

## طراحی و ساخت ماتریس های سبک وزن MDS شبه خودمعکوس بر اساس ساختارهای بازگشتی و ماتریس های خلوت دودویی

علی زاغیان<sup>۱\*</sup>، سید محسن موسوی<sup>۲</sup>

۱- دانشیار، ۲- دانشجوی دکتری، دانشگاه صنعتی مالک اشتر اصفهان

(دریافت: ۹۷/۰۷/۰۹، پذیرش: ۹۷/۱۰/۱۵)

### چکیده

ماتریس های MDS یکی از مهم ترین اجزای طراحی در رمزهای قالبی است. یکی از ویژگی های اصلی یک ماتریس MDS برای ساختارهای SPN، سرعت قابل قبول پیاده سازی ماتریس MDS و معکوس آن، از نظر سخت افزاری است. در این مقاله، نوعی جدید از ماتریس های بلوکی دودویی بنام ماتریس های شبه خودمعکوس استفاده شده است که هزینه پیاده سازی این ماتریس ها و معکوس آن ها برابر است. در ابتدا، با به کارگیری توابع خطی دودویی در ماتریس های خلوت دودویی، یک ماتریس  $4 \times 4$  MDS شبه خودمعکوس پیشنهاد شده است که هزینه پیاده سازی این ماتریس پیشنهادی برای ورودی ۸ بیتی، برابر با  $68$  XOR یک بیتی است. ماتریس  $4 \times 4$  پیشنهادی و معکوس آن، پیاده سازی مناسبی از نظر سخت افزاری دارند زیرا ساختار این ماتریس ها، بر اساس ساختارهای DS1 است. در ادامه، با استفاده از ساختارهای DS1، یک ماتریس  $8 \times 8$  MDS شبه خودمعکوس پیشنهاد شده که برای ورودی ۸ بیتی، با  $320$  XOR یک بیتی پیاده سازی شده است. مهم ترین نتیجه این مقاله، پیشنهاد ماتریس  $8 \times 8$  MDS شبه خودمعکوس با هزینه  $320$  XOR است زیرا بهترین نتیجه برای ساخت ماتریس  $8 \times 8$  MDS با استفاده از الگوریتم های ذاتی و برای ورودی ۸ بیتی  $392$  XOR است. همچنین، با استفاده از ماتریس های مناسب دودویی خلوت، ماتریس  $4 \times 4$  MDS پیشنهاد شده با هزینه  $8m+4$  XOR برای ورودی  $m \geq 4$  بیتی پیاده سازی شده است.

**کلیدواژه ها:** ماتریس MDS، رمزنگاری سبک، لایه های انتشار بازگشتی، شمارش XOR، رمز قالبی

## Design and Construction of Lightweight MDS Semi Involutory Matrices Based on the Recursive Structures and Binary Sparse Matrices

A. Zaghian\*, M. Mousavi

Malek Ashtar University of Technology, Isfahan

(Received: 01/10/2018; Accepted: 05/01/2019)

### Abstract

MDS matrices are one of the most important components in designing block ciphers. Based on the hardware terminologies, the acceptable speed of the implementation of MDS matrix and its inverse is one of the main features of MDS matrix for SPN structures. In this paper, a new type of binary block matrices called semi involutory is used such that the cost of implementation of these matrices and their inverses are equal. At first, by using binary linear functions over binary sparse matrices, a  $4 \times 4$  semi involutory MDS matrix is proposed so that the cost of implementation of the proposed matrix is 68 bitwise XOR for 8 bit input. The structure of proposed  $4 \times 4$  MDS matrix and its inverse are based on the DS1 structures, so they have suitable implementation from hardware point of view. Next, a  $8 \times 8$  semi involutory MDS matrix is proposed by applying DS1 structures such that the proposed  $8 \times 8$  matrix is implemented with the 320 bitwise XOR for 8 bit input. The proposed  $8 \times 8$  semi involutory MDS matrix is the major result, since the best known result in the implementation of a  $8 \times 8$  semi involutory MDS matrix for 8 bit input, based on the heuristic algorithm, is 392 bitwise XOR while 320 bitwise XOR was obtained in this research. Moreover, the proposed  $4 \times 4$  MDS matrix is implemented with  $8m+4$  XOR for  $m \geq 4$  bit input by applying suitable binary sparse matrices.

**Keywords:** MDS Matrix, Lightweight Cryptography, Recursive Diffusion Layers, XOR Count, Block Ciphers

\*Corresponding Author E-mail: a\_zaghian@mut-es.ac.ir

## ۱. مقدمه

اگرچه که ماتریس‌های MDS به شکل‌های مختلفی ساخته شده‌اند اما سه دسته از ماتریس‌های MDS اهمیت و کاربرد بیشتری در رمزنگاری دارند. دسته اول شامل ماتریس‌های MDS ای هستند که از طریق بازگشتی<sup>۹</sup> ساخته شده‌اند [۱۵-۱۲]. این دسته، مناسب پیاده‌سازی برای سخت‌افزار است مانند ماتریس استفاده‌شده در تابع چکیده‌ساز PHOTON [۱۶]. دسته دوم از ماتریس‌های MDS به شکل ماتریس هادامارد است که در توابع چکیده‌سازی مانند Khazad [۱۷] و Maelstrom [۱۸] مورد استفاده قرار گرفته‌اند. دسته سوم از ماتریس‌های MDS به شکل ماتریس چرخشی هستند که در رمزهای قالبی مانند AES [۱۹] و تابع چکیده‌ساز *Grosth* [۲۰] به کار برده شده‌اند.

یکی از دلایل مهم استفاده از دسته‌های بازگشتی در ساخت ماتریس MDS، پیاده‌سازی سخت‌افزاری ساده و یکسان این دسته از ماتریس‌ها در رمزنگاری و رمزگشایی است. برای ساخت ماتریس‌های بازگشتی دو روش معروف وجود دارد. در روش اول با استفاده از توان  $n$  ام یک ماتریس  $n \times n$  LFSR<sup>۱۰</sup>، به شرط غیرصفر بودن درایه‌های سطر آخر آن، یک ماتریس MDS بازگشتی به دست می‌آید [۱۶]. محدودیت روش اول در غیرصفر بودن سطر آخر ماتریس LFSR است که هزینه پیاده‌سازی آن را افزایش می‌دهد. روش دوم در ساخت ماتریس بازگشتی، بر اساس ساختارهای فیستلی مانند ساختار GFS<sup>۱۱</sup> [۲۱] یا DSI<sup>۱۲</sup> است [۲۲] که به ما اجازه می‌دهد از ماتریس‌های خلوت‌تری نسبت به ماتریس LFSR، برای ساخت ماتریس MDS، استفاده نماییم. برای مثال، فرض کنید که میدان  $\mathbb{F}_{p^4}$  با استفاده از چندجمله‌ای اولیه  $f = x^4 + x^3 + 1$  ساخته شده است. همچنین، فرض نمایید که  $\alpha$  یک ریشه از  $f$  است. حال دو ماتریس  $4 \times 4$  زیر را بر روی میدان  $\mathbb{F}_{p^4}$  در نظر بگیرید:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & \alpha + 1 & 1 & \alpha \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & \alpha \\ 1 & 0 & 0 & 0 \\ 0 & 1 & \alpha + 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

ماتریس  $\mathbf{A}$  از نوع LFSR و ماتریس  $\mathbf{B}$  از نوع DSI است. می‌توان بررسی نمود که ماتریس‌های  $\mathbf{A}$  و  $\mathbf{B}$  بر روی میدان  $\mathbb{F}_{p^4}$ ، ماتریس MDS هستند. اگرچه که تمام عناصر ماتریس‌های  $\mathbf{A}$  و  $\mathbf{B}$  یکی هستند، اما برای محاسبه  $\mathbf{A} \cdot \mathbf{x}^T$ ،  $\mathbf{x} = [x_1, x_2, x_3, x_4]$ ، به سه XOR (چهارعنصر غیرصفر سطر آخر) و برای محاسبه  $\mathbf{B} \cdot \mathbf{x}^T$  تنها به دو XOR (یکی در سطر اول و دیگری در سطر سوم) نیاز است. در واقع داریم:

دو مشخصه مهم در طراحی سامانه‌های رمزنگاری مانند رمزهای قالبی و توابع چکیده‌ساز عبارت است از ویژگی‌های پخش<sup>۱</sup> و انتشار<sup>۲</sup>. در بیشتر سامانه‌های رمزنگاری متقارن از جدول‌های جانشینی به‌عنوان مشخصه پخش و ماتریس‌های MDS<sup>۳</sup> به‌عنوان مشخصه انتشار، استفاده می‌شود. یک ماتریس  $n \times n$  مانند  $\mathbf{A}$  را بر روی میدان  $\mathbb{F}$  یک ماتریس MDS گویند اگر و فقط اگر تمام زیر ماتریس‌های مربعی  $\mathbf{A}$  معکوس‌پذیر باشند [۱]. ماتریس‌های MDS نه‌فقط نقش مهمی در طراحی لایه انتشار سامانه‌های رمزنگاری برای ایجاد بیشترین انتشار را دارند، بلکه در جلوگیری از تحلیل رمزهای خطی و تفاضلی نیز مؤثر هستند [۲].

تعریف دیگری از ماتریس MDS که معادل با تعریف اول است به‌صورت زیر بیان می‌شود. به ازای هر بردار  $n$  تایی غیرصفر مانند  $\mathbf{x}$ ، با  $m_1$  درایه غیرصفر، اگر تعداد درایه‌های غیرصفر بردار  $\mathbf{Ax}$  برابر  $m_2$  باشد آن‌گاه  $m_1 + m_2 \geq n + 1$ . به عدد  $n + 1$ ، عدد انشعاب<sup>۴</sup> ماتریس  $\mathbf{A}$  گویند [۲]. ماتریس MDS را به‌جز میدان می‌توان بر روی حلقه‌ها با ویژگی جابه‌جایی به‌صورت زیر تعریف نمود. عدد انشعاب ماتریس  $n \times n$   $\mathbf{A}$  بر روی حلقه جابه‌جایی  $\mathbf{R}$  برابر  $n + 1$  است اگر درترمینان تمام زیر ماتریس‌های مربعی  $\mathbf{A}$  بر روی  $\mathbf{R}$  معکوس‌پذیر باشند [۳]. در گزاره ۱ مقاله [۴] ثابت شده است که اگر درایه‌های  $\mathbf{A}$ ، ماتریس‌های دودویی  $m \times m$  باشند آن‌گاه  $\mathbf{A}$  بر روی  $\mathbf{R}$  یک ماتریس MDS است اگر و فقط اگر درترمینان تمام زیر ماتریس‌های  $\mathbf{A}$ ، ماتریس‌های معکوس‌پذیر در میدان  $\mathbb{F}_p$  باشند.

ماتریس‌های MDS نه‌فقط در رمزنگاری بلکه در ابتدا برای ساخت کدهای MDS استفاده شده‌اند [۵]. در واقع، اگر  $\mathbf{A}$  یک ماتریس  $n \times n$  MDS بر روی یک میدان  $\mathbb{F}$  باشد و  $\mathbf{I}_n$  ماتریس همانی از مرتبه  $n$  باشد آنگاه ماتریس  $n \times 2n$  به فرم  $[\mathbf{I}_n | \mathbf{A}]$  یک مولد کد MDS است [۶]. همچنین، ماتریس‌های MDS چندگانه<sup>۵</sup> را می‌توان در افزایش جعبه‌های جانشینی فعال<sup>۶</sup> در ساختارهای فیستلی<sup>۷</sup> با تابع دور جایگشتی جانشینی<sup>۸</sup>، به کاربرد [۷-۸]. به‌جز روش جستجوی کامل [۹]، می‌توان با استفاده از ماتریس واندرموند [۱۰] و ماتریس کوشی [۱۱]، شکل‌های مختلفی از ماتریس MDS را به‌دست آورد.

<sup>1</sup> Confusion

<sup>2</sup> Diffusion

<sup>3</sup> Maximum Distance Spreadable

<sup>4</sup> Branch Number

<sup>5</sup> Multiple MDS Matrices

<sup>6</sup> Active S-Boxes

<sup>7</sup> Feistel Structures

<sup>8</sup> Substitution Permutation Round Function

<sup>9</sup> Recursive

<sup>10</sup> Linear Feedback Shift Register

<sup>11</sup> Generalized Feistel Structure

<sup>12</sup> Diagonal-Serial Invertible

## ۱-۱. پیش‌نیازها

تعاریف و نمادهای مورد استفاده در این مقاله به شرح زیر است. میدان متناهی با  $q$  عضو را با  $\mathbb{F}_q$  نمایش می‌دهیم. حلقه چندجمله‌ای‌ها با متغیر  $L$  بر پیمانه ۲ با نماد  $\mathbb{F}_2[L]$  نمایش داده می‌شود. ترانپوز و معکوس ماتریس  $A$  به ترتیب با نمادهای  $A^T$  و  $A^{-1}$ ، نمایش داده شده است. ماتریس‌های  $m \times m$  صفر و همانی را به ترتیب با نمادهای  $\mathbf{0}_m$  و  $\mathbf{I}_m$  نشان می‌دهیم. ماتریس  $n \times n$  را بر روی میدان  $\mathbb{F}_q$  خود معکوس گوئیم اگر و فقط اگر داشته باشیم  $A^{-1} = \mathbf{I}_n$  [۱۱].

در سرتاسر این مقاله منظور ما از نمادهای  $L_{(i,m)}$  برای  $1 \leq i \leq n$ ، ماتریس‌های دودویی  $m \times m$  معکوس‌پذیر بر روی میدان  $\mathbb{F}_2$  است. به ماتریس‌های  $L_{(i,m)}$ ، توابع خطی دودویی نیز گفته می‌شود [۱۲]. همچنین اگر داشته باشیم  $A = L_m$ ،  $B = L_m^{-1}$  و  $C = L_m^2$  آن‌گاه ماتریس‌های  $B$  و  $C$  به ترتیب معکوس و توان دوم ماتریس دودویی  $A$  هستند.

در ادامه تعریف ماتریس مربعی  $DSI$  بلوکی  $m$  بیتی آورده می‌شود. در واقع، تعریف ۱، یک ویرایش از تعریف ۵ در مقاله [۲۲] است که بر روی میدان متناهی، ماتریس  $DSI$  تعریف شده است.

**تعریف ۱.** ماتریس  $2n \times 2n$   $DSI$  بلوکی  $m$  بیتی با نماد  $D_{n,m}$  نمایش داده شده و به صورت زیر تعریف می‌شود:

$$D_{n,m} = \begin{pmatrix} L_{(1,m)} & \mathbf{0}_m & \mathbf{0}_m & \cdots & \cdots & \mathbf{0}_m & \mathbf{0}_m & L_{(n+1,m)} \\ L_{(n+\tau,m)} & \mathbf{0}_m & \mathbf{0}_m & \ddots & \cdots & \cdots & \mathbf{0}_m & \mathbf{0}_m \\ \mathbf{0}_m & L_{(n+\tau,m)} & L_{(2,m)} & \mathbf{0}_m & \ddots & \cdots & \cdots & \mathbf{0}_m \\ \mathbf{0}_m & \mathbf{0}_m & L_{(n+\tau,m)} & \mathbf{0}_m & \mathbf{0}_m & \ddots & \cdots & \mathbf{0}_m \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0}_m & \vdots & \cdots & \mathbf{0}_m & L_{(\tau-n,m)} & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m \\ \mathbf{0}_m & \vdots & \cdots & \cdots & \mathbf{0}_m & L_{(\tau-n,m)} & L_{(n,m)} & \mathbf{0}_m \\ \mathbf{0}_m & \mathbf{0}_m & \cdots & \cdots & \cdots & \mathbf{0}_m & L_{(\tau,n,m)} & \mathbf{0}_m \end{pmatrix}$$

که  $L_{(i,m)}$  برای  $1 \leq i \leq 2n$  ماتریس‌های  $m \times m$  دودویی معکوس‌پذیر بر روی میدان  $\mathbb{F}_2$  هستند.

فرض نماییم هزینه پیاده‌سازی ماتریس  $L_{(i,m)}$  را با  $\#L_{(i,m)}$  نشان دهیم. آن‌گاه بر اساس جدول (۱) مقاله [۲۲] هزینه پیاده‌سازی  $D_{n,m}$ ، نشان داده شده با  $\#D_{n,m}$ ، به صورت زیر است:

$$\#D_{n,m} = \sum_{i=1}^{2n} \#L_{(i,m)} + n \times m \quad (1)$$

در رابطه (۱) علت در نظر گرفتن عامل  $n \times m$  به این دلیل است که در  $n$  سطر از ماتریس  $D_{n,m}$  دو ماتریس دودویی  $m$  بیتی با یکدیگر XOR می‌شوند. همچنین، لازم به ذکر است که اگر در یک سطر از ماتریس  $D_{n,m}$  دو ماتریس یکسان داشته باشیم

$$A \cdot x^T = [x_\tau, x_\tau, x_\tau, x_\tau + (\alpha + 1)x_\tau + x_\tau + \alpha x_\tau]^T, \\ B \cdot x^T = [x_\tau + \alpha x_\tau, x_\tau, x_\tau + (\alpha + 1)x_\tau, x_\tau]^T.$$

بنابراین، نتیجه می‌گیریم که ماتریس  $B$  با ۴ XOR (یک بیتی) کمتر از ماتریس  $A$  می‌تواند پیاده‌سازی شود زیرا میدان  $\mathbb{F}_2$  از عناصر ۴ بیتی تشکیل شده است. همچنین، می‌توان نتیجه گرفت که ماتریس MDS  $B^4$ ، با ۱۶ XOR کمتر از ماتریس MDS  $A^4$  می‌تواند از نظر سخت‌افزاری پیاده‌سازی شود.

دیگر محدودیت در ساخت ماتریس MDS با استفاده از ساختار DSI این است که اگر  $A$  یک ماتریس  $n \times n$  از نوع DSI باشد آن‌گاه در قضیه ۲ مقاله [۲۲] ثابت شده است که اگر ماتریس  $A^k$  بخواهد MDS باشد آن‌گاه  $k \geq n$ . به تازگی در مقاله [۲۳] پیشنهاد شده است که برای ساخت یک ماتریس  $n \times n$  MDS از ماتریس‌های متمایز بازگشتی استفاده شود. این راه‌کار پیشنهادی باعث می‌شود که هزینه پیاده‌سازی ماتریس MDS به طور قابل ملاحظه‌ای نسبت به پیاده‌سازی‌های موجود کم شود. به عنوان نمونه در قسمت ۴ این مقاله، با استفاده از چهار ماتریس متمایز  $DSI$   $8 \times 8$ ، یک ماتریس  $MDS$   $8 \times 8$  پیشنهاد شده است که هزینه پیاده‌سازی آن برای ورودی ۸ بیتی برابر ۳۲۰ XOR یک بیتی است و این در حالی است که بهترین نتیجه شناخته شده ۳۹۲ XOR یک بیتی است [۲۴] که با استفاده از جستجو از طریق الگوریتم‌های ذاتی<sup>۱</sup> شناخته شده به دست آمده است.

یکی از ویژگی‌های مهم و کاربردی یک ماتریس MDS، ساختار معکوس آن است. اگر معکوس یک ماتریس برابر با خود ماتریس باشد به آن خودمعکوس<sup>۲</sup> گویند اما اگر هزینه پیاده‌سازی یک ماتریس برابر با هزینه پیاده‌سازی معکوس آن باشد به آن شبه خودمعکوس<sup>۳</sup> گوئیم. یکی از دلایل مهم استفاده از ماتریس‌های خودمعکوس و یا شبه خودمعکوس در رمزهای قالبی یکسان‌سازی سرعت عملیات رمزنگاری و رمزگشایی است.

در قسمت ۴ این مقاله نشان می‌دهیم که ماتریس  $MDS$   $8 \times 8$  پیشنهادی یک ماتریس شبه خودمعکوس است؛ به عبارت دیگر، هزینه پیاده‌سازی معکوس ماتریس MDS پیشنهادی ۳۲۰ XOR یک بیتی برای ورودی ۸ بیتی است. همچنین ماتریس  $MDS$   $4 \times 4$  پیشنهادی نیز یک ماتریس شبه خودمعکوس برای هر ورودی  $m$  بیتی برای  $m \geq 4$  است.

<sup>۱</sup> Heuristic Algorithm

<sup>۲</sup> Involutory

<sup>۳</sup> Semi Involutory

پیشنهادی با هزینه ۳۴۴ XOR بر روی میدان متناهی  $\mathbb{F}_8$  نیز پیاده‌سازی شده است. در پایان، خلاصه نتایج مقاله در بخش پنجم آورده شده است.

## ۲. معرفی ماتریس‌های $D_{n,m}$ شبه خودمعکوس

در این بخش می‌خواهیم ماتریس‌های شبه خودمعکوس را تعریف نماییم. در بخش‌های سوم و چهارم، ماتریس‌های  $4 \times 4$  و  $8 \times 8$  MDS شبه خودمعکوسی پیشنهاد شده است که هزینه پیاده‌سازی این ماتریس‌ها برای ورودی‌های ۸ بیتی، به ترتیب ۶۸ و ۳۲۰ XOR است.

مفهوم شبه خودمعکوسی رابطه مستقیمی با نحوه پیاده‌سازی دارد. برای نمونه اگر روش پیاده‌سازی گفته‌شده در مقاله [۲۵] را ملاک قرار دهیم تمام ماتریس‌ها هزینه پیاده‌سازی خودشان و وارونشان برابر است.

**تعریف ۲.** ماتریس مربعی  $A$  را شبه خودمعکوس توسط روش پیاده‌سازی  $P$  گویند هرگاه هزینه پیاده‌سازی ماتریس  $A$  و معکوس آن از طریق روش  $P$  یکسان باشد.

با توجه به تعریف ۲، آن دسته از ماتریس‌های  $2n \times 2n$  DSI بلوکی  $m$  بیتی را در نظر می‌گیریم که تعداد XORهای لازم برای پیاده‌سازی سخت‌افزاری این ماتریس‌ها و معکوس آن‌ها یکسان باشد.

**مثال ۲.** ماتریس  $D_{r,m}^{(1)}$  و معکوس آن را که در رابطه (۴) داده شده است در نظر بگیرید:

$$D_{r,m}^{(1)} = \begin{bmatrix} L_m & 0_m & 0_m & I_m \\ I_m & 0_m & 0_m & 0_m \\ 0_m & I_m & L_m^T & 0_m \\ 0_m & 0_m & I_m & 0_m \end{bmatrix}, \quad (D_{r,m}^{(1)})^{-1} = \begin{bmatrix} 0_m & I_m & 0_m & 0_m \\ 0_m & 0_m & I_m & L_m^T \\ 0_m & 0_m & 0_m & I_m \\ I_m & L_m & 0_m & 0_m \end{bmatrix}. \quad (۴)$$

حال فرض کنید که  $\mathbf{x} = [x_1, x_2, x_3, x_4]$  یک بردار ۴ تایی است که هر عنصر آن  $m$  بیتی است؛ بنابراین داریم:

$$D_{r,m}^{(1)} \cdot \mathbf{x}^T = [L_m x_1 + x_4, x_1, x_2 + L_m^T x_3, x_3]^T \quad (۵)$$

$$(D_{r,m}^{(1)})^{-1} \cdot \mathbf{x}^T = [x_2, x_3 + L_m^T x_4, x_4, x_1 + L_m x_3]^T$$

رابطه (۵) نتیجه می‌دهد که ماتریس  $D_{r,m}^{(1)}$  شبه خودمعکوس است زیرا اگر  $L_m$  هزینه پیاده‌سازی  $L_m$  باشد آنگاه با استفاده از رابطه (۵) هزینه پیاده‌سازی ماتریس  $D_{r,m}^{(1)}$  و معکوس آن برابر با  $L_m + \#L_m + 2m$  است. حال دو ماتریس زیر را در نظر بگیرید:

$$D_{r,m}^{(2)} = \begin{bmatrix} L_m & 0_m & 0_m & L_m^{-1} \\ I_m & 0_m & 0_m & 0_m \\ 0_m & L_m^{-1} & L_m & 0_m \\ 0_m & 0_m & I_m & 0_m \end{bmatrix}, \quad (D_{r,m}^{(2)})^{-1} = \begin{bmatrix} 0_m & I_m & 0_m & 0_m \\ 0_m & 0_m & L_m & L_m^T \\ 0_m & 0_m & 0_m & I_m \\ L_m & L_m^T & 0_m & 0_m \end{bmatrix}. \quad (۶)$$

آن‌گاه در شمارش هزینه پیاده‌سازی ماتریس  $D_{n,m}$  در رابطه (۱)، فقط هزینه یکی از آن دو ماتریس در نظر گرفته می‌شود.

**مثال ۱.** ماتریس  $4 \times 4$  DSI بلوکی ۸ بیتی زیر را در نظر بگیرید. با توجه به رابطه (۱) هزینه پیاده‌سازی آن عبارت است از:

$$D_{r,8} = \begin{bmatrix} L_8 & 0_8 & 0_8 & L_8 \\ I_8 & 0_8 & 0_8 & 0_8 \\ 0_8 & I_8 & L_8^T & 0_8 \\ 0_8 & 0_8 & L_8^{-1} & 0_8 \end{bmatrix}, \quad \#D_{r,8} = \#L_8 + \#L_8^T + \#L_8^{-1} + 2 \times 8$$

برای سادگی در نمایش ماتریس‌های دودویی، آن‌ها را با یک لیست نمایش می‌دهیم به طوری که هر درایه از این لیست شامل موقعیت‌های عناصر غیرصفر در هر سطر است. برای مثال داریم:

$$L_4 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = [[1,4], [1], [2,3], [3]] \quad (۲)$$

حال فرض کنید  $\mathbf{x} = [x_1, x_2, x_3, x_4]$  که برای  $1 \leq i \leq 4$  یک بیتی هستند. آنگاه داریم:

$$L_4 \cdot \mathbf{x}^T = [x_1 + x_4, x_1, x_2 + x_3, x_3]^T \quad (۳)$$

تفسیر رابطه (۳) به زبان پیاده‌سازی در سخت‌افزار، به این معنی است که هزینه پیاده‌سازی<sup>۱</sup> ماتریس  $L_4$  در رابطه (۲) دو XOR یک بیتی است. همچنین به دلیل ساده‌نویسی، منظور از XOR در ادامه مقاله، XOR یک بیتی است.

ادامه ساختار مقاله به صورت زیر آورده شده است. در بخش دوم، ماتریس شبه خودمعکوس، از نظر پیاده‌سازی بر اساس تعداد XOR، تعریف شده است. در بخش سوم با استفاده از دو ماتریس متمایز  $4 \times 4$  DSI بلوکی ۸ بیتی یک ماتریس  $4 \times 4$  MDS شبه خودمعکوس با هزینه ۶۸ XOR، برای ورودی ۸ بیتی، پیشنهاد شده است. همچنین، ماتریس  $4 \times 4$  DSI پیشنهادی برای هر ورودی  $m \geq 4$  بیتی، با هزینه  $8m + 4$  پیاده‌سازی شده است. در پایان بخش ۳ به این نکته اشاره شده است که ماتریس پیشنهادی در این بخش چه برتری نسبت به ماتریس  $4 \times 4$  MDS داده شده در مقاله [۲۳] دارد. در بخش چهارم با به‌کارگیری چهار ماتریس متمایز  $8 \times 8$  DSI بلوکی ۸ بیتی، یک ماتریس  $8 \times 8$  MDS شبه خودمعکوس پیشنهاد شده است به طوری که هزینه پیاده‌سازی ماتریس پیشنهاد شده برای ورودی ۸ بیتی برابر با ۳۲۰ XOR است که مهم‌ترین نتیجه این مقاله نیز است. همچنین در بخش چهارم، ماتریس  $8 \times 8$  MDS

<sup>۱</sup> The Cost of Implementation

آن‌گاه داریم:

$$\begin{aligned} \mathbf{D}_{\nu,m}^{(\nu)} \cdot \mathbf{x}^T &= [\mathbf{L}_m \mathbf{x}_1 + \mathbf{L}_m^{\nu} \mathbf{x}_\nu, \mathbf{L}_m^{\nu} \mathbf{x}_1 + \mathbf{L}_m \mathbf{x}_\nu, \mathbf{x}_1, \mathbf{x}_\nu]^T \\ (\mathbf{D}_{\nu,m}^{(\nu)})^{-1} \cdot \mathbf{x}^T &= [\mathbf{x}_1, \mathbf{L}_m (\mathbf{x}_1 + \mathbf{L}_m^{\nu} \mathbf{x}_\nu), \mathbf{x}_\nu, \mathbf{L}_m (\mathbf{x}_1 + \mathbf{L}_m^{\nu} \mathbf{x}_\nu)]^T \end{aligned} \quad (7)$$

اگرچه که بر اساس رابطه (7) هزینه پیاده‌سازی ماتریس  $\mathbf{D}_{\nu,m}^{(\nu)}$  و معکوس آن یکی نیست اما اگر هزینه پیاده‌سازی  $\mathbf{L}_m$  و  $\mathbf{L}_m^{-1}$  یکسان باشد آنگاه ماتریس  $\mathbf{D}_{\nu,m}^{(\nu)}$  نیز شبه خودمعموس است.

یک روش برای ساخت ماتریس MDS استفاده از توابع خطی دودویی به جای عناصر میدان است. ایده این روش بر اساس این نکته است که عناصر غیرصفر میدان متناهی با مشخصه 2 را می‌توان با ماتریس‌های دودویی معکوس‌پذیر در میدان  $\mathbb{F}_\nu$  نمایش داد. به عبارت دیگر فرض نماییم  $\alpha$  ریشه چندجمله‌ای اولیه‌ای از درجه  $m$  باشد که میدان با استفاده از آن چندجمله‌ای ساخته شده است. حال اگر  $\mathbf{L}_m$  نمایش ماتریس دودویی مربوط به  $\alpha$  باشد آن‌گاه نمایش ماتریسی سایر عناصر غیر صفر میدان در  $\mathbb{F}_\nu$ ، توانی از ماتریس  $\mathbf{L}_m$  است.

در پایان این بخش یک الگوریتم تصادفی برای ساخت ماتریس  $2n \times 2n$  MDS سبک‌وزن برای ورودی  $m$  بیتی ارائه می‌دهیم.

**مرحله ۱:** انتخاب ماتریس‌های شبه خودمعموس  $\mathbf{D}_{n,m}^{(i)}$  برای  $1 \leq i \leq 2n$  به طوری که ماتریس‌های انتخابی در دو شرط زیر صدق کنند: اول این که حاصل ضرب این ماتریس‌ها که با نماد  $\mathbf{H}_n$  نشان می‌دهیم، بر روی حلقه  $\mathbb{F}_\nu[\mathbf{L}]$  یک ماتریس MDS باشد. شرط دوم این است که در ساخت ماتریس‌های  $\mathbf{D}_{n,m}^{(i)}$  حداقل تعداد توابع خطی دودویی استفاده شود.

نکته مهم در مرحله اول این است که با توجه به گزاره ۱ مقاله [۴] کافی است که دترمینان تمام زیر ماتریس‌های  $\mathbf{H}_n$  بر روی  $\mathbb{F}_\nu[\mathbf{L}]$  غیرصفر شود زیرا در مرحله سوم قصد داریم به جای توابع دودویی خطی، ماتریس‌های  $m \times m$  معکوس‌پذیر قرار دهیم.

**مرحله ۲:** تمام مؤلفه‌های تحویل‌ناپذیر مربوط به دترمینان زیر ماتریس‌های  $\mathbf{H}_n$  را در میدان  $\mathbb{F}_\nu$  به دست می‌آوریم.

**مرحله ۳:** ماتریس‌های  $m \times m$  معکوس‌پذیر مانند  $\mathbf{L}_m$  را در میدان  $\mathbb{F}_\nu$  جستجو می‌کنیم که ویژگی زیر را داشته باشند: تمام مؤلفه‌های تحویل‌ناپذیر به دست آمده در مرحله ۲، به‌ازای  $\mathbf{L}_m$  ماتریس‌های معکوس‌پذیر بر روی میدان  $\mathbb{F}_\nu$  شوند. توجه شود که برای شروع مرحله ۳، ماتریس  $\mathbf{L}_m$  باید حداقل هزینه پیاده‌سازی را داشته باشد و در صورت نداشتن ویژگی ذکر شده، هزینه

پیاده‌سازی ماتریس  $\mathbf{L}_m$  را افزایش می‌دهیم.

### ۳. پیاده‌سازی ماتریس $4 \times 4$ MDS شبه خودمعموس با هزینه $4m + 4$ XOR برای ورودی $m \geq 4$ بیتی

دو ماتریس  $4 \times 4$  DSI شبه خودمعموس بلوکی زیر را بر روی ورودی  $m$  بیتی در نظر بگیریم:

$$\mathbf{D}_{\nu,m}^{(0)} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{I}_m \\ \mathbf{I}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m \\ \mathbf{0}_m & \mathbf{I}_m & \mathbf{I}_m & \mathbf{0}_m \\ \mathbf{0}_m & \mathbf{0}_m & \mathbf{I}_m & \mathbf{0}_m \end{bmatrix}, \quad \mathbf{D}_{\nu,m}^{(\nu)} = \begin{bmatrix} \mathbf{L}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{L}_m^{-1} \\ \mathbf{I}_m & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m \\ \mathbf{0}_m & \mathbf{L}_m^{-1} & \mathbf{L}_m & \mathbf{0}_m \\ \mathbf{0}_m & \mathbf{0}_m & \mathbf{I}_m & \mathbf{0}_m \end{bmatrix}$$

ماتریس  $\mathbf{H}_\nu = \mathbf{D}_{\nu,m}^{(0)} \times \mathbf{D}_{\nu,m}^{(\nu)} \times (\mathbf{D}_{\nu,m}^{(0)})^\nu$  در میدان  $\mathbb{F}_\nu$  به شکل زیر است:

$$\mathbf{H}_\nu = \begin{bmatrix} \mathbf{L}_m + \mathbf{I}_m & \mathbf{L}_m^{-1} + \mathbf{I}_m & \mathbf{L}_m + \mathbf{L}_m^{-1} + \mathbf{I}_m & \mathbf{L}_m \\ \mathbf{L}_m & \mathbf{L}_m^{-1} & \mathbf{L}_m + \mathbf{L}_m^{-1} & \mathbf{L}_m \\ \mathbf{L}_m + \mathbf{L}_m^{-1} + \mathbf{I}_m & \mathbf{L}_m & \mathbf{L}_m + \mathbf{I}_m & \mathbf{L}_m^{-1} + \mathbf{I}_m \\ \mathbf{L}_m + \mathbf{L}_m^{-1} & \mathbf{L}_m & \mathbf{L}_m & \mathbf{L}_m^{-1} \end{bmatrix} \quad (8)$$

می‌توان بررسی نمود که تعداد زیر ماتریس‌های مربعی یک

ماتریس  $n \times n$  برابر است با  $\binom{2n}{n} - 1$ ؛ بنابراین، تعداد

زیر ماتریس‌های مربعی ماتریس  $\mathbf{H}_\nu$  برابر با  $\binom{69}{4} - 1 = 69$  ماتریس

است. همچنین، می‌توان بررسی نمود که دترمینان این ۶۹

ماتریس بر روی  $\mathbb{F}_\nu[\mathbf{L}]$  غیرصفر است. حال اگر دترمینان این ۶۹

ماتریس را به عنوان چندجمله‌ای بر روی  $\mathbb{F}_\nu[\mathbf{L}]$  در نظر گرفته

شود آن‌گاه عامل‌های تحویل‌ناپذیر متمایز این ۶۹ چندجمله‌ای،

شرط‌های داده شده در رابطه (۹) است.

$$\{\mathbf{L}_m, \mathbf{L}_m + \mathbf{I}_m, \mathbf{L}_m^2 + \mathbf{L}_m + \mathbf{I}_m, \mathbf{L}_m^3 + \mathbf{L}_m + \mathbf{I}_m, \mathbf{L}_m^4 + \mathbf{L}_m + \mathbf{I}_m, \mathbf{L}_m^5 + \mathbf{L}_m + \mathbf{I}_m, \mathbf{L}_m^6 + \mathbf{L}_m + \mathbf{I}_m\} \quad (9)$$

در ادامه با استفاده از جستجوی کامپیوتری، ماتریس‌های  $m \times m$

دودویی معکوس‌پذیر را جستجو می‌کنیم که در شرایط زیر صدق

کنند. در ابتدا بنابر رابطه (7) در مثال ۲، هزینه پیاده‌سازی

ماتریس‌های جستجو شده و معکوس آن‌ها باید یکسان باشد تا

اینکه ماتریس  $\mathbf{H}_\nu$  شبه خودمعموس شود. همچنین، با

جایگذاری ماتریس‌های جستجو شده در رابطه (۹)، تمام شش

ماتریس رابطه (۹) در میدان  $\mathbb{F}_\nu$  معکوس‌پذیر شوند.

فرض کنید تعداد XOR موردنیاز برای پیاده‌سازی ماتریس  $\mathbf{L}_m$

و معکوس آن را به ترتیب با  $\#\mathbf{L}_m$  و  $\#\mathbf{L}_m^{-1}$  نشان دهیم. آنگاه با

استفاده از سطر اول رابطه (7)، تعداد XOR موردنیاز برای

پیاده‌سازی ماتریس  $\mathbf{D}_{\nu,m}^{(\nu)}$  عبارت است از  $2m + 2\#\mathbf{L}_m^{-1} + 2\#\mathbf{L}_m$ .

همچنین، هزینه پیاده‌سازی ماتریس  $\mathbf{D}_{\nu,m}^{(0)}$  نیز برابر  $2m$  است.

بنابراین، تعداد XOR‌های موردنیاز برای پیاده‌سازی ماتریس  $\mathbf{H}_\nu$

در ادامه این بخش می‌خواهیم یک ماتریس  $2^n \times 2^n$  دودویی بنام  $R_n$  با هزینه پیاده‌سازی یک XOR معرفی نماییم به طوری که هزینه پیاده‌سازی ماتریس  $H_f$  و معکوس آن برابر با  $4 \times m + \lambda$  XOR برای ورودی  $m = 2^n$  بیتی است. ماتریس  $R_n$  به صورت زیر تعریف می‌شود:

$$R_n = \left[ \left[ 2^{n-2}, 2^n \right], [1], [2], [3], \dots, [2^n - 1] \right] \quad (13)$$

فرض کنید که چندجمله‌ای مشخصه<sup>۱</sup> ماتریس  $R_n$  را با  $\text{Char}(R_n)$  نمایش دهیم. آن‌گاه در ضمیمه الف ثابت شده است که  $\text{Char}(R_n)$  در میدان  $\mathbb{F}_4$ ، به فرم داده شده در رابطه (۱۴) است.

$$\text{Char}(R_n) = (x^4 + x^2 + 1)^{2^{n-2}} \quad (14)$$

حال می‌خواهیم ثابت کنیم با جایگذاری ماتریس  $R_n$  در شرط‌های (۹) ماتریس‌های معکوس‌پذیر بر روی میدان  $\mathbb{F}_4$  به دست می‌آید. می‌توان بررسی نمود که عنصر  $L_m + L_m + I_m$  در بین این شرط‌ها وجود ندارد. از طرفی اگر شش شرط به دست آمده در رابطه (۹) را به عنوان چندجمله‌ای در میدان  $\mathbb{F}_4$  در نظر بگیریم آن‌گاه می‌توان بررسی نمود که این چندجمله‌ای‌ها بر روی میدان  $\mathbb{F}_4$  نیز تحویل‌ناپذیر هستند. فرض کنید این شش چندجمله‌ای تحویل‌ناپذیر را با  $h_k$  برای  $1 \leq k \leq 6$  نشان دهیم. رابطه (۱۴) نتیجه می‌دهد که چندجمله‌ای مینیمال  $R_n$  در میدان  $\mathbb{F}_4$  برابر با  $f = x^4 + x^2 + 1$  است؛ بنابراین اگر به ازای یک  $1 \leq k \leq 6$  در میدان  $\mathbb{F}_4$  داشته باشیم  $h_k(R_n) = 0_{2^n}$ ، آن‌گاه چندجمله‌ای  $f$  باید عاملی از  $h_k$  باشد که این تناقض با تحویل‌ناپذیری چندجمله‌ای‌های  $h_k$  است.

فرض کنید که میدان متناهی  $\mathbb{F}_{4^f}$  با استفاده از چندجمله‌ای  $f$  ساخته شده است. از مباحث مربوط به میدان متناهی می‌دانیم که ریشه‌های  $f$  در میدان  $\mathbb{F}_{4^f}$  قرار دارند. به میدان  $\mathbb{F}_{4^f}$  یک میدان گسترش یافته از طریق  $f$  نیز می‌گویند. اگر  $\alpha$  یک ریشه از  $f$  باشد آن‌گاه چهار ریشه  $f$  در میدان  $\mathbb{F}_{4^f}$  برابر با  $\alpha^i$  برای  $0 \leq i \leq 3$  است زیرا مشخصه میدان ۲ است. حال فرض کنید یک  $1 \leq k \leq 6$  وجود داشته باشد به طوری که ماتریس  $2^n \times 2^n$  دودویی  $C_{(k,n)} = h_k(R_n)$  بر روی میدان  $\mathbb{F}_4$  معکوس‌پذیر نباشد. آن‌گاه نتیجه می‌گیریم که در مینان ماتریس  $C_{(k,n)}$  در میدان  $\mathbb{F}_4$  صفر است؛ بنابراین، سخت نیست که بررسی شود حداقل یک  $0 \leq i \leq 3$  وجود دارد به طوری که  $h_k(\alpha^i) = 0$ ؛ اما این یک تناقض است زیرا دو چندجمله‌ای تحویل‌ناپذیر متمایز نمی‌توانند ریشه

و معکوس آن برای ورودی  $m$  بیتی، برابر است با  $(\#L_m = \#L_m^{-1})$ :

$$2m + (2m + 2\#L_m + 2\#L_m^{-1}) + 2m + 2m = \lambda m + 2\#L_m + 2\#L_m^{-1} \quad (10)$$

حال ماتریس‌های  $4 \times 4$  دودویی زیر را در نظر بگیرید. چهار ماتریس دودویی داده شده، در میدان  $\mathbb{F}_4$  معکوس‌پذیر هستند.

$$L_{(1,4)} = [[1, 4], [1], [2], [3]], \quad L_{(2,4)} = [[4], [1], [2], [3]], \\ L_{(3,4)} = [[4], [1], [2, 3], [3]], \quad L_{(4,4)} = [[4], [1], [2], [3, 4]]. \quad (11)$$

هزینه پیاده‌سازی چهار ماتریس داده شده در رابطه (۱۱) و معکوس آن‌ها یک XOR است. همچنین، شرط‌های داده شده در رابطه (۹) به‌ازای ماتریس‌های  $L_{(i,4)}$  برای  $1 \leq i \leq 4$ ، ماتریس‌های معکوس‌پذیر در میدان  $\mathbb{F}_4$  هستند؛ بنابراین، با به‌کارگیری رابطه (۱۰)، پیاده‌سازی ماتریس‌های  $H_f^{-1}$  و  $H_f$  ورودی ۴ بیتی، با استفاده از ماتریس‌های  $L_{(i,4)}$  برای  $1 \leq i \leq 4$ ، برابر با  $2 \times 1 + 2 \times 1 + 2 \times 1 = 6$  XOR است.

حال می‌خواهیم ماتریس‌های  $H_f$  و  $H_f^{-1}$  را با ۶۸ XOR برای ورودی‌های ۸ بیتی، پیاده‌سازی نماییم. البته به‌تازگی نتیجه ۶۷ XOR برای پیاده‌سازی ماتریس  $4 \times 4$  در حالت ورودی ۸ بیتی در مقاله [۲۳] هم گزارش شده است که در پایان این بخش به مزیت ماتریس پیشنهادی در این قسمت نسبت به ماتریس گزارش شده در [۲۳] اشاره می‌شود. همچنین، کمترین تعداد XOR برای پیاده‌سازی ماتریس‌های شبه خودمعکوس  $4 \times 4$  با استفاده از ساختارهای بازگشتی و برای ورودی‌های ۸ بیتی، ۸۰ XOR است که در مقاله [۲۲] با استفاده از روش زیر میدان پیاده‌سازی شده است.

در ابتدا هفت ماتریس دودویی معکوس‌پذیر بر روی میدان  $\mathbb{F}_4$  پیشنهاد می‌دهیم که هزینه پیاده‌سازی ماتریس‌های پیشنهادی و معکوس آن‌ها یک XOR است.

$$L_{(1,8)} = [[2, 8], [1], [2], [3], [4], [5], [6], [7]], \quad L_{(2,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \\ L_{(3,8)} = [[8], [1], [2, 4], [3], [4], [5], [6], [7]], \quad L_{(4,8)} = [[8], [1], [2], [3, 5], [4], [5], [6], [7]], \\ L_{(5,8)} = [[8], [1], [2], [3], [4, 6], [5], [6], [7]], \quad L_{(6,8)} = [[8], [1], [2], [3], [4], [5, 7], [6], [7]], \\ L_{(7,8)} = [[8], [1], [2], [3], [4], [5], [6, 8], [7]]. \quad (12)$$

می‌توان بررسی نمود که شرط‌های داده شده در (۹)، به‌ازای ماتریس‌های به دست آمده در (۱۲)، ماتریس‌های معکوس‌پذیر بر روی میدان  $\mathbb{F}_4$  هستند؛ بنابراین، با به‌کارگیری رابطه (۱۰) و استفاده از ماتریس‌های داده شده در (۱۲)، ماتریس‌های  $H_f$  و  $H_f^{-1}$  با  $2 \times 1 + 2 \times 1 + 2 \times 1 = 68$  XOR، برای ورودی ۸ بیتی پیاده‌سازی می‌شود.

<sup>۱</sup> Characteristic Polynomial

$$D_{\varphi, m}^{(k)} = \begin{bmatrix} L_m^k & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m & 1_m \\ 1_m & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m \\ 0_m & 1_m & L_m^k & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m \\ 0_m & 0_m & 1_m & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m \\ 0_m & 0_m & 0_m & 1_m & L_m^k & 0_m & 0_m & 0_m & 0_m \\ 0_m & 0_m & 0_m & 0_m & 1_m & 0_m & 0_m & 0_m & 0_m \\ 0_m & 0_m & 0_m & 0_m & 0_m & 1_m & L_m^k & 0_m & 0_m \\ 0_m & 0_m & 0_m & 0_m & 0_m & 0_m & 1_m & 0_m & 0_m \\ 0_m & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m & 1_m & 0_m \end{bmatrix} \quad (15)$$

که در ماتریس رابطه (۱۵)، منظور از عبارت  $L_m^k$ ، توان  $k$ ام ماتریس  $m \times m$  معکوس‌پذیر  $L_m$  بر روی میدان  $\mathbb{F}_\varphi$  است. می‌توان بررسی نمود که معکوس ماتریس  $D_{\varphi, m}^{(k)}$  بر روی میدان  $\mathbb{F}_\varphi$  به شکل زیر است:

$$(D_{\varphi, m}^{(k)})^{-1} = \begin{bmatrix} 0_m & 1_m & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m \\ 0_m & 0_m & 1_m & L_m^k & 0_m & 0_m & 0_m & 0_m & 0_m \\ 0_m & 0_m & 0_m & 1_m & 0_m & 0_m & 0_m & 0_m & 0_m \\ 0_m & 0_m & 0_m & 0_m & 1_m & L_m^k & 0_m & 0_m & 0_m \\ 0_m & 0_m & 0_m & 0_m & 0_m & 1_m & 0_m & 0_m & 0_m \\ 0_m & 0_m & 0_m & 0_m & 0_m & 0_m & 1_m & L_m^k & 0_m \\ 0_m & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m & 1_m & 0_m \\ 1_m & L_m^k & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m & 0_m \end{bmatrix} \quad (16)$$

عناصر ماتریس‌های  $D_{\varphi, m}^{(k)}$  و  $(D_{\varphi, m}^{(k)})^{-1}$ ، داده شده در (۱۵) و (۱۶)، یکسان هستند بنابراین، هزینه پیاده‌سازی ماتریس‌های  $D_{\varphi, m}^{(k)}$  و معکوس آن یکسان است؛ که نتیجه می‌دهد ماتریس  $D_{\varphi, m}^{(k)}$  شبه خودمعکوس می‌باشد. در ابتدا، هزینه پیاده‌سازی ماتریس  $D_{\varphi, m}^{(k)}$  را به ازای ورودی هشت‌تایی  $m$  بیتی، محاسبه می‌شود.

فرض کنید  $\mathbf{x} = [x_1, x_2, \dots, x_\varphi, x_\lambda]^T$  که مؤلفه‌های  $\mathbf{x}$  عناصر  $m$  بیتی هستند. آن‌گاه حاصل  $D_{\varphi, m}^{(k)} \cdot \mathbf{x}^T$  به صورت زیر به دست می‌آید:

$$D_{\varphi, m}^{(k)} \cdot \mathbf{x}^T = [L_m^k x_1 + x_\lambda, x_1, x_2 + L_m^k x_\varphi, x_\varphi, x_\varphi + L_m^k x_\delta, x_\delta, x_\varphi + L_m^k x_\gamma, x_\gamma]^T \quad (17)$$

حال با استفاده از رابطه (۱۷) نتیجه می‌گیریم که هزینه پیاده‌سازی ماتریس  $D_{\varphi, m}^{(k)}$  و معکوس آن برای ورودی  $m$  بیتی برابر با  $4m + 4\#L_m^k$  است. برای مثال، فرض کنید هزینه پیاده‌سازی  $D_{\varphi, m}^{(k)}$  را با نماد  $\#D_{\varphi, m}^{(k)}$  نمایش دهیم آن‌گاه داریم:

$$\begin{aligned} \#D_{\varphi, m}^{(-)} &= 4m + 4\#L_m^{-}, & \#D_{\varphi, m}^{(+)} &= 4m, \\ \#D_{\varphi, m}^{(0)} &= 4m + 4\#L_m, & \#D_{\varphi, m}^{(\varphi)} &= 4m + 4\#L_m^{\varphi}. \end{aligned} \quad (18)$$

در ادامه می‌خواهیم یک ماتریس  $8 \times 8$  MDS با استفاده از ماتریس‌های داده شده در رابطه (۱۸) پیشنهاد دهیم. ماتریس داده‌شده در رابطه (۱۹) را در نظر بگیرید (ضمیمه د):

$$H_8 = D_{\varphi, m}^{(+)} \times D_{\varphi, m}^{(0)} \times D_{\varphi, m}^{(+)} \times D_{\varphi, m}^{(-)} \times D_{\varphi, m}^{(+)} \times D_{\varphi, m}^{(\varphi)} \times D_{\varphi, m}^{(0)} \times D_{\varphi, m}^{(+)} \quad (19)$$

می‌توان بررسی نمود که ماتریس  $H_8$  دارای  $12869$  زیر ماتریس مربعی است و دترمینان‌های این  $12869$  ماتریس بر

مشترک در میدان گسترش یافته یکدیگر داشته باشند. در واقع اگر  $h_k$  ریشه مشترکی با  $f$  داشته باشد آن‌گاه  $f$  باید عاملی از  $h_k$  باشد.

فرم ماتریس  $R_n$  در رابطه (۱۳) نتیجه می‌دهد که هزینه پیاده‌سازی  $R_n$  و معکوس آن، یک XOR است؛ بنابراین، با استفاده از رابطه (۱۰) نتیجه می‌گیریم که هزینه پیاده‌سازی  $H_\varphi$  و معکوس آن با استفاده از ماتریس  $R_n$ ، برابر با  $8 \times m + 4$  XOR برای ورودی  $m = 2^n$  بیتی است. برای مثال، ماتریس  $R_8$  دودویی  $256 \times 256$  معکوس‌پذیر  $R_8$  را در نظر بگیرید. ماتریس  $R_8$ ، در واقع یک ورودی  $256$  بیتی است؛ بنابراین، با استفاده از ماتریس  $R_8$  می‌توان  $H_\varphi^{-1}$  و  $H_\varphi$  را با هزینه  $2052$  XOR برای ورودی  $256$  بیتی پیاده‌سازی نمود. یکی از مزیت‌های استفاده از ماتریس  $R_n$  در پیاده‌سازی سخت‌افزاری این است که دیگر نیازی به استفاده مستقیم از مباحث میدان متناهی در پیاده‌سازی ماتریس‌های MDS نیست.

در پایان این بخش به این نکته مهم اشاره می‌شود که گزارش قبلاً منتشر شده [۲۳]، ماتریس  $4 \times 4$  MDS با هزینه XOR  $67$  پیشنهاد شده است به طوری که بنا بر شکل (۸)، ماتریس پیشنهادی از سه ماتریس متمایز  $DSI$  و یک ماتریس جایگشتی تشکیل شده است. همچنین بنا بر همین شکل (۸)، بیشترین تعداد XOR و توابع دودویی خطی در هر مسیر از ورودی به خروجی عدد  $6$  است؛ اما بنا بر ضمیمه ب، بیشترین تعداد XOR و توابع دودویی خطی در هر مسیر از ورودی به خروجی برابر عدد  $5$  است؛ به عبارت دیگر، تنها مزیت ماتریس پیشنهادی در آن گزارش [۲۳] این است که ما در ساخت ماتریس  $4 \times 4$  MDS تنها از دو ماتریس متمایز  $DSI$  استفاده نمودیم.

### ۳. پیاده‌سازی ماتریس $8 \times 8$ شبه خودمعکوس با تعداد عملیات XOR $320$ برای ورودی $8$ بیتی

نتایج داده شده در این بخش، مهم‌ترین نتایج به دست آمده در این مقاله است. در این بخش، می‌خواهیم با استفاده از چهار ماتریس متمایز  $D_{\varphi, m}$ ، یک ماتریس شبه خودمعکوس  $8 \times 8$  MDS پیشنهاد دهیم که هزینه پیاده‌سازی آن برای ورودی  $8$  بیتی برابر با XOR  $320$  و هزینه پیاده‌سازی آن بر روی  $\mathbb{F}_{2^8}$  XOR  $344$  است. فرض کنید  $k$  یک عدد صحیح باشد. ماتریس  $8 \times 8$   $DSI$  بلوکی  $m$  بیتی زیر را در نظر بگیرید:

پیاده سازی توان دوم ماتریس های  $L_{(i,8)}$  چهار XOR است. حال با استفاده از رابطه (۲۰)، هزینه پیاده سازی ماتریس  $H_8$  و معکوس آن برابر  $320$  XOR است زیرا داریم:

$$32 \times 8 + 8 \times 2 + 8 \times 4 + 8 \times 2 = 320$$

در ادامه، می توان بررسی نمود که چند جمله ای تحویل ناپذیر  $f = x^8 + x^7 + x^5 + x^4 + 1$  در بین شرط های داده شده در ضمیمه ج نیست؛ بنابراین، می توان ماتریس  $H_8$  را بر روی میدان  $\mathbb{F}_{2^8}$  با استفاده از چند جمله ای  $f$  پیاده سازی نمود. فرض کنید  $\alpha$  ریشه  $f$  باشد. نمایش ماتریسی  $\alpha$  در میدان  $\mathbb{F}_2$  به صورت زیر است:

$$L = [[8], [1], [2], [3], [4, 8], [5, 8], [6], [7, 8]]$$

هزینه پیاده سازی ماتریس های  $L$  و  $L^{-1}$  به ترتیب ۳، ۳ و  $5$  XOR است. برای مثال فرض نمایید  $x = [x_7, \dots, x_0]$ . آنگاه هزینه پیاده سازی ماتریس  $L^{-1}$  برابر ۵ است زیرا داریم:

$$L^{-1} \cdot x^T = [x_7 + x_8, x_6 + x_8, x_5 + x_8, x_4 + x_8, x_3 + x_8, x_2 + x_8, x_1 + x_8, x_0 + x_8]^T, \\ u_1 = x_7 + x_8, u_2 = x_6 + x_8, u_3 = x_5 + x_8, u_4 = x_4 + x_8, u_5 = x_3 + x_8, u_6 = x_2 + x_8, u_7 = x_1 + x_8, u_8 = x_0 + x_8.$$

بنابراین، هزینه پیاده سازی ماتریس  $H_8$  در رابطه (۱۶)، با استفاده از ماتریس  $L$  در رابطه (۲۱)، برابر با  $344$  XOR است زیرا با استفاده از رابطه (۲۰) داریم:

$$32 \times 8 + 8 \times 3 + 8 \times 5 + 8 \times 3 = 344$$

به عبارت دیگر، ماتریس  $H_8$  را می توان بر روی میدان متناهی  $\mathbb{F}_{2^8}$ ، ساخته شده با چند جمله ای  $f$ ، با هزینه  $344$  XOR پیاده سازی نمود. مقایسه نتایج به دست آمده در این مقاله، با بهترین نتیجه های موجود در جدول (۱) آورده شده است.

جدول ۱. مقایسه نتایج این مقاله با نتایج موجود، برای ورودی ۸ بیتی

| مرجع   | هزینه پیاده سازی معکوس | هزینه پیاده سازی ماتریس | نوع ماتریس | اندازه ماتریس |
|--------|------------------------|-------------------------|------------|---------------|
| [۲۲]   | ۸۰                     | ۸۰                      | DSI        | $4 \times 4$  |
| [۲۴]   | ---                    | ۷۲                      | Hadamard   | $4 \times 4$  |
| قسمت ۳ | ۶۸                     | ۶۸                      | DSI        | $4 \times 4$  |
| [۲۳]   | ۶۷                     | ۶۷                      | DSI        | $4 \times 4$  |
| [۲۲]   | ۵۷۶                    | ۵۷۶                     | LFSR       | $8 \times 8$  |
| [۴]    | ۵۲۰                    | ۵۲۰                     | LFSR       | $8 \times 8$  |
| [۲۴]   | ---                    | ۳۹۲                     | Cauchy     | $8 \times 8$  |
| قسمت ۴ | ۳۲۰                    | ۳۲۰                     | DSI        | $8 \times 8$  |

روی حلقه چند جمله ای های  $\mathbb{F}_2[L]$  غیر صفر است. همچنین اگر دترمینان های این  $12869$  ماتریس را به عنوان چند جمله ای بر روی  $\mathbb{F}_2[L]$  در نظر بگیریم آنگاه عامل های تحویل ناپذیر متمایز این  $12869$  چند جمله ای، شرط های داده شده در ضمیمه ج است که تعداد آن  $604$  چند جمله ای تحویل ناپذیر است. حال با استفاده از رابطه های (۱۸) و (۱۹) هزینه پیاده سازی ماتریس  $H_8$ ، به صورت زیر به دست می آید:

$$(4m + (4m + 4\#L_m) + (4m + 4\#L_m^{-1}) + (4m + \#L_m^{-1})) \times 2 \\ = 32m + 8\#L_m + 8\#L_m^{-1} + 8\#L_m^{-1} \quad (20)$$

در ادامه با استفاده از جستجوی کامپیوتری به دنبال ماتریس های  $m \times m$  معکوس پذیر روی میدان  $\mathbb{F}_2$  هستیم که ویژگی های زیر را داشته باشند. در ابتدا، پیاده سازی ماتریس های جستجو شده و معکوس آن ها، یک XOR (حداقل مقدار ممکن) هزینه داشته باشد. همچنین، با جایگذاری ماتریس های جستجو شده در  $604$  شرط به دست آمده در ضمیمه ج،  $604$  ماتریس معکوس پذیر بر روی میدان  $\mathbb{F}_2$  حاصل شود.

در یکی از گزارش ها [۲۴]، ابتدا با استفاده از الگوریتم های ذاتی یک ماتریس  $8 \times 8$  MDS بر روی ورودی ۴ بیتی پیاده سازی شده است. سپس با استفاده از روش زیرمیدان یک ماتریس  $8 \times 8$  MDS با هزینه  $392$  XOR برای ورودی ۸ بیتی به دست آورده است. در این قسمت قصد داریم با استفاده از توابع دودویی خطی، ماتریس  $H_8$  در رابطه (۱۹) و معکوس آن را با هزینه  $320$  XOR بر روی ورودی ۸ بیتی پیاده سازی نماییم. همچنین ماتریس  $H_8$  را با هزینه  $344$  XOR بر روی میدان متناهی  $\mathbb{F}_{2^8}$  پیاده سازی نماییم. در ابتدا، ماتریس های  $8 \times 8$  دودویی داده شده در رابطه زیر را در نظر بگیرید. می توان بررسی نمود که ماتریس های  $L_{(i,8)}$  برای  $1 \leq i \leq 18$  بر روی میدان  $\mathbb{F}_2$  معکوس پذیر هستند.

$$L_{(1,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \quad L_{(2,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \\ L_{(3,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \quad L_{(4,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \\ L_{(5,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \quad L_{(6,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \\ L_{(7,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \quad L_{(8,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \\ L_{(9,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \quad L_{(10,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \\ L_{(11,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \quad L_{(12,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \\ L_{(13,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \quad L_{(14,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \\ L_{(15,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \quad L_{(16,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \\ L_{(17,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]], \quad L_{(18,8)} = [[8], [1], [2], [3], [4], [5], [6], [7]].$$

با جستجوی کامپیوتری بررسی شد که با جایگذاری ماتریس های  $8 \times 8$   $L_{(i,8)}$  برای  $1 \leq i \leq 18$  در  $604$  شرط داده شده در ضمیمه ج، ماتریس های معکوس پذیر در میدان  $\mathbb{F}_2$  حاصل می شود. همچنین، هزینه پیاده سازی ماتریس های  $L_{(i,8)}$  برای  $1 \leq i \leq 18$  و معکوس آن ها دو XOR است؛ بنابراین، هزینه



- [10] Sajadieh, M.; Dakhilalian, M.; Mala, H.; Omooni, B. "On Construction of Involutory MDS Matrices from Vandermonde Matrices"; *Design Code Cryptogr.* 2012, 64, 287-308.
- [11] Youssef, A. M.; Mister, S.; Tavares, S. E. "On the Design of Linear Transformations for Substitution Permutation Encryption Networks"; *Selected Areas of Cryptography* 1997, 40-48.
- [12] Sajadieh, M.; Dakhilalian, M.; Mala, H.; Sepehrdad, P. "Recursive Diffusion Layers for Block Ciphers and Hash Functions"; *Fast Software Encryption* 2012, 7549, 385-401.
- [13] Berger, T. P. "Construction of Recursive MDS Diffusion Layers from Gabidulin Codes"; *Int. Conf. Cryptol. India (INDOCRYPT)* 2013, 8250, 274-285.
- [14] Augot, D.; Finiasz, M. "Direct Construction of Recursive MDS Diffusion Layers Using Shortened BCH Codes"; *Fast Software Encryption* 2014, 8540, 3-17.
- [15] Gupta, K. C.; Pandey, S. K.; Venkateswarlu, A. "Almost Involutory Recursive MDS Diffusion Layers"; *Design Code Cryptogr.* 2018, 1-18.
- [16] Guo, J.; Peyrin, T.; Poschmann, A. "The PHOTON Family of Lightweight Hash Functions"; *Adv. Cryptol.* 2011, 684, 222-239.
- [17] Barreto, P.; Rijmen, V. "The Khazad Legacy-Level Block Cipher"; *Proc. of the 1<sup>st</sup> Open NESSIE Workshop, Belgium*, 2000.
- [18] Filho, G. D.; Barreto, P.; Rijmen, V. "The Maelstrom-0 Hash Function"; *Proc. 6<sup>th</sup> Brazilian Sym. Inform. Computer Syst. Secur.* 2006.
- [19] Daemen, J.; Rijmen, V. "The Design of Rijndael: AES-The Advanced Encryption Standard"; Springer-Verlag 2002.
- [20] Gauravaram, P.; Knudsen, L. R.; Matusiewicz, K.; Mendel, F.; Rechberger, C.; Schläffer, M.; Thomsen S. "Groestl a SHA-3 Candidate"; <http://www.groestl.info>, 2008.
- [21] Shibutani K. "On the Diffusion of Generalized Feistel Structures Regarding Differential and Linear Cryptanalysis"; *Selected Areas of Cryptography* 2011, 6544, 211-228.
- [22] Toh, D.; Teo, J.; Khoo, K.; Sim, S. M. "Lightweight MDS Serial-Type Matrices with Minimal Fixed XOR Count"; *Int. Conf. Cryptol. (AFRICACRYPT)* 2018, 10831, 51-71.
- [23] Duval, S.; Leurent, G. "MDS Matrices with Lightweight Circuits"; *IACR Trans. Symmetric Cryptol.* 2018, 48-78.
- [24] Kranz, T.; Leander, G.; Stoelen, K.; Wiemer, F. "Shorter Linear Straight-Line Programs for MDS Matrices"; *IACR Trans. Symmetric Cryptol.* 2017, 188-211.
- [25] Zhao, R.; Wu, B.; Zhang, R.; Zhang, Q. "Designing Optimal Implementations of Linear Layers (Full Version)"; *Cryptology ePrint Archive*, 2016, 1118.

#### ۴. نتیجه‌گیری

در این مقاله با استفاده از به‌کارگیری ماتریس‌های دودویی خلوت در ساختارهای DSI، دو ماتریس  $4 \times 4$  و  $8 \times 8$  پیشنهاد شده است به طوری که هزینه پیاده‌سازی این ماتریس‌ها و معکوس آن‌ها، برای ورودی ۸ بیتی، به ترتیب ۶۸ و ۳۲۰ XOR یک بیتی است. به دلیل اینکه ساختارهای ماتریس‌های MDS پیشنهاد شده، بر اساس ساختارهای DSI است، پیاده‌سازی این ماتریس‌ها و معکوس آن‌ها در سخت‌افزار به‌سادگی انجام می‌شود. همچنین، ماتریس  $4 \times 4$  پیشنهادی در این مقاله را برای ورودی  $m \geq 4$  بیتی با هزینه  $8m + 4$  پیاده‌سازی نمودیم. در پایان، ماتریس  $8 \times 8$  پیشنهادی بر روی میدان متناهی  $\mathbb{F}_8$  با هزینه XOR ۳۴۴ پیاده‌سازی شد.

#### ۵. مرجع‌ها

- [1] Blaum, M.; Roth, R. M. "On Lowest Density MDS Codes"; *IEEE Trans. Inform. Theory* 1999, 45, 46-59.
- [2] Junod, P.; Vaudenay, S. "Perfect Diffusion Primitives for Block Ciphers Building Efficient MDS Matrices"; *Selected Areas in Cryptography* 2004, 3357, 84-99.
- [3] Augot, D.; Finiasz, M. "Exhaustive Search for Small Dimension Recursive MDS Diffusion Layers for Block Ciphers and Hash Functions"; *Proc. IEEE ISIT* 2013, 1551-1555.
- [4] Wu, S.; Wang, M.; Wu, W. "Recursive Diffusion Layers for (Lightweight) Block Ciphers and Hash Functions"; *Selected Areas of Cryptography* 2012, 7707, 355-371.
- [5] MacWilliams, F. J.; Sloane, N. J. A. "The Theory of Error Correcting Codes"; North-Holland 1977.
- [6] Roth, R. M.; Seroussi, G. "On Generator Matrices of MDS Codes"; *IEEE Trans. Inform. Theory* 1985, 31, 826-830.
- [7] Shirai, T.; Shibutani, K. "On Feistel Structures Using a Diffusion Switching Mechanism"; *Fast Software Encryption* 2006, 4047, 41-56.
- [8] Mirghadri, A.; Yosefipour, M.; Khadem, B.; Sajadieh, M. "Two New Methods for Designing 192-bit Block Ciphers Based on Switching Structure and Recursive Diffusion Layers"; *J. Passive Defence Sci. & Technol.* 2016, 7, 251-259.
- [9] Sim, S. M.; Khoo, K.; Oggier, F.; Peyrin, T. "Lightweight MDS Involution Matrices"; *Fast Software Encryption* 2015, 9054, 471-493.

ضمیمه الف

چند جمله‌ای مشخصه ماتریس  $\mathbf{R}_n$  بر روی میدان  $\mathbb{F}_p$  به صورت  $\text{Char}(\mathbf{R}_n) = |x\mathbf{I}_{2^n} + \mathbf{R}_n|$  به دست می‌آید. اگر دترمینان را بر روی ستون آخر ماتریس  $x\mathbf{I}_{2^n} + \mathbf{R}_n$  انجام دهیم آنگاه داریم:

$$\text{Char}(\mathbf{R}_n) = 1 \times |\mathbf{R}_n^1| + x \times |\mathbf{R}_n^n| \quad (22)$$

در رابطه (۲۲) ماتریس  $\mathbf{R}_n^1$  یک ماتریس بالا مثلثی است که روی قطر اصلی آن، عدد یک است که نتیجه می‌دهد  $|\mathbf{R}_n^1| = 1$ . همچنین ماتریس  $\mathbf{R}_n^n$  یک ماتریس  $(2^n - 1) \times (2^n - 1)$  به شکل زیر است:

$$\mathbf{R}_n^n = \begin{bmatrix} x & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 1 & x & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & x & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & x & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & \vdots & \vdots & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 1 & x & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 1 & x & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 & x \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

در ماتریس  $\mathbf{R}_n^n$ ، عدد یک در سطر اول در  $2^{n-2}$  امین موقعیت قرار دارد. حال اگر عمل دترمینان گیری را بر روی سطر اول ماتریس  $\mathbf{R}_n^n$  انجام دهیم آنگاه داریم:

$$|\mathbf{R}_n^n| = x \times |\mathbf{H}_1| + 1 \times |\mathbf{H}_p| \quad (23)$$

در رابطه (۲۳) ماتریس  $\mathbf{H}_1$  یک ماتریس  $(2^n - 2) \times (2^n - 2)$  پایین مثلثی است که روی قطر آن  $x$  واقع شده است؛ بنابراین داریم  $|\mathbf{H}_1| = x^{2^n - 2}$ . همچنین، ماتریس  $\mathbf{H}_p$  به صورت زیر است:

$$\mathbf{H}_p = \begin{bmatrix} 1 & x & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & x & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & x & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 1 & x & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & 1 & x \end{bmatrix}$$

تعداد اعداد یک بر روی قطر اصلی ماتریس  $\mathbf{H}_p$  برابر با  $2^{n-2} - 1$  تا است. اگر عمل دترمینان گیری را روی ستون اول ماتریس  $\mathbf{H}_p$  انجام دهیم آنگاه با کمی محاسبه می‌توان نشان داد:

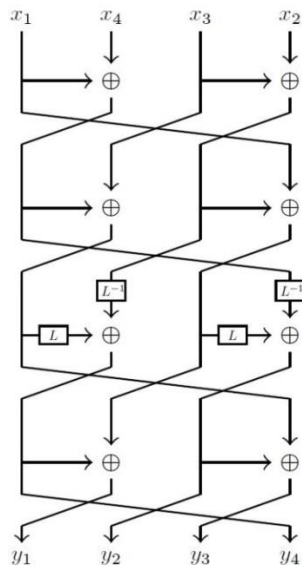
$$|\mathbf{H}_p| = x^{2^n - 2^{n-2} - 1} = x^{3 \times 2^{n-2} - 1} \quad (24)$$

با استفاده از رابطه‌های (۲۳) و (۲۴) در رابطه (۲۲) داریم:

$$\begin{aligned} \text{Char}(\mathbf{R}_n) &= 1 \times 1 + x \times (x^{2^n - 1} + x^{3 \times 2^{n-2} - 1}) \\ &= x^{2^n} + x^{3 \times 2^{n-2}} + 1 = (x^4 + x^3 + 1)^{2^{n-2}} \end{aligned} \quad (25)$$

در رابطه (۲۵) از این واقعیت استفاده نمودیم که در میدان متناهی با مشخصه ۲ داریم  $(a+b)^2 = a^2 + b^2$ .

ضمیمه ب



ضمیمه ج

ماتریس  $\mathbf{H}_8$  در رابطه (۱۹)، به صورت زیر به دست آمده است:

$$\mathbf{H}_8 = \begin{bmatrix} L^4 + 1 & L^2 + L + L^{-1} + 1 & L^4 + 1 & L^5 + L^2 + L^{-2} + 1 \\ L^4 & 1 & L^2 + L + L^{-1} + 1 & L^2 + L^{-2} \\ L^6 & L^4 & L^4 + 1 & L^2 + L + L^{-1} + 1 \\ L^6 + L^3 & L^4 & L^4 & 1 \\ L^{-2} & L^6 + L^3 & L^6 & L^4 \\ L^5 + L^2 + L^{-2} + 1 & L^6 & L^6 + L^3 & L^4 \\ L^4 + 1 & L^5 + L^2 + L^{-2} + 1 & L^{-2} & L^6 + L^3 \\ L^2 + L + L^{-1} + 1 & L^2 + L^{-2} & L^5 + L^2 + L^{-2} + 1 & L^6 \\ L^{-2} & L^6 + L^3 & L^6 & L^4 \\ L^5 + L^2 + L^{-2} + 1 & L^6 & L^6 + L^3 & L^4 \\ L^4 + 1 & L^5 + L^2 + L^{-2} + 1 & L^{-2} & L^6 + L^3 \\ L^2 + L + L^{-1} + 1 & L^2 + L^{-2} & L^5 + L^2 + L^{-2} + 1 & L^6 \\ L^4 + 1 & L^2 + L + L^{-1} + 1 & L^4 + 1 & L^5 + L^2 + L^{-2} + 1 \\ L^4 & 1 & L^2 + L + L^{-1} + 1 & L^2 + L^{-2} \\ L^6 & L^4 & L^4 + 1 & L^2 + L + L^{-1} + 1 \\ L^6 + L^3 & L^4 & L^4 & 1 \end{bmatrix}$$

## ضمیمه د

## ادامه لیست چندجمله‌های تحویل‌ناپذیر داده شده در ضمیمه ج

0x263F, 0x273B, 0x276B, 0x2ADB, 0x2BAD, 0x2D67, 0x2E5B,  
 0x2EB9, 0x2F65, 0x34F9, 0x363B, 0x378B, 0x37E1, 0x3B53,  
 0x3C3B, 0x3D4D, 0x3D69, 0x3DB1, 0x3E69, 0x3F43, 0x3F45,  
 0x2EFD, 0x2FDD, 0x2FED, 0x3BF3, 0x3F37, 0x4021, 0x4941,  
 0x5007, 0x5301, 0x405F, 0x41F1, 0x44D5, 0x45C9, 0x491D,  
 0x4A93, 0x4CD1, 0x505D, 0x5265, 0x5271, 0x58B1, 0x5A25,  
 0x5AC1, 0x5E81, 0x609D, 0x6389, 0x681D, 0x7903, 0x45F3,  
 0x47D3, 0x4BB9, 0x572D, 0x5E47, 0x6697, 0x693D, 0x6BA5,  
 0x6D69, 0x6DC3, 0x715D, 0x719B, 0x722F, 0x754B, 0x761D,  
 0x7653, 0x7655, 0x7695, 0x76A3, 0x786D, 0x7A35, 0x7E61,  
 0x53BF, 0x597F, 0x74FD, 0x79D7, 0x7B5D, 0x8431, 0x84A1,  
 0x9061, 0x9809, 0xA481, 0xAC01, 0x8371, 0x84B5, 0x9075,  
 0x94E1, 0x9919, 0xA06B, 0xA329, 0xA495, 0xB0E1, 0xCC85,  
 0xD0A3, 0x8DE9, 0x96D3, 0x9A6D, 0xA5A7, 0xAB39, 0xAD1B,  
 0xB3A9, 0xB6D1, 0xC6E3, 0xCCA7, 0xD663, 0xE46B, 0xED15,  
 0xF30B, 0xF561, 0xF683, 0x9EED, 0xAC7F, 0xB1DF, 0xB3DD,  
 0xB5BB, 0xBB4F, 0xCF2F, 0xD66F, 0xDFA5, 0xEDB3, 0xF7DB,  
 0x18105, 0x101C7, 0x13911, 0x144A3, 0x1133D, 0x130F5,  
 0x15E0B, 0x15E25, 0x16297, 0x17487, 0x18EA9, 0x1C25D,  
 0x1C969, 0x1D24D, 0x1E42B, 0x10FE7, 0x177A5, 0x1AE5D,  
 0x1AE9B, 0x1B38F, 0x1FAB1, 0x1BF6B, 0x1BFAD, 0x1E777,  
 0x1F56F, 0x1FDD3, 0x2C041, 0x24195, 0x25191, 0x28435,  
 0x28515, 0x217A9, 0x23BC1, 0x26999, 0x26CA9, 0x26D61,  
 0x28CCD, 0x29873, 0x29D29, 0x31987, 0x31B85, 0x3C255,  
 0x29FA9, 0x2BEC9, 0x2D8F5, 0x3558F, 0x37D85, 0x38FA3,  
 0x39FA1, 0x3B9C3, 0x3EEC1, 0x3F4C9, 0x27EBD, 0x2DD77,  
 0x48301, 0x421ED, 0x45B49, 0x5B443, 0x65545, 0x65781,  
 0x68CD1, 0x72B21, 0x46F47, 0x47B93, 0x5135F, 0x58ED9,  
 0x62BD3, 0x660BF, 0x7B169, 0x4F37D, 0x54FF3, 0x5F74B,  
 0x69F3B, 0x6DBE5, 0x7695F, 0x7E07F, 0xA462D, 0xFE009,  
 0xA69A7, 0xAA6C7, 0xC9A97, 0xE283F, 0xE6A4D, 0xFF103,  
 0x9F6D9, 0xF4773, 0xF9C4F, 0xFA6D5, 0xFB751, 0xFF0B3,  
 0xFFD3F, 0x105C05, 0x10507D, 0x1052AB, 0x10EA43, 0x17120B,  
 0x12AE39, 0x151E69, 0x1CE683, 0x1FB043, 0x12AFCB, 0x1CE83F,  
 0x340021, 0x26C125, 0x280DC3, 0x243F83, 0x2B1279, 0x2FA071,  
 0x2FAA11, 0x30387D, 0x3226E5, 0x3815E3, 0x3C2663, 0x3AC57F,  
 0x3F427F, 0x27FEED, 0x400C95, 0x450911, 0x405D45, 0x423195,  
 0x4988C5, 0x43A933, 0x706555, 0x57314F, 0x64DF91, 0x6B44AF,  
 0x71CB6F, 0x75AF2D, 0x7A567D, 0x7ED6C9, 0x7F7833, 0x81503D,  
 0x920BA1, 0xA0B191, 0xD4500B, 0x90570F, 0x921FA1, 0x925D83,  
 0xA24F15, 0x81F1F3, 0xD464F9, 0xF13C8D, 0xF951D1,  
 0x1C09423, 0x10BC10F, 0x11147D1, 0x118499B,  
 0x15138C5, 0x1A6058D, 0x14BA2EF, 0x1C7F193,  
 0x1D741BD, 0x240BC21, 0x31C2C91, 0x3A8CB49,  
 0x30CD3CF, 0x399B917, 0x2FF22CF, 0x3D1E87F,  
 0x52485C5, 0x543B111, 0x464EAC5, 0x550A9D5,  
 0x422A6FF, 0x4603DBF, 0x524FF91, 0xBB07017,  
 0x101B02C1, 0x15CFAB23, 0x3C077C4F, 0x3FDF6D43,  
 0x440408E5, 0x40458B45, 0x401B3BC1, 0x40299DC5,  
 0x417E2371, 0xFC73341F, 0x105580585, 0x105CA9185.

بخاطر ساده‌نویسی، چندجمله‌های تحویل‌ناپذیر را با استفاده از  
 مقادیر هگزایمایش داده‌ایم. برای مثال چندجمله‌ای  $x^8 + x + 1$   
 به صورت 0x13 نمایش داده شده است.

0x2, 0x3, 0x7, 0xB, 0xD, 0x13, 0x19, 0x1F,  
 0x25, 0x29, 0x2F, 0x37, 0x3B, 0x3D, 0x43, 0x49,  
 0x61, 0x57, 0x5B, 0x67, 0x6D, 0x73, 0x75, 0x83,  
 0x89, 0x91, 0xC1, 0x8F, 0x9D, 0xA7, 0xAB, 0xB9,  
 0xCB, 0xD3, 0xD5, 0xE5, 0xF1, 0xBF, 0xEF, 0xF7,  
 0xFD, 0x11B, 0x11D, 0x12B, 0x12D, 0x139, 0x14D,  
 0x163, 0x165, 0x169, 0x171, 0x187, 0x18B, 0x18D,  
 0x1A3, 0x1A9, 0x1C3, 0x13F, 0x15F, 0x177, 0x17B,  
 0x19F, 0x1CF, 0x1D7, 0x1DD, 0x1E7, 0x1F3, 0x1F5,  
 0x1F9, 0x221, 0x301, 0x217, 0x21B, 0x22D, 0x233,  
 0x24B, 0x265, 0x269, 0x287, 0x295, 0x299, 0x2A5,  
 0x2D1, 0x315, 0x323, 0x331, 0x349, 0x361, 0x385,  
 0x3A1, 0x25F, 0x277, 0x27D, 0x2B7, 0x2BD, 0x2CF,  
 0x2DB, 0x2F5, 0x2F9, 0x31F, 0x33B, 0x34F, 0x36B,  
 0x36D, 0x373, 0x38F, 0x3B9, 0x3CB, 0x3CD, 0x3D5,  
 0x3D9, 0x3E3, 0x3E9, 0x37F, 0x3FB, 0x409, 0x481,  
 0x41B, 0x41D, 0x435, 0x453, 0x4A9, 0x50B, 0x519,  
 0x523, 0x543, 0x585, 0x613, 0x623, 0x625, 0x631,  
 0x651, 0x685, 0x6C1, 0x721, 0x781, 0x4D7, 0x4F3,  
 0x557, 0x567, 0x56B, 0x58F, 0x597, 0x59B, 0x5B9,  
 0x637, 0x64F, 0x65B, 0x679, 0x6A7, 0x6CD, 0x6D3,  
 0x70F, 0x747, 0x763, 0x787, 0x78D, 0x793, 0x7A9,  
 0x5F7, 0x5FB, 0x6BF, 0x6F7, 0x6FD, 0x77B, 0x7DB,  
 0x7F9, 0x7FF, 0x805, 0x863, 0x871, 0x895, 0x8A9,  
 0x8B1, 0x8D1, 0x8E1, 0x913, 0x929, 0xA07, 0xA49,  
 0xC0B, 0xC89, 0xE81, 0x89F, 0x8CF, 0x8E7, 0x8F5,  
 0x95B, 0x9AD, 0x9C7, 0xA6D, 0xA9D, 0xAA7, 0xAAB,  
 0xAD5, 0xB4B, 0xB59, 0xBC9, 0xC1F, 0xC57, 0xC73,  
 0xC97, 0xC9B, 0xCD3, 0xCD5, 0xD1D, 0xD27, 0xD93,  
 0xE27, 0xE2B, 0xE4B, 0xEA3, 0xF0B, 0x9F7, 0xB6F,  
 0xBBD, 0xBDD, 0xCBF, 0xCF7, 0xD6F, 0xDBD, 0xDD7,  
 0xDE7, 0xDF5, 0xE5F, 0xE7D, 0xEFC, 0xF37, 0xF5D,  
 0xF6D, 0xF97, 0xFA7, 0xFB5, 0xFFB, 0x1201, 0x120D,  
 0x1225, 0x1245, 0x1431, 0x1449, 0x1849, 0x1883, 0x1C05,  
 0x1D01, 0x1077, 0x111F, 0x11B3, 0x12B9, 0x131D, 0x13A3,  
 0x1467, 0x14E5, 0x154D, 0x1655, 0x18B9, 0x18CB, 0x18F1,  
 0x1999, 0x19C9, 0x1A2B, 0x1A63, 0x1B0D, 0x1B45, 0x1C27,  
 0x1C4D, 0x1CA5, 0x1D43, 0x1D83, 0x1F81, 0x13D7, 0x13DD,  
 0x166F, 0x1775, 0x18FB, 0x196F, 0x19E7, 0x1A5F, 0x1B2F,  
 0x1BD3, 0x1CEB, 0x1D3D, 0x1EB9, 0x1F39, 0x20C9, 0x2451,  
 0x2941, 0x209F, 0x2197, 0x21E3, 0x2563, 0x2571, 0x283D,  
 0x2993, 0x2C4D, 0x2CC5, 0x2D45, 0x2E91, 0x2F05, 0x31C9,  
 0x350D, 0x3705, 0x384B, 0x3A07, 0x3A51, 0x3C83, 0x25D7,