $\mathbf{A}_i$'s can be chosen arbitrarily).

By considering the given five cases, let $\mathbf{B}$ be the multiplication of $\mathbf{A}_i$'s, denoted with $\mathbf{B} = \prod_{i=1}^{k} \mathbf{A}_i$. Without loss of generality we may assume that all nonzero entries of $\mathbf{A}_i$'s are equal to 1. By considering the case 1, the matrix $\mathbf{B}$ is not a primitive matrix over $\mathbb{F}_2[\mathbf{L}]$, since it can be checked that the primitivity of $\mathbf{B}$ implies that $\mathbf{A}_i$'s are primitive matrices which is in contradiction to the assumptions of case 1. Moreover, $\mathbf{B}$ is possibly a primitive matrix over $\mathbb{F}_2[\mathbf{L}]$ by considering the assumptions of cases 2,3,4 and 5 and hence the matrix $\mathbf{B}$ possibly can be an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$. Therefore, to construct MDS matrices over $\mathbb{F}_2[\mathbf{L}]$, by applying search on sparse matrices, cases 2,3,4 and 5 can be used.

Performing a search using the third case has less complexity than cases 2,4 and 5, since $\mathbf{A}_i$'s are with the same structure. In fact, firstly we obtain an $n \times n$ k-primitive matrix $\mathbf{C}$ over $\mathbb{R}$. Then we choose matrices $\mathbf{A}_i$ with $1 \le i \le k'$ over $\mathbb{F}_2[\mathbf{L}]$ provided that $k' \ge k$ and $\mathbf{A}_i$'s are with the same structure as $\mathbf{C}$. Next, we check whether $\mathbf{B} = \prod_{i=1}^{k'} \mathbf{A}_i$ is an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$. Therefore, the main reason for applying the primitive matrices with the same structure, is the issue of reducing the search space.

Example 3. Consider the following 4×4 sparse matrices over $\mathbb{F}_2[\mathbf{L}]$.

$$\widetilde{\mathbf{A}}_1 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \widetilde{\mathbf{A}}_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & \mathbf{L} & 0 & 0 \end{pmatrix}, \widetilde{\mathbf{A}}_3 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \mathbf{L} \\ 0 & 0 & 0 & 1 \\ \mathbf{L} & \mathbf{L} & 0 & 0 \end{pmatrix}, \widetilde{\mathbf{A}}_4 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

The matrices $\widetilde{\mathbf{A}}_i$ with $1 \le i \le 4$ are extracted from Appendix C.2, Fig. 8 in [DL18]. It is shown in [DL18] the multiplication of $\widetilde{\mathbf{A}}_i$'s, denoted with $\widetilde{\mathbf{B}} = \prod_{i=1}^{4} \widetilde{\mathbf{A}}_i$, is an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$. Moreover, it can be checked that $\widetilde{\mathbf{B}}$ is obtained from the Case 5, since $\widetilde{\mathbf{A}}_2$, $\widetilde{\mathbf{A}}_3$ and $\widetilde{\mathbf{A}}_4$ are primitive matrices over $\mathbb{R}$ and $\widetilde{\mathbf{A}}_1$ is a non-primitive matrix over $\mathbb{R}$.

Consider the matrix $\mathbf{A}_1$ given in Example 2. It is observed that $\mathbf{A}_1$ is a 4-primitive matrix over $\mathbb{R}$. Now we select matrices $\widehat{\mathbf{A}}_i$ with $1 \le i \le 4$ provided that $\widehat{\mathbf{A}}_i$'s are with the same structure as $\mathbf{A}_1$. In addition, the multiplication of $\widehat{\mathbf{A}}_i$'s, denoted with $\widehat{\mathbf{B}} = \prod_{i=1}^{4} \widehat{\mathbf{A}}_i$, is an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$. The following matrices are derived from a simple search.

$$\widehat{\mathbf{A}}_1 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \widehat{\mathbf{A}}_2 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \mathbf{L} & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \widehat{\mathbf{A}}_3 = \begin{pmatrix} \mathbf{L} & \mathbf{L} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \mathbf{L} & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \widehat{\mathbf{A}}_4 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Furthermore, by similar structural properties of $\widetilde{\mathbf{A}}_i$ and $\widehat{\mathbf{A}}_i$ with $1 \le i \le 4$, we conclude the implementation cost of the two matrices $\widetilde{\mathbf{B}}$ and $\widehat{\mathbf{B}}$ are the same from hardware perspective.

In the next section, according to the given technique in [DL18] and applying primitive matrices, a probabilistic algorithm for the construction of MDS matrices is proposed.

4 Primitive GFS Matrices

There are two reasons why GFS is used in this paper. The first and most important reason is this fact the inverse of GFS is easy to compute as well. The second one is that combining primitive matrices and GFS reduces search space. In this section, by applying GFS, a type of primitive sparse matrix is proposed which is called primitive GFS matrix. In Section 5 using primitive GFS matrices, 4×4 and 8×8 MDS matrices are proposed such that the implementation cost of these matrices and their inverse are 68 and 264 XOR for 8-bit input, respectively. Consider the following 2×2 block-matrices

$$\mathbf{c}_1^{(m)} = \begin{pmatrix} \mathbf{L} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{c}_2^{(m)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} \end{pmatrix}, \quad \mathbf{z}^{(m)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{2}$$

where $\mathbf{L}$ is an $m \times m$ non-singular matrix over $\mathbb{F}_2$. In addition, $\mathbf{1}$ and $\mathbf{0}$ are $m \times m$ identity and zero matrices, respectively. In Definition (4), by applying $\mathbf{c}_1^{(m)}$, $\mathbf{c}_2^{(m)}$ and $\mathbf{z}^{(m)}$ we propose primitive GFS matrices.

Definition 4 (Primitive GFS Matrices)**.** Suppose that $\mathbf{p}_1 = \{a_1, a_2, \cdots, a_n\}$ and $\mathbf{p}_2 = \{b_1, b_2, \cdots, b_n\}$ are two permutations of integer numbers from 1 to n such that $a_i \neq b_i$ for all $1 \leq i \leq n$. Consider 2×2 block-matrices $\mathbf{c}_1^{(m)}$, $\mathbf{c}_2^{(m)}$ and $\mathbf{z}^{(m)}$ which are given in (2). Suppose that the ith row of an $n \times n$ block-matrix $\mathbf{S} = (s_{i,j})$ with $1 \leq i, j \leq n$, based on the two permutations $\mathbf{p}_1$ and $\mathbf{p}_2$, is filled in the following form:

$$s_{i,j} = \begin{cases} \mathbf{c}_1^{(m)} & j = a_i, \\ \mathbf{c}_2^{(m)} & j = b_i, \\ \mathbf{z}^{(m)} & j \notin \{a_i, b_i\}. \end{cases}$$

The block-matrix $\mathbf{S}$ is called primitive GFS matrix if $\mathbf{S}$ is a primitive matrix over $\mathbb{R}$.

First of all, notice that in Definition 4 to check whether $\mathbf{S}$ is a primitive matrix over $\mathbb{R}$, we assume that $\mathbf{c}_1^{(m)}$, $\mathbf{c}_2^{(m)}$ and $\mathbf{z}^{(m)}$ are 2×2 matrices over $\mathbb{R}$ and $\mathbf{L}$ is a positive integer. Moreover, for simplicity the block-matrix $\mathbf{S}$ is denoted with $\mathbf{S}(n, \mathbf{L}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2])$. Furthermore, it follows from Definition 4 that a primitive GFS matrix can be a non-primitive matrix over $\mathbb{F}_2[\mathbf{L}]$. In addition, primitive GFS matrices are non-singular over $\mathbb{F}_2[\mathbf{L}]$, since it can be proved that $\det(\mathbf{S}) = 1$ over $\mathbb{F}_2[\mathbf{L}]$. In Example 4, it is observed that why the block-matrix $\mathbf{S}$ in Definition 4 should be a primitive matrix over $\mathbb{R}$ and not ask for $\mathbf{S}$ to be primitive over $\mathbb{F}_2[\mathbf{L}]$.

Example 4. Suppose that in Definition 4, we assumed $\mathbf{S}$ be a primitive matrix over $\mathbb{F}_2[\mathbf{L}]$. Consider the following $n \times n$ block-matrix that is constructed from two permutations $\mathbf{p}_1 = \{1, 2, \cdots, n\}$ and $\mathbf{p}_2 = \{2, 3, \cdots, n, 1\}$.

$$\mathbf{S}(n, \mathbf{1}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]) = \left(\begin{array}{cc|cc|cc|cc} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right).$$

For simplicity in representation set $\mathbf{A} = \mathbf{S}(n, \mathbf{1}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2])$. Next, we prove $\mathbf{A}$ is a non-primitive matrix over $\mathbb{F}_2[\mathbf{L}]$. Suppose $\mathbf{A}$ is a primitive matrix over $\mathbb{F}_2[\mathbf{L}]$ which implies that there is a positive integer k such that $\mathbf{A}^k > 0$ over $\mathbb{F}_2[\mathbf{L}]$. Hence, all entries of $\mathbf{A}^k$ are equal to 1. But it is in contradiction to this fact that $\mathbf{A}$ is a non-singular matrix over $\mathbb{F}_2[\mathbf{L}]$ but $\mathbf{A}^k$ is a singular matrix over $\mathbb{F}_2[\mathbf{L}]$, since $\mathbf{A}^k$ has two equal rows. Therefore, in order to use sparse block-matrices such as $\mathbf{A}$, the block-matrix $\mathbf{S}$ in Definition 4 should be asked to be a primitive matrix over $\mathbb{R}$.

It follows from Appendix A that the order of a primitive GFS matrix depends on the two permutations $\mathbf{p}_1$ and $\mathbf{p}_2$. Moreover, the implementation cost of proposed MDS matrices in Section 5, is directly related to the number of iterations required to reach full diffusion. Therefore, for the construction of MDS matrices, GFS matrices with the minimum primitive-order should be used. In Example 5, some types of primitive GFS matrices are made, which are used in Section 5 to construct lightweight 4×4 and 8×8 MDS matrices with the implementation cost 68 and 264 XOR for 8-bit input , respectively.

Example 5. For $n = 2$, the following primitive GFS matrix is used.

$$\mathbf{p}_1 = \{1, 2\}, \quad \mathbf{p}_2 = \{2, 1\}, \quad \mathbf{S}(2, \mathbf{L}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]) = \left(\begin{array}{cc|cc} \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right). \tag{3}$$

In Example 2, it is observed that $\mathbf{S}(2, \mathbf{L}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2])$ is a 4-primitive matrix over $\mathbb{R}$. For $n = 3$, there are 12 permutations such as $\mathbf{p}_1 = \{a_1, a_2, a_3\}$ and $\mathbf{p}_2 = \{b_1, b_2, b_3\}$ from 1 to

3 such that $a_i \neq b_i$ for all $1 \leq i \leq 3$. Consider the following case

$$\mathbf{p}_1 = \{1,3,2\}, \quad \mathbf{p}_2 = \{2,1,3\}, \quad \mathbf{S}(3, \mathbf{L}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]) = \left(\begin{array}{cc|cc|cc} \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{array}\right) \tag{4}$$

The primitive GFS matrix $\mathbf{S}(3, \mathbf{L}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2])$, is an 5-primitive matrix over $\mathbb{R}$. For $n = 4$, there are 216 permutations $\mathbf{p}_1 = \{a_1, a_2, a_3, a_4\}$ and $\mathbf{p}_2 = \{b_1, b_2, b_3, b_4\}$ from 1 to 4 such that $a_i \neq b_i$ for $1 \leq i \leq 4$. These 216 permutations are divided into four cases, which are listed in Appendix A. It follows from Appendix A that the minimum primitive-order of an GFS matrix such as $\mathbf{S}(4, \mathbf{L}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2])$ is 6. Therefore, we select two permutations $\mathbf{p}_1$ and $\mathbf{p}_2$ provided that $\mathbf{S}(4, \mathbf{L}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2])$ is a 6-primitive matrix over $\mathbb{R}$.

$$\mathbf{p}_1 = \{4,3,2,1\}, \quad \mathbf{p}_2 = \{3,2,1,4\}, \quad \mathbf{S}(4, \mathbf{L}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]) = \left(\begin{array}{cc|cc|cc|cc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{array}\right), \tag{5}$$

$$\widehat{\mathbf{p}_1} = \{1,3,2,4\}, \quad \widehat{\mathbf{p}_2} = \{2,1,4,3\}, \quad \mathbf{S}(4, \mathbf{L}_{(m)}, [\widehat{\mathbf{p}_1}, \widehat{\mathbf{p}_2}]) = \left(\begin{array}{cc|cc|cc|cc} \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right). \tag{6}$$

Two primitive GFS matrices such as $\mathbf{A}_1$ and $\mathbf{A}_2$ are considered with the same structure if $\mathbf{A}_1$ and $\mathbf{A}_2$ are constructed from the same permutations. In the rest, we propose a probabilistic algorithm for the construction of MDS matrices.

Algorithm 1: Construction of Lightweight MDS Matrices based on the Primitive GFS Matrices with the same Structure

```
   Input  : Three positive integer $n$, $m$ and $r$.
   Output : An $2n \times 2n$ lightweight MDS matrix over $m$-bit input with $\leq r$ XOR.
1  Select two permutations $\mathbf{p}_1$ and $\mathbf{p}_2$ such that the order of primitive GFS matrix
     $\mathbf{C} = \mathbf{S}(n, \mathbf{1}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2])$ is minimal over $\mathbb{R}$.
2  Let $\mathbf{C}$ be an $k$-primitive matrix over $\mathbb{R}$ and set $u = k$.
3  Select primitive GFS matrices $\mathbf{A}_i = \mathbf{S}(n, \mathbf{f}^{(i)}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2])$ with $1 \leq i \leq u$ such that
     $\mathbf{f}^{(i)} \in \{\mathbf{L}^{-2}, \mathbf{L}^{-1}, \mathbf{1}, \mathbf{L}, \mathbf{L}^2\}$ and $\mathbf{A}_i$'s are constructed from minimal numbers of $\mathbf{L}$.
4  Construct $\mathbf{B} = \prod_{i=1}^{u} \mathbf{A}_i$ over $\mathbb{F}_2[\mathbf{L}]$.
5  *If* $\mathbf{B}$ is an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$ *then* go in Step 8 *end*.
6  *If* all cases in Step 3 are considered *then* set $u = u + 1$.
7  *If* $\mathbf{B}$ is not an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$ *then* go in Step 3 *end*.
8  Get the base set of subdeterminants of $\mathbf{B}$ over $\mathbb{F}_2[\mathbf{L}]$.
9  Obtain an $m \times m$ non-singular binary matrix $\mathbf{L}$ with the minimal implementation
     cost provided that elements of the base set are non-singular matrices over $\mathbb{F}_2$ by $\mathbf{L}$.
10 *If* Step 9 fails to obtain a binary matrix $\mathbf{L}$ *then* go in Step 3 *end*.
11 Obtain the implementation cost of $\mathbf{A}_i$'s, denoted $x$, with respect to the cost of $\mathbf{L}$.
12 *If* $x \leq r$ *then* go in Step 14 *end*.
13 *If* $u \leq 2n$ *then* go in Step 3 *else* go in Step 1 *end*.
14 Return $\mathbf{B}$ , $\mathbf{A}_i$'s and the non-singular binary matrix $\mathbf{L}$.
```

Notice that, Algorithm 1 is a randomized algorithm, since with the proposed approach, there is an extensive search space to construct MDS matrices for the large dimensions.

Example 6. The best implementation of the lightweight 6×6 MDS matrices for 8-bit input is 186 XOR [TTK+18]. Set $n = 3$, $m = 8$ and $r = 186$ as input into Algorithm 1. It follows from (4) that $\mathbf{C} = \mathbf{S}(3, \mathbf{1}_{(8)}, [\mathbf{p}_1, \mathbf{p}_2])$ is an 5-primitive matrices over $\mathbb{R}$ for the two permutations $\mathbf{p}_1 = \{1, 3, 2\}$ and $\mathbf{p}_2 = \{2, 1, 3\}$. Using exhaustive search, it can be verified that there are no primitive GFS matrices $\mathbf{A}_i = \mathbf{S}(3, \mathbf{f}^{(i)}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2])$ with $1 \leq i \leq 5$ such that $\prod_{i=1}^{5} \mathbf{A}_i$ is an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$ provided that $\mathbf{f}^{(i)} \in \{\mathbf{L}^{-2}, \mathbf{L}^{-1}, \mathbf{1}, \mathbf{L}, \mathbf{L}^2\}$. Now consider the following primitive GFS matrices $\mathbf{A}_i$ with $1 \leq i \leq 6$.

$$\begin{array}{lll}
\mathbf{A}_1 = \mathbf{R}(3, \mathbf{1}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]), & \mathbf{A}_2 = \mathbf{R}(3, \mathbf{L}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]), & \mathbf{A}_3 = \mathbf{R}(3, \mathbf{1}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]), \\
\mathbf{A}_4 = \mathbf{R}(3, \mathbf{1}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]), & \mathbf{A}_5 = \mathbf{R}(3, \mathbf{L}^2_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]), & \mathbf{A}_6 = \mathbf{R}(3, \mathbf{1}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]).
\end{array}$$

It can be checked that $\mathbf{B} = \prod_{i=1}^{6} \mathbf{A}_i$ is an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$. Moreover, applying exhaustive search, matrix $\mathbf{B}$ is an optimal result with respect to the number of binary linear functions which are used in the construction of $\mathbf{B}$ (L,L,L,L^2,L^2 and L^2). It is worth mentioning that L^2 is considered two binary linear functions.

$$\mathbf{B} = \begin{pmatrix}
\mathbf{L}^3+\mathbf{1} & \mathbf{L}^3+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^2+\mathbf{1} & \mathbf{1} & \mathbf{L} \\
\mathbf{L}^3+\mathbf{L}+\mathbf{1} & \mathbf{L}^3+\mathbf{1} & \mathbf{L}^2 & \mathbf{L}^2+\mathbf{L} & \mathbf{L}^3+\mathbf{1} & \mathbf{L}^3 \\
\mathbf{1} & \mathbf{L} & \mathbf{L}^3+\mathbf{1} & \mathbf{L}^3+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^2+\mathbf{1} \\
\mathbf{L}^3+\mathbf{1} & \mathbf{L}^3 & \mathbf{L}^3+\mathbf{L}+\mathbf{1} & \mathbf{L}^3+\mathbf{1} & \mathbf{L}^2 & \mathbf{L}^2+\mathbf{L} \\
\mathbf{L}^3+\mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^2+\mathbf{1} & \mathbf{1} & \mathbf{L} & \mathbf{L}^3+\mathbf{1} & \mathbf{L}^3+\mathbf{1} \\
\mathbf{L}^2 & \mathbf{L}^2+\mathbf{L} & \mathbf{L}^3+\mathbf{1} & \mathbf{L}^3 & \mathbf{L}^3+\mathbf{L}+\mathbf{1} & \mathbf{L}^3+\mathbf{1}
\end{pmatrix}.$$

It is easy to verify that the number of subdeterminants of an $n \times n$ matrix is $\sum_{i=1}^{n} \binom{n}{i}\binom{n}{i} = \binom{2n}{n} - 1$. Therefore, $\mathbf{B}$ has $\binom{12}{6} - 1 = 923$ subdeterminants. The base set of subdeterminants of $\mathbf{B}$ is given in (7).

$$\begin{gathered}
\{L, L+1, L^2+L+1, L^3+L+1, L^3+L^2+1, L^4+L+1, L^4+L^3+1, L^4+L^3+L^2+L+1, \\
L^5+L^2+1, L^5+L^3+1, L^5+L^3+L^2+L+1, L^5+L^4+L^2+L+1, L^5+L^4+L^3+L+1, \\
L^5+L^4+L^3+L^2+1, L^6+L+1, L^6+L^3+1, L^6+L^4+L^2+L+1, L^6+L^4+L^3+L+1, \\
L^6+L^5+1, L^6+L^5+L^2+L+1, L^6+L^5+L^3+L^2+1, L^6+L^5+L^4+L+1, \\
L^6+L^5+L^4+L^2+1, L^7+L^3+L^2+L+1, L^7+L^5+L^3+L+1, L^7+L^6+1, \\
L^7+L^6+L^4+L^2+1, L^7+L^6+L^5+L^4+1, L^8+L^6+L^5+L^3+1, \\
L^8+L^7+L^3+L+1, L^8+L^7+L^6+L^3+L^2+L+1\}.
\end{gathered} \tag{7}$$

Assume the XOR cost of a binary linear function ($\mathbf{L}$) is denoted with the symbol $\#\mathbf{L}$. Therefore, the implementation cost of $\mathbf{B}$ for m-bit input is equal to:

$$\overbrace{(3m)}^{\#\mathbf{B}_1} + \overbrace{(3m + 3(\#\mathbf{L})}^{\#\mathbf{B}_2} + \overbrace{(3m)}^{\#\mathbf{B}_3} + \overbrace{(3m)}^{\#\mathbf{B}_4} + \overbrace{(3m + 3(\#\mathbf{L}^2)}^{\#\mathbf{B}_5} + \overbrace{(3m)}^{\#\mathbf{B}_6} = 18m + 3(\#\mathbf{L}) + 3(\#\mathbf{L}^2). \tag{8}$$

Next, we get 8×8 non-singular matrices $\mathbf{L}$ over $\mathbb{F}_2$ such that the given elements in (7) are non-singular matrices over $\mathbb{F}_2$ by applying $\mathbf{L}$. For instance, the following non-singular binary matrices $\mathbf{L}_j$ with $1 \leq j \leq 4$, are obtained.

$$\begin{array}{ll}
\mathbf{L}_1 = [[1,8],[1],[2,7],[3],[4],[5],[6],[7]], & \mathbf{L}_2 = [[1,8],[1],[2],[3],[4,7],[5],[6],[7]], \\
\mathbf{L}_3 = [[3,8],[1],[2],[3],[4],[5,6],[6],[7]], & \mathbf{L}_4 = [[5,8],[1],[2],[3],[4],[5],[6,7],[7]].
\end{array} \tag{9}$$

The implementation cost of $\mathbf{L}_j$'s, given in (9), is two XOR. In addition $\mathbf{L}_j^2$ with $1 \leq j \leq 4$, can be implemented with four XOR. Therefore, it follows from (8) that the implementation cost of $\mathbf{B}$ is $18 \times 8 + 3 \times 2 + 3 \times 4 = 162$ XOR. Moreover, $\mathbf{B}^{-1}$ is implemented with 162 XOR, since the implementation cost of $\mathbf{B}^{-1} = \prod_{i=1}^{6} \mathbf{A}_{7-i}^{-1}$ is equal to (8).

Proposition 1 follows a condition which can be applied to reduce the complexity of the step 9 given in Algorithm 1 when we run Algorithm 1 to construct MDS matrices with the large dimensions.

Proposition 1. Let $\mathbf{r} = \{f_1, f_2, \cdots, f_k\}$ be a set of pairwise distinct irreducible polynomials over $\mathbb{F}_2$ for some positive integer k. Suppose that $\mathbf{A}$ is an $n \times n$ binary matrix such that its minimal polynomial is g. If prime factors of g over $\mathbb{F}_2$ are not elements of the set $\mathbf{r}$, then matrices $f_i(\mathbf{A})$ with $1 \leq i \leq k$ are non-singular matrices over $\mathbb{F}_2$.

Proof. First, we prove the matrices $f_i(\mathbf{A})$ with $1 \leq i \leq k$ are nonzero matrices over $\mathbb{F}_2$. It follows from assumptions that g is minimal polynomial of $\mathbf{A}$. Suppose that $f_i(\mathbf{A})$ is equal to zero matrix for some $1 \leq i \leq k$. Then we get a contradiction to our assumptions, since f_i's are irreducible polynomials over $\mathbb{F}_2$ such that f_i's are not prime factors of g over $\mathbb{F}_2$. Now assume that $f_i(\mathbf{A})$ is equal to a singular matrix for some $1 \leq i \leq k$ which implies that the determinant of $f_i(\mathbf{A})$ is equal to zero. From linear algebra [HJ13] we know that $\det(f_i(\mathbf{A})) = \prod_{j=1}^{n} f_i(\lambda_j)$ where λ_j's are eigenvalues of $\mathbf{A}$. Therefore, there is at least an integer $1 \leq j \leq n$ such that $f_i(\lambda_j) = 0$ and this is in contradiction to this fact that two distinct irreducible polynomials have no common roots. □

It follows from Proposition 1 that in step 9, given in Algorithm 1, it is sufficient to obtain an $m \times m$ non-singular binary matrix such that the prime factors of its minimal polynomial are not elements of the base set of the matrix $\mathbf{B}$ [BKL16, Kol19].

5 Construction of MDS Matrices by Applying Primitive GFS Matrices

In this section, using primitive GFS matrices, we propose 4×4 and 8×8 MDS matrices for 8 bit input. The proposed 4×4 and 8×8 MDS matrices are implemented with 68 and 264 XOR. Moreover, the given results in this section are optimal according to the terminology of XOR cost and the number of binary linear functions. In other words, it is not possible to achieve better results by applying the proposed approach in Section 4. In fact, in order to obtain the results of this section, an exhaustive search has been applied to Algorithm 1. Although the given results in this section are not better than the presented results in Section 6, we intend to show the importance of Algorithm 1 for the construction of lightweight MDS matrices with the large dimensions.

5.1 Construction of 4×4 MDS Matrices

Consider the primitive GFS matrix $\mathbf{C} = \mathbf{R}(2, \mathbf{1}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2])$ over two permutations $\mathbf{p}_1 = \{1, 2\}$ and $\mathbf{p}_2 = \{2, 1\}$. It follows from (3) that $\mathbf{C}$ is a 4-primitive matrix over $\mathbb{R}$. In the rest, based on the structure of matrix $\mathbf{C}$, the following primitive GFS matrices are applied to construct a lightweight 4×4 MDS matrix for m-bit input.

$$\mathbf{A}_1 = \mathbf{R}(2, \mathbf{1}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]), \quad \mathbf{A}_2 = \mathbf{R}(2, \mathbf{L}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]),$$
$$\mathbf{A}_3 = \mathbf{R}(2, \mathbf{L}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]), \quad \mathbf{A}_4 = \mathbf{R}(2, \mathbf{1}_{(m)}, [\mathbf{p}_1, \mathbf{p}_2]).$$

It can be checked that $\mathbf{B} = \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4$, given in (10), is an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$.

$$\begin{aligned}\mathbf{B} &= \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{L}^2+\mathbf{1} & \mathbf{L}^2 & \mathbf{1} & \mathbf{L}+\mathbf{1} \\ \mathbf{L}+\mathbf{1} & \mathbf{1} & \mathbf{L}^2 & \mathbf{L}^2 \\ \mathbf{1} & \mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{1} & \mathbf{L}^2 \\ \mathbf{L}^2 & \mathbf{L}^2 & \mathbf{L}+\mathbf{1} & \mathbf{1} \end{pmatrix}.\end{aligned} \tag{10}$$

Then the implementation cost of $\mathbf{B}$, for m-bit input, is equal to

$$\overbrace{(2m)}^{\#\mathbf{A}_1} + \overbrace{(2m + 2(\#\mathbf{L}))}^{\#\mathbf{A}_2} + \overbrace{(2m + 2(\#\mathbf{L}))}^{\#\mathbf{A}_3} + \overbrace{(2m)}^{\#\mathbf{A}_4} = 8m + 4(\#\mathbf{L}). \tag{11}$$

There are $\binom{8}{4} - 1 = 69$ subdeterminants in $\mathbf{B}$. The base set of these subdeterminants is:

$$\{\mathbf{L}, \mathbf{L}+\mathbf{1}, \mathbf{L}^2+\mathbf{L}+\mathbf{1}, \mathbf{L}^3+\mathbf{L}+\mathbf{1}, \mathbf{L}^3+\mathbf{L}^2+\mathbf{1}, \mathbf{L}^4+\mathbf{L}+\mathbf{1}\} \tag{12}$$

For $m = 4$, consider the following 4×4 non-singular matrices over $\mathbb{F}_2$. It can be verified that using $\mathbf{L}_i$ with $1 \le i \le 4$, the given elements in (12) are non-singular matrices over $\mathbb{F}_2$. Moreover, the implementation cost of $\mathbf{L}_i$'s is one XOR.

$$\mathbf{L}_1 = [[1,4],[1],[2],[3]], \quad \mathbf{L}_2 = [[4],[1,2],[2],[3]], \quad \mathbf{L}_3 = [[4],[1],[2,3],[3]], \quad \mathbf{L}_4 = [[4],[1],[2],[3,4]].$$

Hence, by applying $\mathbf{L}_i$'s and relation (11), $\mathbf{B}$ is implemented with $8 \times 4 + 4 \times 1 = 36$ XOR for 4-bit input. Moreover, $\mathbf{L}^4 + \mathbf{L}^3 + 1$ is not an element of (12). Therefore, $\mathbf{B}$ is implemented with 36 XOR over $\mathbb{F}_{2^4}$/`0x19`. Furthermore, using $\mathbf{L}_i$ with $1 \le i \le 4$, the implementation cost of $\mathbf{B}^{-1}$ is 36 XOR, since we have

$$\mathbf{B}^{-1} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} & \mathbf{1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} & \mathbf{1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \end{pmatrix}. \tag{13}$$

For $m = 8$, consider 8×8 non-singular matrices $\mathbf{L}_j$ with $1 \le j \le 4$ in (14). It can be verified that by applying $\mathbf{L}_j$'s the given elements in (12) are non-singular matrices over $\mathbb{F}_2$.

$$\begin{aligned} &\mathbf{L}_1 = [[2,8],[1],[2],[3],[4],[5],[6],[7]], \quad \mathbf{L}_2 = [[8],[1,3],[2],[3],[4],[5],[6],[7]], \\ &\mathbf{L}_3 = [[8],[1],[2,4],[3],[4],[5],[6],[7]], \quad \mathbf{L}_4 = [[8],[1],[2],[3,5],[4],[5],[6],[7]]. \end{aligned} \tag{14}$$

The implementation cost of $\mathbf{L}_j$'s is one XOR. Therefore, using $\mathbf{L}_j$'s and relation (11), $\mathbf{B}$ is implemented with $8 \times 8 + 4 \times 1 = 68$ XOR for 8-bit input. Moreover, it follows from (13) that the implementation cost of $\mathbf{B}^{-1}$ is 68 XOR for 8-bit input. Furthermore, $\mathbf{B}$ can be implemented with at least $8 \times 8 + 4 \times 2 = 72$ XOR over $\mathbb{F}_{2^8}$, since some roots of irreducible polynomials of degree 8 require only two XOR [BKL16].

5.2 Construction of 8×8 MDS Matrices

First of all, we select two permutations $\widehat{\mathbf{p}}_1 = \{1,3,2,4\}$ and $\widehat{\mathbf{p}}_2 = \{2,1,4,3\}$. Then we construct the primitive GFS matrix $\mathbf{C} = \mathbf{R}(4, \mathbf{1}_{(m)}, [\widehat{\mathbf{p}}_1, \widehat{\mathbf{p}}_2])$. It follows from (6) that $\mathbf{C}$ is a 6-primitive GFS matrix over $\mathbb{R}$. Moreover, it follows from exhaustive search that there are no primitive GFS matrices $\mathbf{A}_i = \mathbf{S}(4, \mathbf{f}^{(i)}_{(m)}, [\widehat{\mathbf{p}}_1, \widehat{\mathbf{p}}_2])$ with $1 \le i \le 6$ such that $\prod_{i=1}^{6} \mathbf{A}_i$ is an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$ provided that $\mathbf{f}^{(i)} \in \{\mathbf{L}^{-2}, \mathbf{L}^{-1}, \mathbf{1}, \mathbf{L}, \mathbf{L}^2\}$. Now consider the following primitive GFS matrices $\mathbf{A}_i$ with $1 \le i \le 7$.

$$\begin{aligned} &\mathbf{A}_1 = \mathbf{R}(4, \mathbf{L}_{(m)}, [\widehat{\mathbf{p}}_1, \widehat{\mathbf{p}}_2]), \quad \mathbf{A}_2 = \mathbf{R}(4, \mathbf{1}_m, [\widehat{\mathbf{p}}_1, \widehat{\mathbf{p}}_2]), \quad \mathbf{A}_3 = \mathbf{R}(4, \mathbf{1}_m, [\widehat{\mathbf{p}}_1, \widehat{\mathbf{p}}_2]), \quad \mathbf{A}_4 = \mathbf{R}(4, \mathbf{L}^2_{(m)}, [\widehat{\mathbf{p}}_1, \widehat{\mathbf{p}}_2]), \\ &\mathbf{A}_5 = \mathbf{R}(4, \mathbf{L}^{-1}_{(m)}, [\widehat{\mathbf{p}}_1, \widehat{\mathbf{p}}_2]), \quad \mathbf{A}_6 = \mathbf{R}(4, \mathbf{L}^{-1}_{(m)}, [\widehat{\mathbf{p}}_1, \widehat{\mathbf{p}}_2]), \quad \mathbf{A}_7 = \mathbf{R}(4, \mathbf{1}_m, [\widehat{\mathbf{p}}_1, \widehat{\mathbf{p}}_2]). \end{aligned}$$

It can be verified that $\mathbf{B} = \prod_{i=1}^{7} \mathbf{A}_i$ is an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$. Moreover, using exhaustive search, $\mathbf{B}$ is one of the optimal results in relation to the number of binary linear functions. Actually, it is not possible to construct an MDS matrix $\mathbf{B}$ by applying primitive GFS matrices $\mathbf{A}_i = \mathbf{S}(4, \mathbf{f}^{(i)}_{(m)}, [\widehat{\mathbf{p}}_1, \widehat{\mathbf{p}}_2])$ with $1 \le i \le 7$ provided that $\mathbf{A}_i$'s are constructed from less than twenty binary linear functions (L,L,L,L,L^{-1},L^{-1},L^{-1},L^{-1}, L^{-1},L^{-1},L^{-1},L^{-1},L^2,L^2,L^2, and L^2). Now, it follows from $\mathbf{B} = \prod_{i=1}^{7} \mathbf{A}_i$ that the implementation cost of $\mathbf{B}$ for m-bit input is equal to

$$\begin{aligned} &\overbrace{(4m + 4(\#\mathbf{L}))}^{\#\mathbf{A}_1} + \overbrace{(4m)}^{\#\mathbf{A}_2} + \overbrace{(4m)}^{\#\mathbf{A}_3} + \overbrace{(4m + 4(\#\mathbf{L}^2))}^{\#\mathbf{A}_4} + \overbrace{(4m + 4(\#\mathbf{L}^{-1}))}^{\#\mathbf{A}_5} + \overbrace{(4m + 4(\#\mathbf{L}^{-1}))}^{\#\mathbf{A}_6} + \overbrace{(4m)}^{\#\mathbf{A}_7} \\ &= 28m + 4(\#\mathbf{L}) + 4(\#\mathbf{L}^2) + 8(\#\mathbf{L}^{-1}) \end{aligned} \tag{15}$$

The base set of subdeterminants of $\mathbf{B}$ has 380 elements. In this base set, there are all irreducible polynomials of degree 4 and 8, except for the irreducible polynomial `0x1A3`. It can be checked that the elements of the base set of $\mathbf{B}$ are non-singular matrices over $\mathbb{F}_2$ by applying the following non-singular 8×8 binary matrices $\mathbf{L}_i$ with $1 \leq i \leq 4$.

$$\begin{array}{ll} \mathbf{L}_1 = [[3],[7],[2],[5],[1,8],[4],[6],[4,8]], & \mathbf{L}_2 = [[6],[3],[1,4],[2,4],[7],[5],[8],[2]], \\ \mathbf{L}_3 = [[7],[1,5],[2],[8],[3,5],[4],[6],[3]], & \mathbf{L}_4 = [[1,5],[7],[2],[8],[6],[1,4],[5],[3]]. \end{array} \tag{16}$$

Notice the implementation cost of $\mathbf{L}_i$'s is two XOR. In addition, $\mathbf{L}_i^{-1}$ and $\mathbf{L}_i^2$ with $1 \leq i \leq 4$ can be implemented with two and four XOR, respectively. Therefore, by applying $\mathbf{L}_i$'s and relation (15), $\mathbf{B}$ is implemented with $28 \times 8 + 4 \times 2 + 4 \times 4 + 8 \times 2 = 264$ XOR for 8-bit input. Moreover, it can be verified that the implementation cost of $\mathbf{B}^{-1}$ is 264 XOR for 8-bit input. Furthermore, it follows from Proposition 1 that the matrix $\mathbf{B}$ is implemented with 264 XOR over $\mathbb{F}_{2^8}/$`0x1A3`, since the minimal polynomial of $\mathbf{L}_i$'s over $\mathbb{F}_2$ is `0x1A3`.

6 Construction of MDS Matrices by Applying Extended Primitive GFS Matrices

In this section, using an extension of Definition 4, we define some types of sparse matrices called EGFS matrices. Then, by applying EGFS matrices we propose 4×4, 6×6 and 8×8 lightweight MDS matrices with the implementation cost 67, 156 and 260 XOR for 8-bit input, respectively. Moreover, we propose an 6×6 MDS matrix for 4-bit input such that its implementation cost is 90 XOR. The proposed MDS matrices are not only suitable by the terminology of implementation cost, but also are efficient with respect to the number of binary linear functions which are used in the construction of the presented matrices. Although, we could not obtain 8×8 MDS matrices from EGFS matrices for 4-bit input, we propose 8×8 near-MDS matrices such that the depth and the implementation cost of the proposed matrices are low and may be used in lightweight cryptography.

The proposed 4×4 MDS matrix is obtained by exhaustive search. But a random search is applied to achieve the proposed 6×6 and 8×8 MDS matrices. Actually, there is an extensive search space to obtain lightweight 6×6 and 8×8 MDS matrices using EGFS matrices. Therefore, by performing a full search on EGFS matrices, better implementation results may be achieved to construct 6×6 and 8×8 lightweight MDS matrices.

Assume that $\mathbf{1}$ and $\mathbf{0}$ are $m \times m$ identity and zero matrices over $\mathbb{F}_2$, respectively. Consider $\mathbf{L}_j$ with $1 \leq j \leq 3n$ such that $\mathbf{L}_j$'s are $m \times m$ non-singular matrices over $\mathbb{F}_2$. In Definition 5, the following 2×2 block-matrices with $1 \leq i \leq n$ are used.

$$\mathbf{c}_i^{(1,m)} = \begin{pmatrix} \mathbf{L}_{3i-2} & \mathbf{L}_{3i-1} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{c}_i^{(2,m)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{3i} & \mathbf{0} \end{pmatrix}, \quad \mathbf{z}^{(m)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{17}$$

Definition 5 (Extended Primitive GFS Matrices)**.** Suppose that $\mathbf{p}_1 = \{a_1, a_2, \cdots, a_n\}$ and $\mathbf{p}_2 = \{b_1, b_2, \cdots, b_n\}$ are two permutations from 1 to n such that $a_i \neq b_i$ for $1 \leq i \leq n$. Consider 2×2 block-matrices $\mathbf{c}_i^{(1,m)}$, $\mathbf{c}_i^{(2,m)}$ and $\mathbf{z}^{(m)}$ with $1 \leq i \leq n$ that are given in (17). Assume that using the two permutations $\mathbf{p}_1$ and $\mathbf{p}_2$, the ith row of an $n \times n$ block-matrix $\mathbf{E} = (e_{i,j})$ with $1 \leq i, j \leq n$, is filled in the following form.

$$e_{i,j} = \begin{cases} \mathbf{c}_i^{(1,m)} & j = a_i, \\ \mathbf{c}_i^{(2,m)} & j = b_i, \\ \mathbf{z}^{(m)} & j \notin \{a_i, b_i\}. \end{cases}$$

The block-matrix $\mathbf{E}$ is called an extended primitive GFS matrix, denoted with EGFS matrix, if $\mathbf{E}$ be a primitive matrix over $\mathbb{R}$.

In Definition 5, it can be verified that $\det(\mathbf{E}) = \prod_{i=1}^{n} \mathbf{L}_{3i-1}\mathbf{L}_{3i}$ over $\mathbb{F}_2[\mathbf{L}]$. Therefore, $\det(\mathbf{E}) \neq 0$ over $\mathbb{F}_2[\mathbf{L}]$, since it is assumed that $\mathbf{L}_j$ with $1 \leq j \leq 3n$ are non-singular matrices over $\mathbb{F}_2$. In other words, EGFS matrices are non-singular matrices over $\mathbb{F}_2[\mathbf{L}]$.

Example 7. For $n = 2$, consider the two permutations $\mathbf{p}_1 = \{1, 2\}$ and $\mathbf{p}_2 = \{2, 1\}$. Then the following two EGFS matrices $\mathbf{E}_1$ and $\mathbf{E}_2$ are 4-primitive matrices over $\mathbb{R}$.

$$\mathbf{c}_1^{(1,m)} = \begin{pmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \mathbf{c}_2^{(1,m)} = \begin{pmatrix} \mathbf{L} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \qquad \mathbf{c}_1^{(1,m)} = \begin{pmatrix} \mathbf{L} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \mathbf{c}_2^{(1,m)} = \begin{pmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

$$\mathbf{c}_1^{(2,m)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} \end{pmatrix}, \mathbf{c}_2^{(2,m)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{L}^{-1} & \mathbf{0} \end{pmatrix}, \qquad \mathbf{c}_1^{(2,m)} = \mathbf{c}_2^{(2,m)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} \end{pmatrix},$$

$$\mathbf{E}_1 = \left(\begin{array}{cc|cc} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} \\ \mathbf{L}^{-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right). \qquad \mathbf{E}_2 = \left(\begin{array}{cc|cc} \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right).$$

6.1 Construction of 4×4 MDS Matrices from EGFS

In this subsection, we propose four EGFS matrices $\mathbf{E}_i$ with $1 \le i \le 4$ such that $\mathbf{E}_i$'s are 4-primitive matrices over $\mathbb{R}$. Moreover, the proposed EGFS matrices are constructed of the two permutations $\mathbf{p}_1 = \{1, 2\}$ and $\mathbf{p}_2 = \{2, 1\}$. In other words, $\mathbf{E}_i$'s are with the same structure. Next, using $\mathbf{E}_i$'s we obtain an MDS matrix $\mathbf{H}$ over $\mathbb{F}_2[\mathbf{L}]$ such that the implementation cost of $\mathbf{H}$ over 4 and 8-bit input are 35 and 67 XOR, respectively.

$$\begin{aligned} \mathbf{H} &= \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} \\ \mathbf{L}^{-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \\ &= \begin{pmatrix} \mathbf{L}+\mathbf{1} & \mathbf{L} & \mathbf{1} & \mathbf{L}+\mathbf{1} \\ \mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^{-1} & \mathbf{L} & \mathbf{L} \\ \mathbf{L}^{-1} & \mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}+\mathbf{1} & \mathbf{L} \\ \mathbf{L} & \mathbf{L} & \mathbf{L}+\mathbf{1} & \mathbf{1} \end{pmatrix}. \end{aligned}$$

Let $\mathbf{H} = \mathbf{E}_1\mathbf{E}_2\mathbf{E}_3\mathbf{E}_4$. Then the implementation cost of $\mathbf{H}$ for m-bit input is equal to

$$\overbrace{(2m)}^{\#\mathbf{E}_1} + \overbrace{(2m + \#\mathbf{L})}^{\#\mathbf{E}_2} + \overbrace{(2m + \#\mathbf{L} + \#(\mathbf{L}^{-1}))}^{\#\mathbf{E}_3} + \overbrace{(2m)}^{\#\mathbf{E}_4} = 8m + 2(\#\mathbf{L}) + \#(\mathbf{L}^{-1}). \tag{18}$$

The base set of subdeterminants of $\mathbf{H}$ is:

$$\{\mathbf{L}, \mathbf{L}+\mathbf{1}, \mathbf{L}^2+\mathbf{L}+\mathbf{1}, \mathbf{L}^3+\mathbf{L}+\mathbf{1}, \mathbf{L}^3+\mathbf{L}^2+\mathbf{1}\}. \tag{19}$$

For $m = 4$, consider 4×4 non-singular binary matrices $\mathbf{L}_i$ with $1 \le i \le 3$ in (20). It can be checked that by applying $\mathbf{L}_i$'s the given elements in (19) are non-singular matrices over $\mathbb{F}_2$. In addition, the implementation cost of $\mathbf{L}_i$ and $\mathbf{L}_i^{-1}$ with $1 \le i \le 6$ are one XOR.

$$\mathbf{L}_1 = [[1,4],[1],[2],[3]], \quad \mathbf{L}_2 = [[3,4],[1],[2],[3]], \quad \mathbf{L}_3 = [[4],[1,2],[2],[3]]. \tag{20}$$

Therefore, by applying $\mathbf{L}_i$'s and relation (18), the implementation cost of $\mathbf{H}$ is $8 \times 4 + 2 \times 1 + 1 \times 1 = 35$ XOR for 4-bit input. Moreover, it follows from (19) that the matrix $\mathbf{H}$ can be implemented with 35 XOR over $\mathbb{F}_{2^4}$, since the two irreducible polynomials `0x13` and `0x19` are not elements of (19). For $m = 8$, the next 8×8 non-singular binary matrices $\mathbf{L}_1 = [[2,8],[1],[2],[3],[4],[5],[6],[7]]$ and $\mathbf{L}_2 = [[6,8],[1],[2],[3],[4],[5],[6],[7]]$ are obtained such that the given elements in (19) are non-singular matrices over $\mathbb{F}_2$ by applying $\mathbf{L}_1$ and $\mathbf{L}_2$. Moreover, the implementation cost of $\mathbf{L}_j$ and $\mathbf{L}_j^{-1}$ with $1 \le j \le 2$ are one XOR. Hence, it follows from (18) that the implementation cost of $\mathbf{H}$ by applying $\mathbf{L}_j$'s is $8 \times 8 + 2 \times 1 + 1 \times 1 = 67$ XOR. Moreover, some roots of irreducible polynomials of degree 8 require only two XOR [BKL16]. Therefore, $\mathbf{H}$ are implemented with 70 XOR over $\mathbb{F}_{2^8}$.

Moreover, the implementation cost of $\mathbf{E}_2^{-1}$ and $\mathbf{E}_3^{-1}$ are $(2m + \#\mathbf{L})$ and $(2m + 2(\#\mathbf{L}))$, respectively. For instance, set $\mathbf{x} = [x_1, x_2, x_3, x_4]$. Consider $\mathbf{E}_3^{-1} \cdot \mathbf{x}^{\mathrm{T}} = [y_1, y_2, y_3, y_4]^{\mathrm{T}}$. Then we get $y_1 = \mathbf{L}x_4$, $y_2 = y_1 + x_1$, $y_3 = x_2$ and $y_4 = \mathbf{L}x_2 + x_3$. Therefore, the implementation cost of $\mathbf{H}^{-1}$ for m-bit input is equal to $(8m + 3(\#\mathbf{L}))$ and hence by applying the given 4×4 and 8×8 non-singular binary matrices, the matrix $\mathbf{H}^{-1}$ can be implemented with 35 and 67 XOR for 4 and 8-bit input, respectively.

6.2 Construction of 6×6 MDS Matrices from EGFS Matrices

In this section, we propose 6×6 MDS matrices for 4,6 and 8-bit input such that the implementation cost of the proposed matrices and their inverses are equal.

6.2.1 Construction of 6×6 MDS Matrix for 4-bit input

In this subsection, by applying distinct EGFS matrices, we propose a 6×6 MDS matrix which is implemented with 90 XOR for 4-bit input. In fact, we could not obtain 6×6 EGFS matrices $\mathbf{E}_i$ with $1 \leq i \leq 6$ provided that $\mathbf{E}_i$'s satisfy the following conditions. First, $\mathbf{E}_i$'s are with the same structure. Second, the multiplication of $\mathbf{E}_i$'s denoted with $\mathbf{H} = \prod_{i=1}^{6} \mathbf{E}_i$, is an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$. Finally, at least one of the two irreducible polynomials `0x13` and `0x19` are not elements of the base set of $\mathbf{H}$. Therefore, we used distinct EGFS matrices to construct a lightweight 6×6 MDS matrix for 4-bit input. Consider the following EGFS matrices $\mathbf{E}_i$ with $1 \leq i \leq 6$.

$$\mathbf{E}_1 = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{pmatrix}, \quad \mathbf{E}_2 = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{L}^{-6} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{L}^{-6} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{L}^{-6} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{pmatrix},$$

$$\mathbf{E}_3 = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{E}_4 = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{L}^{-1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L}^{-1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}^{-1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix},$$

$$\mathbf{E}_5 = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L}^{-1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L}^{-1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}^{-1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{pmatrix}, \quad \mathbf{E}_6 = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{pmatrix}.$$

The two EGFS matrices $\mathbf{E}_1$ and $\mathbf{E}_5$ are with the same structure and are constructed from the two permutations $\mathbf{p}_1 = \{3, 2, 1\}$ and $\mathbf{p}_2 = \{2, 1, 3\}$ and also are 5-primitive matrices over $\mathbb{R}$. The EGFS matrix $\mathbf{E}_2$ is constructed from $\mathbf{p}_1 = \{3, 1, 2\}$ and $\mathbf{p}_2 = \{1, 2, 3\}$ and also is an 6-primitive matrix over $\mathbb{R}$. Moreover, EGFS matrices $\mathbf{E}_3$ and $\mathbf{E}_4$ are 6-primitive matrices and $\mathbf{E}_6$ is a 5-primitive matrix over $\mathbb{R}$. Next, applying EGFS matrices $\mathbf{E}_i$ with $1 \leq i \leq 6$, the proposed lightweight 6×6 MDS matrix $\mathbf{H}$ is given by

$$\begin{aligned} \mathbf{H} &= \mathbf{E}_1\mathbf{E}_2\mathbf{E}_3\mathbf{E}_4\mathbf{E}_5\mathbf{E}_6 \\ &= \begin{pmatrix} \mathbf{L}^{-2}+\mathbf{L}^{-6}+\mathbf{L}^{-7}+\mathbf{1} & \mathbf{L}^{-2}+\mathbf{L}^{-6} & \mathbf{L}^{-2}+\mathbf{1} & \mathbf{L}^{-1}+\mathbf{L}^{-2}+\mathbf{1} & \mathbf{L}^{-8} & \mathbf{L}^{-1}+\mathbf{L}^{-8}+\mathbf{1} \\ \mathbf{1} & \mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^{-8}+\mathbf{1} & \mathbf{L}^{-8} & \mathbf{L}^{-2}+\mathbf{L}^{-6}+\mathbf{L}^{-7} & \mathbf{L}^{-2}+\mathbf{L}^{-6} \\ \mathbf{L}^{-2}+\mathbf{1} & \mathbf{L}^{-1}+\mathbf{L}^{-2}+\mathbf{1} & \mathbf{L}^{-8} & \mathbf{L}^{-1}+\mathbf{L}^{-8}+\mathbf{1} & \mathbf{L}^{-2}+\mathbf{L}^{-6}+\mathbf{L}^{-7}+\mathbf{1} & \mathbf{L}^{-2}+\mathbf{L}^{-6} \\ \mathbf{L}^{-8}+\mathbf{1} & \mathbf{L}^{-8} & \mathbf{L}^{-2}+\mathbf{L}^{-6}+\mathbf{L}^{-7} & \mathbf{L}^{-2}+\mathbf{L}^{-6} & \mathbf{1} & \mathbf{L}^{-1}+\mathbf{1} \\ \mathbf{L}^{-8} & \mathbf{L}^{-1}+\mathbf{L}^{-8}+\mathbf{1} & \mathbf{L}^{-2}+\mathbf{L}^{-6}+\mathbf{L}^{-7}+\mathbf{1} & \mathbf{L}^{-2}+\mathbf{L}^{-6} & \mathbf{L}^{-2}+\mathbf{1} & \mathbf{L}^{-1}+\mathbf{L}^{-2}+\mathbf{1} \\ \mathbf{L}^{-2}+\mathbf{L}^{-6}+\mathbf{L}^{-7} & \mathbf{L}^{-2}+\mathbf{L}^{-6} & \mathbf{1} & \mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^{-8}+\mathbf{1} & \mathbf{L}^{-8} \end{pmatrix}. \end{aligned}$$

The implementation cost of $\mathbf{H}$ for m-bit input is equal to

$$\begin{aligned} \#\mathbf{H} &= \overbrace{(3m)}^{\#\mathbf{E}_1} + \overbrace{(3m + 3(\#\mathbf{L}^{-6}))}^{\#\mathbf{E}_2} + \overbrace{(3m)}^{\#\mathbf{E}_3} + \overbrace{(3m + 3(\#\mathbf{L}^{-1}))}^{\#\mathbf{E}_4} + \overbrace{(3m + 3(\#\mathbf{L}^{-1}))}^{\#\mathbf{E}_5} + \overbrace{(3m)}^{\#\mathbf{E}_6} \\ &= 18m + 6(\#\mathbf{L}^{-1}) + 3(\#\mathbf{L}^{-6}). \end{aligned} \tag{21}$$

The base set of subdeterminants of $\mathbf{H}$ has 149 elements such that $\mathbf{L}^4 + \mathbf{L} + 1(\texttt{0x13})$ is not an element of the base set. For $m = 4$, consider the next 4×4 non-singular matrices over $\mathbb{F}_2$. It can be verified that using $\mathbf{L}_i$ with $1 \le i \le 2$, the elements of base set are non-singular matrices over $\mathbb{F}_2$. Moreover, the implementation cost of $\mathbf{L}_i$ and $\mathbf{L}_i^{-1}$ with $1 \le i \le 2$ are one XOR.

$$\mathbf{L}_1 = [[3,4],[1],[2],[3]], \quad \mathbf{L}_2 = [[4],[1,4],[2],[3]].$$

In addition, $\mathbf{L}_i^{-6}$ with $1 \le i \le 2$ can be implemented with four XOR. For instance, consider $\mathbf{x} = [x_1, x_2, x_3, x_4]$ and assume that $\mathbf{L}_1^{-6} \cdot \mathbf{x}^{\mathrm{T}} = [y_1, y_2, y_3, y_4]^{\mathrm{T}}$. Then we get $y_1 = x_1 + x_3$, $y_2 = y_1 + x_2$, $y_3 = y_2 + x_4$ and $y_4 = y_3 + x_1$. Next, by applying $\mathbf{L}_i$'s and relation (21), $\mathbf{H}$ is implemented with $18 \times 4 + 6 \times 1 + 3 \times 4 = 90$ XOR for 4-bit input. Moreover, $\mathbf{H}$ is implemented with 90 XOR over $\mathbb{F}_{2^4}/\texttt{0x13}$. Furthermore, it is easy to check that $\mathbf{H}^{-1}$ can be implemented with 90 XOR for 4-bit input.

6.2.2 Construction of 6 × 6 MDS Matrix for 6-bit and 8-bit input

In this subsection, using EGFS matrices, we propose a new lightweight 6×6 MDS matrix which is implemented with 156 XOR for 8-bit input. Moreover, the proposed 6×6 MDS matrix is implemented with 114 XOR for 6-bit input. The result of this subsection is not only efficient from hardware perspective, but also are suitable by software terminology. In fact, in order to construct the proposed 6×6 MDS matrix for 8-bit input, six binary linear functions ($\mathbf{L}$) are used such that the cost of $\mathbf{L}$ is very small compared to the total XOR.

Consider the following EGFS matrices $\mathbf{E}_i$ with $1 \le i \le 2$ such that $\mathbf{E}_1$ is constructed from the two permutations $\mathbf{p}_1 = \{1,2,3\}$ and $\mathbf{p}_2 = \{3,1,2\}$ and also $\mathbf{E}_2$ is obtained from $\mathbf{p}_1 = \{3,2,1\}$ and $\mathbf{p}_2 = \{2,1,3\}$. It can be checked that $\mathbf{E}_1$ and $\mathbf{E}_2$ are 6-primitive and 5-primitive matrices over $\mathbb{R}$, respectively.

$$\mathbf{E}_1 = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{E}_2 = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{pmatrix}.$$

Now based on the EGFS matrices $\mathbf{E}_i$ with $1 \le i \le 2$, the proposed lightweight 6×6 MDS matrix $\mathbf{H}$ is constructed as follows.

$$\mathbf{H} = \mathbf{E}_1^2 \times \mathbf{E}_2^2 \times \mathbf{E}_1^2 = \begin{pmatrix} \mathbf{L}^2+\mathbf{L} & \mathbf{L}^2+\mathbf{1} & \mathbf{L}^2 & \mathbf{1} & \mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} \\ \mathbf{L}+\mathbf{1} & \mathbf{L} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{1} \\ \mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L} & \mathbf{L}^2+\mathbf{1} & \mathbf{L}^2 & \mathbf{1} \\ \mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{1} & \mathbf{L}+\mathbf{1} & \mathbf{L} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}+\mathbf{1} \\ \mathbf{L}^2 & \mathbf{1} & \mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L} & \mathbf{L}^2+\mathbf{1} \\ \mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{1} & \mathbf{L}+\mathbf{1} & \mathbf{L} \end{pmatrix}.$$

The implementation cost of $\mathbf{H}$ for m-bit input is equal to

$$\overbrace{2(3m)}^{2(\#\mathbf{E}_1)} + \overbrace{2(3m + 3(\#\mathbf{L}))}^{\#2(\mathbf{E}_2)} + \overbrace{2(3m)}^{2(\#\mathbf{E}_1)} = 18m + 6(\#\mathbf{L}) \tag{22}$$

The base set of subdeterminants of $\mathbf{H}$ has 20 elements that are listed in (23).

$$\begin{gathered} \{\texttt{0x2}, \texttt{0x3}, \texttt{0x7}, \texttt{0xB}, \texttt{0xD}, \texttt{0x13}, \texttt{0x19}, \texttt{0x1F}, \texttt{0x25}, \texttt{0x29}, \texttt{0x2F}, \\ \texttt{0x37}, \texttt{0x3B}, \texttt{0x3D}, \texttt{0x61}, \texttt{0x57}, \texttt{0x5B}, \texttt{0x67}, \texttt{0x6D}, \texttt{0x75}\} \end{gathered} \tag{23}$$

Consider 8×8 non-singular binary matrices $\mathbf{L}_i$ with $1 \le i \le 4$ that are given in (24). It can be verified that by applying $\mathbf{L}_i$'s, the given elements in (23) are non-singular matrices over $\mathbb{F}_2$. Moreover, the implementation cost of $\mathbf{L}_i$ and $\mathbf{L}_i^{-1}$ with $1 \le i \le 4$ are two XOR.

$$\begin{array}{ll} \mathbf{L}_1 = [[1,8],[1],[2,5],[3],[4],[5],[6],[7]], & \mathbf{L}_2 = [[1,8],[1],[2,7],[3],[4],[5],[6],[7]], \\ \mathbf{L}_3 = [[1,8],[1],[2],[3,6],[4],[5],[6],[7]], & \mathbf{L}_4 = [[1,8],[1],[2],[3],[4,7],[5],[6],[7]]. \end{array} \tag{24}$$

Therefore, by applying $\mathbf{L}_i$'s and relation (22), $\mathbf{H}$ is implemented with $18 \times 8 + 6 \times 2 = 156$ XOR for 8-bit input. Moreover, $\mathbf{H}$ can be implemented with 156 XOR over $\mathbb{F}_{2^8}$, since some roots of the irreducible polynomials of degree 8 require only two XOR [BKL16] and there are no irreducible polynomials of degree 8 in (23). Furthermore, it is easy to verify that $\mathbf{H}^{-1}$ is implemented with 156 XOR over $\mathbb{F}_{2^8}$.

Finally, we implement $\mathbf{H}$ for 6-bit input with the low XOR cost. In the base set of $\mathbf{H}$, given in (23), there are all irreducible polynomials of degree 6 except the three irreducible polynomials `0x43`, `0x49` and `0x73`. Next, we search on 6×6 binary matrices provided that the implementation cost of these matrices is one XOR and their minimal polynomials over $\mathbb{F}_2$ are `0x43` or `0x49`. For instance, consider the following 6×6 binary matrices.

$$\mathbf{L}_1 = [[5,6],[1],[2],[3],[4],[5]], \quad \mathbf{L}_2 = [6],[1,4],[2],[3],[4],[5]].$$

The implementation cost of $\mathbf{L}_i$ with $1 \le i \le 2$ is one XOR. Moreover, the minimal polynomial of $\mathbf{L}_1$ and $\mathbf{L}_2$ are `0x43` and `0x49`, respectively. Therefore, by applying $\mathbf{L}_i$'s the given elements in (23) are non-singular matrices over $\mathbb{F}_2$. Hence, using $\mathbf{L}_i$'s in (22), $\mathbf{H}$ and $\mathbf{H}^{-1}$ are implemented with $18 \times 6 + 6 \times 1 = 114$ XOR for 6-bit input. Moreover, based on Proposition 1 the implementation cost of $\mathbf{H}$ over $\mathbb{F}_{2^6}$/`0x43` and $\mathbb{F}_{2^6}$/`0x49` is 114 XOR.

6.3 Construction of 8×8 MDS and Near-MDS Matrices from EGFS

First, we propose two 8×8 near-MDS matrices such that the implementation cost of the proposed matrices are 116 and 108 XOR for 4-bit input. Moreover, the proposed near-MDS matrices are implemented with 212 and 204 XOR for 8-bit input, respectively. The first proposed near-MDS matrix can be applied to hardware implementation, since has low XOR cost and also the depth of its circuit is 6. The second proposed near-MDS matrix can be used in software implementation, since the proposed matrix is constructed from 12 binary linear functions. Next, we propose two 8×8 MDS matrices for 8-bit input. The first proposed MDS matrix is implemented with 272 XOR and the second 8×8 MDS matrix is implemented with 260 XOR for 8-bit input.

6.3.1 Construction of Lightweight 8×8 Near-MDS Matrices for 4 and 8-bit inputs

The best results for the construction of 8×8 near-MDS matrices for 4 and 8-bit input are 216 and 432 XOR which are given in [LW17]. In this subsection, the proposed 8×8 near-MDS matrices are not only with low implementation cost, but also the depth of their circuits are low. Consider the two proposed 8×8 EGFS matrices $\mathbf{E}_1$ and $\mathbf{E}_2$, which are constructed from the two permutations $\mathbf{p}_1 = \{4,3,2,1\}$ and $\mathbf{p}_2 = \{3,2,1,4\}$. It can be checked that $\mathbf{E}_1$ and $\mathbf{E}_2$ are 6-primitive matrices over $\mathbb{R}$.

$$\mathbf{E}_1 = \begin{pmatrix} 0&0&0&0&0&0&1&1\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&1&1&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&1&1&0&0&0&0\\ 1&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0 \end{pmatrix}, \quad \mathbf{E}_2 = \begin{pmatrix} 0&0&0&0&0&0&\mathbf{L}^{-2}&\mathbf{L}\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&\mathbf{L}^{-2}&\mathbf{L}&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&\mathbf{L}^{-2}&\mathbf{L}&0&0&0&0\\ 1&0&0&0&0&0&0&0\\ \mathbf{L}^{-2}&\mathbf{L}&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0 \end{pmatrix}. \tag{25}$$

By applying the two EGFS matrices $\mathbf{E}_1$ and $\mathbf{E}_2$, we propose the following 8×8 matrix.

$$\mathbf{H} = \mathbf{E}_1^3\mathbf{E}_2\mathbf{E}_1^2 =$$

$$\begin{pmatrix} \mathbf{L} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2}+\mathbf{1} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{1} & \mathbf{1} & \mathbf{L}^{-2}+\mathbf{1} \\ \mathbf{L}^{-2}+\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{1} & \mathbf{L} \\ \mathbf{1} & \mathbf{L}^{-2}+\mathbf{1} & \mathbf{L} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2}+\mathbf{1} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{1} \\ \mathbf{L}+\mathbf{1} & \mathbf{L} & \mathbf{L}^{-2}+\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2} \\ \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{1} & \mathbf{1} & \mathbf{L}^{-2}+\mathbf{1} & \mathbf{L} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2}+\mathbf{1} & \mathbf{L}+\mathbf{L}^{-2} \\ \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{1} & \mathbf{L} & \mathbf{L}^{-2}+\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{L}^{-2} \\ \mathbf{L}+\mathbf{L}^{-2}+\mathbf{1} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{1} & \mathbf{1} & \mathbf{L}^{-2}+\mathbf{1} & \mathbf{L} & \mathbf{L}+\mathbf{L}^{-2} \\ \mathbf{0} & \mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{L}^{-2} & \mathbf{L}+\mathbf{1} & \mathbf{L} & \mathbf{L}^{-2}+\mathbf{1} & \mathbf{1} \end{pmatrix}. \tag{26}$$

The implementation cost of $\mathbf{H}$ for m-bit input is equal to

$$\overbrace{3(4m)}^{3(\#\mathbf{E}_1)} + \overbrace{(4m + 4(\#\mathbf{L}) + 4(\#\mathbf{L}^{-2}))}^{\#(\mathbf{E}_2)} + \overbrace{2(4m)}^{2(\#\mathbf{E}_1)} = 24m + 4(\#\mathbf{L}) + 4(\#\mathbf{L}^{-2}). \tag{27}$$

Consider the following 4×4 and 8×8 binary matrices, given in (28). The minimal polynomials of $\mathbf{L}_i$ with $1 \leq i \leq 2$ over $\mathbb{F}_2$ are $L^4 + L + 1$ and $(L^4 + L + 1)^2$. Moreover, the implementation cost of $\mathbf{L}_i$'s and $\mathbf{L}_i^{-2}$'s are one and two XOR, respectively.

$$\mathbf{L}_1 = [[4], [1, 4], [2], [3]], \quad \mathbf{L}_2 = [[8], [1], [2, 8], [3], [4], [5], [6], [7]]. \tag{28}$$

Now, by applying $\mathbf{L}_i$'s it can be check that the matrix $\mathbf{H}$ is a 8×8 near-MDS matrix over $\mathbb{F}_2[\mathbf{L}]$. Therefore, using $\mathbf{L}_i$ with $1 \leq i \leq 2$ and relation (27), the matrix $\mathbf{H}$ can be implemented with $24 \times 4 + 4 \times 1 + 4 \times 2 = 108$ and $24 \times 8 + 4 \times 1 + 4 \times 2 = 204$ XOR for 4 and 8-bit input, respectively. Furthermore, the implementation cost of the matrix $\mathbf{E}_2^{-1}$ are $4m + 4(\#\mathbf{L}^{-1}) + 4(\#\mathbf{L}^{-2})$. In fact, consider $\mathbf{x} = [x_1, x_2, \cdots, x_8]$. Then we get

$$\mathbf{E}_2^{-1} \cdot \mathbf{x}^{\mathrm{T}} = \\ \left[x_6, \mathbf{L}^{-1}(x_7 + \mathbf{L}^{-2}x_6), x_4, \mathbf{L}^{-1}(x_5 + \mathbf{L}^{-2}x_4), x_2, \mathbf{L}^{-1}(x_3 + \mathbf{L}^{-2}x_2), x_8, \mathbf{L}^{-1}(x_1 + \mathbf{L}^{-2}x_8)\right].$$

The above relation implies that the implementation cost of $\mathbf{H}^{-1}$ for m-bit input is equal to the implementation cost of the matrix $\mathbf{H}$, since $\mathbf{L}_i^{-1}$'s are implemented with one XOR.

It follows from Fig. 6 that the depth of $\mathbf{H}$, given in (26), is 7. In the rest, we propose a 8×8 near-MDS matrix such that the depth of its circuit is 6 and has low XOR cost. Consider the given 8×8 EGFS matrices $\widehat{\mathbf{E}}_i$ with $1 \leq i \leq 3$ in (29). The proposed matrices in (29) are with the same structure with $\mathbf{E}_1$ and $\mathbf{E}_2$ which are given in (25).

$$\widehat{\mathbf{E}}_i = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{f}_i & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{f}_i & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{f}_i & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{f}_i & \mathbf{0} \end{pmatrix}, \quad \mathbf{f_1} = \mathbf{1}, \quad \mathbf{f_2} = \mathbf{L}^2, \quad \mathbf{f_3} = \mathbf{L}^3. \tag{29}$$

Based on the given matrices in (29), we propose the following 8×8 matrix $\widehat{\mathbf{H}} = \widehat{\mathbf{E}}_1^3\widehat{\mathbf{E}}_2\widehat{\mathbf{E}}_3\widehat{\mathbf{E}}_1$. It is easy to check that the implementation cost of $\widehat{\mathbf{H}}$ for m-bit input is equal to $24m + 4(\#\mathbf{L}^2) + 4(\#\mathbf{L}^3)$. Moreover, using $\mathbf{L}_1$ and $\mathbf{L}_2$ in (28), it can be checked that $\widehat{\mathbf{H}}$ is a 8×8 near-MDS matrix over $\mathbb{F}_2[\mathbf{L}]$. Therefore, by applying the given binary matrices in (28), the matrix $\widehat{\mathbf{H}}$ can be implemented with $24 \times 4 + 4 \times 2 + 4 \times 3 = 116$ and $24 \times 8 + 4 \times 2 + 4 \times 3 = 212$ XOR for 4 and 8-bit input, since the implementation cost of $\mathbf{L}_i^2$ and $\mathbf{L}_i^3$ with $1 \leq i \leq 2$ are two and three XOR, respectively.

Furthermore, the implementation cost of $\widehat{\mathbf{E}}_i^{-1}$ with $1 \leq i \leq 3$ are equal to $4m$, $4m + 4(\#\mathbf{L}^{-2})$ and $4m + 4(\#\mathbf{L}^{-3})$, respectively. For instance, consider $\mathbf{x} = [x_1, x_2, \cdots, x_8]$ and assume that $\widehat{\mathbf{E}}_3^{-1} \cdot \mathbf{x}^{\mathrm{T}} = [y_1, y_2, \cdots, y_8]^{\mathrm{T}}$. Then we get

$$y_1 = \mathbf{L}^{-3}x_6, \quad y_2 = y_1 + x_7, \quad y_3 = \mathbf{L}^{-3}x_4, \quad y_4 = y_3 + x_5,$$
$$y_5 = \mathbf{L}^{-3}x_2, \quad y_6 = y_5 + x_3, \quad y_7 = \mathbf{L}^{-3}x_8, \quad y_8 = y_7 + x_1.$$

Hence, the implementation cost of $\widehat{\mathbf{H}}^{-1}$ for m-bit input is equal to $24m+4(\#\mathbf{L}^{-2})+4(\#\mathbf{L}^{-3})$. In addition, the implementation cost of $\mathbf{L}_i^{-2}$ and $\mathbf{L}_i^{-3}$ with $1 \leq i \leq 2$ are two and three XOR, respectively. Therefore using $\mathbf{L}_1$ and $\mathbf{L}_2$, given in (28), the implementation cost of $\widehat{\mathbf{H}}^{-1}$ is equal to the implementation cost of the matrix $\widehat{\mathbf{H}}$.

6.3.2 Construction of a 8×8 MDS Matrix with 272 XOR Cost

First of all, by applying Appendix A, we tried to obtain 8×8 EGFS matrices $\mathbf{E}_i$ with $1 \leq i \leq 6$ such that $\mathbf{E}_i$'s satisfy the following conditions. First, $\mathbf{E}_i$'s are with the same structure and are 6-primitive matrices over $\mathbb{R}$. Second, the multiplication of $\mathbf{E}_i$'s denoted with $\mathbf{H} = \prod_{i=1}^{6} \mathbf{E}_i$, is an MDS matrix over $\mathbb{F}_2[\mathbf{L}]$. Finally, the implementation cost of $\mathbf{H}$ is less than 392 XOR for 8-bit input. But we could not get EGFS matrices under the stated conditions. Therefore, we increased the number of 8×8 EGFS matrices.

Consider 8×8 EGFS matrices $\mathbf{E}_i$ with $1 \leq i \leq 5$, given in (30). The structures of $\mathbf{E}_i$'s are the same, since $\mathbf{E}_i$'s are constructed from two permutations $\mathbf{p}_1 = \{4,3,2,1\}$ and $\mathbf{p}_2 = \{3,2,1,4\}$. Moreover, using Appendix A, it can be checked that EGFS matrices $\mathbf{E}_i$ with $1 \leq i \leq 5$ are 6-primitive matrices over $\mathbb{R}$.

$$\mathbf{c}_1^{(1,m)} = \cdots = \mathbf{c}_4^{(1,m)} = \begin{pmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{c}_1^{(2,m)} = \cdots = \mathbf{c}_4^{(2,m)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} \end{pmatrix}, \quad \mathbf{E}_1 = \left(\begin{array}{cc|cc|cc|cc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{array}\right)$$

$$\mathbf{c}_1^{(1,m)} = \cdots = \mathbf{c}_4^{(1,m)} = \begin{pmatrix} \mathbf{1} & \mathbf{L} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{c}_1^{(1,m)} = \cdots = \mathbf{c}_4^{(1,m)} = \begin{pmatrix} \mathbf{L} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

$$\mathbf{c}_1^{(2,m)} = \cdots = \mathbf{c}_4^{(2,m)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} \end{pmatrix}, \quad \mathbf{c}_1^{(2,m)} = \cdots = \mathbf{c}_4^{(2,m)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} \end{pmatrix},$$

$$\mathbf{E}_2 = \left(\begin{array}{cc|cc|cc|cc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{L} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{L} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{L} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{1} & \mathbf{L} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{array}\right), \quad \mathbf{E}_3 = \left(\begin{array}{cc|cc|cc|cc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{L} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{array}\right)$$

$$\mathbf{c}_1^{(1,m)} = \cdots = \mathbf{c}_4^{(1,m)} = \begin{pmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{c}_1^{(1,m)} = \cdots = \mathbf{c}_4^{(1,m)} = \begin{pmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

$$\mathbf{c}_1^{(2,m)} = \cdots = \mathbf{c}_4^{(2,m)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{L}^{-1} & \mathbf{0} \end{pmatrix}, \quad \mathbf{c}_1^{(2,m)} = \cdots = \mathbf{c}_4^{(2,m)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{L} & \mathbf{0} \end{pmatrix},$$

$$\mathbf{E}_4 = \left(\begin{array}{cc|cc|cc|cc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L}^{-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L}^{-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}^{-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L}^{-1} & \mathbf{0} \end{array}\right), \quad \mathbf{E}_5 = \left(\begin{array}{cc|cc|cc|cc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{L} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L} & \mathbf{0} \end{array}\right) \tag{30}$$

Now, using EGFS matrices $\mathbf{E}_i$ with $1 \leq i \leq 5$, the following 8×8 matrix is proposed.

$$
\begin{aligned}
\mathbf{H} &= \mathbf{E}_1\mathbf{E}_2\mathbf{E}_1\mathbf{E}_3\mathbf{E}_4\mathbf{E}_5\mathbf{E}_1 \\
&= \left(\begin{matrix}
\mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^2+\mathbf{L} \\
\mathbf{L}^2+\mathbf{L}+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^2+\mathbf{L}^{-1} \\
\mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^2+\mathbf{L} & \mathbf{L}^2 & \mathbf{L}^2+\mathbf{L}+\mathbf{L}^{-1} \\
\mathbf{L}^2+\mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^2+\mathbf{L}^{-1} & \mathbf{L}^3+\mathbf{L}+\mathbf{1} & \mathbf{L}^3+\mathbf{L} \\
\mathbf{L}^2 & \mathbf{L}^2+\mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^3+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^3+\mathbf{L}+\mathbf{L}^{-1} \\
\mathbf{L}^3+\mathbf{L}+\mathbf{1} & \mathbf{L}^3+\mathbf{L} & \mathbf{L}^2 & \mathbf{L}^2+\mathbf{1} \\
\mathbf{L}^3+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^3+\mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} \\
\mathbf{L}^2 & \mathbf{L}^2+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}+\mathbf{1}
\end{matrix} \right. \\
&\qquad \left. \begin{matrix}
\mathbf{L}^2 & \mathbf{L}^2+\mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^3+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^3+\mathbf{L}+\mathbf{L}^{-1} \\
\mathbf{L}^3+\mathbf{L}+\mathbf{1} & \mathbf{L}^3+\mathbf{L} & \mathbf{L}^2 & \mathbf{L}^2+\mathbf{1} \\
\mathbf{L}^3+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^3+\mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} \\
\mathbf{L}^2 & \mathbf{L}^2+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}+\mathbf{1} \\
\mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{1} & \mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^2+\mathbf{L} \\
\mathbf{L}^2+\mathbf{L}+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}+\mathbf{1} & \mathbf{L}^2+\mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^2+\mathbf{L}^{-1} \\
\mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^2+\mathbf{L} & \mathbf{L}^2 & \mathbf{L}^2+\mathbf{L}+\mathbf{L}^{-1} \\
\mathbf{L}^2+\mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^2+\mathbf{L}^{-1} & \mathbf{L}^3+\mathbf{L}+\mathbf{1} & \mathbf{L}^3+\mathbf{L}
\end{matrix} \right).
\end{aligned}
\tag{31}
$$

It follows from (31) that the implementation cost of matrix $\mathbf{H}$ for m-bit input is equal to

$$
\begin{aligned}
&\overbrace{(4m)}^{\#\mathbf{E}_1} + \overbrace{(4m+4(\#\mathbf{L}))}^{\#\mathbf{E}_2} + \overbrace{(4m)}^{\#\mathbf{E}_1} + \overbrace{(4m+4(\#\mathbf{L}))}^{\#\mathbf{E}_3} + \overbrace{(4m+4(\#\mathbf{L}^{-1}))}^{\#\mathbf{E}_4} + \overbrace{(4m+4(\#\mathbf{L}))}^{\#\mathbf{E}_5} \\
&+ \overbrace{(4m)}^{\#\mathbf{E}_1} = 28m + 12(\#\mathbf{L}) + 4(\#\mathbf{L}^{-1}).
\end{aligned}
\tag{32}
$$

The base set of subdeterminants of $\mathbf{H}$ has 285 elements that is given in Appendix D. In this base set, there are all irreducible polynomials of degrees 4 and 8, except for the primitive polynomial `0x1E7`. It can be checked the elements of Appendix D, are non-singular matrices over $\mathbb{F}_2$ by applying the following non-singular 8×8 binary matrices $\mathbf{L}_i$ with $1 \leq i \leq 4$.

$$
\begin{aligned}
\mathbf{L}_1 &= [[1,8],[1,3,7],[2],[3],[4],[5],[6],[7]], & \mathbf{L}_2 &= [[1,8],[1,3],[2,8],[3],[4],[5],[6],[7]], \\
\mathbf{L}_3 &= [[1,8],[1,7],[2,4],[3],[4],[5],[6],[7]], & \mathbf{L}_4 &= [[1,8],[1,7],[2],[3,5],[4],[5],[6],[7]].
\end{aligned}
\tag{33}
$$

Moreover, the implementation cost of $\mathbf{L}_i$'s, given in (33), is three XOR. Furthermore, it can be verified that the inverse of $\mathbf{L}_i$'s can be implemented with the three XOR. Hence, using $\mathbf{L}_i$ with $1 \leq i \leq 24$ and relation (32), $\mathbf{H}$ is implemented by $28 \times 8 + 12 \times 3 + 4 \times 3 = 272$ XOR for 8-bit input. Also, it can be checked that the minimal polynomial of $\mathbf{L}_i$'s is the irreducible polynomial `0x1E7`. Therefore, using Proposition 1, $\mathbf{H}$ is implemented with 272 XOR over $\mathbb{F}_{2^8}/$`0x1E7`. In addition, the implementation cost of $\mathbf{E}_i^{-1}$ with $1 \leq i \leq 5$ are equal to $4m$, $4m+4(\#\mathbf{L}^{-1})$, $4m+4(\#\mathbf{L})$, $4m+4(\#\mathbf{L})$ and $4m+4(\#\mathbf{L}^{-1})$, respectively. For instance, set $\mathbf{x} = [x_1, x_2, \cdots, x_8]$ and suppose that $\widehat{\mathbf{E}}_5^{-1} \cdot \mathbf{x}^{\mathrm{T}} = [y_1, y_2, \cdots, y_8]^{\mathrm{T}}$. Then

$$
\begin{aligned}
y_1 &= \mathbf{L}^{-1}x_6, & y_2 &= y_1 + x_7, & y_3 &= \mathbf{L}^{-1}x_4, & y_4 &= y_3 + x_5, \\
y_5 &= \mathbf{L}^{-1}x_2, & y_6 &= y_5 + x_3, & y_7 &= \mathbf{L}^{-1}x_8, & y_8 &= y_7 + x_1.
\end{aligned}
$$

Therefore, the implementation cost of $\mathbf{H}^{-1}$ for m-bit input is equal to $28m + 8(\#\mathbf{L}) + 8(\#\mathbf{L}^{-1})$ which implies that the implementation cost of $\mathbf{H}^{-1}$ and $\mathbf{H}$ is equal.

6.3.3 Construction of a 8×8 MDS Matrix with Depth 8

In this subsection, we propose an 8×8 MDS matrix $\mathbf{H}$ such that the proposed matrix is implemented with 260 XOR for 8-bit input. Moreover, the depth of its circuit is 8 which means $\mathbf{H}$ can be considered as a candidate for 8×8 lightweight MDS matrices in the lightweight cryptography. Consider 8×8 EGFS matrices $\mathbf{E}_i$ with $1 \leq i \leq 4$, given in (34) which are constructed from two permutations $\mathbf{p}_1 = \{4,3,2,1\}$ and $\mathbf{p}_2 = \{3,2,1,4\}$.

$$\mathbf{E}_1 = \begin{pmatrix} \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{0} \end{pmatrix}, \quad \mathbf{E}_3 = \begin{pmatrix} \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{L}^3&\mathbf{1}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{L}^3&\mathbf{1}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{L}^3&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{L}^3&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{0} \end{pmatrix},$$

$$\mathbf{E}_2 = \begin{pmatrix} \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{L}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{L}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{L}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{L}&\mathbf{0} \end{pmatrix}, \quad \mathbf{E}_4 = \begin{pmatrix} \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{L}^{-1}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{L}^{-1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{L}^{-1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{1}&\mathbf{1}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{L}^{-1}&\mathbf{0} \end{pmatrix}. \tag{34}$$

Next, using EGFS matrices $\mathbf{E}_i$ with $1 \le i \le 4$, the proposed lightweight 8×8 MDS matrix $\mathbf{H}$ is presented by

$$\begin{aligned} \mathbf{H} &= \mathbf{E}_1\mathbf{E}_2\mathbf{E}_3\mathbf{E}_1\mathbf{E}_1\mathbf{E}_4\mathbf{E}_1 \\ &= \begin{pmatrix} \mathbf{L}^4+\mathbf{L}^2+\mathbf{L} & \mathbf{L}^4+\mathbf{L}^2+\mathbf{1} & \mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^3+\mathbf{L}+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^4 & \mathbf{L}^3 & \mathbf{L}^3+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^4+\mathbf{L}^3+\mathbf{L}^{-1} \\ \mathbf{L}^3+\mathbf{L}^{-1} & \mathbf{L}^{-1}+\mathbf{1} & \mathbf{1} & \mathbf{L}^3+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^{-1} & \mathbf{L}^2 & \mathbf{L}^2+\mathbf{1} \\ \mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^3+\mathbf{L}+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^4 & \mathbf{L}^3 & \mathbf{L}^3+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^4+\mathbf{L}^3+\mathbf{L}^{-1} & \mathbf{L}^4+\mathbf{L}^2+\mathbf{L} & \mathbf{L}^4+\mathbf{L}^2+\mathbf{1} \\ \mathbf{1} & \mathbf{L}^3+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^{-1} & \mathbf{L}^2 & \mathbf{L}^2+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^{-1} & \mathbf{L}^{-1}+\mathbf{1} \\ \mathbf{L}^4 & \mathbf{L}^3 & \mathbf{L}^3+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^4+\mathbf{L}^3+\mathbf{L}^{-1} & \mathbf{L}^4+\mathbf{L}^2+\mathbf{L} & \mathbf{L}^4+\mathbf{L}^2+\mathbf{1} & \mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^3+\mathbf{L}+\mathbf{L}^{-1}+\mathbf{1} \\ \mathbf{L}^3+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^{-1} & \mathbf{L}^2 & \mathbf{L}^2+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^{-1} & \mathbf{L}^{-1}+\mathbf{1} & \mathbf{1} & \mathbf{L}^3+\mathbf{1} \\ \mathbf{L}^3+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^4+\mathbf{L}^3+\mathbf{L}^{-1} & \mathbf{L}^4+\mathbf{L}^2+\mathbf{L} & \mathbf{L}^4+\mathbf{L}^2+\mathbf{1} & \mathbf{L}+\mathbf{L}^{-1} & \mathbf{L}^3+\mathbf{L}+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^4 & \mathbf{L}^3 \\ \mathbf{L}^2 & \mathbf{L}^2+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^{-1} & \mathbf{L}^{-1}+\mathbf{1} & \mathbf{1} & \mathbf{L}^3+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^{-1}+\mathbf{1} & \mathbf{L}^3+\mathbf{L}^{-1} \end{pmatrix}. \end{aligned} \tag{35}$$

By applying (35) the implementation cost of $\mathbf{H}$ for m-bit input is equal to

$$\begin{aligned} &\overbrace{(4m)}^{\#\mathbf{E}_1} + \overbrace{(4m + 4(\#\mathbf{L}))}^{\#\mathbf{E}_2} + \overbrace{(4m + 4(\#\mathbf{L}^3))}^{\#\mathbf{E}_3} + \overbrace{(4m)}^{\#\mathbf{E}_1} + \overbrace{(4m)}^{\#\mathbf{E}_1} + \overbrace{(4m + 4(\#\mathbf{L}^{-1}))}^{\#\mathbf{E}_4} + \overbrace{(4m)}^{\#\mathbf{E}_1} \\ &= 28m + 4(\#\mathbf{L}) + 4(\#\mathbf{L}^{-1}) + 4(\#\mathbf{L}^3). \end{aligned} \tag{36}$$

The base set of subdeterminants of $\mathbf{H}$ has 324 elements that is given in Appendix C. In this base set, there are all irreducible and primitive polynomials of degree 4 and 8, except for the primitive polynomial `0x187`. Now using the following non-singular 8×8 binary matrices $\mathbf{L}_i$ with $1 \le i \le 6$, it can be verified that the elements of Appendix C are non-singular matrices over $\mathbb{F}_2$.

$$\begin{aligned} \mathbf{L}_1 &= [[1,8],[1,7],[2],[3],[4],[5],[6],[7]], & \mathbf{L}_2 &= [[1,8],[1],[2,8],[3],[4],[5],[6],[7]], \\ \mathbf{L}_3 &= [[6,8],[1],[2],[3],[4],[5],[6,7],[7]], & \mathbf{L}_4 &= [[6,8],[1],[2],[3],[4],[5],[6],[7,8]], \\ \mathbf{L}_5 &= [[8],[1,2],[2,8],[3],[4],[5],[6],[7]], & \mathbf{L}_6 &= [[8],[1,7],[2],[3],[4],[5],[6],[7,8]]. \end{aligned}$$

The implementation cost of $\mathbf{L}_i$ with $1 \le i \le 6$ is two XOR. Moreover, $\mathbf{L}_i^{-1}$ and $\mathbf{L}_i^3$ with $1 \le i \le 6$ can be implemented with two and five XOR, respectively. For instance, set $\mathbf{x} = [x_1, x_2, \cdots, x_8]$. Then the implementation cost of $\mathbf{L}_6^3$ can be reduced to five XOR.

$$\begin{gathered} \mathbf{L}_6^3 \cdot \mathbf{x}^{\mathrm{T}} = [x_6 + x_7 + x_8, x_5 + x_7 + x_8, x_6 + x_8, x_1 + x_7, x_2, x_3, x_4, x_5 + x_6 + x_7 + x_8]^{\mathrm{T}}, \\ u_1 = x_1 + x_7, \quad u_2 = x_6 + x_8, \quad u_3 = u_2 + x_7, \quad u_4 = u_3 + x_5, \quad u_5 = u_4 + x_6. \end{gathered}$$

Therefore, by applying $\mathbf{L}_i$'s and relation (36), $\mathbf{H}$ is implemented with $28 \times 8 + 4 \times 2 + 4 \times 2 + 4 \times 5 = 260$ XOR for 8-bit input. Furthermore, it can be checked that the minimal polynomial of $\mathbf{L}_i$'s is the irreducible polynomial `0x187`. Hence, using Proposition 1 the matrix $\mathbf{H}$ is implemented with 260 XOR over $\mathbb{F}_{2^8}/$`0x187`.

Moreover, the implementation cost of $\mathbf{E}_i^{-1}$ with $1 \leq i \leq 4$ are equal to $4m$, $4m + 4(\#\mathbf{L}^{-1})$, $4m + 4(\#\mathbf{L}^3)$ and $4m + 4(\#\mathbf{L})$, respectively. Hence, the implementation cost of $\mathbf{H}^{-1}$ is equal to $28m + 4(\#\mathbf{L}^{-1}) + 4(\#\mathbf{L}^3) + 4(\#\mathbf{L})$. Therefore, the implementation cost of $\mathbf{H}^{-1}$ and $\mathbf{H}$ are the same. A summary of results of this paper is presented in Table 2.

Table 2: A summary of results of this paper.

Iteration	Implementation Cost	Total Cost	Inverse Cost	Depth	Fig.
4×4 MDS Matrices for 4-bit input					
4 Round	8 $\text{XOR}_{\text{4-bit}}$, 4 $\mathbf{L}$	36 $\text{XOR}_{\text{1-bit}}$	36 $\text{XOR}_{\text{1-bit}}$	6	1
4 Round	8 $\text{XOR}_{\text{4-bit}}$, 3 $\mathbf{L}$	35 $\text{XOR}_{\text{1-bit}}$	35 $\text{XOR}_{\text{1-bit}}$	5	2
4×4 MDS Matrices for 8-bit input					
4 Round	8 $\text{XOR}_{\text{8-bit}}$, 4 $\mathbf{L}$	68 $\text{XOR}_{\text{1-bit}}$	68 $\text{XOR}_{\text{1-bit}}$	6	1
4 Round	8 $\text{XOR}_{\text{8-bit}}$, 3 $\mathbf{L}$	67 $\text{XOR}_{\text{1-bit}}$	67 $\text{XOR}_{\text{1-bit}}$	5	2
6×6 MDS Matrices for 4, 6, and 8-bit input					
6 Round	18 $\text{XOR}_{\text{4-bit}}$, 24 $\mathbf{L}$	90 $\text{XOR}_{\text{1-bit}}$	90 $\text{XOR}_{\text{1-bit}}$	9	4
6 Round	18 $\text{XOR}_{\text{6-bit}}$, 6 $\mathbf{L}$	114 $\text{XOR}_{\text{1-bit}}$	114 $\text{XOR}_{\text{1-bit}}$	8	3
6 Round	18 $\text{XOR}_{\text{8-bit}}$, 6 $\mathbf{L}$	156 $\text{XOR}_{\text{1-bit}}$	156 $\text{XOR}_{\text{1-bit}}$	8	3
8×8 Near-MDS Matrices for 4 and 8-bit input					
6 Round	24 $\text{XOR}_{\text{4-bit}}$, 20 $\mathbf{L}$	116 $\text{XOR}_{\text{1-bit}}$	116 $\text{XOR}_{\text{1-bit}}$	6	5
6 Round	24 $\text{XOR}_{\text{4-bit}}$, 12 $\mathbf{L}$	108 $\text{XOR}_{\text{1-bit}}$	108 $\text{XOR}_{\text{1-bit}}$	7	6
6 Round	24 $\text{XOR}_{\text{8-bit}}$, 20 $\mathbf{L}$	212 $\text{XOR}_{\text{1-bit}}$	212 $\text{XOR}_{\text{1-bit}}$	6	5
6 Round	24 $\text{XOR}_{\text{8-bit}}$, 12 $\mathbf{L}$	204 $\text{XOR}_{\text{1-bit}}$	204 $\text{XOR}_{\text{1-bit}}$	7	6
8×8 MDS Matrices for 8-bit input					
7 Round	28 $\text{XOR}_{\text{8-bit}}$, 16 $\mathbf{L}$	272 $\text{XOR}_{\text{1-bit}}$	272 $\text{XOR}_{\text{1-bit}}$	9	7
7 Round	28 $\text{XOR}_{\text{8-bit}}$, 20 $\mathbf{L}$	260 $\text{XOR}_{\text{1-bit}}$	260 $\text{XOR}_{\text{1-bit}}$	8	8

7 Conclusion

This paper proposed a construction heuristic method to design MDS matrices with the low hardware implementation cost using generalized Feistel structures (GFS). Feistel-based structures such as GFS are suitable choices to construct MDS matrices, since their inverses can be implemented with simplicity. First of all, by applying GFS, some types of sparse matrices, called primitive GFS matrices, are proposed. Next, using an extension of primitive GFS matrices we defined another type of sparse matrices called EGFS matrices. Then based on the EGFS matrices, 4×4, 6×6 and 8×8 MDS matrices are implemented with 67, 156 and 260 XOR for 8-bit input, respectively. In addition, we have proved the inverses of the proposed matrices can be implemented such as the proposed matrices. Moreover, we proposed two 8×8 near-MDS matrices such that the implementation cost of the given matrices are 116 and 108 XOR for 4-bit input and also the proposed matrices are implemented with 212 and 204 XOR for 8-bit input, respectively. Furthermore, the proposed 8×8 MDS and near-MDS matrices are not only with low implementation cost, but also the depth of their circuits are low. In other words, the proposed 8×8 matrices may be applied as diffusion layers in the lightweight cryptography.

References

[AF14] D. Augot and M. Finiasz. Direct construction of recursive MDS diffusion layers using shortened BCH codes. In *FSE*, volume 8540, pages 3–17. Springer, 2014. doi:10.1007/978-3-662-46706-0_1.

[Ber13] T. Berger. Construction of recursive MDS diffusion layers from Gabidulin codes. In *INDOCRYPT*, volume 8250, pages 274–285. Springer, 2013. doi:10.1007/978-3-319-03515-4_18.

[BKL16] C. Beierle, T. Kranz, and G. Leander. Lightweight multiplication in GF(2^n) with applications to MDS matrices. In *CRYPTO*, volume 9814, page 625–653. Springer, 2016. doi:10.1007/978-3-662-53018-4_23.

[BR99] M. Blaum and R. Roth. On lowest density MDS codes. *IEEE Trans. Inform. Theory*, 45(1):46–59, 1999. doi:10.1109/18.746771.

[DL18] S. Duval and G. Leurent. MDS Matrices with Lightweight Circuits. *IACR Transactions on Symmetric Cryptology*, 2017(2):48–78, 2017. doi:10.13154/tosc.v2018.i2.48-78.

[DR13] J. Daemen and V. Rijmen. *The Design of Rijndael AES-The Advanced Encryption Standard*. Springer, 2013.

[GPP11] J. Guo, T. Peyrin, and A. Poschmann. The PHOTON family of lightweight hash functions. In *CRYPTO*, volume 684, page 222–239. Springer, 2011. doi:10.1007/978-3-642-22792-9_13.

[GR13] K. Gupta and I. Ray. On constructions of involutory MDS matrices. In *AFRICACRYPT*, volume 7918, pages 43–60. Springer, 2013. doi:10.1007/978-3-642-38553-7_3.

[HJ13] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge U.P, 2013.

[JPS$^+$17] J. Jean, T. Peyrin, S. Sim, and J. Tourteaux. Optimizing Implementations of Lightweight Building Blocks. *IACR Transactions on Symmetric Cryptology*, 2017(4):130–168, 2017. doi:10.13154/tosc.v2017.i4.130-168.

[KLS+17] H. Kranz, G. Leander, K. Stoffelen, and F. Wiemer. Shorter Linear Straight-Line Programs for MDS Matrices. *IACR Transactions on Symmetric Cryptology*, 2017(4):188–211, 2017. doi:10.13154/tosc.v2017.i4.188-211.

[Kol19] L. Kolsch. XOR-Counts and Lightweight Multiplication with Fixed Elements in Binary Finite Fields. In *EUROCRYPT*, volume 11476, pages 285–312. Springer, 2019. doi:10.1007/978-3-030-17653-2_10.

[KPP+14] K. Khoo, T. Peyrin, A. Poschmann, and H. Yap. FOAM: Searching for Hardware-Optimal SPN Structures and Components with a Fair Comparison. In *CHES*, volume 8731, pages 433–450. Springer, 2014. doi:10.1007/978-3-662-44709-3_24.

[LS16] M. Liu and S. Sim. Lightweight MDS Generalized Circulant Matrices. In *FSE*, volume 9783, pages 101–120. Springer, 2016. doi:10.1007/978-3-662-52993-5_6.

[LSL+19] S. Li, S. Sun, C. Li, Z. Wei, and L. Hu. Constructing Low-latency Involutory MDS Matrices with Lightweight Circuits. *IACR Transactions on Symmetric Cryptology*, 2019(1):84–117, 2019. doi:10.13154/tosc.v2019.i1.84-117.

[LW16] Y. Li and M. Wang. On the Construction of Lightweight Circulant Involutory MDS Matrices. In *FSE*, volume 9783, pages 121–139. Springer, 2016. doi:10.1007/978-3-662-52993-5_7.

[LW17] C. Li and Q. Wang. Design of Lightweight Linear Diffusion Layers from Near-Mds Matrices. *IACR Transactions on Symmetric Cryptology*, 2017(1):129–155, 2017. doi:10.13154/tosc.v2017.i1.129-155.

[PSA+18] M. Pehlivanoglu, M. Sakall, S. Akleylek, N. Duru, and V. Rijmen. Generalisation of Hadamard matrix to generate involutory MDS matrices for lightweight cryptography. *IET Information Security*, 12(4):348–355, 2018. doi:10.1049/iet-ifs.2017.0156.

[SDM+12] M. Sajadieh, M. Dakhilalian, H. Mala, and P. Sepehrdad. Efficient diffusion layers for block ciphers and hash functions. In *FSE*, volume 7549, pages 385–401. Springer, 2012. doi:10.1007/978-3-642-34047-5_22.

[Shi11] K. Shibutani. On the Diffusion of Generalized Feistel Structures Regarding Differential and Linear Cryptanalysis. In *SAC*, volume 6544, pages 211–228. Springer, 2011. doi:10.1007/978-3-642-19574-7_15.

[SKO+15] S. M. Sim, K. Khoo, F. Oggier, and T. Peyrin. Lightweight MDS Involution Matrices. In *FSE*, volume 9054, pages 471–493. Springer, 2015. doi:10.1007/978-3-662-48116-5_23.

[SS16] S. Sarkar and H. Syed. Lightweight Diffusion Layer: Importance of Toeplitz Matrices. In *FSE*, volume 2016, pages 95–113. Springer, 2016. doi:10.13154/tosc.v2016.i1.95-113.

[TTK+18] D. Toh, J. Teo, K. Khoo, and S. Sim. Lightweight MDS Serial-Type Matrices with Minimal Fixed XOR Count. In *AFRICACRYPT*, volume 10831, pages 51–71. Springer, 2018. doi:10.1007/978-3-319-89339-6_4.

[WWW12] S. Wu, M. Wang, and W. Wu. Recursive Diffusion Layers for (Lightweight) Block Ciphers and Hash Functions. In *SAC*, volume 7707, pages 355–371. Springer, 2012. doi:10.1007/978-3-642-35999-6_23.

[YMT97] A. Youssef, S. Mister, and S. Tavares. On the design of linear transformations for substitution permutation encryption networks. In *SAC*, pages 40–48, 1997.

Appendix A

Case	$\mathbf{p}_1$	$\mathbf{p}_2$	Order
1	$\{1,2,3,4\}$	$\{2,1,4,3\}$	non
2	$\{1,2,3,4\}$	$\{2,3,4,1\}$	8
3	$\{1,2,3,4\}$	$\{2,4,1,3\}$	8
4	$\{1,2,3,4\}$	$\{3,1,4,2\}$	8
5	$\{1,2,3,4\}$	$\{3,4,1,2\}$	non
6	$\{1,2,3,4\}$	$\{3,4,2,1\}$	8
7	$\{1,2,3,4\}$	$\{4,1,2,3\}$	8
8	$\{1,2,3,4\}$	$\{4,3,1,2\}$	8
9	$\{1,2,3,4\}$	$\{4,3,2,1\}$	non
10	$\{1,2,4,3\}$	$\{2,1,3,4\}$	non
11	$\{1,2,4,3\}$	$\{2,3,1,4\}$	7
12	$\{1,2,4,3\}$	$\{2,4,3,1\}$	7
13	$\{1,2,4,3\}$	$\{3,1,2,4\}$	7
14	$\{1,2,4,3\}$	$\{3,4,1,2\}$	6
15	$\{1,2,4,3\}$	$\{3,4,2,1\}$	7
16	$\{1,2,4,3\}$	$\{4,1,3,2\}$	7
17	$\{1,2,4,3\}$	$\{4,3,1,2\}$	7
18	$\{1,2,4,3\}$	$\{4,3,2,1\}$	6
19	$\{1,3,2,4\}$	$\{2,1,4,3\}$	6
20	$\{1,3,2,4\}$	$\{2,4,1,3\}$	7
21	$\{1,3,2,4\}$	$\{2,4,3,1\}$	7
22	$\{1,3,2,4\}$	$\{3,1,4,2\}$	7
23	$\{1,3,2,4\}$	$\{3,2,4,1\}$	7
24	$\{1,3,2,4\}$	$\{3,4,1,2\}$	6
25	$\{1,3,2,4\}$	$\{4,1,3,2\}$	7
26	$\{1,3,2,4\}$	$\{4,2,1,3\}$	7
27	$\{1,3,2,4\}$	$\{4,2,3,1\}$	non
28	$\{1,3,4,2\}$	$\{2,1,3,4\}$	7
29	$\{1,3,4,2\}$	$\{2,4,1,3\}$	6
30	$\{1,3,4,2\}$	$\{2,4,3,1\}$	6
31	$\{1,3,4,2\}$	$\{3,1,2,4\}$	6
32	$\{1,3,4,2\}$	$\{3,2,1,4\}$	7
33	$\{1,3,4,2\}$	$\{3,4,2,1\}$	6
34	$\{1,3,4,2\}$	$\{4,1,2,3\}$	6
35	$\{1,3,4,2\}$	$\{4,2,1,3\}$	6
36	$\{1,3,4,2\}$	$\{4,2,3,1\}$	7
37	$\{1,4,2,3\}$	$\{2,1,3,4\}$	7
38	$\{1,4,2,3\}$	$\{2,3,1,4\}$	6
39	$\{1,4,2,3\}$	$\{2,3,4,1\}$	6
40	$\{1,4,2,3\}$	$\{3,1,4,2\}$	6
41	$\{1,4,2,3\}$	$\{3,2,1,4\}$	7
42	$\{1,4,2,3\}$	$\{3,2,4,1\}$	6
43	$\{1,4,2,3\}$	$\{4,1,3,2\}$	6
44	$\{1,4,2,3\}$	$\{4,2,3,1\}$	7
45	$\{1,4,2,3\}$	$\{4,3,1,2\}$	6
46	$\{1,4,3,2\}$	$\{2,1,4,3\}$	6
47	$\{1,4,3,2\}$	$\{2,3,1,4\}$	7
48	$\{1,4,3,2\}$	$\{2,3,4,1\}$	7
49	$\{1,4,3,2\}$	$\{3,1,2,4\}$	7
50	$\{1,4,3,2\}$	$\{3,2,1,4\}$	non
51	$\{1,4,3,2\}$	$\{3,2,4,1\}$	7
52	$\{1,4,3,2\}$	$\{4,1,2,3\}$	7
53	$\{1,4,3,2\}$	$\{4,2,1,3\}$	7
54	$\{1,4,3,2\}$	$\{4,3,2,1\}$	6
55	$\{2,1,3,4\}$	$\{1,2,4,3\}$	non
56	$\{2,1,3,4\}$	$\{1,3,4,2\}$	7
57	$\{2,1,3,4\}$	$\{1,4,2,3\}$	7
58	$\{2,1,3,4\}$	$\{3,2,4,1\}$	7
59	$\{2,1,3,4\}$	$\{3,4,1,2\}$	6
60	$\{2,1,3,4\}$	$\{3,4,2,1\}$	7
61	$\{2,1,3,4\}$	$\{4,2,1,3\}$	7
62	$\{2,1,3,4\}$	$\{4,3,1,2\}$	7
63	$\{2,1,3,4\}$	$\{4,3,2,1\}$	6
64	$\{2,1,4,3\}$	$\{1,2,3,4\}$	non
65	$\{2,1,4,3\}$	$\{1,3,2,4\}$	6
66	$\{2,1,4,3\}$	$\{1,4,3,2\}$	6
67	$\{2,1,4,3\}$	$\{3,2,1,4\}$	6
68	$\{2,1,4,3\}$	$\{3,4,1,2\}$	non
69	$\{2,1,4,3\}$	$\{3,4,2,1\}$	8
70	$\{2,1,4,3\}$	$\{4,2,3,1\}$	6
71	$\{2,1,4,3\}$	$\{4,3,1,2\}$	8
72	$\{2,1,4,3\}$	$\{4,3,2,1\}$	non
73	$\{2,3,1,4\}$	$\{1,2,4,3\}$	7
74	$\{2,3,1,4\}$	$\{1,4,2,3\}$	6
75	$\{2,3,1,4\}$	$\{1,4,3,2\}$	7
76	$\{2,3,1,4\}$	$\{3,1,4,2\}$	6
77	$\{2,3,1,4\}$	$\{3,2,4,1\}$	6
78	$\{2,3,1,4\}$	$\{3,4,2,1\}$	6
79	$\{2,3,1,4\}$	$\{4,1,2,3\}$	6
80	$\{2,3,1,4\}$	$\{4,1,3,2\}$	6
81	$\{2,3,1,4\}$	$\{4,2,3,1\}$	7
82	$\{2,3,4,1\}$	$\{1,2,3,4\}$	8
83	$\{2,3,4,1\}$	$\{1,4,2,3\}$	7
84	$\{2,3,4,1\}$	$\{1,4,3,2\}$	7
85	$\{2,3,4,1\}$	$\{3,1,2,4\}$	7
86	$\{2,3,4,1\}$	$\{3,2,1,4\}$	7
87	$\{2,3,4,1\}$	$\{3,4,1,2\}$	8
88	$\{2,3,4,1\}$	$\{4,1,2,3\}$	non
89	$\{2,3,4,1\}$	$\{4,1,3,2\}$	7
90	$\{2,3,4,1\}$	$\{4,2,1,3\}$	7
91	$\{2,4,1,3\}$	$\{1,2,3,4\}$	8
92	$\{2,4,1,3\}$	$\{1,3,2,4\}$	7
93	$\{2,4,1,3\}$	$\{1,3,4,2\}$	7
94	$\{2,4,1,3\}$	$\{3,1,2,4\}$	7
95	$\{2,4,1,3\}$	$\{3,1,4,2\}$	non
96	$\{2,4,1,3\}$	$\{3,2,4,1\}$	7
97	$\{2,4,1,3\}$	$\{4,1,3,2\}$	7
98	$\{2,4,1,3\}$	$\{4,2,3,1\}$	7
99	$\{2,4,1,3\}$	$\{4,3,2,1\}$	8
100	$\{2,4,3,1\}$	$\{1,2,4,3\}$	7
101	$\{2,4,3,1\}$	$\{1,3,2,4\}$	7
102	$\{2,4,3,1\}$	$\{1,3,4,2\}$	6
103	$\{2,4,3,1\}$	$\{3,1,2,4\}$	6
104	$\{2,4,3,1\}$	$\{3,1,4,2\}$	6
105	$\{2,4,3,1\}$	$\{3,2,1,4\}$	7
106	$\{2,4,3,1\}$	$\{4,1,2,3\}$	6
107	$\{2,4,3,1\}$	$\{4,2,1,3\}$	6
108	$\{2,4,3,1\}$	$\{4,3,1,2\}$	6

Case	$\mathbf{p}_1$	$\mathbf{p}_2$	Order
109	$\{3,1,2,4\}$	$\{1,2,4,3\}$	7
110	$\{3,1,2,4\}$	$\{1,3,4,2\}$	6
111	$\{3,1,2,4\}$	$\{1,4,3,2\}$	7
112	$\{3,1,2,4\}$	$\{2,3,4,1\}$	6
113	$\{3,1,2,4\}$	$\{2,4,1,3\}$	6
114	$\{3,1,2,4\}$	$\{2,4,3,1\}$	6
115	$\{3,1,2,4\}$	$\{4,2,1,3\}$	6
116	$\{3,1,2,4\}$	$\{4,2,3,1\}$	7
117	$\{3,1,2,4\}$	$\{4,3,1,2\}$	6
118	$\{3,1,4,2\}$	$\{1,2,3,4\}$	8
119	$\{3,1,4,2\}$	$\{1,3,2,4\}$	7
120	$\{3,1,4,2\}$	$\{1,4,2,3\}$	7
121	$\{3,1,4,2\}$	$\{2,3,1,4\}$	7
122	$\{3,1,4,2\}$	$\{2,4,1,3\}$	non
123	$\{3,1,4,2\}$	$\{2,4,3,1\}$	7
124	$\{3,1,4,2\}$	$\{4,2,1,3\}$	7
125	$\{3,1,4,2\}$	$\{4,2,3,1\}$	7
126	$\{3,1,4,2\}$	$\{4,3,2,1\}$	8
127	$\{3,2,1,4\}$	$\{1,3,4,2\}$	7
128	$\{3,2,1,4\}$	$\{1,4,2,3\}$	7
129	$\{3,2,1,4\}$	$\{1,4,3,2\}$	non
130	$\{3,2,1,4\}$	$\{2,1,4,3\}$	6
131	$\{3,2,1,4\}$	$\{2,3,4,1\}$	7
132	$\{3,2,1,4\}$	$\{2,4,3,1\}$	7
133	$\{3,2,1,4\}$	$\{4,1,2,3\}$	7
134	$\{3,2,1,4\}$	$\{4,1,3,2\}$	7
135	$\{3,2,1,4\}$	$\{4,3,2,1\}$	6
136	$\{3,2,4,1\}$	$\{1,3,2,4\}$	7
137	$\{3,2,4,1\}$	$\{1,4,2,3\}$	6
138	$\{3,2,4,1\}$	$\{1,4,3,2\}$	7
139	$\{3,2,4,1\}$	$\{2,1,3,4\}$	7
140	$\{3,2,4,1\}$	$\{2,3,1,4\}$	6
141	$\{3,2,4,1\}$	$\{2,4,1,3\}$	6
142	$\{3,2,4,1\}$	$\{4,1,2,3\}$	6
143	$\{3,2,4,1\}$	$\{4,1,3,2\}$	6
144	$\{3,2,4,1\}$	$\{4,3,1,2\}$	6
145	$\{3,4,1,2\}$	$\{1,2,3,4\}$	non
146	$\{3,4,1,2\}$	$\{1,2,4,3\}$	6
147	$\{3,4,1,2\}$	$\{1,3,2,4\}$	6
148	$\{3,4,1,2\}$	$\{2,1,3,4\}$	6
149	$\{3,4,1,2\}$	$\{2,1,4,3\}$	non
150	$\{3,4,1,2\}$	$\{2,3,4,1\}$	8
151	$\{3,4,1,2\}$	$\{4,1,2,3\}$	8
152	$\{3,4,1,2\}$	$\{4,2,3,1\}$	6
153	$\{3,4,1,2\}$	$\{4,3,2,1\}$	non
154	$\{3,4,2,1\}$	$\{1,2,3,4\}$	8
155	$\{3,4,2,1\}$	$\{1,2,4,3\}$	7
156	$\{3,4,2,1\}$	$\{1,3,4,2\}$	7
157	$\{3,4,2,1\}$	$\{2,1,3,4\}$	7
158	$\{3,4,2,1\}$	$\{2,1,4,3\}$	8
159	$\{3,4,2,1\}$	$\{2,3,1,4\}$	7
160	$\{3,4,2,1\}$	$\{4,1,3,2\}$	7
161	$\{3,4,2,1\}$	$\{4,2,1,3\}$	7
162	$\{3,4,2,1\}$	$\{4,3,1,2\}$	non
163	$\{4,1,2,3\}$	$\{1,2,3,4\}$	8
164	$\{4,1,2,3\}$	$\{1,3,4,2\}$	7
165	$\{4,1,2,3\}$	$\{1,4,3,2\}$	7
166	$\{4,1,2,3\}$	$\{2,3,1,4\}$	7
167	$\{4,1,2,3\}$	$\{2,3,4,1\}$	non
168	$\{4,1,2,3\}$	$\{2,4,3,1\}$	7
169	$\{4,1,2,3\}$	$\{3,2,1,4\}$	7
170	$\{4,1,2,3\}$	$\{3,2,4,1\}$	7
171	$\{4,1,2,3\}$	$\{3,4,1,2\}$	8
172	$\{4,1,3,2\}$	$\{1,2,4,3\}$	7
173	$\{4,1,3,2\}$	$\{1,3,2,4\}$	7
174	$\{4,1,3,2\}$	$\{1,4,2,3\}$	6
175	$\{4,1,3,2\}$	$\{2,3,1,4\}$	6
176	$\{4,1,3,2\}$	$\{2,3,4,1\}$	6
177	$\{4,1,3,2\}$	$\{2,4,1,3\}$	6
178	$\{4,1,3,2\}$	$\{3,2,1,4\}$	7
179	$\{4,1,3,2\}$	$\{3,2,4,1\}$	6
180	$\{4,1,3,2\}$	$\{3,4,2,1\}$	6
181	$\{4,2,1,3\}$	$\{1,3,2,4\}$	7
182	$\{4,2,1,3\}$	$\{1,3,4,2\}$	6
183	$\{4,2,1,3\}$	$\{1,4,3,2\}$	7
184	$\{4,2,1,3\}$	$\{2,1,3,4\}$	7
185	$\{4,2,1,3\}$	$\{2,3,4,1\}$	6
186	$\{4,2,1,3\}$	$\{2,4,3,1\}$	6
187	$\{4,2,1,3\}$	$\{3,1,2,4\}$	6
188	$\{4,2,1,3\}$	$\{3,1,4,2\}$	6
189	$\{4,2,1,3\}$	$\{3,4,2,1\}$	6
190	$\{4,2,3,1\}$	$\{1,3,2,4\}$	non
191	$\{4,2,3,1\}$	$\{1,3,4,2\}$	7
192	$\{4,2,3,1\}$	$\{1,4,2,3\}$	7
193	$\{4,2,3,1\}$	$\{2,1,4,3\}$	6
194	$\{4,2,3,1\}$	$\{2,3,1,4\}$	7
195	$\{4,2,3,1\}$	$\{2,4,1,3\}$	7
196	$\{4,2,3,1\}$	$\{3,1,2,4\}$	7
197	$\{4,2,3,1\}$	$\{3,1,4,2\}$	7
198	$\{4,2,3,1\}$	$\{3,4,1,2\}$	6
199	$\{4,3,1,2\}$	$\{1,2,3,4\}$	8
200	$\{4,3,1,2\}$	$\{1,2,4,3\}$	7
201	$\{4,3,1,2\}$	$\{1,4,2,3\}$	7
202	$\{4,3,1,2\}$	$\{2,1,3,4\}$	7
203	$\{4,3,1,2\}$	$\{2,1,4,3\}$	8
204	$\{4,3,1,2\}$	$\{2,4,3,1\}$	7
205	$\{4,3,1,2\}$	$\{3,1,2,4\}$	7
206	$\{4,3,1,2\}$	$\{3,2,4,1\}$	7
207	$\{4,3,1,2\}$	$\{3,4,2,1\}$	non
208	$\{4,3,2,1\}$	$\{1,2,3,4\}$	non
209	$\{4,3,2,1\}$	$\{1,2,4,3\}$	6
210	$\{4,3,2,1\}$	$\{1,4,3,2\}$	6
211	$\{4,3,2,1\}$	$\{2,1,3,4\}$	6
212	$\{4,3,2,1\}$	$\{2,1,4,3\}$	non
213	$\{4,3,2,1\}$	$\{2,4,1,3\}$	8
214	$\{4,3,2,1\}$	$\{3,1,4,2\}$	8
215	$\{4,3,2,1\}$	$\{3,2,1,4\}$	6
216	$\{4,3,2,1\}$	$\{3,4,1,2\}$	non

Appendix B

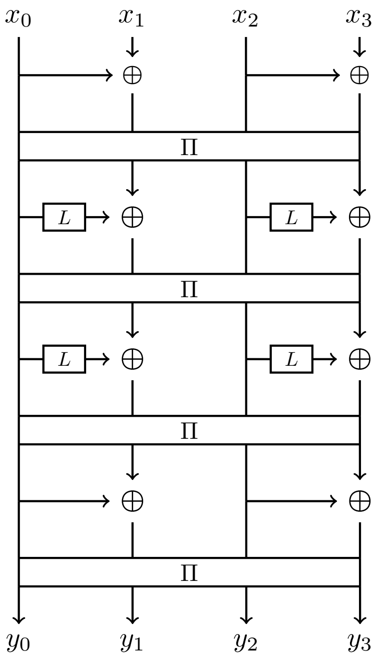

Figure 1: 4×4 MDS matrix with depth 6: $\boldsymbol{\Pi(0,1,2,3) = (3,0,1,2)}$

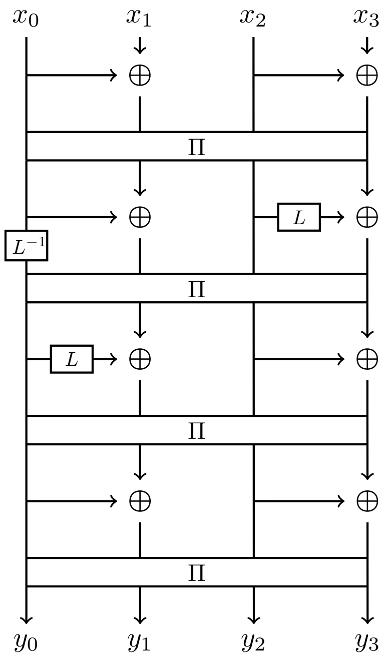

Figure 2: 4×4 MDS matrix with depth 5: $\boldsymbol{\Pi(0,1,2,3) = (3,0,1,2)}$

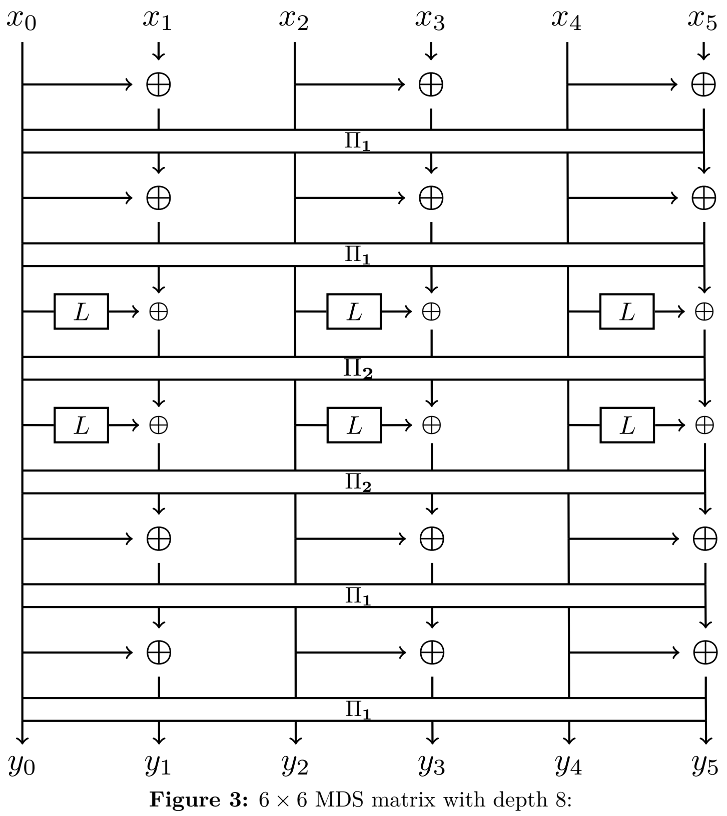

Figure 3: 6×6 MDS matrix with depth 8:
$\Pi_1(0,1,2,3,4,5,6) = (3,0,5,2,1,4)$
$\Pi_2(0,1,2,3,4,5,6) = (3,4,1,2,5,0)$

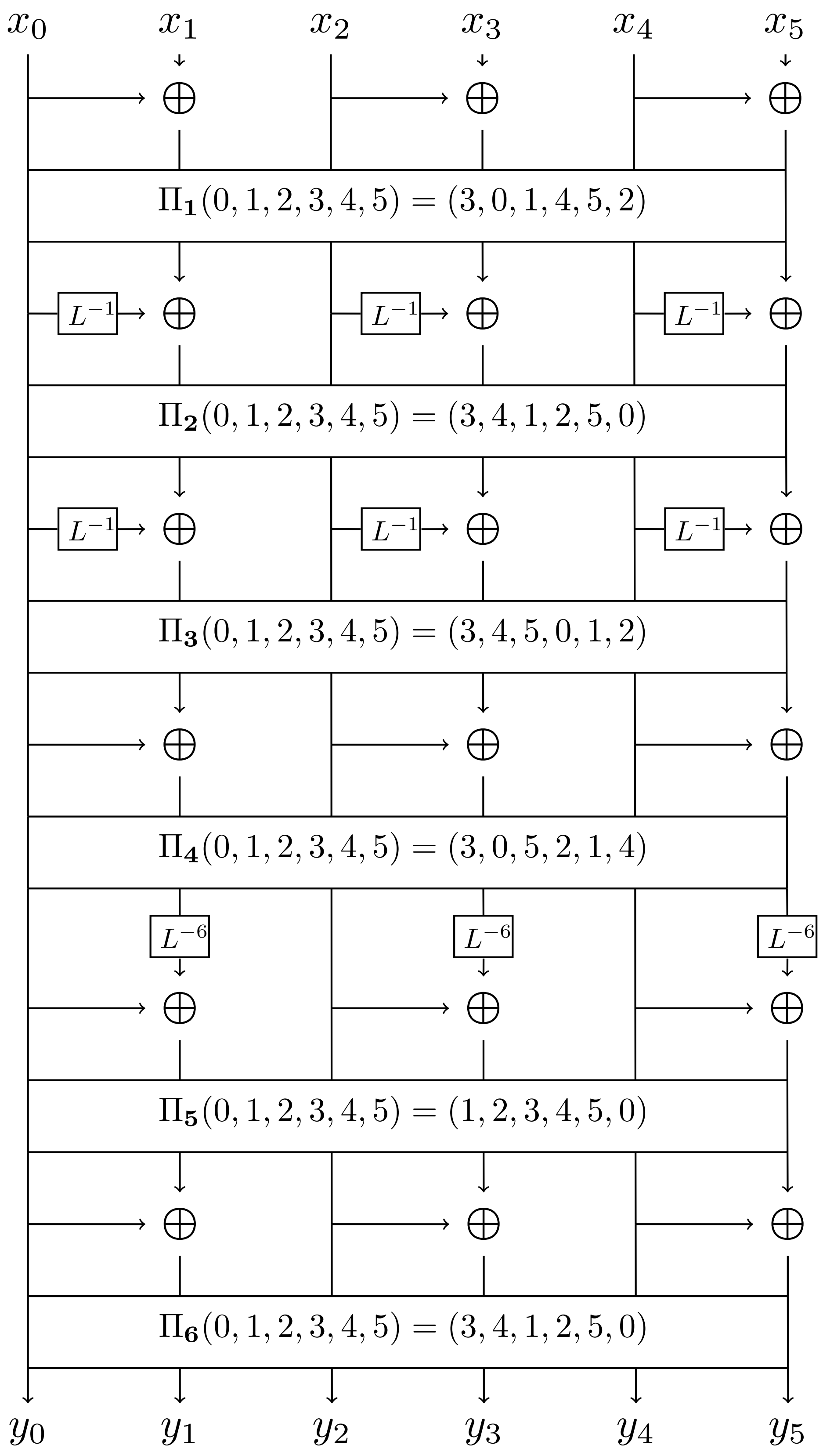
x_0
x_1
x_2
x_3
x_4
x_5
$\Pi_{\mathbf{1}}(0,1,2,3,4,5) = (3,0,1,4,5,2)$
L^{-1}
$\Pi_{\mathbf{2}}(0,1,2,3,4,5) = (3,4,1,2,5,0)$
$\Pi_{\mathbf{3}}(0,1,2,3,4,5) = (3,4,5,0,1,2)$
$\Pi_{\mathbf{4}}(0,1,2,3,4,5) = (3,0,5,2,1,4)$
L^{-6}
$\Pi_{\mathbf{5}}(0,1,2,3,4,5) = (1,2,3,4,5,0)$
$\Pi_{\mathbf{6}}(0,1,2,3,4,5) = (3,4,1,2,5,0)$
y_0
y_1
y_2
y_3
y_4
y_5

Figure 4: 6×6 MDS matrix with depth 9

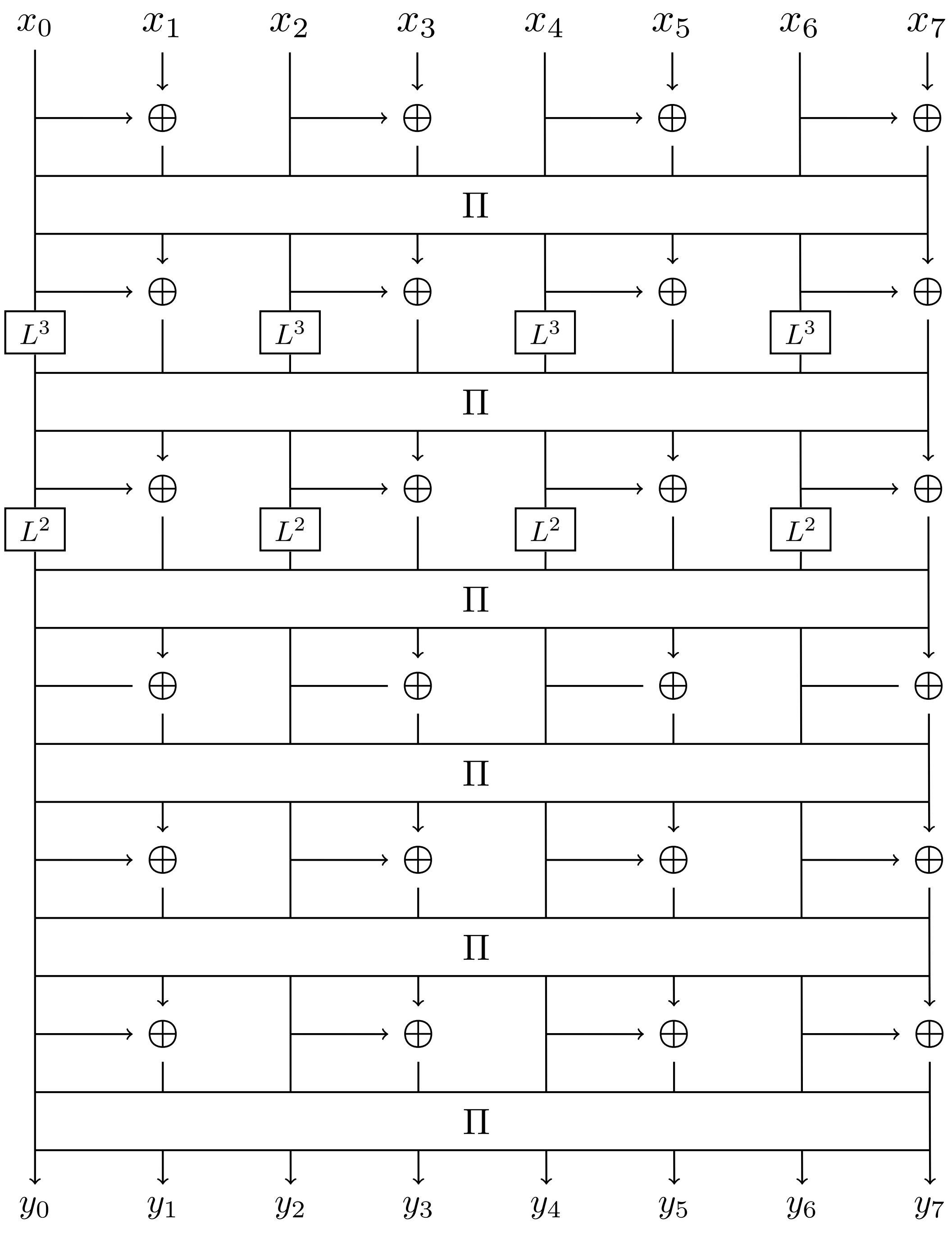

Figure 5: 8×8 Near-MDS matrix with depth **6**: $\mathbf{\Pi(0,1,2,3,4,5,6,7) = (5,6,3,4,1,2,7,0)}$

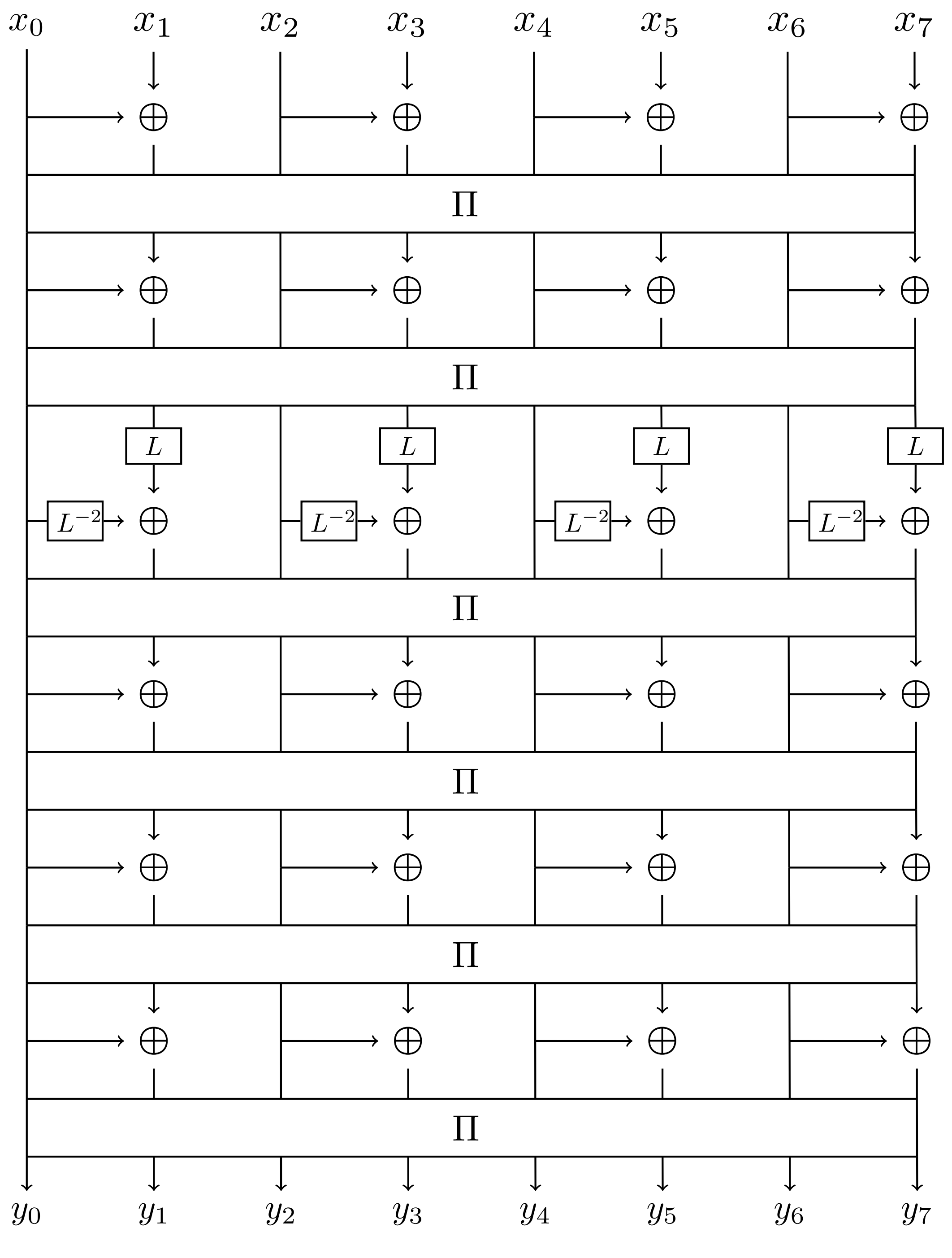

Figure 6: 8×8 Near-MDS matrix with depth $\mathbf{7}$:
$\mathbf{\Pi(0, 1, 2, 3, 4, 5, 6, 7) = (5, 6, 3, 4, 1, 2, 7, 0)}$

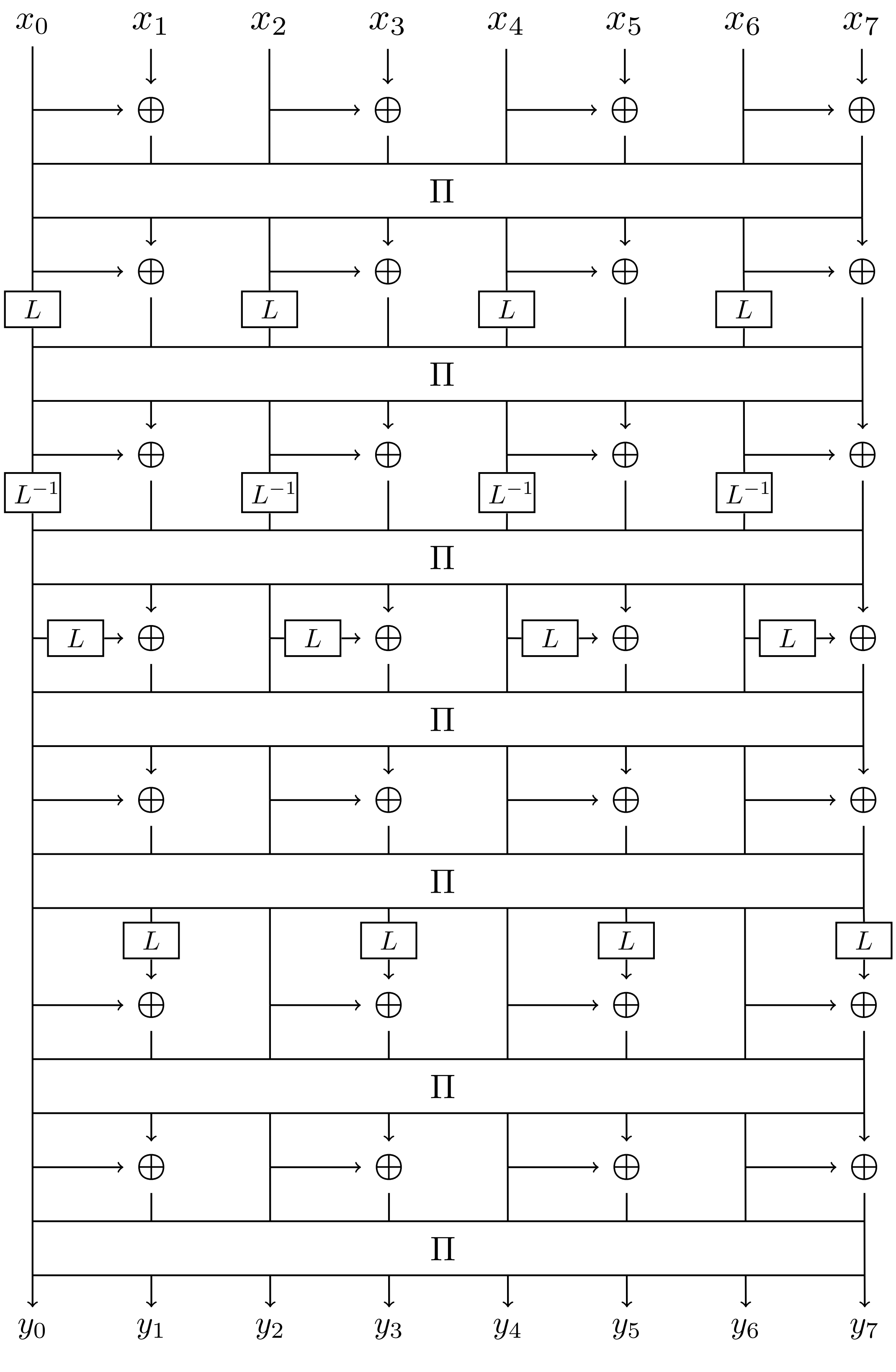

Figure 7: 8×8 MDS matrix with depth $\mathbf{9}$: $\mathbf{\Pi(0,1,2,3,4,5,6,7) = (5,6,3,4,1,2,7,0)}$

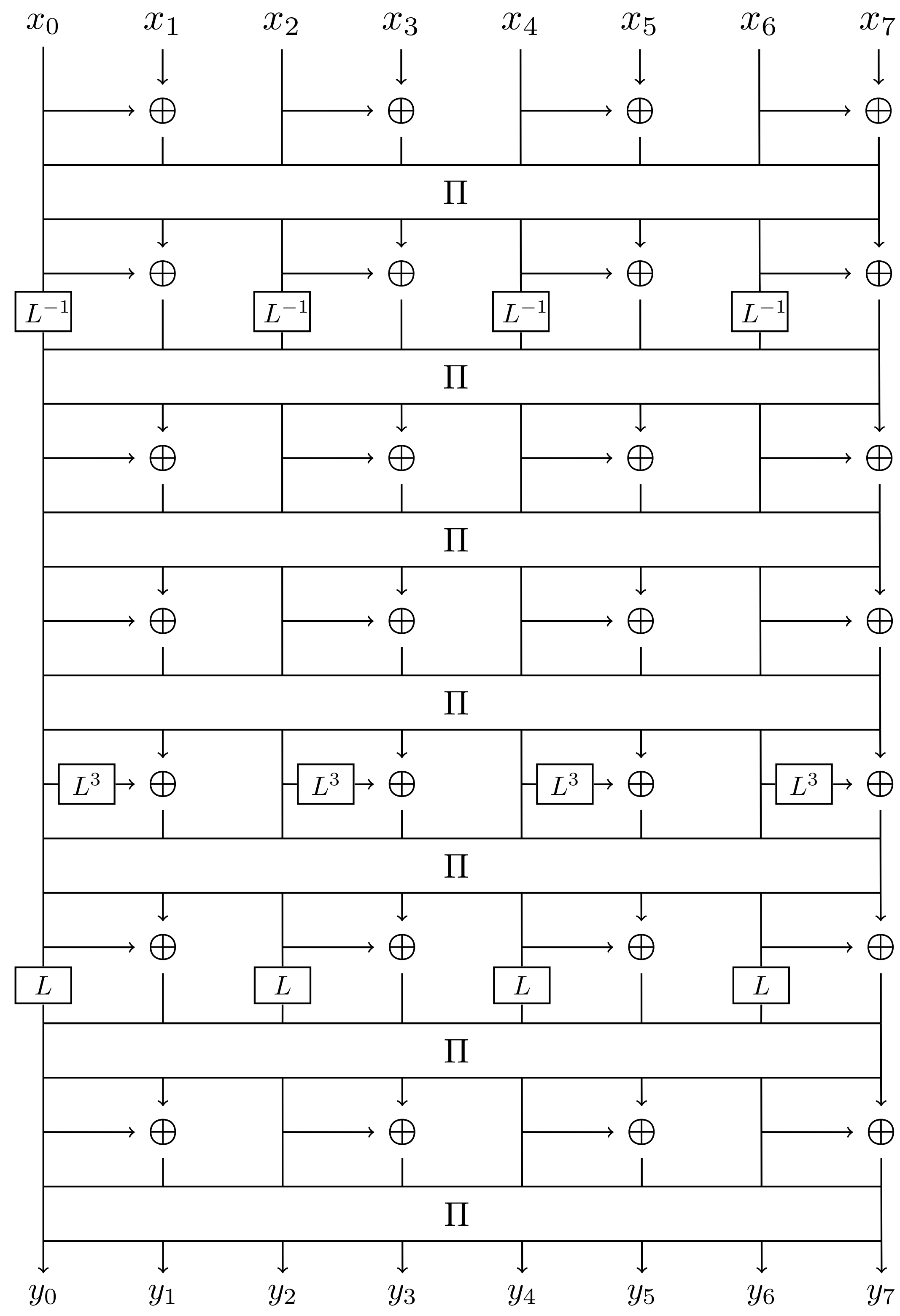

Figure 8: 8×8 MDS matrix with depth $\mathbf{8}$: $\mathbf{\Pi(0,1,2,3,4,5,6,7) = (5,6,3,4,1,2,7,0)}$

Appendix C

{0x2, 0x3, 0x7, 0xB, 0xD, 0x13, 0x19, 0x1F, 0x25, 0x29, 0x2F, 0x37, 0x3B, 0x3D, 0x43, 0x49,
0x61, 0x57, 0x5B, 0x67, 0x6D, 0x73, 0x75, 0x83, 0x89, 0x91, 0xC1, 0x8F, 0x9D, 0xA7, 0xAB,
0xB9, 0xCB, 0xD3, 0xD5, 0xE5, 0xF1, 0xBF, 0xEF, 0xF7, 0xFD, 0x11B, 0x11D, 0x12B, 0x12D,
0x139, 0x14D, 0x163, 0x165, 0x169, 0x171, 0x18B, 0x18D, 0x1A3, 0x1A9, 0x1B1, 0x1C3,
0x13F, 0x15F, 0x177, 0x17B, 0x19F, 0x1BD, 0x1CF, 0x1D7, 0x1DD, 0x1E7, 0x1F3, 0x1F5,
0x1F9, 0x203, 0x221, 0x301, 0x217, 0x21B, 0x233, 0x24B, 0x259, 0x265, 0x269, 0x287,
0x295, 0x299, 0x2A3, 0x2A5, 0x2D1, 0x313, 0x315, 0x323, 0x331, 0x349, 0x361, 0x385,
0x3A1, 0x25F, 0x26F, 0x277, 0x27D, 0x2AF, 0x2B7, 0x2BD, 0x2CF, 0x2DB, 0x2F5, 0x2F9,
0x31F, 0x33B, 0x34F, 0x35B, 0x36D, 0x373, 0x38F, 0x3B5, 0x3B9, 0x3C7, 0x3CB, 0x3D5,
0x3E3, 0x3E9, 0x37F, 0x3FB, 0x409, 0x481, 0x40F, 0x41B, 0x41D, 0x427, 0x42D, 0x435,
0x447, 0x499, 0x4C9, 0x50B, 0x50D, 0x523, 0x531, 0x543, 0x561, 0x5A1, 0x615, 0x631,
0x685, 0x689, 0x6C1, 0x711, 0x721, 0x781, 0x46F, 0x4AF, 0x4D7, 0x51F, 0x567, 0x56B,
0x597, 0x59B, 0x5AB, 0x5B9, 0x5C7, 0x637, 0x6A7, 0x6AD, 0x6B5, 0x6D3, 0x70F, 0x739,
0x747, 0x74D, 0x759, 0x763, 0x793, 0x4FF, 0x5FB, 0x67F, 0x6BF, 0x6DF, 0x6FD, 0x77B,
0x77D, 0x7DB, 0x7FF, 0x805, 0xA01, 0x817, 0x863, 0x871, 0x8D1, 0x90D, 0x913, 0x929,
0x945, 0x949, 0xA29, 0xA61, 0xC0B, 0xD03, 0xE21, 0x87B, 0x89F, 0xAB5, 0xAE3, 0xB33,
0xB4B, 0xB65, 0xB95, 0xBC9, 0xC73, 0xC75, 0xCCD, 0xCD3, 0xD27, 0xD2D, 0xDC9, 0xDE1,
0xE2B, 0xE33, 0xE4B, 0xE55, 0xEC9, 0xF19, 0xF83, 0xF91, 0x97F, 0xAEF, 0xBAF, 0xBBD,
0xBED, 0xCF7, 0xD9F, 0xDBB, 0xE7B, 0xE9F, 0xECF, 0xEDD, 0xF79, 0xFA7, 0xFAD, 0xFD3,
0x1161, 0x1431, 0x1823, 0x1883, 0x18A1, 0x1905, 0x103F, 0x10ED, 0x123D, 0x144F, 0x16A5,
0x1715, 0x186D, 0x192D, 0x1935, 0x1A2B, 0x1B19, 0x1CC9, 0x1DC1, 0x147F, 0x1B2F, 0x1B57,
0x1B8F, 0x1D2F, 0x1D67, 0x1DB3, 0x1E8F, 0x1EB9, 0x1ED3, 0x1EF1, 0x1F39, 0x1FC3, 0x1FC9,
0x17BF, 0x1FAF, 0x1FBD, 0x1FFF, 0x201B, 0x2189, 0x2461, 0x24C1, 0x2603, 0x21E3, 0x290F,
0x2AA9, 0x2E15, 0x320F, 0x324B, 0x3293, 0x3299, 0x344D, 0x3817, 0x3C23, 0x22EF, 0x274F,
0x2A7D, 0x2D37, 0x2EAB, 0x32B7, 0x32DB, 0x36D3, 0x3967, 0x3B63, 0x3F91, 0x2F5F, 0x2FF5,
0x37DD, 0x3BE7, 0x41D9, 0x5271, 0x612D, 0x6549, 0x6CC1, 0x7119, 0x6C3D, 0x6DA3, 0x7FD7,
0x8B3B, 0x94F5, 0x974D, 0xB0F3, 0xB84F, 0xE28F, 0xF40F, 0xB797, 0x1594B, 0x26B0B}

Appendix D

{0x2, 0x3, 0x7, 0xB, 0xD, 0x13, 0x19, 0x1F, 0x25, 0x29, 0x2F, 0x37, 0x3B, 0x3D, 0x43, 0x49,
0x57, 0x5B, 0x61, 0x67, 0x6D, 0x73, 0x75, 0x83, 0x89, 0x8F, 0x91, 0x9D, 0xA7, 0xAB, 0xB9,
0xBF, 0xC1, 0xCB, 0xD3, 0xD5, 0xE5, 0xEF, 0xF1, 0xF7, 0xFD, 0x11B, 0x11D, 0x12B, 0x12D,
0x139, 0x13F, 0x14D, 0x15F, 0x163, 0x165, 0x169, 0x171, 0x177, 0x17B, 0x187, 0x18B,
0x18D, 0x19F, 0x1A3, 0x1A9, 0x1B1, 0x1BD, 0x1C3, 0x1CF, 0x1D7, 0x1DD, 0x1F3, 0x1F5,
0x1F9, 0x203, 0x21B, 0x221, 0x22D, 0x233, 0x24B, 0x25F, 0x265, 0x269, 0x277, 0x27D,
0x287, 0x295, 0x299, 0x2A3, 0x2A5, 0x2B7, 0x2CF, 0x2DB, 0x2F5, 0x2F9, 0x313, 0x315,
0x31F, 0x331, 0x33B, 0x34F, 0x35B, 0x361, 0x36B, 0x36D, 0x373, 0x37F, 0x385, 0x3A1,
0x3B9, 0x3C7, 0x3CB, 0x3CD, 0x3D5, 0x3D9, 0x3E3, 0x3E9, 0x3FB, 0x409, 0x41B, 0x435,
0x447, 0x453, 0x465, 0x46F, 0x481, 0x4A9, 0x4C5, 0x4E7, 0x4F3, 0x4FF, 0x523, 0x53D,
0x543, 0x557, 0x58F, 0x59B, 0x5A1, 0x5AB, 0x5C7, 0x5F7, 0x615, 0x623, 0x631, 0x637,
0x64F, 0x65B, 0x679, 0x67F, 0x685, 0x689, 0x6A7, 0x6AD, 0x6B5, 0x6C1, 0x6CD, 0x711,
0x717, 0x71D, 0x721, 0x72B, 0x735, 0x755, 0x759, 0x77B, 0x77D, 0x781, 0x787, 0x7B1,
0x7C5, 0x7DB, 0x7F3, 0x7F9, 0x7FF, 0x805, 0x82D, 0x88D, 0x8A9, 0x8C3, 0x8CF, 0x8D1,
0x8E7, 0x93B, 0x949, 0x951, 0x973, 0x975, 0x9E5, 0x9EF, 0xA07, 0xA13, 0xA15, 0xA6D,
0xA79, 0xA7F, 0xAD5, 0xADF, 0xB11, 0xB33, 0xB3F, 0xB87, 0xB95, 0xBAF, 0xBBD, 0xBC9,
0xC0D, 0xC97, 0xCBF, 0xCC7, 0xD0F, 0xD1D, 0xD27, 0xD93, 0xDBB, 0xDC9, 0xDD7, 0xE27,
0xE2B, 0xE7B, 0xEA3, 0xEC9, 0xECF, 0xEF9, 0xF0B, 0xF19, 0xF6B, 0x1069, 0x1077,
0x10D1, 0x11EF, 0x1219, 0x13A9, 0x14B5, 0x154D, 0x1593, 0x15BB, 0x15C5, 0x16E7,
0x17FB, 0x1823, 0x1879, 0x197B, 0x19CF, 0x19F9, 0x1A69, 0x1BFD, 0x1C03, 0x1C27,
0x1CBB, 0x1CED, 0x1E3D, 0x1F11, 0x1F1B, 0x1FAF, 0x1FC3, 0x1FE1, 0x227F, 0x232B,
0x2429, 0x25BD, 0x2B2F, 0x2B97, 0x2F5F, 0x329F, 0x33E5, 0x3499, 0x3A61, 0x3FB5,
0x49E1, 0x4A17, 0x549F, 0x5585, 0x6A6B, 0x6D05, 0x7327, 0x74C7, 0x7BB9, 0x7CA3,
0xA6C7, 0xA7D1, 0xAF2F, 0xB08D, 0xB24F, 0xB909, 0xBB15, 0xD91B, 0xE05F, 0xEB97, 0x14ABF}