Journal of Computer Science and Cybernetics, V.42, N. 1 (2026), 25–40
DOI no. 10.15625/1813-9663/22518

INNOVATIVE CONSTRUCTION OF INVOLUTORY MDS MATRICES VIA SELF-RECIPROCAL GENERATOR POLYNOMIALS DERIVED FROM REED-SOLOMON CODES

LUONG TRAN THI[1,*], CUONG NGUYEN NGOC[1], XINH DINH THI[2]

[1] *Academy of Cryptography Techniques, No. 141, Chien Thang Road,*
Thanh Liet Ward, Ha Noi, Viet Nam

[2] *Tay Nguyen University, No. 567, Le Duan Street, Ea Kao Ward,*
Dak Lak Province, Viet Nam

Abstract. Recursive MDS matrices over finite fields optimize diffusion and enable efficient implementation in block ciphers. A key challenge is designing involutory MDS matrices to unify encryption and decryption, reducing costs. Recursive MDS matrices can meet these requirements. In this paper, we present a direct construction method for self-reciprocal recursive MDS matrices of arbitrary sizes, derived from self-reciprocal generator polynomials of Reed–Solomon codes, to generate corresponding involutory MDS matrices. The process for deriving self-reciprocal recursive MDS matrices from Reed–Solomon codes is straightforward. Additionally, we identify self-reciprocal generator polynomials of Reed–Solomon codes over the general finite field $GF(q)$, where $q = p^r$ and p is a prime number, including cases where p is an odd prime. Involutory MDS matrices derived from self-reciprocal matrices are highly efficient for hardware and software, making them ideal for modern cryptographic applications.

Keywords. MDS matrix, companion matrix, recursive MDS matrix, reed-Solomon codes, self-reciprocal polynomials.

1. INTRODUCTION

An MDS (maximum distance separable) matrix originates from MDS codes in error-correcting code theory [1]. Serge Vaudenay [2] was instrumental in introducing MDS matrices to block cipher design, applying them as linear cases of multi-permutations, which achieve ideal diffusion. Due to their capacity to provide optimal diffusion, MDS matrices are integral to the structure of various block ciphers [3–6] and hash functions, including AES [3, 7, 8], SHARK, SQUARE, Twofish, as well as LED, Khazad, Whirlpool, and PHOTON.

A block cipher typically consists of two main components: a substitution layer and a diffusion layer. The substitution layer uses S-boxes [9–11] to introduce nonlinearity by substituting bits to create complex transformations, making it difficult to trace input-output relationships. The diffusion layer often employs MDS matrices [12–14], which spread changes

[*] Corresponding author.
E-mail addresses: luongtran@actvn.edu.vn (L.T. Thi), nguyenngoccuong189@gmail.com (C.N. Ngoc), dinhthixinh@ttn.edu.vn (X.D. Thi).

in any single bit across the entire block. This combination ensures that minor changes in input affect multiple bits in the output, enhancing security. Together, these layers are repeatedly applied to provide strong encryption. Despite their utility, the search for MDS matrices that maintain low implementation costs, particularly those that efficiently support both encryption and decryption, remains a substantial challenge.

A recursive MDS matrix is an MDS matrix that can be represented as a power of some companion matrix [14, 15]. Recursive MDS matrices can be efficiently implemented using LFSR registers, making them well-suited for lightweight cryptographic algorithms.

There has been a significant amount of research in the literature on recursive MDS matrices [14–19]. In [14], the authors presented a construction method for recursive MDS matrices using companion matrices, tailored for lightweight cryptography. This method provides effective diffusion for cryptographic functions with minimal implementation costs, which is particularly advantageous in environments with limited resources. In [15, 16], the authors focused on a direct approach to constructing recursive MDS matrices by BCH codes, which are highly effective for cryptographic diffusion layers. In [17], the authors presented the construction of recursive MDS matrices within finite commutative rings, which offers an alternative framework to traditional finite fields. The authors demonstrate efficient methodologies for building these matrices, emphasizing their value in achieving strong diffusion and compactness in cryptographic systems. This approach broadens the practical applications of MDS matrices in lightweight cryptographic designs, supporting both efficiency and security. In [18], the authors introduced a novel method for constructing recursive MDS matrices using DLS matrices, which enhances the efficiency of cryptographic operations. The authors provide a detailed framework for leveraging DLS matrices to create MDS matrices that offer strong diffusion properties, which are crucial in improving the security and performance of lightweight cryptographic systems.

Overall, to find recursive MDS matrices, authors typically use exhaustive search methods over families of companion matrices, which makes it challenging to discover larger recursive matrices. Other methods based on BCH codes or DLS matrices are also relatively complex. In [19], we employed a method to construct recursive MDS matrices of arbitrary sizes using Reed-Solomon (RS) codes. This approach is both simple and efficient, capable of generating numerous recursive MDS matrices of varying sizes.

A key challenge for block cipher designers is identifying involutory MDS matrices [20,21], which ensure that the encryption and decryption operations are identical, thereby reducing implementation costs. Therefore, recursive MDS matrices are desired to be involutory, allowing for identical diffusion layer calculations in both the encryption and decryption processes.

For this challenge, studies in [14–19] have not addressed the construction of involutory MDS matrices from recursive MDS matrices. In [22], the authors found symmetric generator polynomials of degree $4RS$ codes over the field $GF(q)$. From these, they generate the corresponding 4×4 recursive MDS matrices with inverse matrices of a similar form. They demonstrated that symmetric generator polynomials of RS codes only exist when $p = 2$ but not when p is an odd prime. However, in [23], the authors did not address the construction of involutory MDS matrices from such recursive matrices.

In [24], the authors proved that, except for trivial cases, it's impossible to have an involutory recursive MDS matrix over fields with characteristic 2. To address this limitation, they proposed a method to achieve an involutory structure by multiplying a symmetric recursive MDS matrix with a row-reversal matrix. Their symmetric recursive MDS matrices are derived from BCH codes over $GF(2^S)$, though this BCH-based approach is complex. Moreover, they did not consider generating symmetric recursive MDS matrices over the broader field $GF(p^r)$ with p as a prime.

In [25], the authors proposed a new approach by introducing the concept of self-reciprocal recursive MDS matrices, which serve as an alternative to the symmetric matrices discussed in prior work like [24]. They provided a construction method for 8 × 8 self-reciprocal recursive MDS matrices derived from RS codes over $GF(q)$, leading to the creation of involutory MDS matrices. However, this method was applied only to 8 × 8 matrices, leaving the construction of self-reciprocal recursive MDS matrices in arbitrary sizes unaddressed.

This paper introduces a method to construct self-reciprocal recursive MDS matrices of any size by utilizing self-reciprocal generator polynomials from RS codes, thereby facilitating the creation of involutory MDS matrices. The approach is efficient and streamlined, focu/sing on RS code-based generation, and covers generalized fields $GF(q)$, where $q = p^r$ with p as a prime number, including cases for odd primes. It considers both even-sized and odd-sized matrices. The resulting involutory MDS matrices, applicable across various sizes, offer high efficiency in hardware and software, making them well-suited for contemporary cryptographic applications.

This manuscript is based on our earlier work, which was initially shared as a preprint to promote open research and early visibility [26]. The study proposed a construction of involutory recursive-like MDS matrices utilizing self-reciprocal generator polynomials derived from RS codes.

This paper is organized as follows: Section 2 presents the preliminaries and related works. In Section 3, we introduce our main method for constructing monic, self-reciprocal polynomials of degree m derived from generator polynomials of RS codes, which are used to build self-reciprocal recursive MDS matrices and their corresponding involutory MDS matrices. Section 4 provides a comparative analysis of our approach against several existing methods in the literature. Finally, Section 5 offers our conclusions.

2. PRELIMINARIES AND RELATED WORKS

2.1. Preliminaries

A BCH code over $GF(q)$ of length n and designed distance d is the largest possible cyclic code having zeros $\alpha^b, \alpha^{(b+1)}, \ldots, \alpha^{(b+d-2)}$, where $\alpha \in GF(q^m)$ is a primitive n^{th} root of unity, b is a nonnegative integer, and m is the multiplicative order of q modulo n [1].

RS codes are defined in [1] as follows. An RS code over $GF(q) = GF(p^r)$, where p is a prime number, is a BCH code of length $q - 1$. Let n, k, and d be the length, dimension, and minimum distance of the code, respectively. An RS code over the finite field $GF(q)$ with length n, dimension k, and minimum distance d is denoted as $RS(n, k, d)$, satisfying $n = q - 1$ and $d = n - k + 1$.

Denote α as a primitive element of the field. An $RS(n, k, d)$ code with length $n = q - 1$ designed with a distance d will have the corresponding generator polynomial of degree $d - 1$ as follows

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

where b is a non-negative integer ($b \in \mathbb{N}$, $b \geq 0$).

A recursive MDS matrix [15] is defined as a power of a companion matrix, i.e., a matrix $M = C_g^k$, where C_g is a companion matrix corresponding to the polynomial $g(x) \in GF(q)[X]$ of degree k. We often say that the polynomial $g(x)$ generates a recursive MDS matrix. Reciprocal polynomials are defined as follows.

Definition 1. [1, 10, p. 108] The reciprocal f^* of a polynomial

$$f = a_0 + a_1X + \cdots + a_{k-1}X^{k-1} + a_kX^k \in GF(q)[X], \tag{2}$$

where $a_0, \ldots, a_k \in GF(q)^*$, is defined by

$$f^* = X^k f\left(\frac{1}{X}\right) = a_0X^k + a_1X^{k-1} + \cdots + a_{k-1}X + a_k. \tag{3}$$

A polynomial is called *self-reciprocal* if it coincides with its reciprocal.

From Definition 1, we have the following remarks.

Remark 1. If f^* is the reciprocal polynomial of f and if $\beta\,(\neq 0)$ is a root of f, then β^{-1} is also a root of f^* and vice versa.

If f is a self-reciprocal polynomial, then $a_j = a_{k-j}$ for $j = 0, 1, \ldots, k$, and if $\beta\,(\neq 0)$ is a root of f, then β^{-1} is also a root of f [24].

In [25], the author introduced the definition of a self-reciprocal MDS matrix as follows.

Definition 2. [25] A recursive MDS matrix $M = C_g^k$ is said to be a *self-reciprocal MDS matrix* if the polynomial associated with it g is monic and self-reciprocal.

Remark 2. According to Proposition 2 [19], from an MDS code $C[n, k, d]$, we can build an $(n-k) \times (n-k)$ recursive MDS matrix if $k \geq n-k$. The RS code is a special case of the MDS code, and with the assumption $d = n-k+1$, it is deduced that $n-k = d-1 = m$, and $k \geq n-k \Rightarrow q \geq 2m+1$.

Thus, from an $RS[n, k, d]$ code, we can build an $m \times m$ recursive MDS matrix if $q \geq 2m+1$.

2.2. Related works

With the assumption $n > 2k$, over the field of characteristic 2, it was shown in [15] that the necessary and sufficient condition for the existence of a BCH MDS code of length n and dimension $(n-k)$ is

$$q \equiv \pm 1 \pmod{n}. \tag{4}$$

In [19], we demonstrated that an arbitrary-size recursive MDS matrix can be efficiently constructed via RS codes [1].

It is also desirable that the MDS matrix used for diffusion in a block cipher be *involutory* so that the computations at the diffusion layer during encryption and decryption are the same. Unfortunately, an MDS matrix that is both recursive and involutory does not exist (see [24, p. 5]) over any field of characteristic 2 except for the trivial case $M = (1)$.

However, also in [24], Gupta *et al.* proposed an alternative. More specifically, they proved that over the field of characteristic 2, if f is a monic and self-reciprocal polynomial of degree m, C_f is a companion matrix of f, and $M = C_f^m$ is an MDS matrix, then M is symmetric and recursive, and RM is an involutory MDS matrix, where R is the matrix that rearranges the rows in reverse sequence in the matrix M, i.e.,

$$R = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{5}$$

RM is an involutory MDS matrix, it is convenient to implement RM at the diffusion layer. For hardware implementation, one can use the same LFSR in both encryption and decryption [24], while for software implementation, after calculating $y = Mx$, one can simply reverse the order of y to obtain RMx. This is the same for both encryption and decryption because RM is involutory.

Studying monic, self-reciprocal polynomials is crucial, as they enable the direct construction of involutory recursive-like MDS matrices, which can be efficiently integrated into the diffusion layer of block ciphers. In [24], the authors explored the conditions under which the k-th power of the companion matrix associated with a degree-k polynomial f forms an MDS matrix, and they outlined how to implement symmetric recursive matrices in hardware. We further redefine these matrices, referring to them as *self-reciprocal recursive MDS matrices*, as discussed in [25].

With the assumption $q = 2^s$, in the case $n \mid (q-1)$, it was shown in [15] that the number of BCH MDS codes of length n and dimension $(n-k)$ over $GF(q)$ is precisely $n\varphi(n)/2$, and the number of symmetric solutions (matrices) in $S^{(n)}$ is equal to $\varphi(n)/2$. The corresponding generator polynomials are given by

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

where $\beta_{(n,i)} = \alpha^i$ and α is a generator of the group of n-th roots of unity in $GF(q)$. Here, $\varphi(n)$ is the Euler's totient function, which counts the number of positive integers less than n and coprime to n, and $S^{(n)}$ is the set of generator polynomials of distinct MDS BCH codes of length n [15].

In [23], we introduced a different method to find symmetric polynomials of degree 4 over the field $GF(q)$, which uses the generator polynomials of the RS codes. We also showed that the symmetric polynomial of degree 4 only exists when $p = 2$ but not when p is an odd prime.

In this paper, using yet another different approach from [15] and [23] as discussed above, we manage to find a self-reciprocal polynomial of an arbitrary degree m from a generator polynomial $g(x)$ of degree m of the RS $[n, k, d]$ code over $GF(q)$ and establish a necessary and sufficient condition on the parameters p, r, m so that a self-reciprocal polynomial exists. This constructive approach allows us to efficiently obtain the desired involutory MDS matrices RM for all feasible parameters, thus significantly extending the previous results in [15,23–25].

Note that compared to [15], we focus on RS codes with arbitrary parameters, while they focus on binary BCH codes with a limited parameter range. For example, we show that there is no self-reciprocal polynomial of the form (1) with degree m where m is even over $GF(p^r)$ with p an odd prime. Interestingly, for $n = q - 1$, the formula in [15] gives us only

$\varphi(q-1)/2$ self-reciprocal polynomials, whereas we get $\varphi(q-1)$ polynomials, which is twice as many.

3. CONSTRUCTING SELF-RECIPROCAL MDS MATRICES OF ARBITRARY SIZE FROM *RS* CODES TO GENERATE INVOLUTORY MDS MATRICES

In this section, we present an efficient method to find self-reciprocal polynomials to construct self-reciprocal recursive MDS matrices of arbitrary size m over a finite field $GF(q)$, where $q = p^r$ and p is a prime number. From these self-reciprocal recursive MDS matrices, we can generate involutory MDS matrices of arbitrary size.

Our goal is to find the self-reciprocal generator polynomial $g(x)$ of an arbitrary degree m with the form (1) of the RS codes over the field $GF(q)$.

First, we introduce the following key lemma, which states the necessary and sufficient conditions for the polynomial of the form (1) of the RS codes to be self-reciprocal. These are the initial conditions, which form the basis for us to state and prove Theorem 1 to show specific parameters for this generator polynomial to be self-reciprocal.

Lemma 1. *Assume $q \geq 2m+1$. The necessary and sufficient condition for the generator polynomial of the form (1) of the RS code to be self-reciprocal is simultaneously satisfied*

$$(q-1) \mid (2b+d-2),\ \alpha^{\frac{(2b+d-2)(d-1)}{2}}(-1)^{d-1} = 1.$$

Proof. The polynomial $g(x)$ has degree $d-1$. Consider the reciprocal polynomial of $g(x)$

$$g^*(x) = x^{(d-1)}g(1/x). \tag{6}$$

Since $\alpha^{(b+i)}$ for $0 \leq i \leq d-2$, are all different roots of $g(x)$, then $\alpha^{-(b+i)}$ for $0 \leq i \leq d-2$ are also all different roots of $g^*(x)$ (according to Comment 1).

Now we prove the necessary condition.

Assume that $g(x)$ is self-reciprocal. Then $\alpha^{-(b+i)}$, $0 \leq i \leq d-2$, are also roots of $g(x)$. Specifically α^{-b} is a root of $g^*(x)$, so there exists i_1, such that

$$\alpha^{-b} = \alpha^{b+d-2-i_1},\ \text{for } 0 \leq i_1 \leq d-2. \tag{7}$$

This is equivalent to

$$(q-1) \mid (2b+d-2-i_1). \tag{8}$$

And it is also equivalent to

$$\alpha^{-(b+i)} = \alpha^{b+d-2-i_1-i},\ 0 \leq i \leq d-2-i_1. \tag{9}$$

Because of having the same $(b+d-2-i_1-i)-(-b-i) = 2b+d-2-i_1$.

When $i = d-2-i_1$, (4) becomes

$$\alpha^{-(b+d-2-i_1)} = \alpha^b.$$

Now we will prove $i_1 = 0$.

Suppose $i_1 \geq 1$. Then $b + d - 2 - i_1 < b + d - 2$ so $(b + d - 2 - i_1) + 1 \leq b + d - 2$, it is deduced that $\alpha^{(b+d-2-i_1)+1}$ is a root of $g(x)$, so α^{-b-d+i_1+1} is a root of $g^*(x)$ and does not belong to the left side of (4). With the assumption that $g(x)$ is self-reciprocal, this element is a root of $g(x)$. Then there exists an integer i_2 such that

$$\alpha^{-b-d+i_1+1} = \alpha^{b+d-2-i_2},\ 0 \leq i_2 \leq i_1 \Leftrightarrow \alpha^{(2b+d-2-i_1)+(d-1-i_2)} = 1. \tag{10}$$

It is deduced that $(q - 1) \mid ((2b + d - 2 - i_1) + (d - 1 - i_2))$. From here and (3), it is deduced that $(q-1) \mid (d-1-i_2)$. But $d-1 = n-k = q-1-k$ and $k > 0$ so $d-1 < q-1$, and because $i_2 \leq i_1 \leq d - 2$ then $1 \leq (d - 1 - i_2) < q - 1 - i_2 \leq q - 1$. Therefore, there cannot be $(q - 1) \mid (d - 1 - i_2)$. That is, there cannot be (5). This proves that the element α^{-b-d+i_1+1} is a root of $g^*(x)$ but not a root of $g(x)$ for all i_2 satisfying $0 \leq i_2 \leq i_1$. Then $g(x)$ cannot be self-reciprocal. Contradiction.

Hence $i_1 = 0$.

Then (2) becomes

$$\alpha^{-b} = \alpha^{b+d-2} \tag{11}$$

or $\alpha^{2b+d-2} = 1$. This is equivalent to

$$\alpha^{-(b+i)} = \alpha^{b+d-2-i},\ 0 \leq i \leq d-2. \tag{12}$$

and is also equivalent to $(q - 1) \mid (2b + d - 2)$, or

$$2b + d - 2 = t(q - 1), \tag{13}$$

where t is a positive integer (because the left side is a positive number).

We have

$$\begin{aligned} g^*(x) &= x^{(d-1)}g(1/x) = x^{(d-1)}(1/x - \alpha^b)(1/x - \alpha^{(b+1)})\ldots(1/x - \alpha^{(b+d-2)}) \\ &= (1 - \alpha^b x)(1 - \alpha^{(b+1)}x)\ldots(1 - \alpha^{(b+d-2)}x) \\ &= \alpha^{b+(b+1)+\cdots+(b+d-2)}(\alpha^{-b} - x)(\alpha^{-(b+1)} - x)\ldots(\alpha^{-(b+d-2)} - x) \\ &= \alpha^{(d-1)b+\frac{(d-1)(d-2)}{2}}(\alpha^{-b} - x)(\alpha^{-(b+1)} - x)\ldots(\alpha^{-(b+d-2)} - x). \end{aligned}$$

From (6'), we have

$$\begin{aligned} g^*(x) &= \alpha^{(d-1)b+\frac{(d-1)(d-2)}{2}}(\alpha^{b+d-2} - x)(\alpha^{b+d-3} - x)\ldots(\alpha^b - x) \\ &= \alpha^{(d-1)b+\frac{(d-1)(d-2)}{2}}(-1)^{d-1}g(x). \end{aligned} \tag{14}$$

From (8), since $g(x)$ is self-reciprocal

$$\alpha^{(d-1)b+\frac{(d-1)(d-2)}{2}}(-1)^{d-1} = 1 \Leftrightarrow \alpha^{\frac{(2b+d-2)(d-1)}{2}}(-1)^{d-1} = 1. \tag{15}$$

Next, we prove the sufficient condition. Because $(q - 1) \mid (2b + d - 2)$, it has (11) and (12). Then we have (15). Because (16) is satisfied, $g^*(x) = g(x)$, or $g(x)$ is self-reciprocal. ■

Lemma 1 provides a necessary and sufficient condition for the generator polynomial of the form (1) of the *RS* codes to be self-reciprocal. Next, we use this lemma to find specific parameters for this generator polynomial to be self-reciprocal.

Theorem 1. *Suppose $q \geq 2m+1$, $m \geq 2$. The polynomial $g(x)$ of form (1) over $GF(q)$ is self-reciprocal if and only if:*

1. When m is even: - $p = 2$ and $b = \frac{q-m}{2}$.

2. When m is odd: - $p = 2$ and $b = \frac{2q-m-1}{2}$, - p is odd and $b = \frac{q-m}{2}$.

Proof.

Necessary condition. Suppose $g(x)$ is self-reciprocal. According to Lemma 1, we have $(q-1) \mid (2b+d-2)$, or there is (13) for t as a positive integer. Since $k = n-d+1 = q-d > 0$ or $q > d$, so $q-1 > d-2$. On the other hand, since $0 \leq b \leq q-1$, from (13) it follows $t(q-1) < 3(q-1)$ or $t < 3$. That is $t = 1$ or $t = 2$. From (13), we consider the following cases to determine a set of parameters (p, r, m, b) to make (13) and (15) satisfied.

Case 1: When d is odd (i.e., m is even)

Then $2b+d-2$ is odd, and from (13) it follows that $q-1$ is odd. So q must be even, so q must be of the form $q = 2^r$ (that is, $p = 2$). Note that if q (or p) is odd, there is no (13), so according to Lemma 1, $g(x)$ is not self-reciprocal.

When $p = 2$, it is to have:

- If $t = 1$, then (13) is $q-1 = 2b+d-2$. Since $d-1$ is even, (15) is satisfied. So the solution is $b = q/2-(d-1)/2 = (q-m)/2$. We have $b \geq 0$ because $q \geq 2m+1$. Thus, the set of parameters with $p = 2$, m even, and $b = (q-m)/2$ makes (13) and (15) satisfied for all $r \geq 2$.

- If $t = 2$, then (13) is $2(q-1) = 2b+d-2$. The left side is an even positive number, and the right side is an odd positive number, so the problem has no solution for b. So $g(x)$ cannot be self-reciprocal for $t = 2$.

So in this case, when $g(x)$ is self-reciprocal, the parameters must be satisfied

$$\begin{cases} q = 2^r, \; r \geq 2, \\ b = \frac{q-m}{2}. \end{cases} \tag{16}$$

Case 2: When d is even (i.e., m is odd)

- If $t = 1$, then (13) is $q-1 = 2b+d-2$. The right side is an even positive number, so q must be odd, and p is an odd prime number. When p is odd, since $-1 = \alpha^{(q-1)/2}$ it follows $\alpha^{(2b+d-2)((d-1)/2)}(-1)^{d-1} = \alpha^{(q-1)((d-1)/2)}(-1)^{d-1} = (-1)^{d-1}(-1)^{d-1} = 1$.

  Hence, (15) is satisfied. Then $b = (q-1)/2-(d-2)/2 = (q-d+1)/2 = (q-m)/2$. We have $b > 0$. Thus, the set of parameters with p odd, m odd, and $b = (q-m)/2$ makes (13) and (15) satisfied for all $r \geq 2$.

- If $t = 2$, then (13) is $2(q-1) = 2b+d-2$, it follows $(2b+d-2)((d-1)/2) = (q-1)(d-1)$. Therefore, $\alpha^{((2b+d-2)((d-1)/2))} = 1$. We consider this further.

- If $p = 2$, then $(-1)^{d-1} = 1$ since $-1 = 1$. Therefore, (15) is satisfied and $b = q-1-(d-2)/2 = q-1-(m-1)/2 = (2q-m-1)/2$, so the equations are satisfied.

- If p is an odd prime number, then since $d-1$ is odd, $(-1)^{d-1} = -1$. Therefore, (15) is not satisfied or the problem has no solution.

Sufficient condition. Now we prove the sufficient condition, that is, assuming that there are confirmations 1) and 2) of the theorem, we need to prove $g(x)$ is self-reciprocal.

Considering the case m is even (i.e., d is odd, note that $d-1=m$):

- If $p=2$ and $b=(q-m)/2$, it is to have $2b+d-2=q-1$. Then, $\alpha^{((2b+d-2)((d-1)/2))} = \alpha^{((q-1)((d-1)/2))} = 1$ since $d-1$ is even. We also have $(-1)^{d-1}=1$ because $p=2$. It is inferred that $\alpha^{((2b+d-2)((d-1)/2))}(-1)^{d-1}=1$. Therefore, both (13) and (15) are satisfied, so according to Lemma 1, $g(x)$ is self-reciprocal.

- If p is odd, $q-1$ is even, but the left side of (13) is an odd number, so in this case (13) is not satisfied, Lemma 1 is not satisfied, so $g(x)$ is not self-reciprocal.

Considering the case of m is odd (i.e., d is even):

- If $p=2$ and $b=q-1-(m-1)/2$, it is to have $2(q-1)=2b+d-2$. That means there is (13). It is also to have $(2b+d-2)((d-1)/2)=(q-1)(d-1)$. Therefore, $\alpha^{((2b+d-2)((d-1)/2))}=1$. Because $-1=1$, so $(-1)^{d-1}=1$. That means there is (15). Thus, both (13) and (15) are satisfied. According to Lemma 1, it is inferred that $g(x)$ is self-reciprocal.

- If p is odd and $b=(q-m)/2$, then $q-1=2b+d-2$, that means there is (13). Repeat the argument in case 2 for $t=1$, we have (15). According to Lemma 1, $g(x)$ is self-reciprocal.

■

Consequences 1. *There, the self-reciprocal polynomials of the form (1) with degree m (even) over $GF(p^r)$ for p an odd prime number. Since the highest order coefficient of $g(x)$ is 1, when $g^*(x)=g(x)$ is satisfied, the constant coefficient of $g(x)$ equals 1.*

Based on the result of Theorem 1, we propose the algorithmic flowchart for generating a self-reciprocal generator polynomial for RS codes, as shown in Figure 1.

Remark 3. The field $GF(q)$ has $\varphi(q-1)$ primitive elements, so the above process described in Theorem 1 can be repeated to obtain $\varphi(q-1)$ self-reciprocal monic polynomials. When α is a primitive element of $GF(q)$, then α^s is also a primitive element of $GF(q)$ if s is a positive integer that is co-prime with $q-1$.

Example 1.

For $m=4$, the condition $q \geq 2m+1$ becomes $q \geq 9$.
- When $r=2,\ 3 \rightarrow q=4$ or 8 does not satisfy the condition $q \geq 9$.
- When $r=4 \rightarrow q=16$ satisfies $q \geq 9$, then $b=6$.
- When $r=8 \rightarrow q=256$ satisfies $q \geq 9$, then $b=126$.

For $m=8$, the condition $q \geq 2m+1$ becomes $q \geq 17$.
- When $r=3,\ 4 \rightarrow q=8$ or 16 does not satisfy the condition $q \geq 17$.
- When $r=5 \rightarrow q=32$ satisfies $q \geq 17$, then $b=12$.
- When $r=8 \rightarrow q=256$ satisfies $q \geq 17$, then $b=124$.

For $m=16$, the condition $q \geq 2m+1$ becomes $q \geq 33$.
- When $r=3,\ 4,\ 5 \rightarrow q=8,\ 16,\ 32$ does not satisfy the condition $q \geq 33$.
- When $r=6 \rightarrow q=64$ satisfies $q \geq 33$, then $b=24$.

Example 2.

For $m=3$, the condition $q \geq 2m+1$ becomes $q \geq 7$.
- When $p=2, r=2 \rightarrow q=4$ does not satisfy the condition $q \geq 7$.

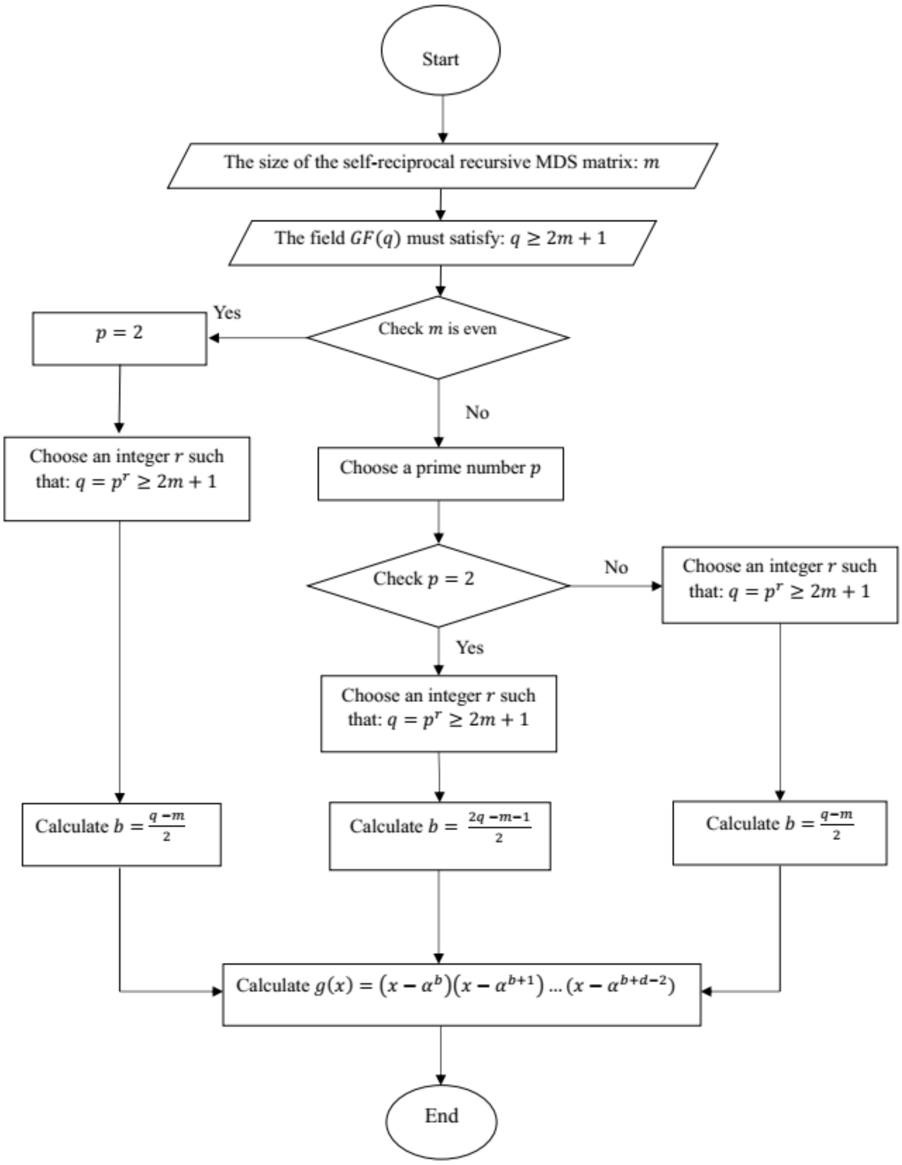
Start
The size of the self-reciprocal recursive MDS matrix: m
The field $GF(q)$ must satisfy: $q \geq 2m+1$
Check m is even
Yes
$p = 2$
No
Choose an integer r such that: $q = p^r \geq 2m+1$
Choose a prime number p
Check $p = 2$
No
Choose an integer r such that: $q = p^r \geq 2m+1$
Yes
Choose an integer r such that: $q = p^r \geq 2m+1$
Calculate $b = \frac{q-m}{2}$
Calculate $b = \frac{2q-m-1}{2}$
Calculate $b = \frac{q-m}{2}$
Calculate $g(x) = (x-\alpha^b)(x-\alpha^{b+1})\ldots(x-\alpha^{b+d-2})$
End

Figure 1: Flowchart for generating a self-reciprocal generator polynomial for RS codes

- When $p = 2, r = 3 \to q = 8$ satisfies $q \geq 7$, then $b = 6$.
- When $p = 3, r = 2 \to q = 9$ satisfies $q \geq 7$, then $b = 3$.
- When $p = 3, r = 4 \to q = 81$ satisfies $q \geq 7$, then $b = 39$.

For $m = 5$, the condition $q \geq 2m+1$ becomes $q \geq 11$.

- When $p = 2, r = 2; 3 \to q = 4; 8$ does not satisfy the condition $q \geq 11$.
- When $p = 2, r = 4 \to q = 16$ satisfies $q \geq 11$, then $b = 13$.
- When $p = 3, r = 2 \to q = 9$ does not satisfy the condition $q \geq 11$.
- When $p = 5, r = 2 \to q = 25$ satisfies $q \geq 11$, then $b = 10$.

For $m = 7$, the condition $q \geq 2m+1$ becomes $q \geq 15$.

- When $p = 2, r = 2; 3 \to q = 4; 8$ does not satisfy the condition $q \geq 15$.

- When $p = 2, r = 4 \rightarrow q = 16$ satisfies $q \geq 15$, then $b = 12$.
- When $p = 3, r = 2 \rightarrow q = 9$ does not satisfy the condition $q \geq 15$.
- When $p = 3, r = 3 \rightarrow q = 27$ satisfies $q \geq 15$, then $b = 10$.

Table 1 and Table 2 show some examples of self-reciprocal polynomials and corresponding self-reciprocal recursive MDS matrices built using RS codes from the above examples.

Thanks to Theorem 1, we can directly construct self-reciprocal generator polynomials $g(x)$ with a constant coefficient equal to 1 of the RS codes, from which the corresponding self-reciprocal recursive MDS matrices of size m can be obtained for every m over $GF(q)$, where $q = p^r$ and p is a prime number. Then it is possible to construct corresponding involutory MDS matrices. We simply need to perform the RM multiplication with the obtained self-reciprocal recursive MDS matrices to yield involutory RM MDS matrices. As analyzed in the Introduction section, these matrices are especially suitable for both software and hardware implementation.

4. COMPARISON

In this section, we provide a comparison of our results with those in [15, 23–25]. These comparisons are provided in Table 3.

The method of generating self-reciprocal recursive matrices from BCH codes in [15, 24] is quite complex. In this paper, we use RS codes to generate self-reciprocal recursive matrices in a much simpler way, and can directly specify the parameters for the self-reciprocal generator matrix g.

The authors in [15, 24] did not consider the case of generating self-reciprocal recursive MDS matrices over a general field $GF(p^r)$, where p is a prime number; they only addressed the case over the field $GF(2^s)$. Our method, on the other hand, also finds self-reciprocal generator polynomials of RS codes over the general field $GF(q)$, where $q = p^r$ and p is a prime number, including the case where p is an odd prime.

In [23], the authors only generated self-reciprocal recursive MDS matrices of size 4×4, and in [25], the authors only generated self-reciprocal recursive MDS matrices of size 8×8. In contrast, this paper demonstrates the ability to generate self-reciprocal recursive MDS matrices of arbitrary sizes.

Implementation and practical evaluation

To validate the practical efficiency of our proposed method, we implemented the construction of self-reciprocal recursive MDS matrices derived from RS codes using SageMath and Python. We evaluated several representative cases with different matrix sizes and finite fields to assess computational time, memory usage, and hardware-related performance indicators.

Table 4 presents the average computation time and memory consumption required to generate self-reciprocal generator polynomials and the corresponding MDS matrices for various matrix sizes.

These results demonstrate that the generation process is computationally lightweight, even for larger matrices over higher-order fields. From Table 4, it can be seen that using self-reciprocal generator polynomials derived from RS codes allows direct construction of recursive MDS matrices, significantly reducing computation time and memory usage.

Table 1: Some examples of self-reciprocal polynomials and corresponding self-reciprocal recursive MDS matrices over $GF(p^r)$ where $p = 2$.

№	Field GF	Primitive polynomial	Size of matrix	RS code, self-reciprocal generator polynomials	Corresponding self-reciprocal recursive MDS matrices	Corresponding Companion matrix	b
1	$GF(2^4)$	x^4+x+1	4×4	RS(15, 11, 5), $g(x)=x^4+\alpha^3x^3+\alpha x^2$ $+\alpha^3x+1$	$\begin{bmatrix}1&8&2&8\\8&D&B&E\\E&1&2&2\\2&D&5&1\end{bmatrix}$	$\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&8&2&8\end{bmatrix}$	$b=6$
	$GF(2^4)$	x^4+x^3+1	4×4	RS(15, 11, 5), $g(x)=x^4+\alpha^{12}x^3$ $+\alpha^{14}x^2+\alpha^{12}x+1$	$\begin{bmatrix}1&3&C&3\\3&4&E&9\\9&1&C&C\\C&4&7&1\end{bmatrix}$	$\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&3&C&3\end{bmatrix}$	$b=6$
3	$GF(2^4)$	x^4+x+1	5×5	RS(15, 10, 6), $g(x)=x^5+\alpha^4x^4+\alpha^{11}x^3$ $+\alpha^{11}x^2+\alpha^4x+1$	$\begin{bmatrix}1&3&E&E&3\\3&4&2&F&B\\B&D&C&A&1\\1&8&3&2&9\\9&9&F&4&A\end{bmatrix}$	$\begin{bmatrix}0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\\1&3&E&E&3\end{bmatrix}$	$b=13$
4	$GF(2^4)$	x^4+x+1	7×7	RS(15, 8, 8), $g(x)=x^7+\alpha^2x^6+\alpha^5x^5$ $+x^4+x^3+\alpha^5x^2$ $+\alpha^2x+1$	$\begin{bmatrix}1&4&6&1&1&6&4\\4&2&F&2&5&A&5\\5&3&F&A&7&8&D\\D&4&B&2&7&F&9\\9&F&7&2&B&4&D\\D&8&7&A&F&3&5\\5&A&5&2&F&2&4\end{bmatrix}$	$\begin{bmatrix}0&1&0&0&0&0&0\\0&0&1&0&0&0&0\\0&0&0&1&0&0&0\\0&0&0&0&1&0&0\\0&0&0&0&0&1&0\\0&0&0&0&0&0&1\\1&4&6&1&1&6&4\end{bmatrix}$	$b=12$
5	$GF(2^5)$	x^5+x^3+1	8×8	RS(31, 23, 9), $g(x)=x^8+\alpha^{17}x^7+\alpha^{18}x^6$ $+\alpha^6x^5+\alpha^{30}x^4+\alpha^6x^3$ $+\alpha^{18}x^2+\alpha^{17}x+1$	$\begin{bmatrix}1&18&19&12&14&12&19&18\\18&9&8&17&1E&1A&2&11\\11&17&17&9&B&1F&4&D\\D&D&6&2&1B&1E&E&18\\18&5&1D&8&E&15&E&6\\6&1A&1&A&B&19&11&C\\C&2&12&6&C&C&11&15\\15&18&3&9&18&17&D&5\end{bmatrix}$	$\begin{bmatrix}0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\\1&18&19&12&14&12&19&18\end{bmatrix}$	$b=12$
6	$GF(2^6)$	x^6+x+1	16×16	RS(63, 47, 17), $g(x)=x^{16}+\alpha^{51}x^{15}+\ldots$ $+\alpha^{51}x+1$	$\begin{bmatrix}1&2B&\ldots&2B\\2B&37&\ldots&27\\\vdots&\vdots&\ddots&\vdots\\2D&30&\ldots&3\end{bmatrix}$	$\begin{bmatrix}0&1&\ldots&0\\\vdots&\vdots&\ddots&\vdots\\0&0&\ldots&1\\1&2B&\ldots&2B\end{bmatrix}$	$b=24$
7	$GF(2^8)$	x^8+x^6 $+x^5+x^3+1$	4×4	RS(255, 251, 5), $g(x)=x^4+\alpha^{174}x^3$ $+\alpha^{234}x^2+\alpha^{174}x+1$	$\begin{bmatrix}1&96&E8&96\\96&92&83&7B\\7B&C&98&19\\19&6A&BE&89\end{bmatrix}$	$\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&96&E8&96\end{bmatrix}$	$b=126$
8	$GF(2^8)$	x^8+x^4 $+x^3+x^2+1$	8×8	RS(255, 247, 9), $g(x)=x^8+\alpha^{44}x^7$ $+\alpha^{231}x^6+\alpha^{70}x^5$ $+\alpha^{235}x^4+\alpha^{70}x^3$ $+\alpha^{231}x^2+\alpha^{33}x+1$	$\begin{bmatrix}1&EE&F5&5E&EB&5E&F5&EE\\EE&FF&5A&CB&D1&D5&EA&B\\B&E2&6&72&E0&F9&2C&E6\\E6&D6&AD&F2&F6&14&B6&F1\\F1&CD&83&57&CB&C&41&9D\\9D&EF&D0&C7&9A&8F&11&5F\\5F&4D&85&A&29&40&E5&C1\\C1&BE&38&F9&9&55&35&4\end{bmatrix}$	$\begin{bmatrix}0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\\1&EE&F5&5E&EB&5E&F5&EE\end{bmatrix}$	$b=124$

Table 2: Some examples of self-reciprocal polynomials and corresponding self-reciprocal recursive MDS matrices over $GF(p^r)$ where p is an odd prime.

№	Field GF	Primitive polynomial	Size of matrix	RS code, self-reciprocal generator polynomials	Corresponding self-reciprocal recursive MDS matrices	Corresponding Companion matrix	b
1	$GF(3^2)$	x^2+2x $+2$	3×3	RS(8, 5, 4), $g(x)=x^3+\alpha^5x^2$ $+\alpha^5x+1$	$\begin{bmatrix} 1 & 2x & 2x \\ 2x & x+2 & 1 \\ 1 & x & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2x & 2x \end{bmatrix}$	$b=3$
2	$GF(3^4)$	x^4+x+2	3×3	RS(80, 77, 4), $g(x)=x^3+\alpha^7x^2$ $+\alpha^7x+1$	$\begin{bmatrix} 1 & x^3+x+2 & x^3+x+2 \\ x^3+x+2 & 2x^2+2x+1 & x^3+2x^2+2 \\ x^3+2x^2+2 & x+1 & 2x^3+2x^2+2x \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & x^3+x+2 & x^3+x+2 \end{bmatrix}$	$b=39$
3	$GF(3^3)$	x^3+2x+1	7×7	RS(26, 19, 8), $g(x)=x^7+\alpha^{12}x^6+\alpha^8x^5$ $+\alpha^{18}x^4+\alpha^{18}x^3+\alpha^8x^2$ $+\alpha^{12}x+1$	$\begin{bmatrix} 1 & x^2+2 & 2x^2+2 & x^2+2x+1 & x^2+2x+1 & 2x^2+2 & x^2+2 \\ x^2+2 & 2x^2+x & \cdots & \cdots & \cdots & \cdots & x^2+x+1 \\ \vdots & \vdots & \ddots & & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & & \ddots & \vdots & \vdots \\ x^2+x+1 & \cdots & \cdots & \cdots & \cdots & \cdots & 2 \\ 2 & 2x^2+2x+2 & 2x^2+2x+1 & x^2+2x+1 & x^2+2x+1 & 2x^2+2 & x^2+2 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & x^2+2 & 2x^2+2 & x^2+2x+1 & x^2+2x+1 & 2x^2+2 & x^2+2 \end{bmatrix}$	$b=10$
4	$GF(5^2)$	x^2+3x+3	5×5	RS(24, 19, 6), $g(x)=x^5+\alpha^{10}x^4+\alpha^6x^3$ $+\alpha^6x^2+\alpha^{10}x+1$	$\begin{bmatrix} 1 & 3x+1 & 3 & 3 & 3x+1 \\ 3x+1 & 4x & 4+2x & 1+4x & 4x+2 \\ 4x+2 & 2x+2 & x+1 & 4x & 3x+2 \\ 3x+2 & x+2 & x+3 & 2 & x \\ x & 3 & 4x+2 & 4x+3 & 2x+3 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 3x+1 & 3 & 3 & 3x+1 \end{bmatrix}$	$b=124$

Table 3: Comparison of our results with those in [16, 23-25]

	Our Method	Method in [16]	Method in [23]	Method in [24]	Method in [25]
Finite Field	- General field $GF(q)$ with $q=p^r$, where p is a prime number, - Considering the case of odd p as well.	$GF(2^S)$	- General field $GF(q)$ with $q=p^r$, where p is a prime number	$GF(2^S)$	- General field $GF(q)$ with $q=p^r$, where p is a prime number
Size of self-reciprocal recursive MDS matrices	Arbitrary size	Even size	4×4	Even size	8×8
Efficiency	Uses very simple *RS* codes	Uses complex BCH codes	Uses very simple *RS* codes	Uses complex BCH codes	Uses complex BCH codes

5. CONCLUSIONS

Recursive MDS matrices are especially suitable for hardware implementation because they can be implemented using LFSRs. Self-reciprocal recursive MDS matrices bring even greater advantage because, based on them, we can create corresponding involutory recursive-like MDS matrices. We can then implement the same LFSR circuit for both encryption and

Table 4: Performance comparison between the proposed and traditional MDS matrix generation methods

Matrix Size	Finite Field GF(q)	Gen. Poly Degree	Time (ms)—Self-reciprocal recursive MDS matrices	Time (ms)—normal MDS matrices	Memory (KB)—Self-reciprocal recursive MDS matrices	Memory (KB)—normal MDS matrices
4×4	$GF(2^4)$	4	1.2	2.5	48	64
8×8	$GF(2^8)$	8	2.9	6.8	112	178
16×16	$GF(2^8)$	16	6.5	14.3	246	375
8×8	$GF(7^4)$	8	3.3	7.9	134	210

decryption, thus achieving a significant saving in terms of the implementation cost.

In this paper, we introduce an approach for constructing self-reciprocal recursive MDS matrices of arbitrary sizes, which are derived from the self-reciprocal generator polynomials of RS codes. This method for generating such matrices from RS codes is efficient and simple. Furthermore, we explore the self-reciprocal generator polynomials of RS codes over the general finite field $GF(q)$, where $q = p^r$ and p is a prime, including cases where p is an odd prime.

The resulting involutory MDS matrices of various sizes, derived from these self-reciprocal matrices, offer high efficiency for both hardware and software implementations, making them highly advantageous for a wide range of cryptographic applications. Our future research direction will focus on discovering new methods to generate additional self-reciprocal recursive MDS matrices that are useful for cryptographic applications.

REFERENCES

[1] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes.* Elsevier Science Publishers, 1977, vol. 2.

[2] S. Vaudenay, "On the need for multipermutations: Cryptanalysis of MD4 and SAFER," in *International Workshop on Fast Software Encryption.* Springer, 1994, pp. 286–297.

[3] K. S. Dhanalakshmi and R. A. Padmavathi, "A survey on VLSI implementation of AES algorithm with dynamic S-Box," *Journal of Applied Security Research*, vol. 17, pp. 241–256, 2022.

[4] K. Achkoun, C. Hanin, A. Sadak, F. Ziani, and F. Omary, "SPF-CA-1.2: An enhanced version of cellular automata-based block cipher system," *International Journal of Computer Mathematics: Computer Systems Theory*, vol. 6, pp. 194–208, 2021.

[5] R. Muthalagu and S. Jain, "A novel modified KASUMI block cipher for global system for mobile communications," *International Journal of Computers and Applications*, vol. 43, no. 8, pp. 805–811, 2021.

[6] H. Mirvaziri, "Cryptanalysis of MRVLK using genetic algorithm," *Journal of Applied Security Research*, vol. 13, pp. 489–501, 2018.

[7] NIST, "Advanced encryption standard (AES)," National Institute of Standards and Technology, Tech. Rep. FIPS PUB 197, November 2001.

[8] D. Mehta, M. Jha, H. Suhagiya, and R. Mangrulkar, "DieRoll: A unique key generation and encryption technique," *Journal of Applied Security Research*, vol. 19, pp. 168–195, 2024.

[9] S. Dey and R. Ghosh, "A smart review and two new techniques using 4-bit Boolean functions for cryptanalysis of 4-bit crypto S-boxes," *International Journal of Computers and Applications*, vol. 43, no. 3, pp. 199–217, 2021.

[10] A. Kuznetsov, S. Kandii, E. Frontoni, and N. Poluyanenko, "SBGen: A high-performance library for rapid generation of cryptographic S-boxes," *SoftwareX*, vol. 27, p. 101788, 2024.

[11] T. Etem and T. Kaya, "Fast image encryption algorithm with random structures," *International Journal of Computers and Applications*, vol. 45, no. 10, pp. 626–637, 2023.

[12] K. C. Gupta, S. K. Pandey, and S. Samanta, "On the construction of near-MDS matrices," *Cryptography and Communications*, vol. 16, no. 2, pp. 249–283, 2024.

[13] A. Kesarwani, S. Sarkar, and A. Venkateswarlu, "Exhaustive search for various types of MDS matrices," *IACR Transactions on Symmetric Cryptology*, pp. 231–256, 2019.

[14] K. C. Gupta and I. G. Ray, "On constructions of MDS matrices from companion matrices for lightweight cryptography," in *International Conference on Availability, Reliability, and Security.* Springer, 2013, pp. 29–43.

[15] K. C. Gupta, S. K. Pandey, and A. Venkateswarlu, "On the direct construction of recursive MDS matrices," *Designs, Codes and Cryptography*, vol. 82, pp. 77–94, 2017.

[16] ——, "Towards a general construction of recursive MDS diffusion layers," *Designs, Codes and Cryptography*, vol. 82, pp. 179–195, 2017.

[17] A. Kesarwani, S. K. Pandey, S. Sarkar, and A. Venkateswarlu, "Recursive MDS matrices over finite commutative rings," *Discrete Applied Mathematics*, vol. 304, pp. 384–396, 2021.

[18] K. C. Gupta, S. K. Pandey, and S. Samanta, "Construction of recursive MDS matrices using DLS matrices," in *International Conference on Cryptology in Africa.* Cham: Springer, 2022, pp. 3–27.

[19] T. T. Luong, "Constructing effectively MDS and recursive MDS matrices by Reed-Solomon codes," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 10–16, 2016.

[20] G. G. Güzel, M. T. Sakallı, S. Akleylek, V. Rijmen, and Y. Çengellenmiş, "A new matrix form to generate all 3 × 3 involutory MDS matrices over $\mathbb{F}_{2^m}$," *Information Processing Letters*, vol. 147, pp. 61–68, 2019.

[21] Y. Kumar, P. R. Mishra, S. Samanta, and A. Gaur, "A systematic construction approach for all 4 × 4 involutory MDS matrices," *Journal of Applied Mathematics and Computing*, pp. 1–21, 2024.

[22] T. T. Luong, N. Van Long, and B. Vo, "Efficient implementation of the linear layer of block ciphers with large MDS matrices based on a new lookup table technique," *PLOS ONE*, vol. 19, no. 6, p. e0304873, 2024.

[23] T. T. Luong, N. N. Cuong, and B. D. Trinh, "4 × 4 recursive MDS matrices effective for implementation from Reed-Solomon code over GF(q) field," in *International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences*, 2021, pp. 386–391.

[24] K. C. Gupta, S. K. Pandey, and A. Venkateswarlu, "Almost involutory recursive MDS diffusion layers," *Designs, Codes and Cryptography*, vol. 87, pp. 609–626, 2019.

[25] T. T. Luong, "On the direct building of 8 × 8 self-reciprocal recursive MDS matrices effective for implementation over GF(q) using Reed-Solomon codes," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 26, no. 4, pp. 1237–1248, 2023.

[26] T. L. Tran, N. C. Nguyen, and T. X. Dinh, "A construction of involutory recursive-like MDS matrices via self-reciprocal generator polynomials of Reed-Solomon codes," *Journal of Science and Technology on Information Security*, vol. 16, no. 2, pp. 50–59, 2022.

Received on March 07, 2025
Accepted on December 24, 2025