On the Construction of Low-latency 32×32 Binary MDS Matrices from GHadamard Matrices

Meltem Kurt Pehlivanoğlu[1], Fatma Büyüksaraçoğlu Sakallı[2], Muharrem Tolga Sakallı[3]

[1]Computer Engineering Department, Kocaeli University, Kocaeli, Turkey
[2,3]Computer Engineering Department, Trakya University, Edirne, Turkey
Corresponding Author: meltem.kurt@kocaeli.edu.tr

Research Paper Received: 01.10.2021 Revised: 11.11.2021 Accepted: 12.11.2021

Abstract—In this paper, we generate new hardware efficient involutory 32×32 binary Maximum Distance Separable (MDS) diffusion layers with branch number 5. In our construction method, the idea used in Generalised Hadamard (GHadamard) matrix form is applied when generating these diffusion layers. We construct lightweight circuits by applying Boyar's global optimization heuristic (BP) to these diffusion layers. Hence, new 32×32 binary involutory MDS matrices with the best-known implementation cost (78 XORs) and depth 4 are generated. The obtained result is the same with the previous result given in [1], and we show that the diffusion layer given in [1] can also be obtained directly by using our construction method. As a result, we give thirteen more new involutory 32×32 binary MDS matrices with the best-known result.

Keywords—GHadamard matrix form, 32×32 binary MDS matrix, Global optimization, Involutory matrix

1. Introduction

Diffusion layers are one of the important structures used in the design of block ciphers. These structures satisfy the principle of diffusion proposed by Shannon [2]. The metric branch number [3] shows the power of a diffusion layer. In this respect, Maximum Distance Separable (MDS) matrices providing the maximum diffusion are used as the main component of a diffusion layer in many ciphers [4]–[8] and have the maximum branch number (an $n \times n$ MDS matrix over $\mathbb{F}_{2^m}$ has branch number $n + 1$). Although MDS matrices provide the best diffusion, they can be compared with other diffusion layers in view of implementation cost. Therefore, the construction of MDS matrices, especially low cost-involutory ones (the same matrix can be used in the process of encryption and decryption), is a challenging open problem in the literature [1]. The implementation cost of MDS matrices is evaluated by two important metrics; XOR (Exclusive-OR) count [3] and circuit depth [3]. On the other hand, there are two important optimization methods for the design of low-cost involutory MDS matrices; local optimization and global optimization. Local optimization focuses on the elements of the matrix and selects matrix elements with a minimum XOR count whereas the main focus in global optimization is the entire diffusion matrix. The aim of the global optimization is to find the shortest linear Straight Line Program (SLP) which implements the circuit of linear functions of a diffusion matrix. In other words, we search the diffusion layer circuits with the minimum depth and the minimum XOR count by using SLP.

There are two different global optimization techniques: cancellation-free technique and heuristic methods. When calculating the shortest SLP program in the cancellation-free technique, it is not allowed to cancel the same elements in the circuit function by adding (XORing) them (e.g. if $a = x_1 \oplus x_3, b = x_3 \oplus x_4, a \oplus b$ is not allowed because the element x_3 is located in both a and b). Heuristic methods find the optimum circuit by allowing cancellations.

PAAR1 and PAAR2 optimization algorithms [9] are cancellation-free methods. These methods give quick results, especially when dealing with large-sized matrices, but they do not guarantee the optimal solution. In 2010, Boyar and Peralta proposed a heuristic optimization method consisting of two steps [10]: nonlinear components (AND gate) are optimized in the first step while linear components (XOR gate) are optimized in the second step. In 2013, Boyar and et. al. proposed a new heuristic [11]. It consists of two steps but in this method, nonlinear components are optimized with an ad-hoc algorithm. The heuristic optimization method given in [12] is similar to the ones given in [10], [11]. But, the method given in [12], as distinct from these heuristics, includes the See-Saw method which is used to obtain lower-depth circuits. The aim of the heuristic method presented in [13] is to produce optimum circuits with respect to the given depth limit value. In order to achieve this purpose, it repeats the See-Saw method steps given in [12].

RNBP (Random-Boyar Peralta), A1, and A2 heuristics are presented in [14]. RNBP heuristic is obtained by adding randomness to the method given in [10]. A1 and A2 algorithms are proposed for low-depth and low XOR circuit optimizations, especially in low-dimensional (e.g. 16×16) matrices. In addition to local and global optimization techniques for circuit optimization, synthesizers such as Yosys [15], ABC [16] and formative tools (e.g. the satisfiability (SAT) [17] and LIGHTER [18] tools) are also used.

The development of global optimization algorithms, especially for 32×32 binary MDS matrices, become a major field of study [19]. In [19], the authors constructed matrices with depth 6 and 67 XORs, addressing 32×32 binary MDS matrices with branch number 5. In [14], the authors proposed 32×32 binary MDS matrices implemented with just depth 5 and 67 XORs, improving the result for depth 6 and 67 XORs given in [19]. However, the matrices given in these studies are not involutory MDS matrices. In [1], Li and et al. proposed a generalized construction method by using Companion matrices to generate 32×32 binary involutory MDS matrices and they presented a new 32×32 binary involutory MDS matrix with depth 4 and 78 XORs, which is the best-known XOR count for depth 4.

In this study, we focus on the construction of involutory 32×32 binary MDS matrices with minimum XOR count, especially for depth 4. Basically, we use the idea of applying the subfield construction method to the involutory MDS matrix given in [1]. The 4×4 GHadamard [20] and involutory MDS matrices over $\mathbb{F}_{2^m}$ are used as generator matrices. Then, we generalize these matrices in $M_4(GL(8, \mathbb{F}_2))$ form to find new lightweight involutory 32×32 binary MDS matrices. Moreover, we obtain lightweight circuits by using the global optimization technique given in [13], and refer to BP heuristic in this study. Finally, we identify some new involutory 32×32 binary MDS matrices with branch number 5, which can be implemented by 78 XORs and depth 4.

This paper is organized as follows, the preliminaries on GHadamard matrix form and MDS matrices are given in Section 2. In Section 3, we show how to construct lightweight involutory 32×32 binary MDS matrices by generalizing the GHadamard matrices as a matrix in $M_4(GL(8,\mathbb{F}_2))$ form and give the experimental results. The conclusion is given in Section 4.

2. Preliminaries

The finite field $\mathbb{F}_{2^m}$ consisting of 2^m elements is defined by an irreducible polynomial $p(x)$ of degree m over $\mathbb{F}_2$ and denoted by $\mathbb{F}_2[x]/p(x)$. In this paper, we use $\mathbb{F}_{2^m}/p(x)$ notation to express the finite field for simplicity. Note that $\mathbb{F}_{2^m}/p(x)$ notation expresses a finite field having 2^m elements. Also, we use hexadecimal notation to denote irreducible polynomial $p(x)$ when expressing $\mathbb{F}_{2^m}/p(x)$. For example, `0x13` stands for the irreducible polynomial $p(x) = x^4 + x + 1$.

Let C be an $[n,k,d]$ code and let $G = [I|A]$ be a generator matrix of C, where A is a $k \times (n-k)$ matrix. Then, A is an MDS matrix, if and only if every square sub-matrix of A is nonsingular. Moreover, if A also satisfies that $A = A^{-1}$, then A is called involutory MDS matrix.

$M_k(GL(n,\mathbb{F}_2))$ defines the set of all $k\times k$ matrices and all elements of these matrices are taken from the general linear group $GL(n,\mathbb{F}_2)$ [7]. The matrix A in $GL(n,\mathbb{F}_2)$ can be denoted as an $nk\times nk$ binary matrix which is the binary representation of A. We also use I_8 to indicate the 8×8 identity matrix.

Definition 1. *(Companion Matrix) [3] Let* $g(x) = a_0+a_1x+a_2x^2+\cdots+a_{k-1}x^{k-1}+x^k \in \mathbb{F}_q[x]$ *be a monic polynomial of degree* k*. The* $k\times k$ *Companion matrix* C_g *is defined as follows* (1)*:*

$$C_g = \begin{bmatrix} 0 & 0 & \dots & 0 & -a_0 \\ 1 & 0 & \dots & 0 & -a_1 \\ 0 & 1 & \dots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -a_{k-1} \end{bmatrix} \tag{1}$$

A recursive MDS matrix is an MDS matrix and can be expressed as a power of a companion matrix. For example, let A^k be a recursive MDS matrix, then A^k is obtained by recursively implementation (k times) of the companion matrix A.

The construction of MDS matrices, especially involutory ones, is a challenging issue. The construction methods of MDS matrices can be grouped into three groups: direct construction methods, search-based methods, and hybrid construction methods. In direct construction methods, MDS matrices are constructed directly (e.g. companion matrices, Vandermonde matrices, Cauchy matrices, and a new matrix form given in [21] to generate all 3×3 involutory and MDS matrices). Search-based methods simply use special matrix forms (such as circulant matrices, Toeplitz matrices, and Hadamard matrices) to search for MDS matrices. However, the cost of checking a matrix is an MDS matrix or not would be expensive, especially for larger dimensions. In the hybrid construction methods, the previous two construction techniques, direct construction, and search-based construction are merged (e.g. GHadamard matrix). In addition to the previous construction methods, a new method based on ground field structure was proposed to generate isomorphic (involutory) MDS matrices in [22].

Definition 2. *(GHadamard Matrix) [20] Let* $H = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ a_1 & a_1 & a_3 & a_2 \\ a_2 & a_3 & a_0 & a_1 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix}$ *be a* 4×4 *Hadamard ma-*

trix. The 4×4 *GHadamard matrix form GH* $=$ *Ghad*$(a_0, a_1; b_1, a_2; b_2, a_3; b_3)$ *with additional parameters* b_1, b_2 *and* $b_3 \in \mathbb{F}_{2^m} - \{0\}$ *can be given as follows* (2)*:*

$$GH = \begin{bmatrix} a_0 & a_1b_1 & a_2b_2 & a_3b_3 \\ a_1b_1^{-1} & a_0 & a_3b_1^{-1}b_2 & a_2b_1^{-1}b_3 \\ a_2b_2^{-1} & a_3b_2^{-1}b_1 & a_0 & a_1b_2^{-1}b_3 \\ a_3b_3^{-1} & a_2b_3^{-1}b_1 & a_1b_3^{-1}b_2 & a_0 \end{bmatrix} \quad (2)$$

GHadamard matrix form starts with an (involutory) Hadamard MDS matrix over $\mathbb{F}_{2^m}$, and then constructs new (involutory) MDS matrices directly. By using GHadamard matrix form, one can generate $(2^m - 1)^{k-1}$ $k \times k$ (involutory) MDS matrices over $\mathbb{F}_{2^m}$ in total from starting with a $k \times k$ (involutory) Hadamard MDS matrix over $\mathbb{F}_{2^m}$ (see the proof in [20]).

3. Our New Construction

In this paper, we first generate 4×4 involutory and MDS matrices in GHadamard matrix form over $\mathbb{F}_{2^m}/\texttt{0x13}$ and use these matrices as generator matrices. We utilize the companion matrix $A_1 \in GL(8, \mathbb{F}_2)$ given in (3) with characteristic polynomial $x^8 + x^2 + 1$ $(= (x^4 + x + 1)^2)$. Then, we generalize the generator matrices in the GHadamard matrix form by using the companion matrix A_1 and construct matrices in $M_4(GL(8, \mathbb{F}_2))$ with 32×32 binary matrix representations. Finally, lightweight 32×32 binary MDS matrices with branch number 5 are obtained. One of these lightweight matrices is given in (5).

$$A_1 = \begin{bmatrix} 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix} \quad (3)$$

Example 1. *Let the finite field* $\mathbb{F}_{2^4}$ *be generated by the primitive element* α*, which is a root of the primitive polynomial* `0x13` $(x^4 + x + 1)$*. Consider the involutory* 4×4 *Hadamard MDS matrix* $H_1 = had(1, \alpha, \alpha^2, \alpha^5) = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^5 \\ \alpha & 1 & \alpha^5 & \alpha^2 \\ \alpha^2 & \alpha^5 & 1 & \alpha \\ \alpha^5 & \alpha^2 & \alpha & 1 \end{bmatrix}$ *over* $\mathbb{F}_{2^4}/\texttt{0x13}$*. Then, the* 4×4 *GHadamard matrix* $GH_1 =$ *Ghad(1,* $\alpha; 1, \alpha^2; \alpha^{14}, \alpha^5; \alpha^{14})$ *corresponding to* H_1 *with the parameters* $b_1 = 1, b_2 = \alpha^{14}$ *and* $b_3 = \alpha^{14}$ *can be obtained as shown in* (4)*:*

$$GH_1 = \begin{bmatrix} 1 & \alpha & \alpha & \alpha^4 \\ \alpha & 1 & \alpha^4 & \alpha \\ \alpha^3 & \alpha^6 & 1 & \alpha \\ \alpha^6 & \alpha^3 & \alpha & 1 \end{bmatrix} \quad (4)$$

which is also an involutory MDS matrix with naive XOR count 80 $(= 32 + 4 \times 3 \times 4)$*. Then, we generalize the generator matrix* GH_1 *by using the matrix* A_1 *and it can be regarded as a matrix in* $M_4(GL(8, \mathbb{F}_2))$ *in the following matrix form* (5)*:*

$$GenGH_1 = \begin{bmatrix} I_8 & A_1 & A_1 & {A_1}^4 \\ A_1 & I_8 & {A_1}^4 & A_1 \\ {A_1}^3 & {A_1}^6 & I_8 & A_1 \\ {A_1}^6 & {A_1}^3 & A_1 & I_8 \end{bmatrix} \quad (5)$$

where I_8 *is the* 8×8 *identity matrix.* $GenGH_1$ *can be transformed into* 32×32 *involutory binary MDS* *matrix of branch number 5 with naive XOR count 128 XORs, which is given in* (6).

$$= \begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0\\
0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&0&0&0&1&0&1&0\\
0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&1\\
0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0\\
0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1\\
0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1\\
1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&1&0&1&0&0&1&0&0&0&0&0&1\\
0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&1&0&0&1&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&1&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&1&0&0&0&0&0&1&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&1&0&0&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&1&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&1&0&0&1&0&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&1\\
1&0&0&0&0&0&1&0&0&0&0&1&0&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0\\
0&1&0&0&0&0&0&1&0&0&0&0&1&0&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0\\
0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&1&0&1&0&0&0&0&0&0&0&0&1&0&1&0&1&0&0&0&0&0&1&0&0&1&0&0&0&0&0\\
0&0&0&1&0&1&0&0&1&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&1&0&1&0&0&1&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&1&0&1&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0\\
1&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0\\
0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1
\end{bmatrix} \tag{6}$$

Moreover, we use BP heuristic for the global optimization of $GenGH_1$ and achieve the lowest bound with 78 XORs and circuit depth 4. The global optimization results of $GenGH_1$ with 78 XORs and depth 4 is given in Table 1 where x_i, y_i and t_j represent input bits, output bits and temporary variables, respectively, and the values are given in parentheses represent the related circuit depth.

Example 2. *Let the finite field* $\mathbb{F}_{2^4}$ *be generated by the primitive element* α*, which is a root of the primitive polynomial* `0x13`. *Consider the involutory* 4×4 *Hadamard MDS matrix* $H_2 = had(1, \alpha^2, \alpha, \alpha^5) = \begin{bmatrix} 1 & \alpha^2 & \alpha & \alpha^5 \\ \alpha^2 & 1 & \alpha^5 & \alpha \\ \alpha & \alpha^5 & 1 & \alpha^2 \\ \alpha^5 & \alpha & \alpha^2 & 1 \end{bmatrix}$ *over* $\mathbb{F}_{2^4}$/`0x13`. *Then, the* $4{\times}4$ *GHadamard matrix* GH_2 = *Ghad(1,* $\alpha^2;\alpha^{13},\alpha;\alpha^{14},\alpha^5;\alpha^{14})$ *corresponding to* H_2 *with the parameters* $b_1 = \alpha^{13}, b_2 = \alpha^{14}, b_3 = \alpha^{14}$

TABLE 1
The global optimization result of $GenGH_1$ with 78 XORs and depth 4

No	Operation	No	Operation	No	Operation
1	$t_1 = x_7 + x_{20}$ (1)	27	$t_{27} = t_3 + t_{14}(2)$	53	$t_{53} = t_1 + t_{40}(2)$
2	$t_2 = x_{10} + x_{31}(1)$	28	$t_{28} = t_{11} + t_{27}[y_{28}](3)$	54	$t_{54} = t_{38} + t_{53}[y_{20}](3)$
3	$t_3 = x_{15} + x_{28}(1)$	29	$t_{29} = t_4 + t_{26}(3)$	55	$t_{55} = t_2 + t_{52}(3)$
4	$t_4 = x_2 + x_{23}(1)$	30	$t_{30} = t_8 + t_{29}[y_{26}](4)$	56	$t_{56} = t_6 + t_{55}[y_{18}](4)$
5	$t_5 = x_3 + x_{14}(1)$	31	$t_{31} = t_{12} + t_{27}(3)$	57	$t_{57} = t_{37} + t_{53}(3)$
6	$t_6 = x_5 + x_{18}(1)$	32	$t_{32} = t_{29} + t_{31}[y_2](4)$	58	$t_{58} = t_{55} + t_{57}[y_{10}](4)$
7	$t_7 = x_6 + x_{11}(1)$	33	$t_{33} = t_{18} + t_{31}[y_{23}](4)$	59	$t_{59} = t_{44} + t_{57}[y_{31}](4)$
8	$t_8 = x_{13} + x_{26}(1)$	34	$t_{34} = t_{18} + t_{20}(3)$	60	$t_{60} = t_{44} + t_{46}(3)$
9	$t_9 = x_0 + x_{21}(1)$	35	$t_{35} = t_{29} + t_{34}[y_{13}](4)$	61	$t_{61} = t_{55} + t_{60}[y_5](4)$
10	$t_{10} = x_8 + x_{29}(1)$	36	$t_{36} = x_{21} + t_{34}[y_{21}](4)$	62	$t_{62} = x_{29} + t_{60}[y_{29}](4)$
11	$t_{11} = x_6 + x_{19}(1)$	37	$t_{37} = x_7 + x_{22}(1)$	63	$t_{63} = x_1 + x_{16}(1)$
12	$t_{12} = x_{15} + x_{30}(1)$	38	$t_{38} = x_{14} + x_{27}(1)$	64	$t_{64} = t_{10} + t_{63}[y_1](2)$
13	$t_{13} = t_{11} + t_{12}[y_{15}](2)$	39	$t_{39} = t_{37} + t_{38[y_7](2)}$	65	$t_{65} = x_9 + x_{24}(1)$
14	$t_{14} = x_4 + x_9(1)$	40	$t_{40} = x_1 + x_{12}(1)$	66	$t_{66} = t_9 + t_{65}[y_9](2)$
15	$t_{15} = x_{24} + t_7(2)$	41	$t_{41} = x_{16} + t_5(2)$	67	$t_{67} = x_{22} + t_5(2)$
16	$t_{16} = x_{17} + t_{15}[y_{17}](3)$	42	$t_{42} = x_{25} + t_{41}[y_{25}](3)$	68	$t_{68} = t_{10} + t_{67}[y_{22}](3)$
17	$t_{17} = x_4 + x_{15}(1)$	43	$t_{43} = x_7 + x_{12}(1)$	69	$t_{69} = x_{30} + t_7(2)$
18	$t_{18} = x_{23} + t_3(2)$	44	$t_{44} = x_{31} + t_1(2)$	70	$t_{70} = t_9 + t_{69}[y_{30}](3)$
19	$t_{19} = x_0 + t_{18}[y_0](3)$	45	$t_{45} = x_8 + t_{44[y_8](3)}$	71	$t_{71} = t_9 + t_{11}(2)$
20	$t_{20} = x_{13} + t_4(2)$	46	$t_{46} = x_5 + t_2(2)$	72	$t_{72} = t_{24} + t_{71}[y_{19}](4)$
21	$t_{21} = x_{24} + t_{20}[y_{24}](3)$	47	$t_{47} = x_{16} + t_{46}[y_{16}](3)$	73	$t_{73} = t_{10} + t_{38}(2)$
22	$\mathrm{t}_{22} = x_{21} + t_8(2)$	48	$t_{48} = x_{29} + t_6(2)$	74	$t_{74} = t_{50} + t_{73}[y_{27}](4)$
23	$\mathrm{t}_{23} = x_6 + t_{22}[y_6](3)$	49	$t_{49} = x_{14} + t_{48}[y_{14}](3)$	75	$t_{75} = t_{13} + t_{15}(3)$
24	$\mathrm{t}_{24} = x_{11} + t_{22}(3)$	50	$t_{50} = x_3 + t_{48}(3)$	76	$t_{76} = t_{17} + t_{75}[y_4](4)$
25	$\mathrm{t}_{25} = t_{20} + t_{24}[y_{11}](4)$	51	$t_{51} = t_{46} + t_{50}[y_3](4)$	77	$t_{77} = t_{39} + t_{41}(3)$
26	$\mathrm{t}_{26} = x_{17} + t_{17}(2)$	52	$t_{52} = x_{25} + t_{43}(2)$	78	$t_{78} = t_{43} + t_{77}[y_{12}](4)$

can be obtained as shown in (7)*:*

$$GH_2 = \begin{bmatrix} 1 & 1 & 1 & \alpha^4 \\ \alpha^4 & 1 & \alpha^6 & \alpha^2 \\ \alpha^2 & \alpha^4 & 1 & \alpha^2 \\ \alpha^6 & 1 & \alpha^2 & 1 \end{bmatrix} \qquad (7)$$

which is also an involutory MDS matrix with naive XOR count 80 $(= 32 + 4 \times 3 \times 4)$. *Then, we generalize the generator matrix GH_2 by using the matrix A_1 and it can be regarded as a matrix in $M_4(GL(8, \mathbb{F}_2))$ in the following matrix form* (8)*:*

$$GenGH_2 = \begin{bmatrix} I_8 & I_8 & I_8 & {A_1}^4 \\ {A_1}^4 & I_8 & {A_1}^6 & {A_1}^2 \\ {A_1}^2 & {A_1}^4 & I_8 & {A_1}^2 \\ {A_1}^6 & 1 & {A_1}^2 & I_8 \end{bmatrix} \qquad (8)$$

$GenGH_2$ is equal to the matrix given in [1] (see the matrix H with 78 XORs and depth 4). Therefore, this shows that our new construction method based on the GHadamard matrix form can generate the lightest matrices directly. Note that in [1], the authors searched through a range of matrices generated by 6 parameters, which increases the search cost.

TABLE 2
The new generator matrices in GHadamard matrix form

No	Parameters $a_0, a_1; b_1, a_2; b_2, a_3; b_3$
1	$Ghad\ (1, \alpha; \alpha, \alpha^2; 1, \alpha^5; \alpha^{14})$
2	$Ghad\ (1, \alpha; \alpha^{14}, \alpha^2; \alpha^{13}, \alpha^5; \alpha^{14})$
3	$Ghad\ (1, \alpha; 1, \alpha^5; \alpha^{14}, \alpha^2; \alpha^{14})$
4	$Ghad\ (1, \alpha; \alpha, \alpha^5; \alpha^{14}, \alpha^2; 1)$
5	$Ghad\ (1, \alpha; \alpha^{14}, \alpha^5; \alpha^{14}, \alpha^2; \alpha^{13})$
6	$Ghad\ (1, \alpha^2; \alpha^2, \alpha; \alpha, \alpha^5; \alpha)$
7	$Ghad\ (1, \alpha^2; \alpha^2, \alpha^5; \alpha, \alpha; \alpha)$
8	$Ghad\ (1, \alpha^2; \alpha^{13}, \alpha^5; \alpha^{14}, \alpha; \alpha^{14})$
9	$Ghad\ (1, \alpha^5; \alpha^{14}, \alpha; \alpha^{14}, \alpha^2; \alpha^{13})$
10	$Ghad\ (1, \alpha^5; \alpha, \alpha^2; \alpha^2, \alpha; \alpha)$
11	$Ghad\ (1, \alpha^5; \alpha^{14}, \alpha^2; \alpha^{13}, \alpha; \alpha^{14})$
12	$Ghad\ (\alpha, 1; 1, \alpha^5; \alpha^{14}, \alpha^2; \alpha^{14})$
13	$Ghad\ (\alpha, 1; 1, \alpha^2; \alpha^{14}, \alpha^5; \alpha^{14})$

3.1. Experimental Results

In this section, we present thirteen more involutory 32×32 binary MDS matrices of branch number 5 (implemented by 78 XORs and depth 4) generated by our construction method. In Table2, we list generator GHadamard matrices (with the parameters a_is and b_is) of these involutory 32×32 binary MDS matrices. After generalizing these generator matrices by using the matrix A_1, each of them can be regarded as a matrix in $M_4(GL(8, \mathbb{F}_2))$. Moreover, they can be implemented by 78 XORs and depth 4 after optimizing with BP heuristic. The optimization results of these matrices are given in [23].

In Table 3, we give a summary of results for involutory and non-involutory 32×32 binary MDS matrices given in the literature. These results show that our new matrices in $M_4(GL(8, \mathbb{F}_2))$ form have the best-known results for the circuit depth 4. Note that the matrix H given in [7] can be generated by our new construction method (recall the matrix $GenGH_2$). We also give thirteen more matrices,

TABLE 3
The summary of results for involutory and non-involutory 32×32 binary MDS matrices given in the literature

Matrix	inv. / non-inv.	XOR Count	Depth
AES [14]	non-inv.	94	6
$M_{4,5}^{9,3}, (A_8, -, -)$ [14]	non-inv.	67	5
$M_{4,4}^{8,4''}, (A_8, -, -)$ [14]	non-inv.	69	4
M_{KLSW} [24]	inv.	84	4
G [1]	inv.	80	4
H [1]	inv.	78	4
$GenGH_1$ (Eq.(6))	inv.	78	4

which can be implemented by 78 XORs and depth 4. Hence, this shows the efficiency of our new construction method.

4. Conclusion

In this paper, we focus on direct construction of lightweight involutory 32×32 binary MDS matrices. Moreover, we consider a new construction method from GHadamard matrix form perspective and give new lightweight involutory 32×32 binary MDS matrices that can be implemented by only 78 XORs and depth 4. This result is the same with the previous result given in [1] and we show that this matrix can be obtained with our new construction method directly, whereas in [1] the authors searched through a range of matrices generated by 6 parameters. Therefore, our construction method actually minimizes the average search complexity. Additionally, we give thirteen more new involutory 32×32 binary MDS matrices with the best-known result. Hence, the obtained matrices can be used in the design of lightweight and low-latency diffusion layers in symmetric-key encryption.

It is still an open problem to design new global SLP heuristics, especially for larger dimensions.

As future work, it would be interesting to focus on designing a new global optimization algorithm, especially for 32×32 matrices.

Acknowledgments

Meltem Kurt Pehlivanoğlu is partially supported by the Scientific Research Project Department of Kocaeli University under the Project No: FHD-2020-2272.

References

[1] S. Li, S. Sun, C. Li, Z. Wei and L. Hu, *Constructing Low-latency Involutory MDS matrices with Lightweight Circuits*, IACR Transactions on Symmetric Cryptology, vol. 1, pp. 84–117, 2019.

[2] C.E. Shannon, *Communication theory of secrecy systems*, Bell Syst. Tech. J., vol. 28, pp. 656-715, 1949.

[3] M.K. Pehlivanoğlu and E.B. Kavun, *On the Design of Maximum Distance Separable Diffusion Layers of Cryptographic Block Ciphers*, in CyberSecurity and Defense, Ankara: Nobel Academic Publishing Education Consultancy, pp. 295-325, 2020.

[4] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-The Advanced Encryption Standard*, 1st ed., Springer-Verlag Berlin Heidelber, pp. 1-7, 2002.

[5] J. Guo, T. Peyrin and A. Poschmann, *The PHOTON Family of Lightweight Hash Functions*, in Advances in Cryptology – CRYPTO 2011, vol. 6841, pp. 222-239, 2011.

[6] J. Guo, T. Peyrin, A. Poschmann and M. Robshaw, *The LED Block Cipher*, in Cryptographic Hardware and Embedded Systems – CHES 2011, vol. 6917, pp. 326-341, 2011.

[7] P.S.L.M. Barreto and V. Rijmen, *The Khazad Legacy-Level Block Cipher*, First Open NESSIE Workshop 2000, Leuven, Belgium, 2000.

[8] K. Shibutani, T. Isobe, H. Hiwatari and et al., *Piccolo: An Ultra-Lightweight Blockcipher*, in Cryptographic Hardware and Embedded Systems – CHES 2011, vol. 6917, pp. 342-357, 2011.

[9] C. Paar, *Optimized Arithmetic for Reed-Solomon Encoders*, 1997 IEEE International Symposium on Information Theory, pp. 250, 1997.

[10] J. Boyar and R. Peralta, *A New Combinational Logic Minimization Technique with Applications to Cryptology*, SEA 2010, LNCS, vol. 6049, pp. 178-189, 2010.

[11] J. Boyar, P. Matthews and R. Peralta, *Logic Minimization Techniques with Applications to Cryptology*, Journal of Cryptology, vol. 26, pp. 280–312, 2013.

[12] J. Boyar, M. G. Find and R. Peralta, *Low-Depth, Low-Size Circuits for Cryptographic Applications*, BFA 2017. 2017.

[13] J. Boyar, M. G. Find and R. Peralta, *Small Low-depth Circuits for Cryptographic Applications*, Cryptography and Communications, vol. 11, no. 1, pp. 109–127, 2019.

[14] Q.Q. Tan and T. Peyrin, *Improved Heuristics for Short Linear Programs*, Cryptology ePrint Archive, Report 2019/847, 2019.

[15] C. Wolf, *Yosys Open Synthesis Suite*, http://www.clifford.at/yosys/, Accessed: November 10, 2021.

[16] R.K. Brayton and A. Mishchenko, *ABC: An Academic Industrial Strength Verification Tool*, CAV 2010, vol. 6174, pp. 24–40, 2010.

[17] K. Stoffelen, *Optimizing S-Box Implementations for Several Criteria Using SAT Solvers*, FSE 2016, LNCS, vol. 9783, pp. 140–160, 2016.

[18] J. Jean, T. Peyrin, S.M. Sim and J. Tourteaux, *Optimizing Implementations of Lightweight Building Blocks*, IACR Trans. Symmetric Cryptol., vol. 2017, no. 4, pp. 130–168, 2017.

[19] S. Duval and G. Leurent, *MDS Matrices with Lightweight Circuits*. IACR Transactions on Symmetric Cryptology, vol. 2018(2), pp. 48-78, 2018.

[20] M.K. Pehlivanoglu, M.T. Sakallı, S. Akleylek, N. Duru and V. Rijmen, *Generalisation of Hadamard Matrix to Generate Involutory MDS Matrices for Lightweight Cryptography*, IET Information Security, vol. 12, pp. 348–355, 2018.

[21] G.G. Guzel, M.T. Sakallı, S. Akleylek, V. Rijmen and Y. Çengellenmis¸ *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.

[22] M.T. Sakallı, S. Akleylek, K. Akkanat and V. Rijmen, *On the automorphisms and isomorphisms of MDS matrices and their efficient implementations*, Turkish Journal of Electrical Computer Sciences, vol.28, no. 1, pp. 275-287, 2020.

[23] M.K. Pehlivanoğlu, https://github.com/mkurtpehlivanoglu/32x32_binarymatrices.git, Accessed: November 11, 2021.

[24] T. Kranz, G. Leander, K. Stoffelen and F. Wiemer, *Shorterlinear straight-line programs for MDS matrices*, IACR Trans. Symmetric Cryptol., vol. 2017(4), pp. 188–211, 2017.