Journal of Informatics and Mathematical Sciences
Vol. 11, No. 2, pp. 155–177, 2019
ISSN 0975-5748 (online); 0974-875X (print)
Published by RGN Publications
DOI: 10.26713/jims.v11i2.957

http://www.rgnpublications.com

Research Article

Constructing Recursive MDS Matrices Effective for Implementation from Reed-Solomon Codes and Preserving the Recursive Property of MDS Matrix of Scalar Multiplication

Tran Thi Luong, Nguyen Ngoc Cuong and Hoang Duc Tho*

Academy of Cryptographic Techniques, Ha Noi, Viet Nam
***Corresponding author:** hdtho@bcy.gov.v

Abstract. MDS matrices from *Maximum Distance Separable codes* (MDS codes) and MDS matrix transformations have important applications in cryptography. However, MDS matrices always have a large description and cannot be sparse, causing costly hardware/software implementations. Recursive MDS matrices allow to solve this problem as they can be a power of a simple serial matrix, so there is a compact description suitable even for constrained processing environments. In this paper, the method for constructing recursive MDS matrices effective for implementation from Reed-Solomon codes is presented. In addition, preserving the recursive property of MDS matrix of scalar multiplication transformation is given. The recursive MDS matrices effective for implementation are meaningful in hardware implementation, and the ability to preserve recursive property of MDS matrix of scalar multiplication transformation also has important applications for efficiently building dynamic block ciphers to improve the security of block ciphers.

Keywords. MDS matrix; Recursive MDS matrices; RS codes

MSC. 11T71; 14G50; 68P25

Received: June 28, 2018 **Accepted:** October 2, 2018

1. Introduction

Using MDS matrices in block ciphers was first introduced by Serge Vaudenay in FSE'95 [13] as a linear case of multipermutations. These multipermutations characterize the notion of perfect diffusion [12] which requires that the change of any t out of m input bits must affect at least $m-t+1$ output bits.

For block ciphers, the security against strong attacks (such as linear and differential attacks) depends on branchnumber [5,15] of diffusion layer. The larger the branch number, the higher the security.As MDS matrices give maximum branch numbers for the linear transformations corresponding with them, they have been used for diffusion layer in many block ciphers such as: AES, SHARK, Square, Twofish, Anubis, Khazad, Manta, Hierocrypt. They are also used in stream ciphers like MUGI and cryptographic hash functions like WHIRLPOOL.

In addition, recursive MDS matrices (powers of serial matrices) [9] have been studied by many authors in the literature because of its important applications in lightweight cryptography, such as [1,2,6,11,14]. However, according to these studies, searching for such recursive MDS matrices required to perform an exhaustive search on families of serial matrices, thus limiting the size of MDS matrices one could look for [2] or to use some other rather complex methods such as constructing recursive MDS matrices from shortened BCH codes [1]. In [8], we gave a method for efficiently and simply constructing recursive MDS matrices from Reed-Solomon (RS) codes, but not to mention to finding recursive MDS matrices effective for implementation from this method.

To further enhance the security of the block ciphers, some MDS matrix transformations have been studied to generate dynamic block ciphers later such as: scalar multiplication [10], permutations of rows and columns [3,4], direct exponent [10]. The scalar multiplication for MDS matrix was first published by Murtaza and Ikram [10] but the authors did not show the preservation of some good cryptographic properties of MDS matrix by the transformation. In [7], we showed that the scalar multiplication is capable of preserving some good cryptographic properties of the MDS matrix, but not to mention to preservation of recursive property of MDS matrix of the scalar multiplication transformation.

In this paper, the method for constructing recursive MDS matrices effective for implementation (meaning that inverse diffusion layer can use the same circuit as the diffusion layer itself in hardware implementation) from Reed-Solomon codes is presented. In addition, preserving the recursive property of MDS matrix of the scalar multiplication transformation is given. The recursive MDS matrices effective for implementation are meaningful in hardware implementation, and the ability to preserve recursive property of MDS matrix of scalar multiplication transformation also has important applications for efficiently building dynamic block ciphers to improve the security of block ciphers.

The paper is organized as follows. In Section 2, preliminaries and related works are introduced. Section 3 presents the method for constructing recursive MDS matrices effective for implementation from Reed-Solomon codes and experimental results. Section 4 provides preserving the recursive property of MDS matrix of the scalar multiplication transformation. Finally, conclusion is given in Section 5.

2. Preliminaries and Related Works

2.1 RS Code

A RS code over $GF(q) = GF(p^m)$ is a BCH code of length $n = q-1$. Suppose α is a primitive element of the field. A RS code of length $n = q-1$ designed with distance d will have a corresponding generator polynomial of degree $d-1$ as follow:

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

where b is a pre-selected value ($b \in N, b \geq 1$).

In [9], the authors showed that a $RS[n,k,d]$ code generated from the polynomial of the form (1) is an MDS code i.e. it satisfies the condition: $d = n-k+1$.

2.2 Recursive MDS Matrix

It can be definited a general recursive MDS matrixanda recursive MDS matrix as a power of aserial matrix as follows:

Definition 1. Let $A = [a_{i,j}]_{m\times m}$, $a_{i,j} \in GF(p^r)$, be an MDS matrix. A is called a recursive MDS matrix if there exists a matrix S of size m over $GF(p^r)$ and a non-negative integer k such that: $A = S^k$.

Definition 2. Let $A = [a_{i,j}]_{m\times m}$, $a_{i,j} \in GF(p^r)$ be an MDS matrix. A is called a recursive MDS matrix as a power of a serial matrix if there exists a serial matrix S of size m over $GF(p^r)$ such that: $A = S^m$, where the serial matrix S associated with a polynomial $c(x) = z_0 + z_1x + z_2x^2 + \dots + z_{d-1}x^{d-1} + x^d$ has the following form:

$$S = Serial(z_0,\dots,z_{m-1}) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ z_0 z_1 z_2 & & \cdots & z_{m-1} & \end{bmatrix} \tag{2}$$

and the inverse matrix of S has the following form:

$$Serial(z_0,\dots,z_{m-1})^{-1} = \begin{bmatrix} \frac{z_1}{z_0} & \frac{z_2}{z_0} & & & \cdots & \frac{1}{z_0} \\ 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{3}$$

Notice that the polynomial $c(x)$ having the constant term equal to 1 ($z_0 = 1$) is particularly interesting as the diffusion layer and its inverse share the same coefficients:

$$Serial(z_0,\dots,z_{m-1})^{-1} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ 1 & z_1 z_2 & \cdots & z_{m-1} & \end{bmatrix}^{-1} = \begin{bmatrix} z_1 z_2 & & & & \cdots & z_{m-1} & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & \ddots & \vdots \\ 0 & 0 & 0 & & \cdots & 1 & 0 \end{bmatrix} \tag{4}$$

Definition 3. A matrix serial-like $\acute{S}$ of size m over $GF(p^r)$ has the following form:

$$\acute{S} = Serial\_like(z_0,\dots,z_{m-1},e) = \begin{bmatrix} 0 & e & 0 & \cdots & 0 \\ 0 & 0 & e & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & e \\ ez_0 & ez_1 & ez_2 & \cdots & ez_{m-1} \end{bmatrix} \tag{5}$$

where $e \neq 0$ is an arbitrary element in $GF(p^r)$.

The inverse matrix of $\acute{S}$ has the following form:

$$Serial\_like(z_0,\dots,z_{m-1},e)^{-1} = \begin{bmatrix} e^{-1}\frac{z_1}{z_0} & e^{-1}\frac{z_2}{z_0} & & \cdots & e^{-1}\frac{z_{m-1}}{z_0} & \frac{e^{-1}}{z_0} \\ e^{-1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & e^{-1} & 0 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & e^{-1} & 0 \end{bmatrix} \tag{6}$$

It can be seen that the serial-like matrix is very sparse similar to the serial matrix. Therefore, the advantage of recursive MDS matrices as powers of such matrices is that the diffusion layer can be implemented as a *linear feedback shift register* (LFSR) that is clocked m times, using a very small number of gates in hardware implementations, or a very small amount of memory for software. The inverse of the diffusion layer also benefits from a similar structure.

In particular, if $c(x)$ is a *symmetric* polynomial (i.e. having coefficients symmetric each other) and having the *constant term equal to* 1, the inverse diffusion layer with the recursive MDS matrix as a power of the serial matrix can use the exact same circuit as the diffusion layer itself by simply reversing the order of the input and output symbols. If the hardware implementation uses LFSR registers, encryption and decryption can use the exact same circuit thus saving hardware resources and implementation cost. Then, the encryption is done from left to right and decryption is done from right to left.

In [8], we presented a method for efficiently constructing recursive MDS matrices from the RS codes based on the following propositions:

Proposition 1. *If $m \times m$ MDS matrices can be generated from a $RS[2^r-1, 2^r-d, d]$ code over $GF(2^r)$ then r, m and d must satisfy: $r \geq \log_2(2m+1)$ and $m+1 \leq d \leq 2^r - m$.*

Proposition 2. *Let $C[n,k,d]$ be an MDS code (i.e. $d = n-k+1$). If $k \geq n-k$ then a recursive MDS matrix of size $n-k$ can be generated from this code.*

2.3 Recursive MDS Matrix

Murtaza and Ikram [10] defined the scalar multiplication for MDS matrix as follows:

Let $A = [A_1,\dots,A_m]^T$ be an MDS matrix and $A_i = [a_{i,1}\cdots a_{i,n}]$, $a_{i,j} \in F_q$. Denote vector $E = [e_i]$, $i = 1,2,dots,m$ where $e_i \neq 0 \in F_q$, $i = 1,2,\dots,m$. Then the scalar multiplication of E and A generates an MDS matrix denoted by:

$$EA = [e_1A_1 \dots e_mA_m]^T,$$

where $e_iA_i = [e_ia_{i,1}\cdots e_ia_{i,n}]$.

In [7], we define our scalar multiplication as follows:

Let $A = [a_{i,j}]_{m\times m}$, $a_{i,j} \in GF(p^r)$ be an MDS matrix. Denote vectors $E = (e_1, e_2, \dots, e_m)$, $F = (f_1, f_2, \dots, f_m)$, (where $e_i, f_i \in GF^*(p^r)$). Then the scalar multiplication of E, F and A generates an MDS matrix denoted by: $(E,F)(A) = [b_{i,j}]_{m\times m}$, where $b_{i,j} = e_i f_j a_{i,j}$.

3. Constructing Recursive MDS Matrices Effective for Implementation from *RS* Codes

In this section, specifying directly recursive MDS matrices effective for implementation from *RS* codes is presented.

Firstly, from (1) it can be found that $b \in N$, $b \geq 1$ is a pre-selected value. On the other hand $\alpha^{q-1} = 1$, it is to have:

$$1 \leq b \leq q-1$$

where $q = 2^r$ in this case.

In order to find the recursive MDS matrices built from the RS codes, just need to find the corresponding generator polynomials of these codes. Essentially these polynomials are the polynomials that generate the serial matrices corresponding to the recursive MDS matrices (see [8]).

The following proposition specifies the specific cases in which symmetric polynomials having the constant term equal to 1 can be directly selected from the *RS* codes without performing an exhaustive search for 4×4 recursive MDS matrices over $GF(2^4)$ and $GF(2^8)$.

Proposition 3. *On constructing* 4×4 *recursive MDS matrices over* $GF(2^4)$ *or* $GF(2^8)$ *from RS codes, the generator polynomial* $g(x)$ *of the form* (1) *is a symmetric polynomial having the constant term equal to* 1 *if and only if* $b = 6$ *or* $b = 126$*, respectively.*

Proof. For the construction of a 4×4 recursive MDS matrix from *RS* codes, the generator polynomial $g(x)$ of the form (1) is a polynomial of degree 4 and has the following form:

$$g(x) = (x+\alpha^b)(x+\alpha^{b+1})(x+\alpha^{b+2})(x+\alpha^{b+3}) \tag{7}$$

where $1 \leq b \leq q-1$, $b \in N$, for $q = 16$ or $q = 256$.

Expand the above expression, it is to have:

$$\begin{aligned} g(x) = x^4 &+ \alpha^b(1+\alpha+\alpha^2+\alpha^3)x^3 + \alpha^{2b+1}(\alpha^4+\alpha^3+\alpha+1)x^2 \\ &+ \alpha^b\alpha^{2b+3}(1+\alpha+\alpha^2+\alpha^3)x + \alpha^{4b+6}. \end{aligned} \tag{8}$$

From (8), $g(x)$ is symmetric and has the constant term equal to 1 if and only if:

$$\begin{cases} \alpha^b\alpha^{2b+3}(1+\alpha+\alpha^2+\alpha^3) = \alpha^b(1+\alpha+\alpha^2+\alpha^3) \\ \alpha^{4b+6} = 1 \end{cases} \tag{9}$$

$$\iff \begin{cases} \alpha^{2b+3} = 1 \\ \alpha^{4b+6} = 1 \end{cases} \tag{10}$$

where $1 \leq b \leq q-1$, $b \in N$.

Since α is a primitive element of the field, it is to infer that:

$$(10) \Longleftrightarrow \begin{cases} (q-1)|(2b+3) \\ (q-1)|(4b+6) \end{cases}$$

$$\Longleftrightarrow (q-1)|(2b+3) \tag{11}$$

where $1 \le b \le q-1$, $b \in N$.

- For $GF(2^4)$, it is to have:

$$(11) \Longleftrightarrow \begin{cases} 2b+3 = 15k \\ 1 \le b \le 15;\ k \ge 1;\ b,k \in N \end{cases}$$

$$\Longleftrightarrow \begin{cases} 2b+3 = 15k \\ 1 \le b = \frac{15k-3}{2} \le 15;\ k \ge 1;\ b,\ k \in N \end{cases}$$

$$\Longleftrightarrow \begin{cases} 2b+3 = 15k \\ 1 \le k \le 2;\ b,\ k \in N \end{cases}$$

$$\Longleftrightarrow \begin{cases} b = 6 \\ k = 1 \end{cases} \tag{12}$$

- For $GF(2^8)$, prove similarly from (11), it is to have:

$$\begin{cases} b = 126 \\ k = 1 \end{cases} \tag{13}$$

From (12), (13), the proposition is proven. □

Similarly, the following proposition can be proven for the cases of 8×8, 16×16 and 32×32 recursive MDS matrices over $GF(2^8)$ from the RS codes.

Proposition 4. *On constructing* 8×8, 16×16 *or* 32×32 *recursive MDS matrices over* $GF(2^8)$ *from RS codes, the generator polynomial* $g(x)$ *of the form* (1) *is a symmetric polynomial having the constant term equal to* 1 *if and only if* $b = 124$ *or* $b = 120$ *or* $b = 112$, *respectively.*

From Proposition 3 and Proposition 4, we found immediately the polynomials are both symmetric and have the constant term equal to 1. Table 1 shows 66 such polynomials that we found after experimenting on Maple for sizes of 4, 8, 16, 32 (the symbol a in Table 1 is a primitive element of the field).

For larger sizes, it is possible to use the *RS* codes to find the corresponding recursive MDS matrices in our way. It can be said that these are recursive MDS matrices effective for hardware implementation which may have important applications for cryptographic applications in general and in particular lightweight cryptography. With such matrices, just use the exact same circuit for encryption and decryption in hardware implementation, thus saving resources and implementation cost.

Table 1. Symmetric polynomials havingthe constant term equal to 1 from *RS*

№	Field GF	Primitive polynomial	Size of matrix	*RS* code	Symmetric polynomials havingthe constant term equal to 1
1	$GF(2^4)$	x^4+x^3+1	4×4	RS(15,11,5)	$g_6 = x^4 + a^{12}x^3 + a^{14}x^2 + a^{12}x + 1$
2	$GF(2^4)$	x^4+x+1	4×4	RS(15,11,5)	$g_6 := x^4 + a^3x^3 + ax^2 + a^3x + 1$
3	$GF(2^8)$	$x^8+x^4+x^3+x^2+1$	4×4	RS(255, 251, 5)	$g_{126} := x^4 + a^{201}x^3 + a^{246}x^2 + a^{201}x + 1$
4	$GF(2^8)$	$x^8+x^6+x^5+x^3+1$	4×4	RS(255, 251, 5)	$g_{126} := x^4 + a^{174}x^3 + a^{234}x^2 + a^{174}x + 1$
5	$GF(2^8)$	$x^8+x^7+x^6+x^5+x^2+x+1$	4×4	RS(255, 251, 5)	$g_{126} := x^4 + a^{216}x^3 + a^{68}x^2 + a^{216}x + 1$
6	$GF(2^8)$	$x^8+x^5+x^3+x+1$	4×4	RS(255, 251, 5)	$g_{126} := x^4 + a^{90}x^3 + a^{53}x^2 + a^{90}x + 1$
7	$GF(2^8)$	$x^8+x^6+x^5+x^2+1$	4×4	RS(255, 251, 5)	$g_{126} := x^4 + a^{60}x^3 + a^{128}x^2 + a^{60}x + 1$
8	$GF(2^8)$	$x^8+x^6+x^4+x^3+x^2+x+1$	4×4	RS(255, 251, 5)	$g_{126} := x^4 + a^{237}x^3 + a^{168}x^2 + a^{237}x + 1$
9	$GF(2^8)$	$x^8+x^7+x^6+x+1$	4×4	RS(255, 251, 5)	$g_{126} := x^4 + a^{87}x^3 + a^{208}x^2 + a^{87}x + 1$
10	$GF(2^8)$	$x^8+x^6+x^5+x+1$	4×4	RS(255, 251, 5)	$g_{126} := x^4 + a^{207}x^3 + a^{245}x^2 + a^{207}x + 1$
11	$GF(2^8)$	$x^8+x^7+x^2+x+1$	4×4	RS(255, 251, 5)	$g_{126} := x^4 + a^{168}x^3 + a^{47}x^2 + a^{168}x + 1$

Table Contd.

12	$GF(2^8)$	$x^8+x^7+x^6+x^3+x^2+x+1$	4×4	RS(255, 251, 5)	$g_{126}:=x^4+a^{39}x^3+a^{187}x^2+a^{39}x+1$
13	$GF(2^8)$	$x^8+x^6+x^3+x^2+1$	4×4	RS(255, 251, 5)	$g_{126}:=x^4+a^{195}x^3+a^{127}x^2+a^{195}x+1$
14	$GF(2^8)$	$x^8+x^7+x^3+x^2+1$	4×4	RS(255, 251, 5)	$g_{126}:=x^4+a^{48}x^3+a^{10}x^2+a^{48}x+1$
15	$GF(2^8)$	$x^8+x^7+x^6+x^5+x^4+x^2+1$	4×4	RS(255, 251, 5)	$g_{126}:=x^4+a^{18}x^3+a^{87}x^2+a^{18}x+1$
16	$GF(2^8)$	$x^8+x^7+x^5+x^3+1$	4×4	RS(255, 251, 5)	$g_{126}:=x^4+a^{165}x^3+a^{202}x^2+a^{165}x+1$
17	$GF(2^8)$	$x^8+x^5+x^3+x^2+1$	4×4	RS(255, 251, 5)	$g_{126}:=x^4+a^{81}x^3+a^{21}x^2+a^{81}x+1$
18	$GF(2^8)$	$x^8+x^6+x^5+x^4+1$	4×4	RS(255, 251, 5)	$g_{126}:=x^4+a^{54}x^3+a^{9}x^2+a^{54}x+1$
19	$GF(2^8)$	$x^8+x^4+x^3+x^2+1$	8×8	RS(255, 254, 9)	$g_{124}:=x^8+a^{44}x^7+a^{231}x^6+a^{70}x^5+a^{235}x^4+a^{70}x^3+a^{231}x^2+a^{44}x+1$
20	$GF(2^8)$	$x^8+x^6+x^5+x^3+1$	8×8	RS(255, 247, 9)	$g_{124}:=x^8+a^{236}x^7+a^{175}x^6+a^{11}x^5+a^{115}x^4+a^{11}x^3+a^{175}x^2+a^{236}x+1$
21	$GF(2^8)$	$x^8+x^7+x^6+x^5+x^2+x+1$	8×8	RS(255, 247, 9)	$g_{124}:=x^8+a^{164}x^7+a^{242}x^6+a^{68}x^5+a^{158}x^4+a^{68}x^3+a^{242}x^2+a^{164}x+1$

Table Contd.

22	$GF(2^8)$	$x^8+x^5+x^3+x+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{40}x^7 + a^{107}x^6 + a^{45}x^5 + a^{154}x^4 + a^{45}x^3 + a^{107}x^2 + a^{40}x + 1$
23	$GF(2^8)$	$x^8+x^6+x^5+x^2+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{225}x^7 + a^{179}x^6 + a^{202}x^5 + a^{131}x^4 + a^{202}x^3 + a^{179}x^2 + a^{225}x + 1$
24	$GF(2^8)$	$x^8+x^7+x^6+x+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{203}x^7 + a^{118}x^6 + a^{42}x^5 + a^{224}x^4 + a^{42}x^3 + a^{118}x^2 + a^{203}x + 1$
25	$GF(2^8)$	$x^8+x^6+x^4+x^3+x^2+x+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{213}x^7 + a^{14}x^6 + a^{188}x^5 + a^{199}x^4 + a^{188}x^3 + a^{14}x^2 + a^{213}x + 1$
26	$GF(2^8)$	$x^8+x^6+x^5+x+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{228}x^7 + a^{130}x^6 + a^{51}x^5 + a^{221}x^4 + a^{51}x^3 + a^{130}x^2 + a^{228}x + 1$
27	$GF(2^8)$	$x^8+x^7+x^2+x+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{52}x^7 + a^{137}x^6 + a^{213}x^5 + a^{31}x^4 + a^{213}x^3 + a^{137}x^2 + a^{52}x + 1$
28	$GF(2^8)$	$x^8+x^7+x^6+x^3+x^2+x+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{91}x^7 + a^{13}x^6 + a^{187}x^5 + a^{97}x^4 + a^{187}x^3 + a^{13}x^2 + a^{91}x + 1$

Table Contd.

29	$GF(2^8)$	$x^8+x^6+x^3+x^2+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{30}x^7 + a^{76}x^6 + a^{53}x^5 + a^{124}x^4 + a^{53}x^3 + a^{76}x^2 + a^{30}x + 1$
30	$GF(2^8)$	$x^8+x^7+x^3+x^2+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{27}x^7 + a^{125}x^6 + a^{204}x^5 + a^{34}x^4 + a^{204}x^3 + a^{125}x^2 + a^{27}x + 1$
31	$GF(2^8)$	$x^8+x^7+x^6+x^5+x^4+x^2+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{42}x^7 + a^{241}x^6 + a^{67}x^5 + a^{56}x^4 + a^{67}x^3 + a^{241}x^2 + a^{42}x + 1$
32	$GF(2^8)$	$x^8+x^7+x^5+x^3+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{215}x^7 + a^{148}x^6 + a^{210}x^5 + a^{101}x^4 + a^{210}x^3 + a^{148}x^2 + a^{215}x + 1$
33	$GF(2^8)$	$x^8+x^5+x^3+x^2+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{19}x^7 + a^{80}x^6 + a^{244}x^5 + a^{140}x^4 + a^{244}x^3 + a^{80}x^2 + a^{19}x + 1$
34	$GF(2^8)$	$x^8+x^6+x^5+x^4+1$	8×8	RS(255, 247, 9)	$g_{124} := x^8 + a^{211}x^7 + a^{24}x^6 + a^{185}x^5 + a^{20}x^4 + a^{185}x^3 + a^{24}x^2 + a^{211}x + 1$
35	$GF(2^8)$	$x^8+x^4+x^3+x^2+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16} + a^{240}x^{15} + a^{89}x^{14} + a^{212}x^{13} + a^{79}x^{12} + a^{192}x^{11} + a^{116}x^{10} + a^{151}x^9 + a^{198}x^8 + a^{151}x^7 + a^{116}x^6 + a^{192}x^5 + a^{79}x^4 + a^{212}x^3 + a^{89}x^2 + a^{240}x + 1$

Table Contd.

36	$GF(2^8)$	$x^8+x^6+x^5+x^3+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{105}x^{15}+a^{193}x^{14}+a^{42}x^{13}+a^{58}x^{12}+ax^{11}+a^{10}x^{10}+a^{117}x^9+a^{51}x^8+a^{117}x^7+a^{10}x^6+ax^5+a^{58}x^4+a^{42}x^3+a^{193}x^2+a^{105}x+1$
37	$GF(2^8)$	$x^8+x^7+x^6+x^5+x^2+x+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{60}x^{15}+a^{18}x^{14}+a^{41}x^{13}+a^{19}x^{12}+a^{50}x^{11}+a^{153}x^{10}+a^{177}x^9+a^{127}x^8+a^{177}x^7+a^{153}x^6+a^{50}x^5+a^{19}x^4+a^{41}x^3+a^{18}x^2+a^{60}x+1$
38	$GF(2^8)$	$x^8+x^5+x^3+x+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{195}x^{15}+a^{169}x^{14}+a^{60}x^{13}+a^{243}x^{12}+a^{191}x^{11}+a^{17}x^{10}+a^{93}x^9+a^{90}x^8+a^{93}x^7+a^{17}x^6+a^{191}x^5+a^{243}x^4+a^{60}x^3+a^{169}x^2+a^{195}x+1$
39	$GF(2^8)$	$x^8+x^6+x^5+x^2+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{45}x^{15}+a^{39}x^{14}+a^{89}x^{13}+a^{243}x^{12}+a^{241}x^{11}+a^{201}x^{10}+a^{225}x^9+a^{217}x^8+a^{225}x^7+a^{201}x^6+a^{241}x^5+a^{243}x^4+a^{89}x^3+a^{39}x^2+a^{45}x+1$
40	$GF(2^8)$	$x^8+x^6+x^4+x^3+x^2+x+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{165}x^{15}+a^{160}x^{14}+a^{74}x^{13}+a^{230}x^{12}+a^{174}x^{11}+a^{216}x^{10}+a^{146}x^9+a^{178}x^8+a^{146}x^7+a^{216}x^6+a^{174}x^5+a^{230}x^4+a^{74}x^3+a^{160}x^2+a^{165}x+1$
41	$GF(2^8)$	$x^8+x^7+x^6+x+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{180}x^{15}+a^{20}x^{14}+a^{42}x^{13}+a^{16}x^{12}+a^{179}x^{11}+a^{142}x^{10}+x^9+a^{11}x^8+x^7+a^{142}x^6+a^{179}x^5+a^{16}x^4+a^{42}x^3+a^{20}x^2+a^{180}x+1$

Table Contd.

42	$GF(2^8)$	$x^8+x^6+x^5+x+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{15}x^{15}+a^{166}x^{14}+a^{70}x^{13}+a^{135}x^{12}+a^{138}x^{11}+a^{154}x^{10}+a^{241}x^9+a^{87}x^8+a^{241}x^7+a^{154}x^6+a^{138}x^5+a^{135}x^4+a^{70}x^3+a^{166}x^2+a^{15}x+1$
43	$GF(2^8)$	$x^8+x^7+x^2+x+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{75}x^{15}+a^{235}x^{14}+a^{213}x^{13}+a^{239}x^{12}+a^{76}x^{11}+a^{113}x^{10}+x^9+a^{244}x^8+x^7+a^{113}x^6+a^{76}x^5+a^{239}x^4+a^{213}x^3+a^{235}x^2+a^{75}x+1$
44	$GF(2^8)$	$x^8+x^7+x^6+x^3+x^2+x+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{195}x^{15}+a^{237}x^{14}+a^{214}x^{13}+a^{236}x^{12}+a^{205}x^{11}+a^{102}x^{10}+a^{78}x^9+a^{128}x^8+a^{78}x^7+a^{102}x^6+a^{205}x^5+a^{236}x^4+a^{214}x^3+a^{237}x^2+a^{195}x+1$
45	$GF(2^8)$	$x^8+x^6+x^3+x^2+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{210}x^{15}+a^{216}x^{14}+a^{166}x^{13}+a^{12}x^{12}+a^{14}x^{11}+a^{54}x^{10}+a^{30}x^9+a^{38}x^8+a^{30}x^7+a^{54}x^6+a^{14}x^5+a^{12}x^4+a^{166}x^3+a^{216}x^2+a^{210}x+1$
46	$GF(2^8)$	$x^8+x^7+x^3+x^2+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{240}x^{15}+a^{89}x^{14}+a^{185}x^{13}+a^{120}x^{12}+a^{117}x^{11}+a^{101}x^{10}+a^{14}x^9+a^{168}x^8+a^{14}x^7+a^{101}x^6+a^{117}x^5+a^{120}x^4+a^{185}x^3+a^{89}x^2+a^{240}x+1$
47	$GF(2^8)$	$x^8+x^7+x^6+x^5+x^4+x^2+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{90}x^{15}+a^{95}x^{14}+a^{181}x^{13}+a^{25}x^{12}+a^{81}x^{11}+a^{39}x^{10}+a^{109}x^9+a^{77}x^8+a^{109}x^7+a^{39}x^6+a^{81}x^5+a^{25}x^4+a^{181}x^3+a^{95}x^2+a^{90}x+1$

Table Contd.

48	$GF(2^8)$	$x^8+x^7+x^5+x^3+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{60}x^{15}+a^{86}x^{14}+a^{195}x^{13}+a^{12}x^{12}+a^{64}x^{11} + a^{238}x^{10}+a^{162}x^9+a^{165}x^8+a^{162}x^7+a^{238}x^6+a^{64}x^5 + a^{12}x^4+a^{195}x^3+a^{86}x^2+a^{60}x+1$
49	$GF(2^8)$	$x^8+x^5+x^3+x^2+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{150}x^{15}+a^{62}x^{14}+a^{213}x^{13}+a^{197}x^{12}+a^{254}x^{11} + a^{245}x^{10}+a^{138}x^9+a^{204}x^8+a^{138}x^7+a^{245}x^6+a^{254}x^5 + a^{197}x^4+a^{213}x^3+a^{62}x^2+a^{150}x+1$
50	$GF(2^8)$	$x^8+x^6+x^5+x^4+1$	16×16	RS(255, 239, 17)	$g_{120} := x^{16}+a^{15}x^{15}+a^{166}x^{14}+a^{43}x^{13}+a^{176}x^{12}+a^{63}x^{11} + a^{139}x^{10}+a^{104}x^9+a^{57}x^8+a^{104}x^7+a^{139}x^6+a^{63}x^5 + a^{176}x^4+a^{43}x^3+a^{166}x^2+a^{15}x+1$
51	$GF(2^8)$	$x^8+x^4+x^3+x^2+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{122}x^{31}+a^{230}x^{30}+a^{187}x^{29}+a^{128}x^{28}+a^{44}x^{27} + a^{74}x^{26}+a^{23}x^{25}+a^{198}x^{24}+a^{197}x^{23}+a^{238}x^{22}+a^{95}x^{21} + a^{101}x^{20}+a^{168}x^{19}+a^{161}x^{18}+a^{239}x^{17}+a^{34}x^{16}+a^{239}x^{15} + a^{161}x^{14}+a^{168}x^{13}+a^{101}x^{12}+a^{95}x^{11}+a^{238}x^{10}+a^{197}x^9 + a^{198}x^8+a^{23}x^7+a^{74}x^6+a^{44}x^5+a^{128}x^4+a^{187}x^3+a^{230}x^2 + a^{122}x+1$
52	$GF(2^8)$	$x^8+x^6+x^5+x^3+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{98}x^{31}+a^{127}x^{30}+a^{19}x^{29}+a^{192}x^{28}+a^{161}x^{27} + a^{15}x^{26}+a^{201}x^{25}+a^{241}x^{24}+a^{146}x^{23}+a^{164}x^{22}+a^{100}x^{21} + a^{152}x^{20}+a^{228}x^{19}+a^{41}x^{18}+a^{38}x^{17}+a^{113}x^{16}+a^{38}x^{15} + a^{41}x^{14}+a^{228}x^{13}+a^{152}x^{12}+a^{100}x^{11}+a^{164}x^{10}+a^{146}x^9 + a^{241}x^8+a^{201}x^7+a^{15}x^6+a^{161}x^5+a^{192}x^4+a^{19}x^3 + a^{127}x^2+a^{98}x+1$

Table Contd.

53	$GF(2^8)$	$x^8+x^7+x^6+x^5+x^2+x+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{107}x^{31}+a^{12}x^{30}+a^{50}x^{29}+a^{175}x^{28}+a^{90}x^{27}$ $+a^{67}x^{26}+a^{122}x^{25}+a^{113}x^{24}+a^{150}x^{23}+a^{148}x^{22}+a^{158}x^{21}$ $+a^{111}x^{20}+a^{82}x^{19}+a^{197}x^{18}+a^{212}x^{17}+a^{233}x^{16}+a^{212}x^{15}$ $+a^{197}x^{14}+a^{82}x^{13}+a^{111}x^{12}+a^{158}x^{11}+a^{148}x^{10}+a^{150}x^{9}$ $+a^{113}x^{8}+a^{122}x^{7}+a^{67}x^{6}+a^{90}x^{5}+a^{175}x^{4}+a^{50}x^{3}+a^{12}x^{2}$ $+a^{107}x+1$
54	$GF(2^8)$	$x^8+x^5+x^3+x+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{250}x^{31}+a^{89}x^{30}+a^{49}x^{29}+a^{221}x^{28}+a^{75}x^{27}$ $+a^{195}x^{26}+a^{164}x^{25}+a^{227}x^{24}+a^{89}x^{23}+a^{182}x^{22}+a^{139}x^{21}$ $+a^{189}x^{20}+a^{37}x^{19}+a^{125}x^{18}+a^{225}x^{17}+a^{238}x^{16}+a^{225}x^{15}$ $+a^{125}x^{14}+a^{37}x^{13}+a^{189}x^{12}+a^{139}x^{11}+a^{182}x^{10}+a^{89}x^{9}$ $+a^{227}x^{8}+a^{164}x^{7}+a^{195}x^{6}+a^{75}x^{5}+a^{221}x^{4}+a^{49}x^{3}+a^{89}x^{2}$ $+a^{250}x+1$
55	$GF(2^8)$	$x^8+x^6+x^5+x^2+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{195}x^{31}+a^{133}x^{30}+a^{8}x^{29}+a^{179}x^{28}+a^{231}x^{27}$ $+a^{96}x^{26}+a^{60}x^{25}+a^{73}x^{24}+a^{190}x^{23}+a^{140}x^{22}+a^{146}x^{21}$ $+a^{111}x^{20}+a^{168}x^{19}+a^{54}x^{18}+a^{239}x^{17}+a^{21}x^{16}+a^{239}x^{15}$ $+a^{54}x^{14}+a^{168}x^{13}+a^{111}x^{12}+a^{146}x^{11}+a^{140}x^{10}+a^{190}x^{9}$ $+a^{73}x^{8}+a^{60}x^{7}+a^{96}x^{6}+a^{231}x^{5}+a^{179}x^{4}+a^{8}x^{3}+a^{133}x^{2}$ $+a^{195}x+1$
56	$GF(2^8)$	$x^8+x^6+x^4+x^3+x^2+x+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{69}x^{31}+a^{214}x^{30}+a^{6}x^{29}+a^{209}x^{28}+a^{143}x^{27}$ $+a^{81}x^{26}+a^{46}x^{25}+a^{32}x^{24}+a^{165}x^{23}+a^{93}x^{22}+a^{228}x^{21}$ $+a^{190}x^{20}+a^{6}x^{19}+a^{85}x^{18}+a^{70}x^{17}+a^{234}x^{16}+a^{70}x^{15}$ $+a^{85}x^{14}+a^{6}x^{13}+a^{190}x^{12}+a^{228}x^{11}+a^{93}x^{10}+a^{165}x^{9}$ $+a^{32}x^{8}+a^{46}x^{7}+a^{81}x^{6}+a^{143}x^{5}+a^{209}x^{4}+a^{6}x^{3}+a^{214}x^{2}$ $+a^{69}x+1$

Table Contd.

57	$GF(2^8)$	$x^8+x^7+x^6+x+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{134}x^{31}+a^{232}x^{30}+a^{104}x^{29}+a^{176}x^{28}+a^{25}x^{27}$ $+a^{55}x^{26}+a^{7}x^{25}+a^{16}x^{24}+a^{185}x^{23}+a^{73}x^{22}+a^{139}x^{21}$ $+a^{145}x^{20}+a^{227}x^{19}+a^{171}x^{18}+a^{249}x^{17}+a^{108}x^{16}+a^{249}x^{15}$ $+a^{171}x^{14}+a^{227}x^{13}+a^{145}x^{12}+a^{139}x^{11}+a^{73}x^{10}+a^{185}x^{9}$ $+a^{16}x^{8}+a^{7}x^{7}+a^{55}x^{6}+a^{25}x^{5}+a^{176}x^{4}+a^{104}x^{3}+a^{232}x^{2}$ $+a^{134}x+1$
58	$GF(2^8)$	$x^8+x^6+x^5+x+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{99}x^{31}+a^{221}x^{30}+a^{113}x^{29}+a^{107}x^{28}+a^{76}x^{27}$ $+a^{15}x^{26}+a^{147}x^{25}+a^{94}x^{24}+a^{84}x^{23}+a^{77}x^{22}+a^{190}x^{21}$ $+a^{72}x^{20}+a^{240}x^{19}+a^{19}x^{18}+a^{16}x^{17}+a^{119}x^{16}+a^{16}x^{15}$ $+a^{19}x^{14}+a^{240}x^{13}+a^{72}x^{12}+a^{190}x^{11}+a^{77}x^{10}+a^{84}x^{9}$ $+a^{94}x^{8}+a^{147}x^{7}+a^{15}x^{6}+a^{76}x^{5}+a^{107}x^{4}+a^{113}x^{3}+a^{221}x^{2}$ $+a^{99}x+1$
59	$GF(2^8)$	$x^8+x^7+x^2+x+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{121}x^{31}+a^{23}x^{30}+a^{151}x^{29}+a^{79}x^{28}+a^{230}x^{27}$ $+a^{200}x^{26}+a^{248}x^{25}+a^{239}x^{24}+a^{70}x^{23}+a^{182}x^{22}+a^{116}x^{21}$ $+a^{110}x^{20}+a^{28}x^{19}+a^{84}x^{18}+a^{6}x^{17}+a^{147}x^{16}+a^{6}x^{15}$ $+a^{84}x^{14}+a^{28}x^{13}+a^{110}x^{12}+a^{116}x^{11}+a^{182}x^{10}+a^{70}x^{9}$ $+a^{239}x^{8}+a^{248}x^{7}+a^{200}x^{6}+a^{230}x^{5}+a^{79}x^{4}+a^{151}x^{3}$ $+a^{23}x^{2}+a^{121}x+1$
60	$GF(2^8)$	$x^8+x^7+x^6+x^3+x^2+x+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{148}x^{31}+a^{243}x^{30}+a^{205}x^{29}+a^{80}x^{28}+a^{165}x^{27}$ $+a^{188}x^{26}+a^{133}x^{25}+a^{142}x^{24}+a^{105}x^{23}+a^{107}x^{22}+a^{97}x^{21}$ $+a^{144}x^{20}+a^{173}x^{19}+a^{58}x^{18}+a^{43}x^{17}+a^{22}x^{16}+a^{43}x^{15}$ $+a^{58}x^{14}+a^{173}x^{13}+a^{144}x^{12}+a^{97}x^{11}+a^{107}x^{10}+a^{105}x^{9}$ $+a^{142}x^{8}+a^{133}x^{7}+a^{188}x^{6}+a^{165}x^{5}+a^{80}x^{4}+a^{205}x^{3}$ $+a^{243}x^{2}+a^{148}x+1$

Table Contd.

61	$GF(2^8)$	$x^8+x^6+x^3+x^2+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{60}x^{31}+a^{122}x^{30}+a^{247}x^{29}+a^{76}x^{28}+a^{24}x^{27}$ $+a^{159}x^{26}+a^{195}x^{25}+a^{182}x^{24}+a^{65}x^{23}+a^{115}x^{22}+a^{109}x^{21}$ $+a^{144}x^{20}+a^{87}x^{19}+a^{201}x^{18}+a^{16}x^{17}+a^{234}x^{16}+a^{16}x^{15}$ $+a^{201}x^{14}+a^{87}x^{13}+a^{144}x^{12}+a^{109}x^{11}+a^{115}x^{10}+a^{65}x^{9}$ $+a^{182}x^{8}+a^{195}x^{7}+a^{159}x^{6}+a^{24}x^{5}+a^{76}x^{4}+a^{247}x^{3}$ $+a^{122}x^{2}+a^{60}x+1$
62	$GF(2^8)$	$x^8+x^7+x^3+x^2+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{156}x^{31}+a^{34}x^{30}+a^{142}x^{29}+a^{148}x^{28}+a^{179}x^{27}$ $+a^{240}x^{26}+a^{108}x^{25}+a^{161}x^{24}+a^{171}x^{23}+a^{178}x^{22}+a^{65}x^{21}$ $+a^{183}x^{20}+a^{15}x^{19}+a^{236}x^{18}+a^{239}x^{17}+a^{136}x^{16}+a^{239}x^{15}$ $+a^{236}x^{14}+a^{15}x^{13}+a^{183}x^{12}+a^{65}x^{11}+a^{178}x^{10}+a^{171}x^{9}$ $+a^{161}x^{8}+a^{108}x^{7}+a^{240}x^{6}+a^{179}x^{5}+a^{148}x^{4}+a^{142}x^{3}$ $+a^{34}x^{2}+a^{156}x+1$
63	$GF(2^8)$	$x^8+x^7+x^6+x^5+x^4+x^2+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{186}x^{31}+a^{41}x^{30}+a^{249}x^{29}+a^{46}x^{28}+a^{112}x^{27}$ $+a^{174}x^{26}+a^{209}x^{25}+a^{223}x^{24}+a^{90}x^{23}+a^{162}x^{22}+a^{27}x^{21}$ $+a^{65}x^{20}+a^{249}x^{19}+a^{170}x^{18}+a^{185}x^{17}+a^{21}x^{16}+a^{185}x^{15}$ $+a^{170}x^{14}+a^{249}x^{13}+a^{65}x^{12}+a^{27}x^{11}+a^{162}x^{10}+a^{90}x^{9}$ $+a^{223}x^{8}+a^{209}x^{7}+a^{174}x^{6}+a^{112}x^{5}+a^{46}x^{4}+a^{249}x^{3}$ $+a^{41}x^{2}+a^{186}x+1$
64	$GF(2^8)$	$x^8+x^7+x^5+x^3+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32}+a^{5}x^{31}+a^{166}x^{30}+a^{206}x^{29}+a^{34}x^{28}+a^{180}x^{27}$ $+a^{60}x^{26}+a^{91}x^{25}+a^{28}x^{24}+a^{166}x^{23}+a^{73}x^{22}+a^{116}x^{21}$ $+a^{66}x^{20}+a^{218}x^{19}+a^{130}x^{18}+a^{30}x^{17}+a^{17}x^{16}+a^{30}x^{15}$ $+a^{130}x^{14}+a^{218}x^{13}+a^{66}x^{12}+a^{116}x^{11}+a^{73}x^{10}+a^{166}x^{9}$ $+a^{28}x^{8}+a^{91}x^{7}+a^{60}x^{6}+a^{180}x^{5}+a^{34}x^{4}+a^{206}x^{3}+a^{166}x^{2}$ $+a^{5}x+1$

Table Contd.

65	$GF(2^8)$	$x^8+x^5+x^3+x^2+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32} + a^{157}x^{31} + a^{128}x^{30} + a^{236}x^{29} + a^{63}x^{28} + a^{94}x^{27}$ $+ a^{240}x^{26} + a^{54}x^{25} + a^{14}x^{24} + a^{109}x^{23} + a^{91}x^{22} + a^{155}x^{21}$ $+ a^{103}x^{20} + a^{27}x^{19} + a^{214}x^{18} + a^{217}x^{17} + a^{142}x^{16} + a^{217}x^{15}$ $+ a^{214}x^{14} + a^{27}x^{13} + a^{103}x^{12} + a^{155}x^{11} + a^{91}x^{10} + a^{109}x^{9}$ $+ a^{14}x^{8} + a^{54}x^{7} + a^{240}x^{6} + a^{94}x^{5} + a^{63}x^{4} + a^{236}x^{3} + a^{128}x^{2}$ $+ a^{157}x + 1$
66	$GF(2^8)$	$x^8+x^6+x^5+x^4+1$	32×32	RS(255, 223, 33)	$g_{112} := x^{32} + a^{133}x^{31} + a^{25}x^{30} + a^{68}x^{29} + a^{127}x^{28} + a^{211}x^{27}$ $+ a^{181}x^{26} + a^{232}x^{25} + a^{57}x^{24} + a^{58}x^{23} + a^{17}x^{22} + a^{160}x^{21}$ $+ a^{154}x^{20} + a^{87}x^{19} + a^{94}x^{18} + a^{16}x^{17} + a^{221}x^{16} + a^{16}x^{15}$ $+ a^{94}x^{14} + a^{87}x^{13} + a^{154}x^{12} + a^{160}x^{11} + a^{17}x^{10} + a^{58}x^{9}$ $+ a^{57}x^{8} + a^{232}x^{7} + a^{181}x^{6} + a^{211}x^{5} + a^{127}x^{4} + a^{68}x^{3} + a^{25}x^{2}$ $+ a^{133}x + 1$

Compare our Results with Results in [1]

In [1], the authors proposed the construction of recursive MDS matrices using shortened BCH codes. In general, this construction is complicated because finding the generator polynomial of BCH codes is not straightforward. However, for the RS codes, it is very straightforward to compute the generator polynomial of these codes according to the formula (1). In addition, in [1] the authors found a number of symmetric polynomials, however there is a case where the constant termis not equal to 1. Table 2 shows these polynomials (the symbol α in Table 2 is a root element of the field).

Table 2. Some symmetric polynomials are the results in [1]

No.	Field GF	Primitive polynomial	Size of matrix	Symmetric polynomial
1	$GF(2^4)$	x^4+x^3+1	4×4	$g_1=x^4+\alpha^3x^3+\ \alpha x^2+\alpha^3x+1$
2	$GF(2^4)$	x^4+x^3+1	4×4	$g_2=x^4+\alpha^3x^3+\ \alpha x^2+x+\alpha^3+\alpha$
3	$GF(2^8)$	$x^8+x^4+x^3+x^2+1$	4×4	$g_3=x^4+\alpha^3x^3+\ \alpha^{-3}x^2+\alpha^3x+1$
4	$GF(2^8)$	$x^8+x^4+x^3+x^2+1$	4×4	$g_4=x^4+(\alpha^2+\alpha^3)x^3+\ \alpha^3x^2+(\alpha^3+\alpha^2)x+1$
5	$GF(2^8)$	$x^8+x^4+x^3+x^2+1$	4×4	$g_4=x^4+\alpha^{202}x^3+(\alpha^{202}+1)x^2+x+\alpha+1$

In Table 1, we found 66 symmetric polynomials having the constant term equal to 1. Thereby, there will be many options to choose recursive MDS matrices effective for implementation for cryptographic applications.

4. Preserving the Recursive Property of MDS Matrix of Scalar Multiplication

This section presents the ability to preserve the recursive property of MDS matrix of the scalar multiplication transformation. Specifically, from a recursive MDS matrix which is a power of a serial matrix, by scalar multiplication many other recursive MDS matrices as powers of serial-like matrices can be generated. Serial-like matrices are very sparse similar to serial matrices. Thus, the recursive MDS matrices from such matrices are very meaningful in implementation, especially in hardware implementation when they are applied to design the diffusion layer of block ciphers.

It is to have the following proposition:

Proposition 5. *Let $A=[a_{i,j}]_{m\times m}$, $a_{i,j}\in GF(p^r)$, be an MDS matrix such that $A=S^m$, where S is a serial matrix of the form* (2). *Let matrix $\acute{A}=c.A$ for $c\in GF(p^r)\backslash 0$. If there exists $e\in GF(p^r):e^m=c$ then $\acute{A}=\acute{S}^m$, where $\acute{S}$ is a serial-like matrix of the form* (5).

Proof. By assumption, it is to have:

$$A=S^m \tag{14}$$

where S is a serial matrix of the form (2).

If there exists $e\in GF(p^r):e^m=c$, it is to have:

$$\acute{A}=c.A=e^mA \tag{15}$$

From (14) and (15) it is to infer that:

$$\acute{A} = e^m S^m = (eS)^m = \acute{S}^m \tag{16}$$

where$\acute{S} = eS$. Therefore,it is to obtain $\acute{S}$ of the form (5). □

On the other hand, two vectors of m elements can be defined as follows: $E = [e^m, e^m, \dots, e^m]$, $F = [1, 1, \dots, 1]$. From (15), it is to infer:

$$\acute{A} = (E, F)A \tag{17}$$

Hence, it is possible to generate $\acute{A}$ from A by the scalar multiplication, then $\acute{A}$ is also an MDS matrix.

Therefore, from a recursive MDS matrix which is a power of a serial matrix, by scalar multiplication another recursive MDS matrix as a power of a serial-like matrix can be generated.

Example 1. Consider the field $GF(2^8)$with the primitive polynomial: $x^8 + x^7 + x^6 + x + 1$.
Let A be a recursive MDS matrix as a power of a serial matrix of size $m = 4$ over the field:
$A = \begin{bmatrix} 8A & 46 & D8 & 1E \\ 17 & 42 & C2 & 4F \\ F5 & 5D & 78 & E4 \\ A2 & 4B & F & 11 \end{bmatrix}$

The corresponding serial matrix is: $S = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 8A & 46 & D8 & 1E \end{bmatrix}$ such that: $A = S^4$.

Consider element $c = 0 \times 10$. It is to have $c = (0 \times 02)^4$ then it has the form $c = e^4$.

Compute the matrix $\acute{A} = cA$. It is to obtain: $\acute{A} = \begin{bmatrix} 77 & EA & 1E & 23 \\ B3 & AA & 7D & 7A \\ 8B & 99 & 8C & 58 \\ B2 & 3A & F0 & D3 \end{bmatrix}$.

Compute the serial-like matrix $\acute{S}$ as follow: $\acute{S} = (0 \times 02).S = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \\ D7 & 8C & 73 & 3C \end{bmatrix}$.

Check and see that $\acute{A} = \acute{S}^4$, so $\acute{A}$ is a recursive MDS matrix which is a power of the serial-like matrix $\acute{S}$.

Comment 1. *To be able to select a matrix $\acute{A} \neq A$, it is to need to select the element $c \neq 1$ or $e^m \neq 1$ or $ord(e)$ is not a divisor of m.*

Comment 2. *Suppose there exists two elements $e_1, e_2 \in GF(p^r)$ such that: $(e_1)^m = (e_2)^m$ (i.e. $A_1 = (e_1)^m A = A_2 = (e_2)^m A$). It is to infer that: $(e_1 e_2^{-1})^m = 1$ or $ord(e_1 e_2^{-1}) | m$. Then, in order to obtain $A_1 \neq A_2$, it is to need to choose an element $a = e_1 e_2^{-1} \in GF(p^r) \backslash \{0, 1\}$ such that $ord(a)$ is not a divisor of m where $e_1, e_2 \in GF(p^r) \backslash \{0, 1\}$.*

The question is that how many recursive MDS matrices (powers of serial-like matrices) can be generated from an original recursive MDS matrix (power of a serial matrix) by the scalar multiplication transformation?

Let $A = [a_{i,j}]_{m\times m}$, $a_{i,j} \in GF(p^r)$, be a recursive MDS matrix as a power of a serial matrix. Choose an element $a \in GF(p^r)\backslash\{0,1\}$ such that $ord(a)$ is not a divisor of m and $a = e_1 e_2^{-1}$, where $e_1, e_2 \in GF(p^r)\backslash\{0,1\}$.

Select and fix an element e_1 other than $0,1 \in GF(p^r)$ then compute $e_2 = e_1 a^{-1}$. Obviously $e_2 \neq 0$, so if $e_2 = 1$ or e_2 does not satisfy the condition: $ord(e_2)$ is not a divisor of m (to make $(e_2)^m A \neq A$) then choose another element e_1 until e_2 satisfies the above condition (i.e. $e_2 \neq 1$ and $ord(e_2)$ is not a divisor of m).

Consider a sequence of matrices: $A_0 = (e_2)^m A$, $A_1 = (e_2 a)^m A$, $A_2 = (e_2 a^2)^m A, \dots, A_k = (e_2 a^k)^m A$, and so on. It is to have the following result:

Proposition 6. *The sequence of matrices $A_0, A_1, A_2, \dots$ has a finite cycle, that is $d = ord(a)$.*

Proof. By assumption, $ord(a)$is not a divisor of m, so according to Comment 2, the sequence of matrices $A_0, A_1, A_2, \dots$ are different matrices in pairs.

For $d = ord(a)$, it is to have $(e_2 a^d)^m A = (e_2)^m A$ or $A_d = A_0$.

Now, suppose that $\exists\ d_1 \in N^+ : (e_2 a^{d_1})^m A = (e_2)^m A$. Then, $a^{md_1} = 1$. Therefore $d|(md_1)$. By assumption, d is not a divisor of m, it is to infer that $d|d_1$ or d is the smallest positive integer that satisfies the condition $(e_2 a^d)^m A = (e_2)^m A$. □

Consequently, for an element $a \in GF(p^r)\backslash\{0,1\}$ such that $ord(a)$ is not a divisor of m, one can obtain $ord(a)$ recursive MDS matrices (powers of serial-like matrices) from the original recursive MDS matrix A (power of a serial matrix).

The next question is how to select elementa so that the cycle of the above sequence of matrices is as large as possible?

Suppose that the size of the matrices satisfies $m < p^r - 1$. As $\leq p^r - 1$, so if $d = p^r - 1$ (i.e. a is a root element of the field), it is always to have $d$is not a divisor of m, so the cycle of the above matrices reaches a maximum value that is $p^r - 1$.

The following Algorithm 1 will show how to find the sequence including $p^r - 1$ different recursive MDS matrices (powers of serial-like matrices) from an original recursive MDS matrix (power of a serial matrix) over $GF(p^r)$.

Algorithm 1 (Finding a set of p^{r-1} different recursive MDS matrices from an original recursive MDS matrix A).

Input: the recursive MDS matrix$A$ (power of a serial matrix) over $GF(p^r)$; C = {Set of root elements of the field$GF(p^r)$}.

Output: a set T of $p^r - 1$ different recursive MDS matrices (powers of serial-like matrices) generated from A.

Step 1: Select any element $a \in C$.

Step 2: *Step* 2.1: Select an element $e_1 \in GF(p^r)\backslash\{0,1\}$.

Step 2.2: Compute $e_2 = e_1 a^{-1}$.

Step 2.3: If $e_2 \neq 1$ and $ord(e_2)$ is not a divisor of m then go to *Step* 3, otherwise go back to *Step* 2.1.

Step 3: Return $T = \{A_0 = (e_2)^m A, A_1 = (e_2 a)^m A, A_2 = (e_2 a^2)^m A, \ldots, A_{p^r-2} = (e_2 a^{p^r-2})^m A\}$.

Example 2. Consider the field $GF(2^8)$ with the primitive polynomial: $x^8 + x^5 + x^3 + x + 1$. Let A be a recursive MDS matrix as a power of a serial matrix of size $m = 4$ over the field:
$A = \begin{bmatrix} AC & BD & D8 & 1E \\ 97 & EE & 75 & A7 \\ 21 & 44 & 85 & 30 \\ 4 & 68 & D8 & F3 \end{bmatrix}$.

The corresponding serial matrix is: $S = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ AC & BD & D8 & 1E \end{bmatrix}$.

Matrices A and S satisfy: $A = S^4$.

Consider an element $a = 0 \times 80 \in GF(2^8)$ is a root element of the field, i.e. its order is 255.

Next, select $e_1 \in GF(2^8) \backslash \{0,1\}$, for example $e_1 = 0 \times 53$. Compute $e_2 = e_1 a^{-1} = (0 \times 53).(0 \times 60) = 0 \times CF$.

It can be seen that $e_2 \neq 1$ and $ord(e_2) = 85$ is not a divisor of 4. Then, a set of 255 recursive MDS matrices can be built as follows:

$$T = \{A_0 = (e_2)^4 A, A_1 = (e_2 a)^4 A, A_2 = (e_2 a^2)^4 A, \ldots, A_{254} = (e_2 a^{254})^4 A\}.$$

Let S_i be the serial-like matrix corresponding to matrix A_i such that: $A_i = (S_i)^4$, $0 \leq i \leq 254$. Table 3 gives some of recursive MDS matrices (powers of serial-like matrices) from the set T.

Table 3. Some of recursive MDS matrices obtained over $GF(2^8)$

No.	**Recursive MDS matrix**	**Corresponding serial-like matrix**
1	$A_0 = \begin{bmatrix} BE & 3 & B0 & D6 \\ 60 & 9C & D7 & CC \\ 11 & A2 & 9D & AC \\ 19 & 41 & B0 & A \end{bmatrix}$	$S_0 = \begin{bmatrix} 0 & CF & 0 & 0 \\ 0 & 0 & CF & 0 \\ 0 & 0 & 0 & CF \\ CE & 2E & 42 & 59 \end{bmatrix}$
2	$A_1 = \begin{bmatrix} 36 & E9 & B8 & 73 \\ 32 & 96 & CD & BD \\ 50 & 1 & 28 & 8F \\ 27 & 7B & B8 & 20 \end{bmatrix}$	$S_1 = \begin{bmatrix} 0 & 53 & 0 & 0 \\ 0 & 0 & 53 & 0 \\ 0 & 0 & 0 & 53 \\ D3 & 37 & CC & B8 \end{bmatrix}$
3	$A_2 = \begin{bmatrix} E0 & 5A & CF & 35 \\ 4E & B6 & 3 & DF \\ 2B & BE & 80 & 91 \\ B0 & 42 & CF & F7 \end{bmatrix}$	$S_2 = \begin{bmatrix} 0 & 3F & 0 & 0 \\ 0 & 0 & 3F & 0 \\ 0 & 0 & 0 & 3F \\ DA & 68 & F8 & DC \end{bmatrix}$
	$\vdots$	$\vdots$
255	$A_{254} = \begin{bmatrix} 1 & CD & 27 & 88 \\ 1E & 64 & 2A & 75 \\ A7 & 4D & C6 & 6B \\ 30 & B6 & 27 & F8 \end{bmatrix}$	$S_{254} = \begin{bmatrix} 0 & E2 & 0 & 0 \\ 0 & 0 & E2 & 0 \\ 0 & 0 & 0 & E2 \\ 82 & C9 & 55 & 72 \end{bmatrix}$

5. Conclusion

In this paper, the method for constructing recursive MDS matrices effective for implementation from Reed-Solomon codes is presented. In addition, the ability to preserve the recursive property of MDS matrix of the scalar multiplication transformation is given. The recursive MDS matrices effective for implementation are meaningful in hardware implementation, and preserving recursive property of MDS matrix of the scalar multiplication transformation also has important applications for efficiently building dynamic block ciphers later. The strength of the ciphers against developing cryptanalytic techniques can be enhanced by the dynamic block ciphers.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

All the authors contributed significantly in writing this article. The authors read and approved the final manuscript.

References

[1] D. Augot and M. Finiasz, Direct construction of recursive MDS diffusion layers using shortened BCH codes, *21st International Workshop on Fast Software Encryption*, *FSE 2014*, Springer (2014), DOI: 10.1007/978-3-662-46706-0_1.

[2] D. Augot and M. Finiasz, 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, DOI: 10.1109/ISIT.2013.6620487.

[3] K. C. Gupta and I. G. Ray, *On Constructions of MDS Matrices from Companion Matrices for Lightweight Cryptography*, Applied Statistics Unit, Indian Statistical Institute 203, B. T. Road, Kolkata 700108, India (2013), https://link.springer.com/content/pdf/10.1007/978-3-642-40588-4_3.pdf.

[4] K. C. Gupta and 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, https://www.isical.ac.in/~asu/TR/TechRepASU201401.pdf.

[5] L. Keliher, *Linear Cryptanalysis of Substitution-Permutation Networks*, Queen's University, Kingston, Ontario, Canada (2003), http://www.madchat.fr/crypto/codebreakers/keliherPhD.pdf.

[6] S. Kolay and D. Mukhopadhyay, Lightweight diffusion layer from the kth root of the MDS matrix, *IACR Cryptology ePrint Archive* **498** (2014), https://eprint.iacr.org/2014/498.pdf.

[7] T. T. Luong and 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* **32** (1) (2016), 1 – 17, DOI: 1813-9663/32/1/7732.

[8] 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*, **3**(2) (2016), 10 – 16, http://antoanthongtin.vn/Portals/0/NewsAttach/2017/01/MDS%20matric.pdf.

[9] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, (Bell Laboratories, Murray Hill, NJ, USA), North-Holland Publishing Company, Amsterdam, pp. 100 – 350, New York, Oxford (1977), `http://www.academia.edu/download/43668701/linear_codes.pdf`.

[10] G. Murtaza and N. Ikram, *Direct Exponent and Scalar Multiplication Classes of an MDS Matrix*, [EB/OL], National University of Sciences and Technology, Pakistan, (2011-01-10), pp. 2 – 5, `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.400.8450&rep=rep1&type=pdf`.

[11] M. Sajadieh, M. Dakhilalian, H. Mala and P. Sepehrdad, Recursive diffusion layers for block ciphers and hash functions, in *Fast Software Encryption*, Springer (2012), pp. 385 – 401, DOI: 10.1007/978-3-642-34047-5_22.

[12] C. Schnorr and S. Vaudenay, Black box cryptanalysis of hash networks based on multipermutations, in *Advances in Cryptology - EU-ROCRYPT '94.Proceedings*, A. De Santis (editor), Vol. 950 of LNCS, pp. 47 – 57, Springer-Verlag (1995), DOI: 10.1007/BFb0053423.

[13] S. Vaudenay, On the need for multipermutations: cryptanalysis of MD4 and SAFER, in *Fast Software Encryption.Proceedings*, B. Preneel (editor), Vol. 1008 of LNCS, pp. 286 – 297, Springer-Verlag (1995), DOI: 10.1007/3-540-60590-8_22.

[14] S. Wu, M. Wang and W. Wu, Recursive diffusion layers for (lightweight) block ciphers and hash functions, in *Selected Areas in Cryptography*, Springer (2013), pp. 43 – 60, `http://ir.iscas.ac.cn/handle/311060/15899`.

[15] M. R. Z'aba, *Analysis of Linear Relationships in Block Ciphers*, Ph.D Thesis, Queensland University of Technology, Brisbane, Australia (2010), `http://eprints.qut.edu.au/35725/1/Muhammad_Z%27aba_Thesis.pdf`.