Estimation for the Number of MDS Matrices, Recursive MDS Matrices and Symmetric Recursive MDS Matrices from the Reed-Solomon Codes

Tran Thi Luong
Academy of Cryptography Techniques, No.141 Chien Thang road, Tan Trieu, Thanh Tri, Hanoi, Vietnam
Correspondence: Tran Thi Luong, luongtranhong@gmail.com
Communication: received 18 Oct 2022, revised 28 Jan 2023, accepted 16 Feb 2023
DOI: 10.32913/mic-ict-research.v2023.n1.1105

***Abstract*: The diffusion layer of the SPN block ciphers is usually built on the basis of the MDS (Maximum Distance Separable) matrices which is the matrix of the maximum distance separable code (MDS code). MDS codes have long been studied in error correcting code theory and have applications not only in coding theory but also in the design of block ciphers and hash functions. Thanks to that important role, there have been many studies on methods of building MDS matrices. In particular, the recursive MDS matrices and the symmetric recursive MDS matrices have particularly important applications because they are very efficient for execution. In this paper, we will give an estimate of the number of MDS matrices, recursive MDS matrices and symmetric recursive MDS matrices built from Reed-Solomon codes. This result is meaningful in determining the efficiency from this method of building matrices based on the Reed-Solomon codes. From there, this method can be applied to find out many MDS matrices, secure and efficient symmetric recursive MDS matrices for execution to apply in current block ciphers. Furthermore, recursive MDS matrices can be efficiently implemented using Linear Feedback Shift Registers (LFSR), making them well suited for lightweight cryptographic algorithms, so suitable for limited resources application.**

Keywords: *MDS matrix, recursive MDS matrix, RS codes, number of MDS matrices.*

I. INTRODUCTION

In [17], C. Shannon introduced that diffusion and confusion are two must-have properties for the operation of any secure cryptographic algorithm. In classical ciphers, these two properties are achieved using substitution or permutation ciphers. In modern cryptography, the ciphers also use substitution and permutation in the form of SPN block ciphers.

The diffusion layer of SPN block ciphers is often built on top of MDS matrices because they provide maximum diffusion, making the block cipher resistant to many strong attacks such as: linear attack, differential attack. That is why MDS matrices have been used in many well-known block ciphers such as: AES, Twofish, Square, Shark, GOST R34.12-2015, Kalyna. etc

Currently, many methods have been studied to build MDS matrices such as: methods based on MDS codes (Reed-Solomon codes [5, 15], Gabidulin codes [4]), or based on particular matrices such as: Cauchy matrix, Vandermonde matrix, Hadamard matrix, circulant matrix, etc.

Recently, researchers are interested in recursive MDS matrices (power of a companion matrix) because they are very efficient for hardware and software implementations. There have been many works in the literature about recursive MDS matrices as in [1, 2, 3, 6, 7, 8, 9, 10, 16, 18]. In particular, Gupta et al., and Augot et al. have studied the construction of recursive MDS matrices and symmetric recursive MDS matrices [2, 6, 7, 8, 9] based on on the BCH codes. The authors also give an estimate for the number of symmetric recursive MDS matrices obtained by the BCH codes [6]. In general, the construction based on BCH codes is very complicated. In [11], we present a very simple and efficient method of building MDS matrices and recursive MDS matrices using Reed-Solomon (RS) codes that can obtain matrices of arbitrary size. In [12, 13], we presented methods to build symmetric recursive MDS matrices using RS codes. These matrices are very efficient for execution and our approach based on RS codes is extremely simple. Recursive MDS matrices can be efficiently implemented using LFSR registers, making them well suited for lightweight cryptographic algorithms. On

the other hand, symmetric recursive matrices are extremely efficient for hardware implementation, since one can use exactly the same LFSR in both encryption and decryption, saving on execution costs [9].

In this paper, we will give an estimate of the number of MDS matrices, recursive MDS matrices and symmetric recursive MDS matrices built from RS codes. This result is meaningful in determining the efficiency from this method of building matrices based on the RS codes. We also present some examples and experiments illustrating the above estimates and methods. From there, these methods can be applied to find out many MDS matrices, secure and efficient symmetric recursive MDS for execution to apply in current block ciphers.

Our contribution: In [19], we gave an estimate for the number of MDS matrices of size m over the general field $GF(p^r)$, but we have not given an estimate for the MDS matrices in general, or recursive MDS matrices, symmetric recursive MDS matrices from a particular construction method. In this paper, we make these estimates when building those matrices from RS codes, and compare how they are built from RS codes with other codes such as BCH codes or Gabidulin codes. From this work, it can be seen that our main contribution is to give a specific estimate for the number of MDS matrices, recursive MDS, symmetric recursive MDS matrices from RS codes. From there, we evaluate and compare our method with related works to see that building these matrices from RS codes is very efficient and simple. This makes a lot of sense in practice when the designer needs an efficient method and tool to find MDS matrices with good cryptographic properties, especially with low implementation cost in hardware and software, such as symmetric recursive MDS matrices from RS code.

This paper is organized as follows. In Section 2 presents preliminaries and related works. Section 3 gives an estimate of the number of MDS matrices, recursive MDS matrices and symmetric recursive MDS matrices built from RS codes and some examples. Section 4 is the conclusion.

II. PRELIMINARIES AND RELATED WORKS

1. MDS matrix

In error correcting code theory, for any linear code $C(n,k,d)$ where n is the length, k is the dimension, d is the minimum distance of the code, then there is a limit to be called Singleton bound: $d \le n-k+1$ [14]. And when this minimum distance d is equal to $n-k+1$ then the code is called MDS code. In coding theory, there is the following important theorem:

Theorem 1 ([14, 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, the MDS matrix is a very special matrix whose all square sub-matrices are nonsingular.

2. Recursive MDS matrices

The recursive MDS matrix [6, 9] is a matrix defined as a power of a companion matrix, that is, $A = S_f^l$, where S_f is a companion matrix corresponding to the polynomial $f(x) \in GF(q)[X]$ of degree l.

If the polynomial $g(x)$ is monic and has symmetric coefficients, then A is called a symmetric recursive MDS matrix [9].

In [11], we presented the **Algorithm 1** to build MDS matrices and the **Algorithm 2** to build recursive MDS matrices over $GF(2^m)$.

These two algorithms are presented again below.

Algorithm 1 ([11]).

INPUT: Size of the MDS matrices is l, the finite field is $GF(2^m)$ where $m \ge log_2(2l+1)$.

OUTPUT:$l \times l$ MDS matrices over $GF(2^m)$.

Detail of steps as follows:

Step 1: Constructing a $RS[n,k,d]$ code where $n = 2^m - 1$ by the following way:

- Select a value d satisfying: $l+1 \le d \le 2^m - l$.
- Calculate the dimension of the code: $k = n - d + 1 = 2^m - d, (\ge l)$.
- Find the generator matrix of the code.

Step 2: Change the generator matrix of the RS $[2^m-1, 2^m-d, d]$ code to the echelon form $G = [I|A]$. Matrix A obtained is a $(2^m-d)\times(d-1)$ one. Then, it can be taken any $l\times l$ submatrices of A. They are all MDS matrices of size l over $GF(2^m)$.

The following is the **Algorithm 2** in [11].

Algorithm 2 ([11]).

INPUT: Size of the MDS matrices is l, the finite field is $GF(2^m)$ where $m \ge log_2(2l+1)$.

OUTPUT: a $l \times l$ recursive MDS matrix over $GF(2^m)$.

Detail of steps as follows:

Step 1: Select $d = l+1$ (satisfying the condition $l+1 \le d \le 2^m - l$ of the **Algorithm 1**). Construct a $RS[2^m-1, 2^m-l-1, l+1]$ code (where $2^m - l - 1 \ge l$ or $2^m - 1 \ge 2l$ according to the Proposition 2).

Step 2: Change the generator matrix of this code to the echelon form $G = [I|A]$ where A is a $(2^m - d) \times (d - 1) = (2^m - l - 1) \times l$ matrix.

Step 3: Get the square submatrix A_1 of size l at l first rows of A.

Then A_1 is the $l \times l$ recursive MDS matrix over $GF(2^m)$ that we are looking for. In addition, denote the serial matrix corresponding with A_1 is S of size l. This matrix has the elements in the last row coinciding with the elements in the first row of A_1 and these matrices satisfy: $A_1 = S^l$.

III. ESTIMATING THE NUMBER OF MDS MATRICES, RECURSIVE MDS MATRICES, AND SYMMETRIC RECURSIVE MDS MATRICES FROM RS CODES

In [11], we gave the **Algorithm 1** to construct $l \times l$ MDS matrices using RS codes over $GF(2^m)$, provided that $m \geq log_2(2l + 1)$ and $l + 1 \leq d \leq 2^m - l$. Now, the number of $l \times l$ MDS matrices obtained from a class of RS codes according to the **Algorithm 1** is estimated by the following proposition:

Proposition 1. *Let $C[2^m - 1, 2^m - d, d]$ be a RS code class over $GF(q)$ for $q = 2^m$. Then, the number of $l \times l$ MDS matrices (for $m \geq log_2(2l + 1)$ and $l + 1 \leq d \leq 2^m - l$) obtained from the above code class is: $\leq \mathcal{M}(m) \times (q-1) \times \varphi(q - 1) \times C^l_{(2^m-d)} C^l_{(d-1)}$.*

Proof.

First, calculate the number of MDS codes (RS codes) obtained from the code class $C[2^m - 1, 2^m - d, d]$. The RS code $C[2^m - 1, 2^m - d, d]$ over the field $GF(q)$ where $q = 2^m$ and its primitive polynomial of degree m and α is a primitive element, will have the following generator polynomial :

$$g(x) = (x - \alpha^b)(x - \alpha^{b+1})...(x - \alpha^{b+d-2}) \quad (1)$$

where b is a preselected value ($b \in \mathbb{N}, b \geq 0$).

Denote the number of primitive polynomials of degree m is $\mathcal{M}(m)$. (2)

For each of these primitive polynomials of degree m, there will be a corresponding number of primitive elements is $\varphi(q - 1)$. (3)

Since the degree of α is $q - 1$, so $b \leq q - 1$, however the case $b = q - 1$ coincides with the case$b = 0$ in (1), we consider: $0 \leq b \leq q - 2$. Then, for a preselected value of d, it is possible to generate $q-1$ RS codes $RS[2^m-1, 2^m-d, d]$ with different generator polynomials as (1) corresponding to $q - 1$ values of b. (4)

From (2), (3) and (4), the number of MDS codes (RS codes) obtained over $GF(q)$ is denoted by $\mathcal{L}_{RS}$ calculated as follows:

$$\mathcal{L}_{RS} = \mathcal{M}(m) \times (q - 1) \times \varphi(q - 1) \quad (5)$$

Next, for a particular RS code $[RS[2^m-1, 2^m-d], d]$, we find the number of $l \times l$ MDS matrices that can be obtained from this code.

According to the assumption, $m \geq log_2(2l+1)$ and $l+1 \leq d \leq 2^m - l$, apply the **Algorithm 1** [11], we get a matrix A of size $(2^m - d) \times (d - 1)$. Then, any $l \times l$ square submatrix of A can be taken, which are all $l \times l$ MDS matrices over $GF(q)$.

The number of square submatrices of size $l \times l$ (in any l rows and l columns) of A is given by:

$$C^l_{(2^m-d)} C^l_{(d-1)} \quad (6)$$

From (5) and (6), it follows that the maximum number of $l \times l$ MDS matrices that can be generated from RS codes over $GF(q)$ is $\mathcal{M}(m) \times (q-1) \times \varphi(q-1) \times C^l_{(2^m-d)} C^l_{(d-1)}$.

Since among these MDS matrices, there may also be cases where there are duplicate matrices, so the number above is the largest possible number. □

Corollary 1. Let $C[2^m - 1, 2^m - d, d]$ be a RS code class over $GF(q)$ where $q = 2^m$. Then, the total number of MDS matrices of size $\leq l$ (for $m \geq log_2(2l + 1)$ and $l + 1 \leq d \leq 2^m - l$) obtained from the above code class is: $\leq \mathcal{M}(m) \times (q - 1) \times \varphi(q - 1) \times \sum_{i=1}^{l} \left(C^i_{(2^m-d)} C^i_{(d-1)} \right)$.

In the following, we will estimate the number of recursive MDS matrices and symmetric recursive matrices that can be obtained using RS codes through the following proposition.

Example 1. Consider the RS [255, 241, 15] code class over $GF(2^8)$. (Note that, in this example $d = 15$, consider $l = 8 < d$). Choose a primitive polynomial of degree 8 of $GF(2^8)$ is: $x^8 + x^4 + x^3 + x^2 + 1$, where α is a primitive element of $GF(2^8)$. Construct a RS [255, 241, 15] code over $GF(2^8)$ with this primitive polynomial according to the **Algorithm 1** [11]. We get the generator matrix of this code in the echelon form $G = [I|A]$. In which, A is a 241×14 matrix. Then, any 8×8 submatrices of A can be taken, which are all 8×8 MDS matrices over $GF(2^8)$.

Figure 1 is a 14×14 submatrix in the top corner of the matrix A. This matrix itself is also an MDS matrix. We can also take any 8×8 submatrix of this matrix to obtain 8×8 MDS matrices.

With this primitive polynomial, we can choose another primitive element of $GF(2^8)$ and construct the corresponding RS code, to obtain a different matrix A.

We can also choose other primitive polynomials of $GF(2^8)$ and do the same. Therefore, the number of 8×8 MDS matrices that can be obtained from the RS code is very large, as shown in **Proposition 1**.

Proposition 2. *Let $C[2^m - 1, 2^m - d, d]$ be an RS code class over $GF(q)$ where $q = 2^m$. Then, the number of $l \times l$*

recursive MDS matrices (for $m \geq log_2(2l+1)$ *and* $l + 1 \leq d \leq 2^m - l$*) obtained from the above code class is:* $\mathcal{M}(m) \times (q-1) \times \varphi(q-1)$.

Proof.

According to the proof of **Proposition 1**, the number of MDS codes (RS codes) obtained over $GF(q)$ is $\mathcal{L}_{RS} = \mathcal{M}(m) \times (q-1) \times \varphi(q-1)$. On the other hand, according to the **Algorithm 2** in [11], from each RS code like this, we can generate a recursive MDS matrix of size l. So the proposition is proved. □

The number of symmetric recursive MDS matrices that can be obtained using RS codes will be estimated by the following proposition.

$$\begin{bmatrix}
1A & 9C & BC & 2B & AF & 24 & 4D & 24 & 2E & 64 & 14 & B7 & D8 & 1C \\
5 & F1 & D0 & DF & 9E & 78 & FB & 9A & 2B & AA & C9 & 9C & 7F & 95 \\
4E & 53 & F0 & F9 & D9 & A7 & F1 & C2 & 48 & BC & 61 & B6 & A & 68 \\
79 & 64 & F8 & AE & 24 & E5 & A & CD & 49 & 6B & CF & 15 & 3E & 1E \\
31 & B7 & 4D & CD & 58 & BB & A0 & 95 & 9E & 5 & EE & 34 & 70 & 4B \\
30 & C8 & DB & 67 & B8 & FC & 94 & 4 & C5 & D & F0 & 3D & CB & E7 \\
6E & C8 & 81 & 35 & CC & B2 & B4 & 9E & B6 & B0 & 60 & E6 & 1 & CD \\
ED & 4D & 73 & 75 & 88 & 47 & 67 & 3F & B4 & 10 & EF & B7 & FB & 78 \\
C4 & F2 & E9 & A7 & 8A & CE & 4E & 21 & 6E & 99 & 3E & 24 & 3E & 32 \\
EE & BE & 86 & 3C & 82 & 4C & A5 & 88 & E & B4 & 56 & 91 & 8A & 7C \\
AC & BB & D0 & FE & 45 & 54 & 6C & 73 & 61 & AE & CA & 7B & 5F & 33 \\
F4 & 4A & 73 & 2E & 74 & A7 & 72 & 8E & 72 & DF & 75 & D2 & D & 1 \\
1A & 68 & F6 & 58 & 81 & 50 & EA & 56 & A0 & 16 & CB & C2 & A & 11 \\
A7 & B3 & DB & 57 & D5 & DF & B9 & B4 & A2 & CA & 5F & C3 & 1B & CB
\end{bmatrix}$$

Figure 1. A 14×14 MDS matrix

Proposition 3. *Let* $C[2^m - 1, 2^m - d, d]$ *be an RS code class over* $GF(q)$ *where* $q = 2^m$*. Then, the number of* $l \times l$ *symmetric recursive MDS matrices (for* $m \geq log_2(2l+1)$ *and* $l+1 \leq d \leq 2^m - l$*) obtained from the above code class is:* $\mathcal{M}(m) \times \varphi(q-1)$.

Proof.

According to the results of [12, 13], we showed that, from a particular RS code over $GF(q)$, there is only a single value of b that makes the generatore polynomial of the RS code of the form (1) symmetric. Combined with the proof of Proposition 1, inferring that the number of symmetric recursive MDS matrices that can be obtained using the RS code class $[2^m - 1, 2^m - d, d]$ over $GF(q)$ is $\mathcal{M}(m) \times \varphi(q-1)$. □

Example 2. Consider the RS [255, 247, 9] code class over $GF(2^8)$. (Note that, in this example, $d = 9, l = 8$ satisfy the condition $d = l + 1$ of the **Algorithm 2** in [11]). We have the number of primitive polynomials of degree 8 over $GF(2)$ is: $M(8) = 16$. The number of primitive elements of $GF(2^8)$ is $\varphi(255) = 200$. Then the number of possible RS codes of this code class is: $\mathcal{L}_{RS} = 16 \times 255 \times 200 = 816000$. It follows that the number of 8×8 recursive MDS matrices generated using these RS codes is equal to the number of RS codes and is equal to 816000 matrices.

The number of 8×8 symmetric recursive MDS matrices generated using these RS codes is $16 \times 200 = 3200$ matrices.

Example 3. To construct 4×4 recursive MDS matrices using the RS codes. We build the RS $[255, 251, 5]$ codes over $GF(2^8)$ with the generator polynomial of the following form: $g(x) = (x - \alpha^b)(x - \alpha^{b+1})(x - \alpha^{b+1})(x - \alpha^{b+3})$ where $b \in \mathbb{N}, 0 \leq b \leq q - 2$. Because the number of possible recursive matrices in this case is 816000, so it is very large. Therefore, for illustration, we choose $b = 1$. Since the field $GF(2^8)$ has 16 primitive polynomials, we experiment on Maple to build recursive MDS matrices using RS [255,251,5] codes over $GF(2^8)$ corresponding to these 16 primitive polynomials with the value of b is $1(b = 1)$ according to the **Algorithm 2** in [11]. We obtain 16 recursive MDS matrices of size 4 presented in the Table I.

Example 4. To construct 8×8 symmetric recursive MDS matrices from the RS codes, we build the $RS[255, 247, 9]$ code class over $GF(2^8)$. Then the value of b for the generator polynomial of the RS code being monic and having the symmetric coefficients is $b = 124$ [12]. Choose the primitive polynomial of $GF(2^8)$ to be $x^8+x^6+x^5+x^3+1$, where α is a primitive element of $GF(2^8)$. We build a corresponding RS code $[255, 247, 9]$ according to the **Algorithm 2** [11]. Then we can obtain a symmetric recursive matrix and the corresponding companion matrix as follows:

$$\begin{bmatrix}
01 & 1B & 45 & F3 & 8E & F3 & 45 & 1B \\
1B & 2D & B3 & 62 & A6 & A9 & 5B & 69 \\
69 & DC & 67 & AB & 3B & BE & E3 & 9C \\
9C & D3 & E3 & 47 & 77 & 1B & 81 & 59 \\
59 & 59 & 57 & 4D & 2B & D9 & 9F & 44 \\
44 & EA & 82 & 26 & EB & 5A & 02 & 2C \\
2C & 2B & 3E & 18 & 57 & 71 & 8E & 6D \\
6D & 87 & 1C & 51 & AB & 38 & 46 & 25
\end{bmatrix}$$

and

$$\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 1B & 45 & F3 & 8E & F3 & 45 & 1B
\end{bmatrix}$$

We can choose 15 other primitive polynomials of $GF(2^8)$

TABLE I
LIST OF 16 RECURSIVE MDS MATRICES USING RS CODES

No	**Primitive polynomial**	**4×4 recursive MDS matrices**	**Corresponding Companion matrix**
1	$x^8+x^4+x^3+x^2+1$	$\begin{bmatrix} 74 & E7 & D8 & 1E \\ B1 & A1 & 82 & 91 \\ FF & 7E & 70 & DA \\ A8 & 10 & 50 & 29 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 74 & E7 & D8 & 1E \end{bmatrix}$
2	$x^8+x^6+x^5+x^3+1$	$\begin{bmatrix} CD & 7B & D8 & 1E \\ E5 & 1B & 71 & E5 \\ F2 & 72 & 9F & 1F \\ 28 & 5F & A0 & BC \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ CD & 7B & D8 & 1E \end{bmatrix}$
3	$x^8+x^7+x^6+x^5+x^2+x+1$	$\begin{bmatrix} 52 & 0E & D8 & 1E \\ 27 & E6 & DD & 6B \\ FF & 2C & 7A & 1D \\ D1 & 59 & 70 & EB \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 52 & 0E & D8 & 1E \end{bmatrix}$
4	$x^8+x^5+x^3+x+1$	$\begin{bmatrix} 97 & EE & 75 & A7 \\ 21 & 44 & 85 & 30 \\ 04 & 68 & D8 & F3 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ AC & BD & D8 & 1E \end{bmatrix}$
5	$x^8+x^6+x^5+x^2+1$	$\begin{bmatrix} F1 & 6F & D8 & 1E \\ F3 & EA & 1D & E9 \\ 28 & 58 & 65 & 9B \\ 1A & E6 & 54 & EB \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ F1 & 6F & D8 & 1E \end{bmatrix}$
6	$x^8+x^6+x^4+x^3+x^2+x+1$	$\begin{bmatrix} 23 & 21 & D8 & 1E \\ 03 & 1C & A8 & D3 \\ 3E & C4 & 6B & F3 \\ 7D & FA & 98 & 11 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 23 & 21 & D8 & 1E \end{bmatrix}$
7	$x^8+x^7+x^6+x+1$	$\begin{bmatrix} 8A & 46 & D8 & 1E \\ 17 & 42 & C2 & 4F \\ F5 & 5D & 78 & E4 \\ A2 & 4B & 0F & 11 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 8A & 46 & D8 & 1E \end{bmatrix}$
8	$x^8+x^6+x^5+x+1$	$\begin{bmatrix} EF & 65 & D8 & 1E \\ F2 & 26 & 2B & EF \\ C7 & 3F & 06 & D9 \\ CF & BC & B9 & 56 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ EF & 65 & D8 & 1E \end{bmatrix}$
9	$x^8+x^7+x^2+x+1$	$\begin{bmatrix} 95 & CE & D8 & 1E \\ 37 & 0C & 74 & 0B \\ 05 & 8E & 3F & A6 \\ 5A & C3 & D6 & 83 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 95 & CE & D8 & 1E \end{bmatrix}$
10	$x^8+x^7+x^6+x^3+x^2+x+1$	$\begin{bmatrix} A2 & 09 & 90 & 61 \\ EF & 55 & CE & EA \\ 57 & 68 & BE & A5 \\ AC & B7 & 11 & B9 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ A2 & 9 & 90 & 61 \end{bmatrix}$
11	$x^8+x^6+x^3+x^2+1$	$\begin{bmatrix} 79 & 14 & 0D & 29 \\ 1A & D7 & 9C & 35 \\ 92 & 49 & D3 & 0F \\ 3D & 5E & 02 & 09 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 79 & 14 & 0D & 29 \end{bmatrix}$
12	$x^8+x^7+x^3+x^3+1$	$\begin{bmatrix} 25 & 23 & 1D & 7E \\ 6F & 39 & 6E & 4F \\ 78 & FA & 2F & C3 \\ 2F & 4E & 80 & A2 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 25 & 23 & 1D & 7E \end{bmatrix}$
13	$x^8+x^7+x^6+x^5+x^4+x^2+1$	$\begin{bmatrix} 3E & 2A & D8 & 1E \\ 8B & D8 & 33 & 79 \\ 0D & D3 & 64 & B6 \\ CA & F6 & 08 & B7 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3E & 2A & D8 & 1E \end{bmatrix}$
14	$x^8+x^7+x^5+x^3+1$	$\begin{bmatrix} 31 & 59 & 2C & CE \\ 40 & 81 & CD & FD \\ AB & AA & CF & 6E \\ 7C & 72 & F4 & 10 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 31 & 59 & 2C & CE \end{bmatrix}$
15	$x^8+x^5+x^3+x^2+1$	$\begin{bmatrix} B1 & AA & 16 & E4 \\ 6C & 43 & 80 & 48 \\ DC & F2 & 48 & 25 \\ DA & A8 & 1F & 6E \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ B1 & AA & 16 & E4 \end{bmatrix}$
16	$x^8+x^6+x^5+x^4+1$	$\begin{bmatrix} 97 & 71 & 42 & 7E \\ AB & 5F & 81 & 18 \\ B4 & DF & DD & 14 \\ EE & FA & 1C & BA \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 97 & 71 & 42 & 7E \end{bmatrix}$

and do the same, and for each primitive polynomial we can choose other primitive elements. Then according to **Proposition 3**, we can obtain 3200 symmetric recursive MDS matrices like this. These matrices are very efficient

for execution.

Remark. Although the number of recursive MDS matrices and symmetric recursive MDS matrices are significantly smaller than that of MDS matrices when they are all generated using RS codes. However, it can be seen that the number of recursive MDS matrices and symmetric recursive MDS matrices is quite large. It is very convenient to construct these matrices using RS codes . It also shows that building these matrices from RS codes is extremely simple and efficient. Designers can select many good candidates from these matrices, especially those that are efficient for implementation and have low implementation costs.

IV. COMPARE WITH THE RESULTS IN [6]

In [6], the authors constructed symmetric recursive and recursive MDS matrices based on BCH codes. With the assumption $q = 2^s$, in the case of $n|(q-1)$, [6] showed the number of BCH MDS codes of length n and dimension $(n-k)$ over $\mathbb{F}_q$ equal to $n\frac{\varphi(n)}{2}$ and the number of symmetric (matrix) solutions in S^n equal to $\frac{\varphi(n)}{2}$, and also showed the corresponding generator polynomial given by (detailed in [6]):

$$g_{(i,l)}^{(n)}(X) = \prod_{j=0}^{k-1}\left(X - \beta_{n,i}^{l+j}\right) \qquad (*)$$

where $\beta_{n,i} = \alpha^i$ and α is the root element of the group of $n-th$ roots of unity in $\mathbb{F}_q$, $\varphi(n)$ is the Euler's totient function that is the number of positive integers less than n and co-prime to n .

In general, the construction based on the BCH codes is very complicated (as shown in [4]), while the construction based on the RS codes is very simple, only needing a primitive element α of the field, and a given parameter b to be able to build the generator polynomial of the RS code according to the formula (1). From this, it is possible to generate a large number of recursive and symmetric recursive MDS matrices.

On the other hand, by the method of [6], for $n = q-1$, the number of BCH MDS codes of [6] obtained is $(q-1).\frac{\varphi(q-1)}{2}$, the number of the monic and symmetric polynomials in [6] is only $\frac{\varphi(q-1)}{2}$, while our result is $\varphi(q-1)$. Therefore, our monic and symmetric polynomials are twice as high as theirs for the case $n = q-1$. Furthermore, according to the **Proposition 3**, we also obtain $\mathcal{M}(m) \times \varphi(q-1)$ recursive and symmetric recursive MDS matrices by RS codes. The efficiency and simplicity of building these matrices from RS codes can be demonstrated through **Algorithm 1** and **Algorithm 2** ([11]). Both of these algorithms have the complexity of approximately $O(dlogd) + O\left((n-d+1)^3\right)$ where $n = 2^m - 1$ with the finite field is $GF(2^m)$ and $d < n$. Therefore, these two algorithms run very fast. On the other hand, the implementation steps of these two algorithms are also quite simple.

Therefore, in practice of generating recursive MDS matrices, symmetric recursive MDS matrices from RS codes is very convenient and efficient. However, in [6], the authors use BCH codes to generate recursive MDS matrices but their algorithms are very complex. In [3, 4], the authors used Gabidulin codes to generate the MDS matrices and recursive MDS ones. However, the Gabidulin codes are built over $GF(q^m)$ that is a m-order extension of the finite field $GF(q)$. Therefore, the construction of this code is also much more complicated than the RS codeS. With our method, designers can select many good candidates from these matrices, especially those that are efficient for implementation and have low implementation costs.

V. CONCLUSION

In this paper, we give estimates for the number of MDS matrices, recursive MDS matrices and symmetric recursive MDS matrices built from Reed-Solomon codes. We also present some examples and experiments illustrating the above estimates and methods.

These results show the efficiency from this method of building MDS matrices, recursive MDS matrices and symmetric recursive MDS matrices based on Reed-Solomon codes. From there, this method can be applied to find out many MDS matrices, secure and efficient symmetric recursive MDS matrices for execution to apply in current block ciphers. In particular, recursive MDS matrices can be implemented efficiently using LFSR registers, in addition symmetric recursive matrices are extremely efficient for hardware implementation, since one can use exactly the same LFSR in both encryption and decryption. Therefore, they are well suited for lightweight cryptographic algorithms, suitable for limited resources applications.

REFERENCES

[1] Augot D., Finiasz M., "Exhaustive search for small dimension recursive MDS diffusion layers for block ciphers and hash functions", *in 2013 IEEE International Symposium on Information Theory Proceedings (ISIT)*. IEEE, 2013, pp.1551-1555.

[2] Augot D., Finiasz M., "Direct construction of recursive MDS diffusion layers using shortened BCH codes". In: *FSE* 2014, LNCS, vol. 8540, pp. 3–17. Springer (2015).

[3] Berger T. P., "Construction of recursive MDS diffusion layers from Gabidulin codes". In: *INDOCRYPT* 2013, LNCS, vol. 8250, pp. 274–285. Springer (2013).

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

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

[6] Gupta K.C., Pandey S.K., Venkateswarlu A., "On the direct construction of recursive MDS matrices". *Des. Codes Cryptogr*. 82(1–2), 77–94 (2017).
[7] Gupta K.C., Pandey S.K., Venkateswarlu A., "Towards a general construction of recursive MDS diffusion layers". *Des. Codes Cryptogr*. 82(1–2), 179–195 (2017).
[8] Gupta K.C., Ray I.G., "On constructions of MDS matrices from companion matrices for lightweight cryptography". In: *CD-ARES Workshops* 2013, LNCS, vol. 8128, pp. 29–43. Springer (2013).
[9] Gupta K.C., Pandey S.K., Venkateswarlu, "Almost involutory recursive MDS diffusion layers", *Design, Codes and Cryptography*, 87 (2018), 609-626.
[10] Kolay S., Mukhopadhyay D., "Lightweight diffusion layer from the kth root of the mds matrix", *IACR Cryptology ePrint Archive*, vol. 498, 2014.
[11] Luong T. T., "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.
[12] Luong T. T., Cuong N. N., and Tho H. D., "Constructing Recursive MDS Matrices Effective for Implementation from Reed-Solomon Codes and Preserving the Recursive Property of MDS Matrix of Scalar Multiplication", *Journal of Informatics and Mathematical Sciences*, Vol. 11, No. 2, pp. 155–177, 2019.
[13] Luong T. T., Cuong N. N., and Trinh B. D., "4×4 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.
[14] MacWiliams F.J, Sloan N.J., "The Theory of Error-Correcting Codes", *North-holland Publishing Company Amsterdam-New York- Oxford*, Third Printing, 1981.
[15] Rijmen V., Daemen J., Preneel B., Bosselaers A., De Win E., "The cipher Shark, in Fast Software Encryption". Springer, 1996, pp. 99-111.
[16] Sajadieh M., Dakhilalian M., Mala H., Sepehrdad P., "Recursive diffusion layers for block ciphers and hash functions". In: *FSE* 2012, LNCS, vol. 7549, pp. 385–401. Springer (2012).
[17] Shannon C. E., "Communication theory of secrecy systems", *Bell System Technical Journal*, vol. 28, no. 4, pp. 656-715, 1949.
[18] Wu S., Wang M., Wu W., "Recursive diffusion layers for (lightweight) block ciphers and hash functions". In: *SAC* 2013, LNCS, vol. 7707, pp. 355–371. Springer (2013).
[19] T. T. Luong, N. N. Cuong, "Direct exponent and scalar multiplication transformations of mds matrices: some good cryptographic results for dynamic diffusion", *Journal of Computer Science and Cybernetics*, vol.32, no.1, pp. 1–17, 2016.

Tran Thi Luong received Bachelor of Mathematics and Informatics of The Ha Noi university of Sicence in 2006; Master degreee in cryptographic technique at Academy of Cryptographic Techniques in 2012; Ph.D degree in cryptographic technique at Academy of Cryptographic Techniques in 2019;
Recent research direction: Cryptogaphy, Coding theory and Information Security.
Email: luongtranhong@gmail.com