On The Efficient Construction of Lightweight MDS Matrices

Lijing Zhou, Licheng Wang and Yiru Sun

State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing,
China.379739494@qq.com

Abstract. Firstly, by analyzing non-singular matrices with few XORs in the matrix polynomial residue ring, we present an efficient method for building lightweight maximum distance separable (MDS) matrices with elements chosen from a fixed matrix polynomial residue ring. Compared with that constructions of previous methods usually cost several days or several weeks, our new method only cost within several minutes. With this method, many different types of lightweight MDS matrices can be quickly constructed. This method has a significance for researching the lightweight MDS matrix. Surprisingly, it did not receive much attention previously. We give 5 matrix templates which are suitable to construct lightweight MDS matrices. Secondly, we investigate the existence of involutory MDS matrix for several matrix templates. Besides, we present an efficient necessary-and-sufficient condition for judging whether a Hadamard matrix is involutory. With this condition, an extremely efficient algorithm for constructing lightweight Hadamard involutory MDS matrices is given. By doing experiments, we get a lot of new Hadamard involutory MDS matrices with much fewer XORs than previously optimal results. Thirdly, in theory, we discuss reasons about why our methods work very efficiently. Finally, we prove a series of propositions about the parity of XORs of element-matrix and entirety-matrix.

Keywords: MDS matrix, XOR count, matrix polynomial residue ring, involutory matrix

1 Introduction

Background. In block cipher, the linear diffusion layer is a significant component required for the security of the cipher. The linear diffusion layer with bigger branch number can more effectively resist differential and linear cryptanalysis. The diffusion layer is often constructed by a matrix. For any $n \times n$ matrix, the maximum possible branch number of the corresponding diffusion layer is $n + 1$. Maximum distance separable (MDS) matrices can indeed reach this limitation and thus are broadly used in many ciphers like PHOTON [1], SQUARE [2], LED [3], AES

[4]. For lightweight cryptography, the cost of implementing a linear diffusion layer will influence the efficiency of cryptography largely. Therefore, constructions of lightweight MDS matrices are meaningful to design the lightweight cryptography. Moreover, from the perspective of hardware implementations, an efficient lightweight MDS matrix is extremely useful for saving logical gates. Considering that the sum of XORs [15] is the most important index for measuring the efficiency of MDS matrices, MDS matrices with fewer XORs are more efficient.

Currently, a major method of constructing lightweight MDS matrices is to use the recursive matrix. That is, we can firstly choose a special non-singular matrix, and then compose it k times to get a MDS matrix A^k, the so-called serial matrices. This method was successfully used in constructions of hash function PHOTON [1], block cipher LED [3] and authenticated encryption scheme PRIMATEs [10]. Further investigation on this method can be found in [11–15]. However, this method has a drawback: It is not suitable for low-latency implementations, since it has to run several rounds to get results.

Sim et al.[18] constructed lightweight Hadamard involutory MDS matrices over the finite field. Over the finite field, the newest lightweight circulant MDS matrices are constructed by Beierle et al. [23] at CRYPTO 2016. At FSE 2016, Li et al. [19] construct many new involutory and non-involutory lightweight MDS matrices over $GL(m, \mathbb{F}_2)$. Although Nakahara et al.[16] and Gupta et al.[17] proved that circulant MDS matrices can not be involutory over the finite field, Li et al. [19] successfully get circulant MDS matrices over $GL(m, \mathbb{F}_2)$.

Hadamard matrix, circulant matrix and Optimal matrix [27] are usually used as templates in building MDS matrices. Since the elements of these templates are repeatedly used, the searching space can be reduced obviously. Liu et al. [22] and Sim et al. [18] employed the equivalence of matrices to further reduce the searching space. Many constructions of MDS matrices over the finite field were proposed [18, 23, 20, 21]. By investigating the multiplication of special element in $GF(2^m)$, Christof et al. [23] got lightweight circulant MDS matrices over $GF(2^m)$.

Motivations. Although $GF(2^m)$ is suitable to efficiently construct MDS matrices, it is not suitable to construct the lightest results. Although $GL(m, \mathbb{F}_2)$ is suitable to construct the lightest results, the construction usually takes a large amount of time. For finding a efficient method to construct MDS matrices with as few XORs as possible, we discover that MDS matrices can be constructed over the matrix polynomial residue ring.

Contribution. We investigate the feasibility of building lightweight MDS matrices over the matrix polynomial residue ring. By analyzing non-singular matrices with few XORs in the matrix polynomial residue ring, we propose an efficient method to construct lightweight MDS matrices. Compared with that constructions of previous methods usually cost several days or several weeks, our new method only use within several minutes. It has a significance for researching the lightweight MDS matrix. To our best knowledge, it is the first time to construct MDS matrices over the matrix polynomial residue ring. With the matrix polynomial residue ring, MDS matrices ont only have favourable XORs, but also construction is very efficient. Our contributions are summarized as follows

- We search each $T \in GL(m, \mathbb{F}_2)$ that satisfies $\#T{=}1$ and $T + I$ non-singular. For each T, we find its minimum polynomial. We analyze the distribution of the minimum polynomials and the distribution of XOR count for all elements in the matrix polynomial residue ring. Based on these work, we recommend 5 matrix templates which are suitable to construct non-involutory lightweight MDS matrices.
- For constructing lightweight MDS matrices, an efficient algorithm is given. Results are shown as follows
  (1) When elements are 4×4 binary matrices, 288 4×4 MDS matrices with 10 XORs are built within 2 minutes.
  (2) When elements are 8×8 binary matrices, 40320 MDS matrices with 10 XORs are built within 2 minutes.
  (3) When elements are 16×16 binary matrices, one 4×4 MDS matrix with 10 XORs is found within 1 minute.
- We extend some results about the existence of involutory MDS matrix as follows
  (1) Over the matrix polynomial residue ring, $n \times n (n \geq 3)$ circulant MDS matrices can not be involutory.
  (2) Over $GL(m, \mathbb{F}_2)$, $n \times n (n \geq 2)$ special MDS matrices as mentioned in Section 5 can not be involutory.
  (3) We give an efficient necessary-and-sufficient condition for judging whether a Hadamard matrix is involutory. With this condition, another extremely efficient algorithm for constructing lightweight Hadamard involutory MDS matrices is proposed. With this algorithm, over 8×8 matrix over $\mathbb{F}_2$, we only use 1 minute and 4 second to construct 80640 4×4 Hadamard involutory MDS matrices with 20 XORs, which are much lighter than previous optimal results.
- In the computation efficiency of matrix polynomial residue ring, search space and theory, we discuss reasons about why our methods work very efficiently.

– We prove a series of propositions about the parity of XOR count of element-matrix and entirety-matrix.

ROADMAP. In Sec. 2, necessary preliminaries are presented. In Sec. 3, we investigate the distributions of the minimum polynomial and distributions of XOR count on matrix polynomial residue rings, and then introduce 5 matrix templates. In Sec. 4, we design an algorithm for efficiently constructing lightweight non-involutory MDS matrices. In Sec. 5, we investigate the involutory MDS matrix. In Sec. 6, we discuss reasons about why our methods work very efficiently. In Sec. 7, we prove a series of properties about the parity of XOR count. A short conclusion is given in Sect. 8.

2 Preliminaries

In this section, we introduce the basic definitions and theorems about the lightweight MDS matrix.

2.1 MDS Matrices

Let R be a ring with identity and $x \in R^m$. The *bundle weight* of x is defined as the number of nonzero entries of x and is expressed by $\omega_b(x)$. Let M be a $n \times n$ matrix over R. The *branch number* of M is the minimum number of nonzero components in the input vector v and output vector $u = M \cdot v$ as we search all nonzero $v \in R^n$. I.e. the branch number of M is $B_M = min_{v \neq 0}\{\omega_b(v) + \omega_b(Mv)\}$, and $B_M \leq n + 1$. A *maximum distance separable* (MDS) $n \times n$ matrix is a matrix that has the maximum branch number $n+1$. $GL(n, \mathbb{F}_2)$ denotes the set of all non-singular $n \times n$ matrices over $\mathbb{F}_2$.

Eevry linear diffusion layer is a linear map and can be represented by a matrix as follow

$$L = \begin{pmatrix} L_{1,1} & L_{1,2} & \cdots & L_{1,n} \\ L_{2,1} & L_{2,2} & \cdots & L_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ L_{n,1} & L_{n,2} & \cdots & L_{n,n} \end{pmatrix}$$

where $L_{i,j}$ $(1 \leq i, j \leq n)$ is an $m \times m$ non-singular matrix over $\mathbb{F}_2$. Denote $M(n, m)$ be all matrices, which are $n \times n$ matrices over $GL(m, \mathbb{F}_2)$. For

$X = (x_1, x_2, ..., x_n)^T \in (\mathbb{F}_2{}^m_2)^n$,

$$L(X) = \begin{pmatrix} L_{1,1} & L_{1,2} & \cdots & L_{1,n} \\ L_{2,1} & L_{2,2} & \cdots & L_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ L_{n,1} & L_{n,2} & \cdots & L_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} L_{1,i}(x_i) \\ \sum_{i=1}^{n} L_{2,i}(x_i) \\ \vdots \\ \sum_{i=1}^{n} L_{n,i}(x_i) \end{pmatrix},$$

where $L_{i,j}(x_k) = L_{i,j} \cdot x_k$, for $1 \leq i, j \leq n, 1 \leq k \leq n$.

Theorem 1. *[19] Let L be a matrix, then L is MDS if and only if all square sub-matrices of L are of full rank.*

2.2 XOR Count

Let $a, b \in \mathbb{F}_2$, $a + b$ is called a bit XOR operation. Let $A \in GL(m, \mathbb{F}_2)$, $x = (x_1, x_2, ..., x_m)^T \in \mathbb{F}_2{}^m$, $\#A$ denotes the number of XOR operations required to evaluate Ax directly. Let $\omega(A)$ be the number of 1 in A. $\#A$ denotes the *XOR count* of A and $\#A = \omega(A) - m$. For $L \in M(n, m)$, $\#(L)$ denotes *the sum of XORs* of L and $\#(L) = \sum_{i,j=1}^{n} \#(L_{ij})$. For instance, let $x = (a, b, c, d)^T \in \mathbb{F}_2{}^4_2$, and the following matrix with 4 XOR count.

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

$$Ax = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} d \\ c + d \\ b + c + d \\ a + c \end{pmatrix}.$$

For $A \in GL(m, \mathbb{F}_2)$, a simplified representation of A is given by extracting the non-zero positions in each of row of A. For example, [3,2,4,[1,3]] is the representation of the following matrix with 1 XOR count.

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

3 Matrix Polynomial Residue Ring

In this section, we analyze distributions of the minimum polynomials and distributions of XOR count on matrix polynomial residue rings. We introduce 5 matrix templates of matrices which are suitable to construct lightweight MDS matrices.

Let T be an $n \times n$ matrix over $\mathbb{F}_2$, and $f(x)$ be the minimum polynomial of T. Let the order of $f(x)$ be k, then $k \leq n$. $\mathbb{F}_2[T] \cong \mathbb{F}_2[x]/(f(x))$ since T satisfies $f(T) = 0$, where $\mathbb{F}_2[T]$ denotes the matrix polynomial residue ring generated by T. Therefore matrix computations is equivalent to polynomial computations in $\mathbb{F}_2[T]$.

For example, let $B, C \in \mathbb{F}_2[T]$,

$$\begin{aligned} B &= b_{k-1}T^{k-1} + \cdots + b_1 T + b_0 I, \\ C &= c_{k-1}T^{k-1} + \cdots + c_1 T + c_0 I, \\ b(x) &= b_{k-1}x^{k-1} + \cdots + b_1 x + b_0, \\ c(x) &= c_{k-1}x^{k-1} + \cdots + c_1 x + c_0. \end{aligned}$$

Then $B + C = b(x) + c(x)|_{x=T}$, $BC = b(x)c(x)|_{x=T}$.

3.1 Analyzing the 4×4 Matrix Polynomial Residue Ring

In this subsection, we analyze distributions of the minimum polynomial and distributions of XOR count on 4×4 matrix polynomial residue rings.

We search every T satisfying $T \in GL(4, \mathbb{F}_2)$, $\#T$=1 and $I + T$ non-singular. The number of T is 72. Let $f(x)$ be the minimum polynomial of T, $b(x) \in \mathbb{F}_2[x]/(f(x))$. We search all $b(x)$ satisfying $1 \leq \#b(T) \leq 3$. Results are as follows

(1) $f(x)$ must be one of following polynomials

$$x^4 + x + 1, x^4 + x^2 + 1, x^4 + x^3 + 1.$$

(2) For $\#b(T)$=1, $b(x)$ must be one of following polynomials

$$x, x^3 + 1, x^3 + x, x^3 + x^2.$$

(3) For $\#b(T)$=2, $b(x)$ must be one of following polynomials

$$x^2, x^2 + 1, x^2 + x, x^3.$$

(4) For $\#b(T)$=3, $b(x)$ must be one of following polynomials

$$x + 1, x^2, x^3, x^3 + x^2 + 1.$$

Distributions of $f(x)$ and $b(x)$ are described in Table 1, where

– MP_4 denotes the set of the minimum polynomials of 4×4 matrices with 1 XOR,
– Xi_4 means the set of $b(x)$ satisfying $\#b(T) = i$,
– Num indicates the number of T satisfying $\#b(T)$ =1, 2 or 3.

Table 1: Distributions of Polynomials on The 4×4 Binary Matrix Polynomial Residue Ring

Minimum Polynomial		
	$f(x)$	Num
MP_4	x^4+x+1	24
	x^4+x^2+1	24
	x^4+x^3+1	24

2 XORs		
	$b(x)$	Num
$X2_4$	x^2	48
	x^2+1	24
	x^2+x	24
	x^3	24

1 XOR		
	$b(x)$	Num
$X1_4$	x	72
	x^3+1	24
	x^3+x	24
	x^3+x^2	24

3 XORs		
	$b(x)$	Num
$X3_4$	$x+1$	24
	x^2	24
	x^3	24
	x^3+x^2+1	24

3.2 Analyzing the 8×8 Matrix Polynomial Residue Ring

In this subsection, we analyze the distributions of the minimum polynomial and XOR count on 8×8 matrix polynomial residue rings.

We search all matrix T satisfying $T \in GL(8, \mathbb{F}_2)$, $\#T$=1 and $I+T$ nonsingular. The number of T is 282240. Let $f(x)$ be the minimum polynomial of T, $b(x) \in \mathbb{F}_2[x]/(f(x))$. We search every T to find every $f(x)$ and all $b(x)$, where $b(x)$ satisfies $1 \leq \#b(T) \leq 3$. Search results are as follows

(1) $f(x)$ must be one of following polynomials

$$x^8+x+1, x^8+x^2+1, x^8+x^3+1, x^8+x^4+1, x^8+x^5+1, x^8+x^6+1, x^8+x^7+1.$$

(2) For $\#b(T)$=1, $b(x)$ must be one of following polynomials

$$x, x^7+1, x^7+x, x^7+x^2, x^7+x^3, x^7+x^4, x^7+x^5.$$

(3) For $\#b(T)$=2, $b(x)$ must be one of following polynomials

$$x^2, x^6+1, x^6+x, x^6+x^2, x^6+x^3, x^6+x^4.$$

(4) For $\#b(T)$=3, $b(x)$ must be one of following polynomials

$$x^3, x^5+1, x^5+x, x^5+x^2, x^5+x^3, x^7+x^6+1$$

Distributions of $f(x)$ and $b(x)$ are described in Table 2, where

- MP_8 denotes the set of the minimum polynomials of 8×8 matrices with 1 XOR,
- Xi_8 means the set of $b(x)$ satisfying $\#b(T) = i$,
- Num indicates the number of T satisfying $\#b(T)$ =1, 2 or 3.

Table 2: Distributions of Polynomials on on The 8×8 Binary Matrix Polynomial Residue Ring

Minimum Polynomial		
	$f(x)$	Num
MP_8	x^8+x+1	40320
	x^8+x^2+1	40320
	x^8+x^3+1	40320
	x^8+x^4+1	40320
	x^8+x^5+1	40320
	x^8+x^6+1	40320
	x^8+x^7+1	40320

2 XORs		
	$b(x)$	Num
$X2_8$	x^2	241920
	x^6+1	40320
	x^6+x	40320
	x^6+x^2	40320
	x^6+x^3	40320
	x^6+x^4	40320
	x^6+x^5	40320

1 XOR		
	$b(x)$	Num
$X1_8$	x	282240
	x^7+1	40320
	x^7+x	40320
	x^7+x^2	40320
	x^7+x^3	40320
	x^7+x^4	40320
	x^7+x^5	40320
	x^7+x^6	40320

3 XORs		
	b(x)	Num
$X3_8$	x^2	40320
	x^3	201600
	x^5+1	40320
	x^5+x	40320
	x^5+x^2	40320
	x^5+x^3	40320
	x^5+x^4	40320
	x^7+x^6+1	40320

Remark 1. Let $T \in GL(m, \mathbb{F}_2)$, $\#T$=1. $T + I$ is non-singular and $f(x)$ is the minimum polynomial of T. Advantages of the matrix polynomial residue ring for constructing the lightest MDS matrices are as follows

(I) *Over the matrix polynomial residue ring, the non-singular matrix with 1 XOR can be used to be an entry of a MDS matrix. But in the matrix representation of finite field F_{2^8}, there does not exist any matrix with 1 XOR count.*

By searching all non-zero and non-identity elements in all matrix representations of F_{2^8}, we discover that the XOR count must be greater than 1. Fortunately, over matrix polynomial residue rings, if we want to use a matrix T with 1 XOR to be an entry in a MDS matrix, we just need to let T be an entry of MDS matrix, and other entries are chosen from $\mathbb{F}_2[T]$. In this way, T is successfully used to construct MDS matrix, and this MDS matrix is over $\mathbb{F}_2[T]$.

(II) *Computation of the matrix polynomial residue ring is more efficient than $GL(m, \mathbb{F}_2)$.*

Since the matrix polynomial residue ring is isomorphic to polynomial residue ring. Therefore computation of the matrix polynomial residue ring is more efficient than $GL(m, \mathbb{F}_2)$.

3.3 5 Templates of Matrix

In this subsection, we introduce 5 matrix templates used in constructing algorithms.

Let $L_1, L_2 \in M(n, m)$, if L_1 can be transformed to become L_2 by exchanging rows or columns, then L_1 is equivalent to L_2. For constructing the lightest MDS matrix, the lightest MDS matrix should have as many identity matrices to be entries as possible since identity matrix over $\mathbb{F}_2$ has 0 XOR count. However, any sub-matrix of order 2, in MDS matrix, must not be $\begin{pmatrix} I & I \\ I & I \end{pmatrix}$.Otherwise, such matrix is not MDS.

In our algorithms, we only use 5 matrix templates as follows

$$S_1 = \begin{pmatrix} & I & I & I \\ I & I & & \\ I & & I & \\ I & & & I \end{pmatrix}, S_2 = \begin{pmatrix} & I & I & I \\ I & I & & \\ I & & I & \\ & & & I \end{pmatrix}, S_3 = \begin{pmatrix} & I & I & I \\ I & I & & \\ I & & I & \\ & & I & \end{pmatrix},$$

$$S_4 = \begin{pmatrix} & I & I & I \\ I & I & & \\ I & & I & \\ I & & & \end{pmatrix}, S_5 = \begin{pmatrix} I & I & & \\ & I & I & \\ & & I & I \\ I & & & I \end{pmatrix},$$

where I is the identity matrix over $\mathbb{F}_2$ and others can be any other non-singular matrices over $\mathbb{F}_2$.

According to [27], in a MDS matrix of degree n, there exist at most $3(n-1)$ identity matrices to be entries. This matrix is called the *Optimal matrix*. For example, the following matrix is an Optimal matrix.

$$\begin{pmatrix} A_{1,1} & I & I & \cdots & I \\ I & I & A_{2,3} & \cdots & A_{2,n} \\ I & A_{3,2} & I & \cdots & A_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I & A_{n,2} & A_{n,3} & \cdots & I \end{pmatrix}$$

In previous papers, circulant matrix, Hadamard matrix and Optimal matrix are usually used to construct lightweight MDS matrices. They are as follows

$$Circ(I,I,A,B) = \begin{pmatrix} I & I & A & B \\ B & I & I & A \\ A & B & I & I \\ I & A & B & I \end{pmatrix}, Had(I,A,B,C) = \begin{pmatrix} I & A & B & C \\ A & I & C & B \\ B & C & I & A \\ C & B & A & I \end{pmatrix},$$

$$Optimal\ matrix = \begin{pmatrix} A & I & I & I \\ I & I & A & B \\ I & B & I & A \\ I & A & B & I \end{pmatrix}.$$

It should be pointed that $Circ(I,I,A,B)$ is the particular case of S_5 and the Optimal matrix is the particular case of S_1.

Generally, when we construct the lightest MDS matrices, if A, which is not identity matrix, is an entry in one of 5 equivalence classes, then $A+I$ should be non-singular. The reason is that there must exist a sub-determinant of order 2 like $\begin{vmatrix} I & I \\ I & A \end{vmatrix} = A+I$ in such matrix. Because of the requirement of MDS, $A+I$ should be non-singular.

4 Lightweight Non-involutory MDS Matrices

In this section, we investigate the efficient algorithm for constructing lightweight MDS matrices.

4.1 Entries Expression

In this subsection, we investigate entries expression in the constructing algorithm.

Entries of MDS matrices are chosen from the $m \times m$ matrix polynomial residue ring, m=4, 8 or 16. For example, like Optimal matrix

$$Optimal\ Matrix = \begin{pmatrix} A & I & I & I \\ I & I & A & B \\ I & B & I & A \\ I & A & B & I \end{pmatrix}.$$

In such Optimal matrix, T is a non-singular matrix, $\#T$=1, and $f(x)$ is the minimum polynomial of T. $A, B \in \mathbb{F}_2[T]$, $a(T) = A$, $b(T) = B$ and $a(x), b(x) \in \mathbb{F}_2[x]/(f(x))$. In our algorithms, x replaces T, 1 replaces I, $a(x)$ replaces A and $b(x)$ replaces B. Therefore this Optimal matrix is replaced as the following matrix

$$\begin{pmatrix} a(x) & 1 & 1 & 1 \\ 1 & 1 & a(x) & b(x) \\ 1 & b(x) & 1 & a(x) \\ 1 & a(x) & b(x) & 1 \end{pmatrix}.$$

4.2 MDS Judgment

In this subsection, we investigate how to judge whether a matrix is MDS in our constructing algorithms.

Necessary and sufficient condition of MDS According to Theorem 1, $L \in M(n, m)$, L is MDS if and only if all square sub-matrices of L are full rank. That a sub-matrix is full rank is equivalent to that the corresponding sub-determinant is non-singular since entries are $m \times m$ matrices over $\mathbb{F}_2$. Therefore the necessary-and-sufficient condition of MDS can also be described as follow

Theorem 2. *[19] Let $L \in M(n, m)$, L is MDS if and only if all sub-determinants of L are non-singular.*

Above theorem is the method to judge whether matrix is MDS in our algorithms.

Sub-determinant calculation For instance, because entries are expressed as polynomials in our algorithms, so a matrix can be expressed as follow

$$\begin{pmatrix} x & 1 & 1 & 1 \\ 1 & 1 & x & x^2+1 \\ 1 & x^2+1 & 1 & x \\ 1 & x & x^2+1 & 1 \end{pmatrix}.$$

Sub-determinants are calculated according to the determinant complete expansion formula. In above matrix, a sub-determinant of order 3 can be calculated as follow

$$\begin{vmatrix} x & 1 & 1 \\ 1 & 1 & x \\ 1 & x^2+1 & 1 \end{vmatrix} = x + x + (x^2+1) + 1 + (x^4+x^2) + 1 = x^4 + 1.$$

Then let T be substituted into $x^4 + 1$ to get $T^4 + I$.

Finally, judge whether $T^4 + I$ is non-singular. $T^4 + I$ is non-singular if and only if $x^4 + 1$ is relatively prime to $f(x)$, where $f(x)$ is the minimum polynomial of T. We just need to find the greatest common factor of $x^4 + 1$ and $f(x)$. If the greatest common factor equals to 1, then $T + I$ is non-singular. Otherwise, it is singular.

4.3 Algorithm for Constructing the Lightest MDS matrices

For constructing lightweight 4×4 MDS matrices over the $m \times m (m = 4, 8$ or $16)$ matrix polynomial residue ring, Algorithm 1 is given below. MP_m is the set of the minimum polynomials. Xi_m is the set of $b(x)$ satisfying that, for some matrix T, $\#b(T) = i$. S_i is the template of MDS matrices mentioned in Section 3.

The platform for running Algorithm 1 is specified as follows: Intel i5-5300 CPU with 2.30GHz, 4GB memory, Windows 10 OS. The programming language is the C language. By running Algorithm 1, results are organized as follows:

1. Entries are 4×4 matrices over $\mathbb{F}_2$. We use 1 minute 42 seconds to construct 288 MDS matrices with 10 XORs by using S_1 matrix template. It takes about 13 minutes to verify that there does not exist MDS matrices with 10 XORs in S_2, S_4 or S_5. An example is given as follow:

Algorithm 1 Construct Lightweight MDS matrices

for Search all T, $\#T$=1, T and $T+I$ are non-singular **do**
  Find the minimum polynomial of T in MP_m.
  Find polynomials $b_1(x), \cdots, b_k(x)$ in $X1_m$, $X2_m$ and $X3_m$, which satisfy that $\#b_t(T) \leq 3$.
  for i from 1 to 5 **do**
    for In S_i, every place, which is not 1, searches in $\{b_1(x), \cdots, b_k(x)\}$ **do**
      if Matrix is MDS **then**
        Record this MDS matrix and its sum of XORs.
      end if
    end for
  end for
end for

Example 1. m=4. $T = [[1,2],3,4,1]$. The following matrix is a MDS matrix with 10 XORs.

$$\begin{pmatrix} T^2+T & I & I & I \\ I & I & T & T^2+T \\ I & T^2+T & I & T^3+T^2 \\ I & T & T^3+T^2 & I \end{pmatrix}$$

2. Entries are 4×4 matrices over $\mathbb{F}_2$. We use 1 minute 16 seconds to construct 40320 MDS matrices with 10 XORs. An example is given as follow:
   Example 2. m=8. $T = [[2,4],3,4,5,6,7,8,1]$. The following matrix is a MDS matrix with 10 XORs.

$$\begin{pmatrix} T^2 & I & I & I \\ I & I & T & T^2 \\ I & T & I & T^7+T \\ I & T^7+T & T^2 & I \end{pmatrix}$$

3. Over 16×16 matrix polynomial residue rings, we use about 1 minute to construct Circulant MDS matrix with 12 XORs and Optimal MDS matrix with 10 XORs. Let $T \in GL(16, \mathbb{F}_2)$ and $T = [[1,2],3,4,5,6,7,8,9,10,11,12,13,14,15,16,1]$. The minimum polynomial of T is $x^{16} + x^{15} + 1$. Two examples are given below:
   Example 3. L_1 is a circulant MDS matrix with 12 XORs.

$$L_1 = \begin{pmatrix} I & I & T & T^{14}+T^{13} \\ T^{14}+T^{13} & I & I & T \\ T & T^{14}+T^{13} & I & I \\ I & T & T^{14}+T^{13} & I \end{pmatrix}$$

Example 4. L_2 is an Optimal MDS matrix with 10 XORs.

$$L_2 = \begin{pmatrix} T & I & I & I \\ I & I & T & T^{14}+T^{13} \\ I & T^{14}+T^{13} & I & T \\ I & T & T^{14}+T^{13} & I \end{pmatrix}$$

Details of constructions of Algorithm 1 is shown at Table 3.

Table 3: Number of Lightweight Non-involutory MDS Matrices and Running Time

Matrix type	Element	Sum of XORs	Number	Running time
$Circ(I, I, A, B)$	$\mathbb{F}_2[T_{4\times4}]$	12	96	00:00:01
$Had(I, A, B, C)$	$\mathbb{F}_2[T_{4\times4}]$	20	288	00:00:04
Optimal	$\mathbb{F}_2[T_{4\times4}]$	13	48	00:00:01
S_1	$\mathbb{F}_2[T_{4\times4}]$	10	288	00:01:42
S_3	$\mathbb{F}_2[T_{4\times4}]$	10	48	00:05:05
$Circ(I, I, A, B)$	$\mathbb{F}_2[T_{8\times8}]$	12	96	00:01:27
$Had(I, A, B, C)$	$\mathbb{F}_2[T_{8\times8}]$	20	241920	00:07:00
Optimal	$\mathbb{F}_2[T_{8\times8}]$	10	40320	00:01:16
S_1	$\mathbb{F}_2[T_{8\times8}]$	10	1128960	14:00:00

5 Lightweight Involutory MDS Matrices

In this section, we investigate the existence of involutory MDS matrix for some matrix structures. Then we prove an efficient necessary-and-sufficient condition for judging whether a Hadamard matrix is involutory. With this condition, we propose an extremely efficient algorithm to construct lightweight involutory MDS matrices.

5.1 Existence of Involutory MDS Matrices

In this subsection, we investigate the existence of involutory MDS matrix for some matrix structures.

Theorem 3. *Let L be a $n \times n (n \geq 2)$ MDS matrix over $GL(m, \mathbb{F}_2)$ as the following matrix. In L, the number of identity matrices is greater than or equal to $2n-1$. Then L is not involutory.*

$$L = \begin{pmatrix} A_{1,1} & \cdots & A_{1,i-1} & I & A_{1,i+1} & \cdots & A_{1,n} \\ \vdots & & \vdots & \vdots & \vdots & I & \vdots \\ A_{i-1,1} & \cdots & A_{i-1,i-1} & I & A_{i-1,i+1} & \cdots & A_{i-1,n} \\ I & \cdots & I & A_{i,i} & I & \cdots & I \\ A_{i+1,1} & \cdots & A_{i+1,i-1} & I & A_{i+1,i+1} & \cdots & A_{i+1,n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ A_{n,1} & \cdots & A_{n,i-1} & I & A_{n,i+1} & \cdots & A_{n,n} \end{pmatrix} \tag{1}$$

Proof. For proving that L is not involutory, we assume that L is involutory. According to this assumption, if we find a contradiction in following process of proof, then L is not involutory. Now we prove this theorem.

When $n = 2k$, k=1,2,3$\cdots$. Then

$$L^2 = \begin{pmatrix} * \cdots & * & \cdots * \\ \vdots & \vdots & \vdots \\ * \cdots & A_{i,i}^2 + I & \cdots * \\ \vdots & \vdots & \vdots \\ * \cdots & * & \cdots * \end{pmatrix} = \begin{pmatrix} I\ 0 \cdots 0 \\ 0\ I \cdots 0 \\ \vdots\ \vdots\ \ddots\ \vdots \\ 0\ 0 \cdots I \end{pmatrix}$$

$$\Rightarrow A_{i,i}^2 = 0 \Rightarrow A_{i,i}\ is\ singular.$$

Because L is MDS, so $A_{i,i}$ is non-singular. This is a contradiction. Therefore in this case, L can not be involutory.

When $n = 2k + 1$, k=1,2,3$\cdots$. Then

$$L^2 = \begin{pmatrix} * \cdots & * & \cdots * \\ \vdots & \vdots & \vdots \\ * \cdots & A_{i,i}^2 & \cdots * \\ \vdots & \vdots & \vdots \\ * \cdots & * & \cdots * \end{pmatrix} = \begin{pmatrix} I\ 0 \cdots 0 \\ 0\ I \cdots 0 \\ \vdots\ \vdots\ \ddots\ \vdots \\ 0\ 0 \cdots I \end{pmatrix} \Rightarrow A_{i,i}^2 = I$$

$$\Rightarrow A_{i,i}^2 + I = 0 \Rightarrow (A_{i,i} + I)^2 = 0 \Rightarrow A_{i,i} + I\ is\ singular.$$

Because L is as Eq. 1, there must exist a sub-determinant like $\begin{vmatrix} I & I \\ I & A_{i,i} \end{vmatrix} = A_{i,i} + I$ in $|L|$. Becuase L is MDS, so $A_{i,i} + I$ should be non-singular. This is a contradiction. Therefore in this case, L must not be involutory.

In a word, L mentioned in this theorem is not involutory.

□

Theorem 4. *Let L be a MDS matrix of degree $2k+1 (k = 1, 2, \cdots)$ over $GL(m, \mathbb{F}_2)$ as the following matrix. Then L is not involutory.*

$$L = \begin{pmatrix} A_{1,1} & \cdots & A_{1,j-1} & I & A_{1,j+1} & \cdots & A_{1,2k+1} \\ \vdots & & \vdots & \vdots & \vdots & I & \vdots \\ A_{i-1,1} & \cdots & A_{i-1,j-1} & I & A_{i-1,j+1} & \cdots & A_{i-1,2k+1} \\ I & \cdots & I & A_{i,j} & I & \cdots & I \\ A_{i+1,1} & \cdots & A_{i+1,j-1} & I & A_{i+1,j+1} & \cdots & A_{i+1,2k+1} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ A_{2k+1,1} & \cdots & A_{2k+1,j-1} & I & A_{2k+1,j+1} & \cdots & A_{2k+1,2k+1} \end{pmatrix} \tag{2}$$

Proof. For proving that L is not involutory, we assume that L is involutory. According to this assumption, if we find a contradiction in following process of proof, then L is not involutory. Now we prove this theorem.

According to Eq. 2, then

$$L^2 = \begin{pmatrix} * & * & * & * & * \\ * & \ddots & * & * & * \\ * & * & \ddots & * & * \\ * & I & * & \ddots & * \\ * & * & * & * & * \end{pmatrix} \tag{3}$$

Where I is at ith row and jth column.

According to the assumption, L is involutory. Then

$$L^2 = \begin{pmatrix} I & 0 & \cdots & 0 \\ 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I \end{pmatrix} \tag{4}$$

According Eq. 3, at ith row and jth column, this element is I. But according to Eq. 4, at ith row and jth column, this element is 0. It is a contradiction. So the assumption is wrong. Therefore L is not involutory. □

Theorem 5. *Let $T \in GL(m, \mathbb{F}_2)$, A_1, A_2, $\cdots$, $A_n \in \mathbb{F}_2[T]$. If $Circ(A_1, A_2, ..., A_n)$ is MDS, then $Circ(A_1, A_2, ..., A_n)$ is not involutory, where $n \geq 3$.*

Proof. $L = Circ(A_1, A_2, ..., A_n)$ is a MDS matrix as the following matrix, where $A_1,\ A_2,\ \cdots,\ A_n \in \mathbb{F}_2[T]$.

$$Circ(A_1, A_2, ..., A_n) = \begin{pmatrix} A_1 & A_2 & \cdots & A_n \\ A_n & A_1 & \cdots & A_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_2 & A_3 & \cdots & A_1 \end{pmatrix}$$

For proving that $Circ(A_1, A_2, ..., A_n)$ is not involutory, we assume that $Circ(A_1, A_2, ..., A_n)$ is involutory. According to this assumption, if we find a contradiction in following process of proof, then $Circ(A_1, A_2, ..., A_n)$ is not involutory. Now we prove this theorem.

When $n = 2k + 1,\ k = 1, 2, 3 \cdots$. Then

$$L^2 = \begin{pmatrix} A_1 & \cdots & A_{k+1} & \cdots & A_{2k+1} \\ \vdots & & \vdots & & \vdots \\ * & \cdots & * & \cdots & A_{k+1} \\ \vdots & & \vdots & & \vdots \\ * & \cdots & * & \cdots & A_1 \end{pmatrix} \begin{pmatrix} A_1 & \cdots & A_{k+1} & \cdots & A_{2k+1} \\ \vdots & & \vdots & & \vdots \\ * & \cdots & * & \cdots & A_{k+1} \\ \vdots & & \vdots & & \vdots \\ * & \cdots & * & \cdots & A_1 \end{pmatrix}$$

$$= \begin{pmatrix} * & * & \cdots & A_{k+1}^2 \\ * & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & * \end{pmatrix} = \begin{pmatrix} I & 0 & \cdots & 0 \\ 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I \end{pmatrix} \Rightarrow A_{k+1}^2 = 0 \Rightarrow A_{k+1}\ is\ singular.$$

Because L is MDS, so A_{k+1} is non-singular. This is a contradiction. Therefore in this case, L can not be involutory.

When $n = 2k,\ k = 2, 3, 4 \cdots$. Then

$$L^2 = \begin{pmatrix} A_1 & \cdots & A_k & \cdots & A_{2k-1} & A_{2k} \\ \vdots & & \vdots & & \vdots & \vdots \\ * & \cdots & * & \cdots & A_k & A_{k+1} \\ \vdots & & \vdots & & \vdots & \vdots \\ * & \cdots & * & \cdots & A_1 & A_2 \\ * & \cdots & * & \cdots & A_{2k} & A_1 \end{pmatrix} \begin{pmatrix} A_1 & \cdots & A_k & \cdots & A_{2k-1} & A_{2k} \\ \vdots & & \vdots & & \vdots & \vdots \\ * & \cdots & * & \cdots & A_k & A_{k+1} \\ \vdots & & \vdots & & \vdots & \vdots \\ * & \cdots & * & \cdots & A_1 & A_2 \\ * & \cdots & * & \cdots & A_{2k} & A_1 \end{pmatrix}$$

$$= \begin{pmatrix} * & \cdots & A_k^2 + A_{2k}^2 & 0 \\ * & \cdots & * & * \\ \vdots & \cdots & \vdots & \vdots \\ * & \cdots & * & * \end{pmatrix} = \begin{pmatrix} I & 0 & \cdots & 0 \\ 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I \end{pmatrix} \Rightarrow A_k^2 + A_{2k}^2 = 0.$$

There is a 2×2 sub-matrix $\begin{pmatrix} A_k & A_{2k} \\ A_{2k} & A_k \end{pmatrix}$ in L.

$$L = \begin{pmatrix} A_1 & \cdots & A_k & \cdots & A_{2k} \\ \vdots & & \vdots & & \vdots \\ A_{k+1} & \cdots & A_{2k} & \cdots & A_k \\ \vdots & & \vdots & & \vdots \\ * & \cdots & * & \cdots & * \end{pmatrix}$$

According above discussions, $A_k^2 + A_{2k}^2 = 0$. Because L is MDS, so $\begin{vmatrix} A_k & A_{2k} \\ A_{2k} & A_k \end{vmatrix} = A_k^2 + A_{2k}^2$ should be non-singular. This is a contradiction. Therefore in this case, L can not be involutory. □

5.2 Hadamard Involutory Matrices

In this subsection, we investigate the Hadamard involutory matrix.

Theorem 6. *Let $T \in GL(m, \mathbb{F}_2)$. $f(x)$ is the minimum polynomial of T. $a_1(x)$, $a_2(x)$, $\cdots$, $a_{2^k}(x) \in \mathbb{F}_2[x]/(f(x))$. $L = Had(a_1(T)$, $a_1(T)$, $\cdots$, $a_{2^k}(T))$ is involutory if and only if*

$$(\sum_{i=1}^{2^k} a_i(x))^2 \equiv 1 \pmod{f(x)}$$

Proof. Because $T \in GL(m, \mathbb{F}_2)$ and $L = Had(a_1(T), a_1(T), \cdots, a_{2^k}(T))$ is involutory, so

$$L^2 = \begin{pmatrix} \sum_{i=1}^{2^k}(a_i(T))^2 & & & \\ & \sum_{i=1}^{2^k}(a_i(T))^2 & & \\ & & \ddots & \\ & & & \sum_{i=1}^{2^k}(a_i(T))^2 \end{pmatrix} = \begin{pmatrix} I & & & \\ & I & & \\ & & \ddots & \\ & & & I \end{pmatrix}$$

$$\Leftrightarrow \sum_{i=1}^{2^k}(a_i(x))^2 \equiv (\sum_{i=1}^{2^k} a_i(x))^2 \equiv 1 \pmod{f(x)}$$

□

Corollary 1. *Let $T \in GL(m, \mathbb{F}_2)$. $f(x)$ is the minimum polynomial of T. $a(x)$, $b(x)$ and $c(x) \in \mathbb{F}_2[x]/(f(x))$. $L = Had(I, a(T)$, $b(T)$, $c(T))$ is involutory if and only if*

$$(a(x) + b(x) + c(x))^2 \equiv 0 \pmod{f(x)}$$

Proof. According to Theorem 6, $Had(I, a(T), b(T), c(T))$ is involutory if and only if $(1 + a(x) + b(x) + c(x))^2 \equiv 1 \pmod{f(x)}$. $(1 + a(x) + b(x) + c(x))^2 \equiv 1 \pmod{f(x)} \Leftrightarrow (a(x) + b(x) + c(x))^2 \equiv 0 \pmod{f(x)}$. □

We construct lightweight Hadamard involutory MDS matrices as $Had(I, A, B, C)$. In our experiments, $A \in GL(8, \mathbb{F}_2)$, $\#A$=1, $A+I$ is non-singular. $f(x)$ is the minimum polynomial of A. $b(x)$, $c(x) \in \mathbb{F}_2[x]/(f(x))$ and $B = b(A)$, $C = c(A)$. According to above theorem, $Had(I, A, B, C)$ is involutory if and only if $(x + b(x) + c(x))^2 \equiv 0 \pmod{f(x)}$. So $x^2 \equiv (b(x) + c(x))^2 \pmod{f(x)}$. As mentioned in section 4, the minimum polynomial of A must be one of the following polynomials

$$x^8+x+1, x^8+x^2+1, x^8+x^3+1, x^8+x^4+1, x^8+x^5+1, x^8+x^6+1, x^8+x^7+1.$$

We find all $g(x)$ satisfying $g^2(x) \equiv x^2 \pmod{f(x)}$, where $f(x)$ is one of above the minimum polynomials. Each of $x^8 + x + 1,\ x^8 + x^3 + 1,\ x^8 + x^5 + 1$ *and* $x^8 + x^7 + 1$ only has one solution. Each of $x^8 + x^2 + 1,\ x^8 + x^4 + 1$ *and* $x^8 + x^6 + 1$ has 16 solutions.

Specifically, solutions of $g(x)$ satisfying $g^2(x) \equiv x^2 \pmod{x^8+x^2+1}$ are as follows

$$x,\ x^4+1,\ x^5+x^2,\ x^5+x^4+x^2+x^1+1,\ x^6+x^3+x^2+x^1,\ x^6+x^4+x^3+x^2+1,$$

$$x^6 + x^5 + x^3,\ x^6 + x^5 + x^4 + x^3 + x^1 + 1,\ x^7 + x^3 + 1,\ x^7 + x^4 + x^3 + x^1,$$

$$x^7+x^5+x^3+x^2+x^1+1,\ x^7+x^5+x^4+x^3+x^2,\ x^7+x^6+x^2+1,\ x^7+x^6+x^4+x^2+x^1,$$

$$x^7 + x^6 + x^5 + x^1 + 1,\ x^7 + x^6 + x^5 + x^4.$$

Solutions of $g(x)$ satisfying $g^2(x) \equiv x^2 \pmod{x^8 + x^4 + 1}$ are as follows

$$x,\ x^4+x^2+x^1+1,\ x^5+x^3,\ x^5+x^4+x^3+x^2+1,\ x^6+x^1+1,\ x^6+x^4+x^2+x^1,$$

$$x^6+x^5+x^3+1,\ x^6+x^5+x^4+x^3+x^2,\ x^7,\ x^7+x^4+x^2+1,\ x^7+x^5+x^3+x^1,$$

$$x^7 + x^5 + x^4 + x^3 + x^2 + x^1 + 1,\ x^7 + x^6 + 1,\ x^7 + x^6 + x^4 + x^2,$$

$$x^7 + x^6 + x^5 + x^3 + x^1 + 1,\ x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x^1.$$

Solutions of $g(x)$ satisfying $g^2(x) \equiv x^2 \pmod{x^8 + x^6 + 1}$ are as follows

$$x,\ x^4 + x^3 + x^1 + 1,\ x^5 + x^3 + 1,\ x^5 + x^4,\ x^6 + x^3 + x^2 + 1,\ x^6 + x^4 + x^2,$$

$$x^6+x^5+x^2+x^1,\ x^6+x^5+x^4+x^3+x^2+x^1+1,\ x^7+x^2+1,\ x^7+x^4+x^3+x^2,$$

$$x^7 + x^5 + x^3 + x^2 + x^1,\ x^7 + x^5 + x^4 + x^2 + x^1 + 1,\ x^7 + x^6 + x^3 + x^1,$$

$$x^7+x^6+x^4+x^1+1,\ x^7+x^6+x^5+1,\ x^7+x^6+x^5+x^4+x^3.$$

Algorithm 2 is specially designed to construct lightweight 4×4 Hadamard involutory MDS matrices over the matrix polynomial residue ring. The platform of Algorithm 2 is the same as Algorithm 1. By running Algorithm 2, results are organized as follows:

(I) Over 8×8 matrix polynomial residue rings, constructing 80640 Hadamard involutory MDS matrices with 20 XORs only takes about 1 minutes and 4 seconds.

(II) When entries are 4×4 matrices over $\mathbb{F}_2$, the lightest Hadamard involutory MDS matrices with 24 XORs.

(III)When entries are 8×8 matrices over $\mathbb{F}_2$, the lightest Hadamard involutory MDS matrices with 20 XORs.

Algorithm 2 Construct lightweight Hadamard involutory MDS matrices

Define matrix structure as $Had(I,A,B,C)$.
for Search all $A\in GL(8,F_2)$, $\#A=1$, A and $A+I$ are non-singular **do**
    x replaces A.
    Find $f(x)$, which is the minimum polynomial of A in MP_8.
    Find polynomials $b_1(x),\cdots,b_k(x)$ in $X1_8$, $X2_8$ and $X3_8$, which satisfy that XOR count is less than 4.
    Find all quadratic congruences of $x^2 \pmod{f(x)}$.
    for i from 1 to k **do**
        $b_i(x)$ replaces B.
        for j from 1 to 16 **do** $b_i(x)+q_j(x)$ replace C, where q_j is a quadratic congruence of $x^2 \pmod{f(x)}$.
            if Matrix is MDS **then**
                Record this MDS matrix and its sum of XORs.
            end if
        end for
    end for
end for

Example 3

(1) m=4. $T=[[1,2],3,4,1]$. The following matrix is a Hadamard involutory MDS matrix with 24 XORs.

$$\begin{pmatrix} I & T & T^2 & T^2+T \\ T & I & T^2+T & T^2 \\ T^2 & T^2+T & I & T \\ T^2+T & T^2 & T & I \end{pmatrix}$$

(2) m=8. $T = [4, 1, 2, 8, 6, 3, [5, 8], 7]$. The following matrix is a Hadamard involutory MDS matrix with 20 XORs.

$$\begin{pmatrix} I & T & T^6+T^4 & T^2 \\ T & I & T^2 & T^6+T^4 \\ T^6+T^4 & T^2 & I & T \\ T^2 & T^6+T^4 & T & I \end{pmatrix}$$

Comparisons with previous constructions of lightweight involutory MDS matrices are shown at table 4. Comparisons with [19] are at table 5. In table 4 and table 5, the *sum of XORs* denotes the sum of XORs of the entirety-matrix.

Table 4: Comparisons with previous constructions of lightweight involutory MDS matrices

Matrix type	Element	Sum of XORs	Ref.
$Had(I, A, A^{-1}, A + A^{-1})$	$GL(4, \mathbb{F}_2)$	24	[19]
$Had(0 \times 1, 0 \times 4, 0 \times 9, 0 \times d)$	$F_{2^4}/0 \times 13$	24	[26][18]
$Had(0 \times 1, 0 \times 2, 0 \times 6, 0 \times 4)$	$F_{2^4}/0 \times 19$	24	[10]
$Had(I, A, B, C)$	$\mathbb{F}_2[T_{4\times4}]$	24	Ours
$Hadamard - Cauchy(0 \times 01, 0 \times 02, 0 \times fc, 0 \times fe)$	$F_{2^8}/0 \times 11b$	296	[17]
$Had(0 \times 01, 0 \times 02, 0 \times 04, 0 \times 06)$	$F_{2^8}/0 \times 11d$	88	[25]
$Had(0 \times 01, 0 \times 02, 0 \times b0, 0 \times b2)$	$F_{2^8}/0 \times 165$	64	[18]
$Subfield - Had(0 \times 1, 0 \times 4, 0 \times 9, 0 \times d)$	$F_{2^4}/0 \times 13$	48	[18]
$Had(I, A, A^{-1}, A + A^{-1})$	$GL(8, \mathbb{F}_2)$	40	[19]
$Had(I, A, B, C)$	$\mathbb{F}_2[T_{8\times8}]$	20	Ours

6 Reasons of Construction Efficiency

In this section, we discuss reasons of efficiently constructing lightweight MDS matrices.

6.1 Efficiency of Constructing Lightweight Non-involutory MDS Matrices

In this subsection, we introduce reasons of efficiently constructing lightweight non-involutory MDS matrices.

Table 5: Comparisons of construction efficiency with [19]

Matrix type	Element	Sum of XORs	Number	Running time	Ref.
$Optimal$	$GL(8,\mathbb{F}_2)$	10	40320	no mentioned	[19]
$Optimal$	$\mathbb{F}_2[T_{8\times 8}]$	10	40320	1min 16sec	Ours
S_1	$\mathbb{F}_2[T_{8\times 8}]$	10	1128960	14hours	Ours
$Circ(I,I,A,B)$	$GL(8,\mathbb{F}_2)$	12	80640	3days	[19]
$Circ(I,I,A,B)$	$\mathbb{F}_2[T_{8\times 8}]$	12	80640	1min 27sec	Ours
$Had(I,A,A^T,B)$	$GL(8,\mathbb{F}_2)$	20	622	4weeks	[19]
$Had(I,A,B,C)$	$\mathbb{F}_2[T_{8\times 8}]$	20	241920	7min	Ours
$InvolutoryHad(I,A,A^{-1},A+A^{-1})$	$GL(8,\mathbb{F}_2)$	40	80640	1day	[19]
$InvolutoryHad(I,A,B,C)$	$\mathbb{F}_2[T_{8\times 8}]$	20	80640	1min 04sec	Ours

In previous papers, lightweight MDS matrices are usually constructed with templates like Circulant matrix, Hadamard matrix or Optimal matrix. The following matrix is a circulant matrix.

$$Cir(I,I,A,B) = \begin{pmatrix} I & I & A & B \\ B & I & I & A \\ A & B & I & I \\ I & A & B & I \end{pmatrix}.$$

With $Cir(I,I,A,B)$, we and [19] get the same results as mentioned in Table 5. Next we take $m = 8$ as an example. [19] use about 3 days. But we use only 1 minute and 27 seconds. In [19], elements are from $GL(m,\mathbb{F}_2)$. But in our method, elements are from the matrix polynomial residue ring $\mathbb{F}_2[T]$. Deeper reasons of this difference are as follows

(I) With $GL(8,\mathbb{F}_2)$, computations of elements do not have a obviously efficient way. But with $\mathbb{F}_2[T]$, computations are isomorphic to the polynomial. So computations in $\mathbb{F}_2[T]$ is obviously more efficient than $GL(8,\mathbb{F}_2)$.

(II) For $GL(8,\mathbb{F}_2)$. There are 1048320 matrices with 1, 2 or 3 XORs in $GL(8,\mathbb{F}_2)$. Then over $GL(8,\mathbb{F}_2)$, A and B, in $Cir(I,I,A,B)$, have 1048320 choices respectively. So over $GL(8,\mathbb{F}_2)$, the search space of $Cir(I,I,A,B)$ is 1048320×1048320.

For the matrix polynomial residue ring. If non-singular 8×8 T is fixed, $T+I$ non-singular and $\#(T)=1$, then T has 282240 choices. And there are only at most 4 elements with 1 or 2 XORs in $\mathbb{F}_2[T]$. So, with $\mathbb{F}_2[T]$, if we want to construct lightest results, then A and B have at most

4 different choices respectively. So with $\mathbb{F}_2[T]$, the search space of H_1 is only 282240×4×4.

Remark 2. Because the matrix polynomial residue ring has obvious efficiency, so we can use it to construct more general templates like S_1, S_2, S_3, S_4 or S_5 mentioned in Sec. 3.3. The following matrix is a matrix of template S_5.

$$S_5 = \begin{pmatrix} I & I & A_1 & A_2 \\ A_3 & I & I & A_4 \\ A_5 & A_6 & I & I \\ I & A_7 & A_8 & I \end{pmatrix}.$$

In most previous papers, lightweight MDS matrices are constructed only with templates like circulant matrix or Hadamard matrix, where non-identity elements are re-used. The reason is that if all non-identity elements are independent with each other, then the search space is too huge to complete the construction within acceptable time. However, because of the efficiency of our method, lightweight MDS matrices as S_1, S_2, S_3, S_4 or S_5 can be constructed within acceptable time. For example, with $GL(8, \mathbb{F}_2)$, the non-identity element has 1048320 choices. So, with $GL(8, \mathbb{F}_2)$, the search space of S_5 is $(1048320)^8$. But with the matrix polynomial residue ring, the search space is only $282240 \times (4)^8$. With our method and S_5, we use 14 hours to construct 1128960 results with 10 XORs. But with S_1 and $GL(8, \mathbb{F}_2)$, the time of construction will be unacceptable.

6.2 Efficiency of Constructing Involutory MDS Matrices

In this subsection, we introduce reasons of efficiently constructing lightweight involutory MDS matrices. Besides advantages as mentioned in Sec. 6.1, for the lightweight involutory MDS matrix, our second method has some theoretical optimizations for constructing involutory Hadamard MDS matrices.

As mentioned at table 5, paper [19] only construct the involutory Hadamard matrix as $Had(I, A, A^{-1}, A + A^{-1})$. In this matrix, only A is changed. However, we construct the involutory Hadamard matrix as $Had(I, A, B, C)$. In our matrix, A, B and C are all changed. More importantly, they use 1 day to construct 80640 results with 40 XORs. However, we only use 1 minute and 4 second to construct 40320 results with 20 XORs, which is much fewer than 40. Besides advantages as mentioned in Sec. 6.1, we have some theoretical optimizations for constructing involutory Hadamard MDS matrices as follow.

According to Corollary 1, if $Had(1, x, b(x), c(x)$ is involutory, then $(b(x) + c(x))^2 = x^2$. As mentioned at Sec. 5.2, for each adaptive minimum polynomial, there are only 16 solutions satisfying $g(x)^2 = x^2$. Let these 16 solutions be $g_1(x)$, $g_2(x)$, ..., $g_{16}(x)$. When $b(x)$ is fixed, then it must be that $c(x) = b(x) + g_i(x)$. So if $b(x)$ is fixed, $c(x)$ has only 16 choices. In our construction, let non-singular A be adaptive to construct involutory Hadamard MDS matrices, $\#(A) = 1$, $A + I$ non-singular then A has 282240 choices. For constructing lightweight 4×4 involutory MDS matrices, our search space is $120960 \times 2^8 \times 16$.

In a word, by using theoretical optimizations, we largely reduce the search space. So we use only very little time to construct satisfactory results.

7 Propositions about the Parity of XOR Count

In this section, we prove properties about the parity of XORs.

Proposition 1. *Let A, B, $A + B \in GL(m, \mathbb{F}_2)$, then*

$$\#(A + B) \equiv \#(A) + \#(B) + m \pmod 2.$$

Proof. It is obviously that $\omega(A + B) \equiv \omega(A) + \omega(B) \pmod 2$.
Because $\#A = \omega(A) - m, \#B = \omega(B) - m$ and $\#(A+B) = \omega(A+B) - m$.
Then $\#(A + B) \equiv \#(A) + \#(B) + m \pmod 2$. □

Proposition 2. *Let $\alpha = (a_1, a_2, ..., a_m)^T$ and $\beta = (b_1, b_2, ..., b_m)^T$, where $a_i, b_i \in \mathbb{F}_2$. Then*

$$\omega(\alpha\beta^T) = \omega(\alpha)\omega(\beta).$$

Proof. Because $\alpha = (a_1, a_2, ..., a_m)^T, \beta = (b_1, b_2, ..., b_m)^T$, then

$$\omega(\alpha\beta^T) = \omega\begin{pmatrix} a_1b_1 & a_1b_2 & \cdots & a_1b_m \\ a_2b_1 & a_2b_2 & \cdots & a_2b_m \\ \vdots & \vdots & \ddots & \vdots \\ a_mb_1 & a_mb_2 & \cdots & a_mb_m \end{pmatrix} = \sum_{i=1}^{m}\sum_{j=1}^{m} a_ib_j$$

$$= \sum_{i=1}^{m} a_i \sum_{i=1}^{m} b_j = \omega(\alpha)\omega(\beta).$$

□

Proposition 3. *Let $A, B \in GL(m, \mathbb{F}_2)$ and $A = (\alpha_1, \alpha_2, ..., \alpha_m)$ and $B = (\beta_1, \beta_2, ..., \beta_m)^T$. Then*

$$\#(AB) \equiv \sum_{i=1}^{m} \omega(\alpha_i)\omega(\beta_i) \pmod 2.$$

Proof. Because $A = (\alpha_1, \alpha_2, ..., \alpha_m)$ and $B = (\beta_1, \beta_2, ..., \beta_m)^T$, so $AB = \sum_{i=1}^{m} \alpha_i \beta_i^T$. According to proposition 2,

$$\omega(AB) \equiv \sum_{i=1}^{m} \omega(\alpha_i \beta_i^T) \equiv \sum_{i=1}^{m} \omega(\alpha_i)\omega(\beta_i^T) \pmod 2.$$

Because $\#(AB) = \omega(AB) - m$, so

$$\#(AB) \equiv \sum_{i=1}^{m} \omega(\alpha_i)\omega(\beta_i^T) + m \pmod 2.$$

□

Proposition 4. *Let $L_1, L_2, L_1 + L_2 \in M(n, m)$. Then*

$$\#(L_1 + L_2) \equiv \#(L_1) + \#(L_2) + nm \pmod 2.$$

Proof. It is obviously that $\omega(L_1 + L_2) \equiv \omega(L_1) + \omega(L_2) \pmod 2$. Because

$$\#(L_1 + L_2) = \omega(L_1) - n^2 m, \#(L_1) = \omega(L_1) - n^2 m,$$

$$\#(L_2) = \omega(L_2) - n^2 m,$$

so

$$\#(L_1 + L_2) \equiv \#(L_1) + \#(L_1) + n^2 m \equiv \#(L_1) + \#(L_1) + nm \pmod 2$$

. □

Proposition 5. *Let $A_i, B_i \in GL(m, \mathbb{F}_2)$ and $i = 1, 2, ..., n$. Then*

$$\omega\left(\begin{pmatrix} A_1 & A_2 & \cdots & A_n \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{pmatrix} \right) \equiv \omega(\sum_{i=1}^{n} A_i \sum_{j=1}^{n} B_j) \pmod 2.$$

Proof.

$$\omega\left(\begin{pmatrix} A_1 & A_2 & \cdots & A_n \end{pmatrix}\begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{pmatrix}\right) = \omega\begin{pmatrix} A_1B_1 & A_1B_2 & \cdots & A_1B_n \\ A_2B_1 & A_2B_2 & \cdots & A_2B_n \\ \vdots & \vdots & \ddots & \vdots \\ A_nB_1 & A_nB_2 & \cdots & A_nB_n \end{pmatrix}$$

$$\equiv \omega(\sum_{i,j=1}^{n} A_iB_j) \equiv \omega(\sum_{i=1}^{n} A_i \sum_{j=1}^{n} B_j) \pmod 2.$$

□

Proposition 6. *Let* $L_1, L_2, L_1L_2 \in M(n,m)$ *and*

$$L_1 = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{pmatrix},\ L_2 = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1n} \\ B_{21} & B_{22} & \cdots & B_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ B_{n1} & B_{n2} & \cdots & B_{nn} \end{pmatrix}.$$

Then

$$\#(L_1L_2) \equiv \sum_{k=1}^{n} \omega(\sum_{i=1}^{n} A_{ik} \sum_{j=1}^{n} B_{kj}) + nm \pmod 2.$$

Proof.

$$\omega(L_1L_2) \equiv \omega \sum_{k=1}^{n} \left(\begin{pmatrix} A_{1k} \\ A_{2k} \\ \vdots \\ A_{nk} \end{pmatrix}\begin{pmatrix} B_{k1} & B_{k2} & \cdots & B_{kn} \end{pmatrix}\right)$$

$$\equiv \sum_{k=1}^{n} \omega \left(\begin{pmatrix} A_{1k} \\ A_{2k} \\ \vdots \\ A_{nk} \end{pmatrix}\begin{pmatrix} B_{k1} & B_{k2} & \cdots & B_{kn} \end{pmatrix}\right) \pmod 2.$$

According to proposition 5, then

$$\omega(L_1L_2) \equiv \sum_{k=1}^{n} \omega(\sum_{i=1}^{n} A_{ik} \sum_{j=1}^{n} B_{kj}) \pmod 2.$$

Because $\#(L_1L_2) = \omega(L_1L_2) - n^2m$, so

$$\#(L_1L_2) \equiv \sum_{k=1}^{n} \omega(\sum_{i=1}^{n} A_{ik} \sum_{j=1}^{n} B_{kj}) + n^2m$$

$$\equiv \sum_{k=1}^{n} \omega(\sum_{i=1}^{n} A_{ik} \sum_{j=1}^{n} B_{kj}) + nm \pmod 2.$$

□

8 Conclusions

In the present paper, we mainly investigate constructions of 4×4 lightweight MDS matrices over the matrix polynomial residue ring, where m=4, 8 or 16. According to distributions of the minimum polynomial and distributions of XOR count, we propose an efficient algorithm to construct the lightest MDS matrices. Besides, we prove that some special MDS matrices can not be involutory. According to the quadratic congruence, we propose another efficient algorithm to construct lightweight Hadamard involutory MDS matrices, which are much lighter than previous papers. We discuss reasons about why our methods work very efficiently. Finally, we prove a series of properties about the parity of XOR count.

References

1. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON family of lightweight hash functions. In Annual Cryptology Conferencepp. 222-239. Springer Berlin Heidelberg (2011)
2. Daemen, J., Knudsen, L., Rijmen, V.: The block cipher Square. In International Workshop on Fast Software Encryption, pp. 149-165. Springer Berlin Heidelberg (1997)
3. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED block cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326C341. Springer, Heidelberg (2011)
4. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer, Heidelberg (2002)
5. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404 (2013)
6. Bogdanov, A.A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450C466. Springer, Heidelberg (2007)
7. Yang, G., Zhu, B., Suder, V., Aagaard, M.D., Gong, G.: The Simeck family of lightweight block ciphers. In: Guneysu, T., Handschuh, H. (eds.) CHES 2015. LNCS, vol. 9293, pp. 307C329. Springer, Heidelberg (2015)
8. Aumasson, J. P., Henzen, L., Meier, W., Naya-Plasencia, M.: Quark: A lightweight hash. In International Workshop on Cryptographic Hardware and Embedded Systems, pp. 1-15. Springer Berlin Heidelberg(2010)

9. Bogdanov, A., Kne?evi?, M., Leander, G., Toz, D., Var?c?, K., Verbauwhede, I.: SPONGENT: A lightweight hash function. In International Workshop on Cryptographic Hardware and Embedded Systems, pp. 312-325. Springer Berlin Heidelberg(2011)
10. Andreeva, E., Bilgin, B., Bogdanov, A., Luykx, A., Mendel, F., Mennink, B., Mouha, N., Wang, Q., Yasuda, K.: PRIMATEs v1. Submission to the CAESAR Competition. http://competitions.cr.yp.to/round1/primatesv1.pdf(2014)
11. Augot, D., Finiasz, M.: Direct construction of recursive MDS diffusion layers using shortened BCH codes. In: Cid, C., Rechberger, C. (eds.) FSE 2014. LNCS, vol. 8540, pp. 3C17. Springer, Heidelberg (2015)
12. Augot, D., Finiasz, M.: Exhaustive search for small dimension recursive MDS diffusion layers for block ciphers and hash functions. In Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on, pp. 1551-1555. IEEE.(2013)
13. Berger, T.P.: Construction of recursive MDS diffusion layers from Gabidulin codes. In: Paul, G., Vaudenay, S. (eds.) INDOCRYPT 2013. LNCS, vol. 8250, pp. 274C285. Springer, Heidelberg (2013)
14. Sajadieh, M., Dakhilalian, M., Mala, H., Sepehrdad, P.: Recursive diffusion layers for block ciphers and hash functions. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 385C401. Springer, Heidelberg (2012)
15. Wu, S.,Wang,M.,Wu,W.: Recursive diffusion layers for (lightweight) block ciphers and hash functions. In: Knudsen, L.R., Wu, H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 355C371. Springer, Heidelberg (2013)
16. Nakahara Jr., J., Abraho, I.: A new involutory mds matrix for the aes. I. J Netw. Secur. 9(2), pp. 109C116 (2009)
17. Chand Gupta, K., Ghosh Ray, I.: On constructions of circulant MDS matrices for lightweight cryptography. In: Huang, X., Zhou, J. (eds.) ISPEC 2014. LNCS, vol. 8434, pp. 564C576. Springer, Heidelberg (2014)
18. Sim, S. M., Khoo, K., Oggier, F., Peyrin, T.: Lightweight MDS involution matrices. In International Workshop on Fast Software Encryption, pp. 471-493. Springer Berlin Heidelberg(2015)
19. Li, Y., Wang, M.: On the construction of lightweight circulant involutory MDS matrices. In International Conference on Fast Software Encryption, pp. 121-139. Springer Berlin Heidelberg(2016)
20. Berger, T. P., El Amrani, N.: Codes over $L(GF(2)^m, GF(2)^m)$, MDS Diffusion Matrices and Cryptographic Applications. In International Conference on Codes, Cryptology, and Information Security, pp. 197-214. Springer International Publishing(2015)
21. Gupta, K. C., Ray, I. G.: On constructions of MDS matrices from companion matrices for lightweight cryptography. In International Conference on Availability, Reliability, and Security, pp. 29-43. Springer Berlin Heidelberg(2013)
22. Liu, M., Sim, S. M.: Lightweight MDS generalized circulant matrices. In International Conference on Fast Software Encryption, pp. 101-120. Springer Berlin Heidelberg(2016)
23. Beierle, C., Kranz, T., Leander, G.: Lightweight Multiplication in $GF(2^n)$ with Applications to MDS Matrices. In Annual Cryptology Conference, pp. 625-653. Springer Berlin Heidelberg(2016)
24. Li T., Bai J., Sun Y., Wang D., Lin D.: The Lightest 4x4 MDS Matrices over GL (4, F2) http://eprint.iacr.org/2016/686.pdf(2016)
25. Barreto, P., Rijimen, V.: The anubis block cipher. Submission to the NESSIE Project(2000)

26. Jean, J., Nikolic, I., Peyrin, T.:Joltik v1.1. Submission to the CAESAR competition(2014) http://www1.spms.ntu.edu.sg/ syllab/Joltik
27. Junod, P., Vaudenay, S.: Perfect diffusion primitives for block ciphers. In International Workshop on Selected Areas in Cryptography, pp. 84-99. Springer Berlin Heidelberg(2004)