Exploring the Feasibility of Meeting the MDS Criterion in Random Matrices and Searching for Efficient Random Involutory Hadamard MDS Matrices

Tran Thi Luong
Academy of Cryptography Techniques, Tan Trieu, Thanh Tri, Hanoi, Vietnam
Correspondence: Tran Thi Luong, luongtranhong@gmail.com
Communication: received 20 sep 2023, revised 11 Jan 2024, accepted 21 Jan 2024
DOI: 10.32913/mic-ict-research.v2024.n1.1231

***Abstract*: Maximum distance separable codes (MDS codes) have been studied for a long time in error correcting code theory and have important applications in cryptography. There are many different methods have been studied to build MDS matrices such as: from MDS codes, Cauchy matrices, Hadamard matrices, Vandermonde matrices, circulant and circulant-like matrices etc. In this paper, the construction of MDS matrix from random matrices is studied, and the possibility of satisfying the MDS condition of any random matrix is presented. Then, the method of random search of MDS matrices will be given. In addition, we propose algorithms to randomly search for efficient involutory Hadamard MDS matrices of size 4, and 8. These results will be meaningful to determine the possibility that MDS matrices can be obtained from random matrices and diversify the methods of searching for MDS matrices. The involutory Hadamard MDS matrices make the encryption and decryption processes identical, which makes them very efficient for both hardware and software implementation.**

Keywords: *MDS matrix, random matrices, Hadamard MDS matrix, Circulant-like MDS matrix*

I. INTRODUCTION

MDS matrices [5, 11, 16, 17] play an important role in cryptographic applications. They are often used for the diffusion layer of block ciphers and hash functions to provide high diffusion. Therefore, many well-known ciphers have used MDS matrices in their diffusion layer such as AES, Twofish, KHAZAD, Anubis, SHARK, Square Hierocrypt, Camellia, Manta, MUGI, WHIRLPOOL, etc.

Because of the benefits of MDS matrices, many different methods have been studied to generate them, including: from MDS codes [5, 13, 14, 15], from suitable matrices such as Cauchy matrices [1, 3], Hadamard matrices [2, 7, 10, 12], Vandermonde matrices [10], Companion matrix [6, 9], recursive MDS matrices [18-22], etc.

The method of constructing MDS matrices from MDS codes [5, 13, 14, 15] has the advantage of being able to generate a large number of MDS matrices, but the disadvantage is that we cannot control the elements within the matrix. On the other hand, constructing MDS matrices from the Vandermonde matrix [10] and the Cauchy matrix [1, 3] allows for the advantage of building large-sized MDS matrices. However, the elements in these matrices often have high Hamming weights, leading to significant execution costs.

The method of constructing MDS matrices from recursive matrices [18-22] and Companion matrices [6, 9] has the advantage of being convenient for hardware implementation. However, recursive matrices are not necessarily MDS matrices, so they always need to be checked for the MDS condition. On the other hand, checking this condition for large-sized recursive matrices can be challenging. As for Hadamard matrices, they are a special type of matrix with many advantages in execution. Works on constructing MDS matrices from Hadamard matrices [2, 7, 10, 12] also provide various methods to generate matrices of this form. However, these works do not approach the construction randomly, as well as the ability to control the elements in this matrix.

Recently, in [24], the authors shared findings from the all-encompassing exploration of (iterative) MDS matrices over $GL(4, F2)$. The investigation specifically delves into circulant MDS matrices ranging from orders 4 to 8, Hadamard MDS matrices of orders 4 and 8, as well as recursive MDS matrices derived from companion matrices of order 4, and recursive MDS matrices stemming from

sparse DSI matrices spanning orders 4 to 8. In this paper, the authors sought involutory matrices but over the field $GL(4, F2)$; furthermore, they have not considered the fixed points of these matrices. In [25], the authors introduced novel construction frameworks, specifically, transposition-permutation path configurations for 3×3 involutory and MDS permutation-equivalent matrices across F_{2^3} and F_{2^4}. They created 3×3 involutory and MDS matrices across F_{2^3} and F_{2^4}, subjecting all these matrices to analysis by identifying their permutation-equivalent matrices. Subsequently, an examination was conducted to identify any distinctive permutation patterns, particularly within matrices of this size. Consequently, they identified a total of 28,088 unique transposition-permutation path configurations, offering a direct means to construct 3×3 involutory and MDS matrices from any representative 3×3 involutory and MDS matrix over F_{2^3} and F_{2^4}. However, this work only focuses on matrices of size 3×3. In [26], a fresh hybrid approach is introduced, amalgamating search-based techniques and direct construction methods, aiming to produce the complete set of 4×4 involutory maximum distance separable (MDS) matrices over F_{2^m}. The innovative method significantly reduces the complexity of the search space to the square root of n, with n denoting the count of all 4×4 invertible matrices over F_{2^m} subject to exploration. Consequently, this approach facilitates the generation of all 4×4 involutory MDS matrices over and . In this work, the authors solely concentrated on matrices of size 4×4 over F_{2^3} and F_{2^4}.

One of the other methods to search for MDS matrices is to search from random matrices. A random matrix is a matrix whose elements are chosen randomly, independently, with uniform probability in some finite field $GF(p^r)$, where p is a prime number. In [23], the authors presened an algorithm of generating random MDS matrices, but the article is quite sketchy, and the authors cannot control the coefficients of the generated matrix according to this algorithm.

Our contributions: In this paper, a lower bound for the probability of satisfying the MDS condition of any random matrix is presented. Then, a method of random search of MDS matrices is given and based on specific examples to present an estimate for this possibility. In addition, we propose efficient random search algorithms for involutory Hadamard MDS matrices of size 4, 8. These results will be meaningful to determine the possibility that MDS matrices can be obtained from random matrices and diversify the methods of searching for MDS matrices. With our search approach, involutory Hadamard matrices can be found efficiently for implementation due to the presence of elements with small Hamming weights. In addition, the involutory Hadamard MDS matrices make the encryption and decryption processes identical, which makes them very efficient for both hardware and software implementation.

The paper is organized as follows. In Section II, preliminaries are introduced. Section III presents a lower bound for the possibility of satisfying the MDS condition of any random matrix and a method for random search of MDS matrices along with specific examples to present an estimate for this probability. Then is to search for efficient random Hadamard MDS matrices for executions of size 4, and 8. Section IV is the conclusion.

II. PRELIMINARIES AND RELATED WORKS

2.1. MDS matrix. The MDS matrices come from the theory of error correction codes with maximal distance separable codes or MDS codes. To better understand the MDS matrix, we can see the important theorems about the MDS matrix in the code theory as follows:

Theorem 1: ([5, p. 33]). If C is a $[n, k, d]$ code, $n-k \geq d-1$.

The code C is called an MDS code if $n - k = d - 1$ (or $d = n - k + 1$).

Theorem 2: ([5, page 321]). A code $[n, k, d]$ with a generator matrix $G[I|A]$ where A is a $k\times(n-k)$ matrix is MDS if and only if every square submatrix (generated from any i rows and any i columns, for any $i = 1, 2, ..., min\{k, n-k\}$ of A is nonsingular.

Thus, it can be seen that the MDS matrix is a very special matrix whose every square sub-matrix is nonsingular.

2.2. Hadamard MDS matrix. The authors in [2, 7, 10, 12] use Hadamard matrices to construct MDS matrices for their block cipher design.

Elumalai and Reddy in [12] defined the Hadamard matrix as follows:

Definition 1: With k elements $\alpha_0, \alpha_1, ..., \alpha_{k-1}$, a matrix $A = [a_{i,j}]$ is called a Hadamard matrix if each element in this matrix is constructed as: $a_{ij} = \alpha_{i \oplus j}$

The authors in [8] also defined a Hadamard matrix of size 2^n in a different way as follows:

A matrix H of size $2^n\times2^n$ is a Hadamard matrix over the finite field $GF(2^q)$ (FFHadamard) if it can be represented as follows: $H = \begin{bmatrix} U & V \\ V & U \end{bmatrix}$, and two submatrices U, V are matrices over $GF(2^p)$.

An example of a 4×4 FFHadamard matrix is as follow:

$$\mathbf{H} = had(a_0, a_1, a_2, a_3) = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_3 & a_0 & a_1 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} (*)$$

where $h_{ij} = a_{i \oplus j}$

Comment 1. *Each Hadamard matrix A over a finite field will have the following property:* $A^2 = \gamma \cdot I$ *where* γ *is a constant. When* $\gamma = 1$ *, then A is an involutory matrix.*

III. SOME RESULTS

In this section, some results will be presented on the method of finding MDS matrices from random matrices, in which a lower bound for the possibility of satisfying the MDS condition of any random matrix is given. Then, a method of random search of MDS matrices is presented and an estimate for this probability is given on specific examples.

3.1. MDS matrices of size 2

Let $A = [a_{i,j}]_{m\times m}$ be an any random matrix over $GF(p^r)$, where p is a prime number.

Then, it can be easily seen that the number of *sub-matrices* of size $i(1 \le i \le m)$ created by any i rows and i columns of the matrix A will be equal to $(C_m^i)^2$.

In the case of $p = 2$, $r = 1$ we see that there is only one MDS matrix of size 1, and there are no MDS matrices of size 2 or higher.

For the case $GF(p^r)$ and $r > 1$, we have the following lemma:

Lemma 1: The number of MDS matrices of size 2 over $GF(p^r)$ for $r > 1$, is equal to $P^3(P-1)$, where $P = p^r - 1$.

Proof: Indeed, the number of singular matrices of size 2 over $GF(p^r)$ (with all four elements being nonzero) is equal to P^3. Also, the number of possible matrices of size 2 with all four elements being non-zero over $GF(p^r)$ is equal to P^4. Therefore, the number of MDS matrices of size 2 is equal to $P^4 - P^3 = P^3(P-1)$. ■

3.2. The ability to satisfy the MDS condition of random matrices

In this section, an any random matrix is partitioned into disjoint square matrices of size 2, producing a series of Bernoulli's tests on these matrices. On this basis, the probability that an any random matrix not being MDS is estimated.

Let $A = [a_{i,j}]_{m\times n}$, $(m \ge 2)$ be an any random matrix over $GF(p^r)$.

The partition of the disjoint submatrices of size 2 of the matrix A **is as follows:**

- Consider pairs of consecutively disjointed rows and columns of the matrix A .

- Denote N is the number of disjoint sub-matrices of size 2 created by the above mentioned pairs of rows and columns of the matrix A. Then, $N = [\frac{m}{2}]^2$. These matrices are N completely random and independent matrices of size 2.

- According to the proof of the Lemma 1, the probability that each of these N matrices being MDS is equal to $\frac{P^3(P-1)}{P^4} = 1 - \frac{1}{P}$; the probability that each matrix not being MDS is equal to $\frac{1}{P}$. Note that, the larger P is, the higher the probability to choose the MDS matrix is.

Series of Bernoulli's tests for the N above matrices of size 2

Consider the following event: Ω =There are at least $i(1 \le i \le N)$ matrices in N above matrices not being MDS. Obviously, if Ω occurs, then the matrix A is not MDS.

Then, the probability of the event Ω is determined as follows:

$$P(\Omega) = \sum_{i=1}^{N} C_N^i \left(\frac{1}{p}\right)^i \left(1 - \frac{1}{p}\right)^{N-i} \tag{1}$$

On the other hand, it is to have:

$$\sum_{i=1}^{N} C_N^i \left(\frac{1}{p}\right)^i \left(1 - \frac{1}{p}\right)^{N-i} = 1 \tag{2}$$

Then:

$$P(\Omega) = 1 - \left(1 - \frac{1}{p}\right)^N \tag{3}$$

Comment 2. *The larger N is, the smaller* $(1 - \frac{1}{P})$ *is, that is, the larger* $P(\Omega)$ *is.*

Probability that the random matrix $A = [a_{i,j}]_{m\times m}$ **over** $GF(p^r)$ **is not MDS**

Denote the event $\overline{\Omega}$ = The matrix A is not MDS.

It is to have:

$$P(\overline{\Omega}) \ge P(\Omega) \tag{4}$$

Because, as suppose matrix A has N sub-matrices of order 2, but in addition to sub-matrices of order 2, A can also have sub-matrices of level 3, level 4, ..., of order m. So in addition to the submatrices of order 2 which may not be MDS, the higher order submatrices may not be MDS either. Therefore, we have formula (4).

Obviously, it is difficult to directly compute the value of $P(\overline{\Omega})$, so we can estimate the lower bound of this value by computing $P(\Omega)$.

Comment 3. *From the Comment 1 and the formulas (3), (4), it can be seen that, when N is larger or when the size of the matrix m is larger, the probability for the matrix A being not MDS is larger.*

These results are also significant for estimating the number of MDS matrices of any size m.

Example 1.

- For $m = 4$ and $GF(2^8)$ it is to have: $P = 255$ and $N = \frac{m^2}{4} = 4$. Then, $P(\Omega) = 1 - (1 - \frac{1}{P})^4 = 0,0156$, that is, the probability that the matrix A is not MDS $\geq 0,0156$.

- For $m = 8$ and $GF(2^8)$ it is to have: $P = 255$ and $N = \frac{m^2}{4} = 16$. Then, $P(\Omega) = 1 - (1 - \frac{1}{P})^4 = 0,0609$, that is, the probability that the matrix A is not MDS $\geq 0,0609$.

- For $m = 64$ and $GF(2^8)$ it is to have: $P = 255$ and $N = \frac{m^2}{4} = 1024$. Then, $P(\Omega) = \cong 0.984$.

Thus, the larger the matrix size is, the smaller the probability that the matrix being MDS is. In the case of $m \geq 64$, it is almost certain that a random matrix obtained will not be an MDS matrix.

3.3. Searching for MDS matrix from random matrices and probability

Consider $A = [a_{i,j}]_{m\times m}$ is a matrix with elements taken independently, randomly with uniform probability over $GF(p^r)$. The method of random search of MDS matrices is implemented through Algorithm 1 as follows:

Algorithm 1 $m \times m$ base

Input: An empty matrix A of size $m \times m$ over the finite field $GF(p^r)$.

Output: An MDS matrix A of size $m \times m$ over $GF(p^r)$.

Step 1: Construct a matrix **A** by randomly taking the elements of $GF(p^r)$ to fill in the cells of the matrix A.

Step 2: Check if this matrix is MDS:

+ If true, then save this matrix to a file.

+ If false, then go back to Step 1 to create another random matrix A

Experiment:

We experimented this method to find the MDS matrices over the field $GF(2^8)$, and obtained the following results:

- *Searching for MDS matrices of size 4 from random matrices of size 4: with 500 random matrices, 56 MDS matrices of size 4 are obtained.*

Assume the ratio that a random matrix of size 4 is not MDS is P_0.

With a sample size of $n = 500$, we have the ratio that a random matrix of size 4 on a particular sample is not an MDS is: $f_0 = \frac{444}{500} = 0.888$. For accuracy is $1 - \alpha = 95\% \rightarrow u_{1-\frac{\alpha}{2}} = u_{0,975} = 1.96$. Then, the precision of the confidence interval is:

$\varepsilon = u_{1-\frac{\alpha}{2}}\sqrt{\frac{f_0(1-f_0)}{n}} = 1.96\sqrt{\frac{0.888(1-0.888)}{500}} \cong 0.0276$. And, the confidence interval of p_0 is: $(f_0 - \varepsilon; f_0 + \varepsilon) = (0.8604; 0.9156)$

We see that the interval estimate for the true p_0 is larger than the lower bound value 0.0156 (in the example) theoretically calculated.

-Searching for MDS matrices of size 8 from random matrices of size 8: with 500 random matrices, 34 MDS matrices of size 8 are obtained.

Assume the ratio that a random matrix of size 8 is not MDS is p_1 .

With a sample size of , we have the ratio that a random matrix of size 8 on a particular sample is not an MDS is: $f_1 = \frac{466}{500} = 0.932$. For accuracy is $1 - \alpha = 95\%$. Then, $u_{1-\frac{\alpha}{2}} = u_{0.975} = 1.96$. Then, the precision of the confidence interval is:

$\varepsilon = u_{1-\frac{\alpha}{2}}\sqrt{\frac{f_0(1-f_0)}{n}} = 1.96\sqrt{\frac{0,932(1-0.932)}{500}} \cong 0.0221$

And, the confidence interval of p_1 is: $(f_1 - \varepsilon; f_1 + \varepsilon) = (0.9099; 0.9541)$.

It can be seen that, almost certainly, the resulting random matrices will not be MDS matrices.

Comment 4. These results help us to estimate the efficiency of the method of choosing MDS matrices from random matrices for cryptographic applications, and can also help determine the quantity of MDS matrices of any size m. In addition, these results show that the probability that a random matrix is MDS is very small, from which it can be seen that other methods of constructing MDS matrices are more efficient than this one. *However, when we apply the method of random search of MDS matrices with specific matrix types, this method becomes very useful. For example, with Hadamard matrices, the nature of these matrices is not necessarily satisfied by themselves that they are MDS matrices, so our random search for elements in the these matrices, then checking whether the obtained matrices satisfy the MDS condition can help us to find a lot of efficient Hadamard MDS matrices for execution, and have good use in cryptography.*

In the next section, we will present in detail the random search method for involutory Hadamard MDS matrices for execution.

3.4. Finding random 4×4 and 8×8 Hadamard MDS matrices efficient for execution

From Algorithm 1, we improve by not only finding a random MDS matrix A of any form, but we also randomly searching for MDS matrices of a particular form. For example, here we can fix the initial matrix A to be of Hadamard form. Hadamard matrices are capable of being self-inverting matrices, so it is very convenient to implement when the encryption and decryption processes at the linear layer using MDS matrices are exactly the same. Therefore, in random search, we will find elements such that the sum of the elements in a row of the Hadamard matrix is 1 so that the matrix is involutory.

If A is a Hadamard matrix of size m, then from the Definition 2, it can be seen that there are only m distinct

elements in the matrix A, and these m elements will be filled into matrix A in the Hadamard form in Definition 2. For example, these matrices have the form (*) for $m = 4$, and these matrices have the following form for $m = 8$.

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

where the elements $a_i (0 \geq i \geq 7)$ are distinct.

To find the efficient Hadamard matrices, we restrict the elements $a_i \in [01_x, ..., 08_x]$.

We propose **Algorithm 2** to find efficient 4×4 involutory Hadamard MDS matrices, **Algorithm 3** to find efficient 8×8 involutory Hadamard MDS matrices for execution.

Algorithm 2 4×4 base

Input: An empty matrix A of size 4×4 over the finite field $GF(2^8)$; The set T includes a list of tuples of the form $(a_0, a_1, a_2, a_3), T = \emptyset$.

Output: An MDS matrix A of size 4×4 over $GF(2^8)$.

Step 1: Set up the matrix A in the form of a Hadamard matrix $(*)$ with 4 different symbolic elements $a_i \in GF(2^8), 0 \geq i \geq 3$.

Step 2: Select three distinct elements $a_i \in \{01_x, ..., 07_x\} (0 \geq i \geq 2)$. Calculate $a_3 = 1 - \sum_{i=0}^{2} a_i$.

Step 3: If a_3 matches any $a_i (0 \geq i \geq 2)$ element, return to Step 2. Otherwise:

+ If the tuple (a_0, a_1, a_2, a_3) does not belong to T, then fill these elements in the Hadamard matrix A; and add this tuple (a_0, a_1, a_2, a_3) to T.

+ Otherwise, go back to Step 2.

Step 4: Check if the Hadamard matrix obtained after Step 3 is MDS.

+ If true, then save this matrix to a file.

+ If false, then go back to Step 2.

Note that, with the construction as in Algorithm 2, Algorithm 3, the resulting Hadamard matrices from these algorithms will be involutory Hadamard MDS matrices.

Note that, as the Hadamard matrix is of size 4×4 in the form $(*)$, on the other hand, to ensure the sum of elements in a row of the Hadamard matrix A equals 1, in Step 2 of Algorithm 2, we only select 3 distinct elements, and the remaining element in the row of the Hadamard matrix is computed as: $a_3 = 1 - \sum_{i=0}^{2} a_i$. Similarly, we only select 7 distinct elements in Step 2 of **Algorithm 3**.

Algorithm 3 8×8 base

Input: An empty matrix A of size 8×8 over the finite field $GF(2^8)$; The set T includes a list of tuples of the form $(a_0, a_1, a_2, a_3, ..., a_6), T = \emptyset$.

Output: An MDS matrix A of size 8×8 over $GF(2^8)$.

Step 1: Set up the matrix A in the form of a Hadamard matrix $(**)$ with 8 different symbolic elements $a_i \in GF(2^8), 0 \geq i \geq 7$.

Step 2: Select three distinct elements $a_i \in \{01_x, ..., 08_x\} (0 \geq i \geq 6)$. Calculate $a_7 = 1 - \sum_{i=0}^{6} a_i$.

Step 3: If a_7 matches any $a_i (0 \geq i \geq 6)$ element, return to Step 2. Otherwise:

+ If the tuple $(a_0, a_1, a_2, a_3, ..., a_6)$ does not belong to T, then fill these elements in the Hadamard matrix A; and add this tuple $(a_0, a_1, a_2, a_3, ..., a_6)$ to T.

+ Otherwise, go back to Step 2.

Step 4: Check if the Hadamard matrix obtained after Step 3 is MDS.

+ If true, then save this matrix to a file.

+ If false, then go back to Step 2.

We choose the primitive polynomial for $GF(2^8)$ as: $x^8 + x^6 + x^5 + x^2 + 1$.

Since the elements $a_i (0 \leq i \leq m - 1)$ are distinct, for matrices of size 4, we try $7 \times 6 \times 5 = 210$ involutory Hadamard matrices and then obtain 16 MDS matrices. Thus, the ratio of 4×4 involutory Hadamard MDS matrices obtained from randomly generated involutory Hadamard matrices according to Algorithm 2 is $16/210 \approx 0.076$.

The proposed algorithms, including Algorithms 2 and 3, can generate involutory Hadamard MDS matrices. Therefore, the encryption and decryption processes using these matrices are performed identically, resulting in significant execution cost savings. In software, if the encryption process is represented as $y = A.x$, then the decryption process is $x = A.y$. Thus, the encryption and decryption processes are identical, allowing the use of the same lookup table for both encryption and decryption. In hardware, it is possible to design encryption and decryption circuits to be identical, resulting in cost savings in implementation.

In addition, we proceed to calculate the number of fixed points and the number of XORs of the obtained matrices. The results are shown in Table I.

Table I is a list of 16 involutory Hadamard MDS matrices of size 4 obtained. The resulting matrices have a small number of XORs that range from 192 to 200. The number of fixed points equal to 4 is quite small. Compared with AES's Mixcolumn matrix, this matrix has the number of fixed points equal to 2^{16} which is very large. Therefore, the proposed MDS matrices have a much smaller number

TABLE I
LIST OF 16 INVOLUTORY HADAMARD MDS MATRICES OF SIZE 4 WITH PRIMITIVE POLYNOMIAL $x^8 + x^6 + x^5 + x^2 + 1$

No	Involutory Hadamard MDS Matrix of size 4×4	Number of XORs	Number of fixed points
1	$\begin{bmatrix}2&4&6&1\\4&2&1&6\\6&1&2&4\\1&6&4&2\end{bmatrix}$	192	4
2	$\begin{bmatrix}2&5&7&1\\5&2&1&7\\7&1&2&5\\1&7&5&2\end{bmatrix}$	224	4
3	$\begin{bmatrix}3&4&7&1\\4&3&1&7\\7&1&3&4\\1&7&4&3\end{bmatrix}$	240	4
4	$\begin{bmatrix}1&2&4&6\\2&1&6&4\\4&6&1&2\\6&4&2&1\end{bmatrix}$	192	4
5	$\begin{bmatrix}1&2&5&7\\2&1&7&5\\5&7&1&2\\7&5&2&1\end{bmatrix}$	224	4
6	$\begin{bmatrix}1&2&6&4\\2&1&4&6\\6&4&1&2\\4&6&2&1\end{bmatrix}$	192	4
7	$\begin{bmatrix}1&2&7&5\\2&1&5&7\\7&5&1&2\\5&7&2&1\end{bmatrix}$	224	4
8	$\begin{bmatrix}1&3&4&7\\3&1&7&4\\4&7&1&3\\7&4&3&1\end{bmatrix}$	240	4
9	$\begin{bmatrix}1&3&5&6\\3&1&6&5\\5&6&1&3\\6&5&3&1\end{bmatrix}$	240	4
10	$\begin{bmatrix}1&3&6&5\\3&1&5&6\\6&5&1&3\\5&6&3&1\end{bmatrix}$	240	4
11	$\begin{bmatrix}1&3&7&4\\3&1&4&7\\7&4&1&3\\4&7&3&1\end{bmatrix}$	240	4
12	$\begin{bmatrix}1&4&6&2\\4&1&2&6\\6&2&1&4\\2&6&4&1\end{bmatrix}$	240	4
13	$\begin{bmatrix}1&4&7&3\\4&1&3&7\\7&3&1&4\\3&7&4&1\end{bmatrix}$	240	4
14	$\begin{bmatrix}1&5&6&3\\5&1&3&6\\6&3&1&5\\3&6&5&1\end{bmatrix}$	240	4
15	$\begin{bmatrix}1&5&7&2\\5&1&2&7\\7&2&1&5\\2&7&5&1\end{bmatrix}$	224	4
16	$\begin{bmatrix}3&5&6&1\\5&3&1&6\\6&1&3&5\\1&6&5&3\end{bmatrix}$	240	4

of fixed points, in favor of safety.

If we bring back the same method of evaluating the number of XORs that we perform, the AES matrix has a number of XORs of 152, Hierocrypt's MDS matrix is 448. Thus, the 4×4 involutory Hadamard MDS matrices we propose not only have a small number of fixed points, but also a smaller number of XORs than the AES and Hierocrypt MDS matrices.

With matrices of size 8, we try $8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 40.320$ involutory Hadamard MDS matrices and then we obtain only one MDS matrix. That's the matrix $Had(0x1, 0x2, 0x3, 0x4, 0x5, 0x6, 0x7, 0x8, 0x13)$. This matrix has the number of fixed points is 1, the number of XORs is 1.272. This matrix has the smallest possible number of fixed points, so it is very good one, but the number of XORs is quite large, larger than the MDS matrix of the Kalyna block cipher when we perform the same evaluation method. Kalyna's is 952.

In general, the Hadamard MDS matrices we propose have a small number of fixed points, and the number of XOR operations is quite small. Their biggest advantage, on the other hand, is that they are involutory MDS matrices, so they're really useful for execution.

Compare with other MDS matrix generation methods.

In Table II, we present a comparison of our methods for generating MDS matrices with other approaches. With the proposed algorithms, we fixed the matrix format as Hadamard, and the coefficients in these matrices can be controlled. Specifically, in Algorithm 2 and Algorithm 3, we select the elements in the set , so we can create MDS matrices with low execution cost. Meanwhile, with matrices generated from MDS codes such as Reed-Solomon codes, and other methods we cannot control the coefficients of the matrices. The methods for generating MDS matrices from Cauchy [1, 3] and Vandermonde matrices [10] often produce matrices with elements having high Hamming weight and do not control the number of fixed points.

In comparison with the work [24], the authors of the current paper focus solely on the execution cost of involutory matrices, without considering their fixed points. In this study, we prioritize matrices with a small number of fixed points, and then select matrices with low execution costs. Meanwhile, works [25, 26] primarily concentrate on matrices of sizes 3×3 and 4×4 over F_{2^3} and F_{2^4}. The authors in [25] and [26] also did not calculate the fixed points of the obtained matrices.

On the other hand, in Algorithm 2 and Algorithm 3, we select elements from the set $\{01_x, ..., 08_x\}$ rather than exhaustively searching through all possible elements, making the algorithms run quite quickly. The time taken to search

TABLE II
COMPARISON OF OUR MDS MATRIX GENERATION METHODS WITH OTHER APPROACHES

No.	Our algorithms (Algorithm 2, Algorithm 3)	Method for generating MDS matrices from MDS code	Methods for generating MDS matrices from Cauchy matrices [1, 3], Vandermonde matrices [10]	Method for generating MDS matrices in [24]	Method for generating MDS matrices in [25]	Method for generating MDS matrices in [26]
Hamming weight of elements in the matrix	Small	Large	Large	Small	Small	Small
Matrix size	4x4 and 8x8	4x4, 8x8, 16x16	4x4, 8x8, 16x16	4x4 and 8x8	3x3	4x4
Checking the number of fixed points	Yes	No (number of fixed points may be large)	No (number of fixed points may be large)	No	No	No
Involutory property of the resulting matrix	Yes	No	No	Yes	Yes	Yes
Ability to control elements in the matrix	Yes	No	No	Yes	Yes	Yes

for 210 4×4 matrices and verify the MDS condition was 87.5 minutes. The time taken to search for 40,320 8×8 matrices and verify the MDS condition was 896 hours.

Furthermore, the most crucial objective is to find MDS matrices with a small number of fixed points and a low number of XOR operations, ensuring both security and minimizing execution costs. In this paper, we are not only concerned with matrices that have low execution cost, but we also need to select matrices with the small number of fixed point to ensure the security of the block cipher in the future.

IV. CONCLUSION

In this paper, the possibility of satisfying the MDS condition of any random matrices is presented, and the method of finding MDS matrices from random matrices is given. In addition, we propose algorithms to randomly search for involutory Hadarmard MDS matrices. These results will be meaningful to determine the possibility that MDS matrices can be obtained from random matrices and diversify the methods of searching for MDS matrices. The obtained involutory Hadamard MDS matrices are those in which both the encryption and decryption matrices are efficient for execution, so they are very significant in the implementation of today's cryptographic algorithms. On the other hand, they also have small number of fixed points, which is highly favorable for the security of the block cipher and cryptographic algorithms that use them in the future.

REFERENCES

[1] M. Elhosary, N. Hamdy, I.A. Farag, A.E Rohiem, "Optimum Dynamic Diffusion of Block Cipher Based on Maximum Distance Separable Matrices", *International Journal of Information & Network Security (IJINS)*, Vol.2, No.4, August 2013, pp. 327–332, ISSN: 2089–3299.

[2] A. M. Youssef, S. E. Tavares and H. M. Heys, "A New Class of Substitution Permutation Networks", *Workshop on Selected Areas in Cryptography*, SAC '96, Workshop Record, pp. 132–147, 1996.

[3] A. M. Youssef, S.Mister and S.E. Tavares, "On the Design of Linear Transformations for Substitution Permutation Encryption Networks", *in Workshop on Selected Areas in Cryptography* (SAC'97), pp. 40–48, 1997.

[4] A. R. Rao, P. Bhimasankaram, "*Linear Algebra*", Second Edition, Hindustan Book Agency.

[5] F. J. MacWilliams, N. J. A. Sloane, Bell Laboratories, Murray Hill, NJ 07974, U. S. A, "*The Theory of Error-Correcting Codes*", North-holland publishing company amsterdam, New York, Oxford, 1977.

[6] J. Daemen, L. Knudsen, and V. Rijmen, "The block cipher Square", *Fast Software Encryption* (FSE'97), LNCS 1267, pp. 149–165, Springer-Verlag, 1997.

[7] K. C. Gupta and I. G. Ray, "On Constructions of Involutory MDS Matrices", In *AFRICACRYPT* 2013, pp. 43–60, Springer 2013.

[8] K. C. Gupta, I. G. Ray, "On Constructions of MDS Matrices From Circulant-Like Matrices For Lightweight Cryptography", *Technical Report* No. ASU/2014/1, Dated : 14th February, 2014.

[9] K. C. Gupta, I. G. Ray, "On constructions of mds matrices from companion matrices for lightweight cryptography". In A. Cuzzocrea, C. Kittl, D. E. Simos, E. Weippl, L. Xu, A. Cuzzocrea, C. Kittl, D. E. Simos, E. Weippl, and L. Xu, editors, CD-ARES Workshops, volume 8128 of *Lecture Notes in Computer Science*, pp. 29–43. Springer, 2013.

[10] M. Sajadieh, M. Dakhilalian, H. Mala, B. Omoomi, "On construction of involutory MDS matrices from Vandermonde Matrices in GF(2q)", *Designs, Codes and Cryptography*, 64, 287–308.

[11] P. Junod and S. Vaudenay, "Perfect diffusion primitives for block cipher – Building efficient MDS matrices", *in Selected Areas in Cryptology (SAC 2004)*, LNCS 3357, pp. 84–99, Springer-Verlag, 2004.

[12] R. Elumalai, Dr. A. R. Reddy, "Improving Diffusion Power of AES Rijndael with 8×8 MDS Matrix", *International Journal of Scientific & Engineering Research*, Vol. 2, Issue 3, March-2011.

[13] R. Klima, N. Sigmon, E. Stitzinger, "Applications of abstract algebra with maple", *CRC Press*, Boca Raton London New York Washington, D.C, 1999.

[14] S. Lin and D. Costello, "*Error Control Coding: Fundamentals and Applications*", Prentice Hall, 2004.

[15] T. P. Berger, A.V. Ourivski, "Construction of new MDS codes from Gabidulin codes", *LACO, University of Limoges, France*, 2013.

[16] A. Venkateswarlu, A. Kesarwani and S. Sarkar, "On the Lower Bound of Cost of MDS Matrices", *IACR Transactions on Symmetric Cryptology*, pp. 266–290, 2022.

[17] X. Zhou and T. Cong, "Construction of generalized-involutory MDS matrices", *Cryptology ePrint Archive*, 2022.
[18] K. C. Gupta, S. K. Pandey and Venkateswarlu, "Almost involutory recursive MDS diffusion layers", *Design, Codes and Cryptography*, 87 (2018), 609–626.
[19] S. Kolay and D. Mukhopadhyay, "Lightweight diffusion layer from the kth root of the mds matrix", *IACR Cryptology ePrint Archive*, vol. 498, 2014.
[20] T. T Luong, "Constructing effectively mds and recursive mds matrices by reed-solomon codes", *Journal of Science and Technology on Information Security of Viet Nam Government Information Security Commission*, vol.3, no. 2, pp. 10–16, 2016.
[21] T. T. Luong, N. N. Cuong and B. D. Trinh, "4x4 Recursive MDS Matrices Effective for Implementation from Reed-Solomon Code over GF(q) Field", *International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences – MCO 2021*, pp 386–391, 2021.
[22] K. C. Gupta, S. K. Pandey and S. Samanta, "Construction of Recursive MDS Matrices Using DLS Matrices", *In Progress in Cryptology-AFRICACRYPT 2022: 13th International Conference on Cryptology in Africa*, AFRICACRYPT 2022, Springer Nature Switzerland, pp. 18–22, 2022.
[23] P. Freyre, N. Díaz, R. Díaz and C. Pérez, "Random generation of MDS matrices", *Proceedings of Current Trends in Cryptology CTCrypt2014*, Russia, 2014. DOI:10.13140/RG.2.1.5111.2160
[24] A. Kesarwani, S. Sarkar and A. Venkateswarlu, "Exhaustive search for various types of MDS matrices", *IACR Transactions on Symmetric Cryptology*, 2019, 231–256.
[25] M. Kurt Pehlivanoğlu, M. Ali Demir, F. Büyüksaraçoğlu Sakallı, S. Akleylek and M. Tolga Sakallı, "On the Construction Structures of Involutory MDS Matrices over ", *In Nonlinear Dynamics and Applications: Proceedings of the ICNDA*, Cham: Springer International Publishing, pp. 587–595, 2022.
[26] G. Tuncay, F. B. Sakallı, M. K. Pehlivanoğlu, G. G. Yılmazgüç, S. Akleylek and M. T. Sakallı, "A new hybrid method combining search and direct based construction ideas to generate all involutory maximum distance separable (MDS) matrices over binary field extensions", *PeerJ Computer Science*, 9, 2023, e1577.

Tran Thi Luong received a Bachelor's degree from Hanoi University of Science, Hanoi, Vietnam, in 2006; a Master's degree in cryptographic technique at the Academy of Cryptography Techniques, Hanoi, Vietnam, in 2012; Ph.D. degree in cryptographic technique at the Academy of Cryptography Techniques, Hanoi, Vietnam in 2019. Now, she is Vice Dean of Information Security Department of the Academy of Cryptography Techniques. Her research interests include Cryptography, Coding theory, and Information Security.
Email: luongtranhong@gmail.com