IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2019, No. 1, pp. 84–117. DOI:10.13154/tosc.v2019.i1.84-117

Constructing Low-latency Involutory MDS Matrices with Lightweight Circuits

Shun Li[1,2,4], Siwei Sun[1,2,4]✉, Chaoyun Li[3], Zihao Wei[1,2,4] and Lei Hu[1,2,4]

[1] State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{lishun,sunsiwei,hulei,weizihao}@iie.ac.cn
[2] Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing 100093, China
[3] imec - Computer Security and Industrial Cryptography (COSIC) research group, Department of Electrical Engineering (ESAT), KU Leuven, Leuven, Belgium
chaoyun.li@esat.kuleuven.be
[4] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. MDS matrices are important building blocks providing diffusion functionality for the design of many symmetric-key primitives. In recent years, continuous efforts are made on the construction of MDS matrices with small area footprints in the context of lightweight cryptography. Just recently, Duval and Leurent (ToSC 2018/FSE 2019) reported some 32×32 binary MDS matrices with branch number 5, which can be implemented with only 67 `XOR` gates, whereas the previously known lightest ones of the same size cost 72 `XOR` gates.

In this article, we focus on the construction of lightweight *involutory* MDS matrices, which are even more desirable than ordinary MDS matrices, since the same circuit can be reused when the inverse is required. In particular, we identify some involutory MDS matrices which can be realized with only 78 `XOR` gates with depth 4, whereas the previously known lightest involutory MDS matrices cost 84 `XOR` gates with the same depth. Notably, the involutory MDS matrix we find is much smaller than the AES MixColumns operation, which requires 97 `XOR` gates with depth 8 when implemented as a block of combinatorial logic that can be computed in one clock cycle. However, with respect to latency, the AES MixColumns operation is superior to our 78-XOR involutory matrices, since the AES MixColumns can be implemented with depth 3 by using more `XOR` gates.

We prove that the depth of a 32×32 MDS matrix with branch number 5 (e.g., the AES MixColumns operation) is at least 3. Then, we enhance Boyar's SLP-heuristic algorithm with circuit depth awareness, such that the depth of its output circuit is limited. Along the way, we give a formula for computing the minimum achievable depth of a circuit implementing the summation of a set of signals with given depths, which is of independent interest. We apply the new SLP heuristic to a large set of lightweight involutory MDS matrices, and we identify a depth 3 involutory MDS matrix whose implementation costs 88 `XOR` gates, which is superior to the AES MixColumns operation with respect to both lightweightness and latency, and enjoys the extra involution property.

Keywords: Lightweight cryptography · MDS matrix · Involutory matrix · Low latency

1 Introduction

The development of pervasive computing and the demand for low-cost security have stimulated intensive researches on the design of lightweight symmetric-key cryptographic

Received: 2018-11-23, Accepted: 2019-01-23, Published: 2019-03-08

algorithms. This often boils down to the search for lightweight yet cryptographically strong diffusion and confusion components.

In practice, the diffusion components are typically realized with linear operations, whose functionality, loosely speaking, is to spread the internal dependencies as much as possible. The so-called Maximal Distance Separable (MDS) matrices are probably the most preferable diffusion building blocks. When using MDS matrices as the diffusion layers in iterative block ciphers, it is possible to achieve a desired number of differentially or linearly active non-linear elements with a relatively small number of rounds, and therefore leading to low-latency designs. Moreover, designs with MDS matrices typically enjoy simple and clear security proofs, such as the case of AES [DR02]. Actually, it is exactly the elegant security proof offered by AES that initiates the widely application of MDS matrix in the design of symmetric-key primitives.

However, it is not an easy task to find lightweight MDS matrices, and it may be too luxury to use an MDS matrix in a design targeting resource constrained devices. In such situations, the designers compromise by employing almost MDS matrices [BBI+15, Ava17], or linear operations that can be realized with several bitwise XORs [BJK+16], or even bit-level permutations which can be implemented with a proper wiring [BKL+07]. Such design strategy more often than not leads to a significant increase of the number of rounds, and complicates the security proof remarkably. Therefore, it is an important endeavor to construct lightweight MDS matrices. In particular, lightweight *involutory* MDS matrices would be more preferable, since the same circuit can be reused when the inverse is required. Actually, the idea of reusing involutory components in both encryption and decryption has already been applied in some designs [BR00, SPR+04, BCG+12].

1.1 Related work

If the chip area is the sole consideration, one promising approach proposed by Guo, Peyrin, and Poschmann to reduce the implementation footprint is to find a lightweight matrix A such that A^k is MDS [GPP11, GPPR11]. The implementation of A^k can be obtained by recursively "executing" the implementation of A k times. Then no matter how complex A^k is, the cost is determined by A completely. However, this approach comes at the expense of an increased number of clock cycles, which is not desirable in low-latency applications. Therefore, in this work, we focus on the lightweight constructions, where the full MDS matrix is implemented as a block of combinatorial logic circuit such that it can be computed in one clock cycle. We refer the reader to [GPP11, TTKS18, AF14, Ber13, GPV17, WWW12, CLM16] for more information on the recursive constructions.

The initial attempts to find lightweight MDS matrices where the full matrix is implemented mainly focus on the selection of matrix entries enjoying low hardware footprints [SKOP15, BKL16, LS16, LW16, LW17, SS16a, SS16b, SS17, JPST17, ZWS18, GLWL16]. This line of work makes a great step forward for our ability of constructing lightweight MDS matrices and can be categorized as local optimizations. In particular, with the knowledge of which kind of entries are better, one can construct MDS matrices from some special classes of matrices, such as circulant, Hadamard, or Toeplitz matrices [SKOP15, LS16, SS16b]. Some of these constructions lead to involutory MDS matrices. In particular, Sim et al. observed that involutory MDS matrices can be implemented with almost the same cost as non-involutory ones under some specific metric, the latter being usually non-lightweight when the inverse matrix is required [SKOP15]. Note that here the entries of a matrix are not restricted to finite field elements, and can be general linear transformations. Actually, the idea of using general linear transformations leads to notable improvement at the time [BKL16, LW16].

So far, we have a fairly deep understanding of the problem with respect to local optimizations. Hence recent work tend to deal with the problem at a more essential level,

viewing it as the well-known Shortest Linear straight-line Problem (SLP) and optimizing globally. Indeed, this approach results in more accurate estimations of the cost of hardware implementations. In [KLSW17], Kranz et al. shows that the AES MixColumns matrix can be implemented with only 97 $\mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2$ `XOR` gates with Boyar's tool [BMP13] based on SLP heuristic, while the previous best implementation costs 103 `XOR` gates [JPST17]. Just recently in ToSC 2018/FSE 2019, Duval and Leurent reported some 32×32 binary MDS matrices which can be implemented with only 67 `XOR` gates by searching through a set of circuits ordered by hardware cost and optimizing globally [DL18], whereas the previously known lightest ones of the same size cost 72 `XOR` gates [KLSW17].

1.2 Our Contribution

First, we slightly generalize the structure of the involutory MDS matrix $M_{\texttt{KLSW}}$ (costs 84 `XOR` gates) proposed by Kranz, Leander, Stoffelen, and Wiemer [KLSW17], and try to construct an involutory MDS matrix G of the generalized form with less 1's than $M_{\texttt{KLSW}}$ in its binary form based on some educated guesses. After applying the SLP heuristic [BMP13] to G, it turns out that G can be implemented with only 80 `XOR` gates.

Then we further generalize the structure of G to a family of 4×4 matrices whose entries are powers of a given 8×8 binary matrix A. We show that every involutory matrix in this family can be completely determined by 6 parameters taking integer values. We search through a restricted range of matrices generated by these 6 parameters, and identify some involutory MDS matrices which can be implemented with only 78 `XOR` gates, while the previous best result requires 84 `XOR` gates.

Finally, we prove that the depth of a 32×32 MDS matrix with branch number 5 (e.g., the AES MixColumns operation) is at least 3. Then we augment Boyar's SLP-heuristic algorithm [BMP13] with circuit depth awareness to limit the depths of its output circuits. Along the way, we give a formula for computing the minimum achievable depth of a circuit implementing the summation of a set of signals with given depths, which is of independent interest. By applying this tool, we search through a large set of lightweight involutory MDS matrices and identify one which can be implemented with 88 `XOR` gates, whose circuit depth reaches the lower bound 3. A summary of the optimal matrices we find is given in Table 1. We also try to synthesize the matrices from Table 1 with three different technology libraries (`NanGate` 45 `nm`, `SMIC` 65`nm` and `TSMC` 28`nm`). In all cases, our matrices exhibit lower area footprint. Taking the 97-XOR AES MDS matrix for example, it takes 154.811996 $\texttt{um}^2$ when synthesized with `NanGate` 45`nm` technology (194 `GE`), while our 88-XOR matrix takes 140.447996 $\texttt{um}^2$ (176 `GE`). Hence, our 88-XOR matrix enjoys three advantages over the AES MDS matrix: it is involutory; its depth is 3 (the depth of the 97-XOR AES MDS is 8; and its area footprint is lower. Moreover, we make all of our code and results (matrices in binary representations with their actual implementations) publicly available at

https://github.com/siweisun/involutory_mds

1.3 Organization

In Sect. 2, we give some preliminaries on finite fields and MDS matrices. Then metrics used in this work for measuring the circuit cost are given in Sect. 3. In Sect. 4 we show how to construct a lighter involutory matrix by generalizing a previously known involutory MDS matrix. In Sect. 5, we consider further generalizations and search through a large set of matrices to find lighter involutory MDS matrices. We prove a theorem on the lower bound of the circuit depth of an 32×32 MDS matrix with branch number 5, and enhance Boyar's SLP-heuristic algorithm to find lightweight involutory MDS matrices whose depths reach the lower bound. Section 7 concludes the paper.

Table 1: A summary of the results. All matrices shown in the table are 32×32 binary matrices, and $\mathbf{M}_k(\mathbb{R})$ is the set of all $k \times k$ matrices whose entries are drawn from $\mathbb{R}$. The SLP column is obtained by applying Boyar's SLP heuristic [BMP13], and SLP* means that the result is obtained by applying a modified version of Boyar's SLP heuristic with circuit depth awareness presented in Sect. 6.

Matrix	MDS	Involutory	SLP	Depth	Source
$M_{\mathsf{AES}} \in \mathbf{M}_4(\mathbb{F}_{2^8})$	✓	✗	97	8	[KLSW17]
$M_{\mathsf{AES}} \in \mathbf{M}_4(\mathbb{F}_{2^8})$	✓	✗	105 (SLP*)	3	Sect. 6
$M_{\mathtt{KLSW}} \in \mathbf{M}_4(\mathbf{M}_2(\mathbb{F}_{2^4}))$	✓	✓	84	4	[KLSW17]
$G \in \mathbf{M}_4(\mathbf{M}_8(\mathbb{F}_2))$	✓	✓	80	4	Sect. 4
$H \in \mathbf{M}_4(\mathbf{M}_8(\mathbb{F}_2))$	✓	✓	78	4	Sect. 5
$Q \in \mathbf{M}_4(\mathbf{M}_8(\mathbb{F}_2))$	✓	✓	88 (SLP*)	3	Sect. 6

2 Preliminaries

Let $\mathbb{R}$ be an arbitrary ring, and $\mathbf{M}_k(\mathbb{R})$ be the set of all $k \times k$ matrices whose entries are drawn from $\mathbb{R}$. Therefore, $\mathbf{M}_k(\mathbb{F}_{2^n})$ denotes the set of all $k \times k$ matrices over the finite field of 2^n elements, and $\mathbf{M}_k(\mathrm{GL}(n, \mathbb{F}_2))$ is the set of all $k \times k$ matrices whose elements are taken from the general linear group $\mathrm{GL}(n, \mathbb{F}_2)$ formed by all invertible $n \times n$ matrices over $\mathbb{F}_2$. Every matrix A in $\mathbf{M}_k(\mathbb{F}_{2^n})$ or $\mathbf{M}_k(\mathrm{GL}(n, \mathbb{F}_2))$ can be represented as an $nk \times nk$ binary matrix, which we call the binary representation of A. We use I_n and O_n to denote the $n \times n$ identity matrix and zero matrix over $\mathbb{F}_2$ respectively. We will omit the subscript n whenever it is obvious from the context.

Given a vector x in $\mathbb{F}_2^{nk}$, we denote by $\omega_n(x)$ the number of non-zero n-bit chunks in x. When $n = 1$, we simply write $\omega_1(x)$ as $\omega(x)$, which is the well known Hamming weight of x. The branch number $\mathcal{B}_n(A)$ of $A \in \mathbf{M}_{nk}(\mathbb{F}_2)$ is defined as $\min_{x \in \mathbb{F}_{2^{nk}} \setminus \{0\}} \{\omega_n(x) + \omega_n(Ax)\}$.

Definition 1. An invertible $nk \times nk$ binary matrix A is MDS over k n-bit words if and only if $\mathcal{B}_n(A) = k + 1$. Furthermore, if an MDS matrix A satisfies that $A = A^{-1}$, then we call it an involutary MDS matrix.

Definition 2 (Characteristic polynomial [Wan03])**.** The characteristic polynomial f of a binary matrix $A \in \mathbf{M}_m(\mathbb{F}_2)$ is defined as $f(x) = |xI + A| \in \mathbb{F}_2[x]$.

Lemma 1 ([DF04])**.** *If f is a characteristic polynomial of $A \in \mathbf{M}_m(\mathbb{F}_2)$, then $f(A) = 0$.*

Definition 3 ([Con14])**.** Let $A \in \mathbf{M}_m(\mathbb{F}_2)$, $f \in \mathbb{F}_2[x]$ is the minimal polynomial of A if and only if $f(A) = 0$, and for any $g \in \mathbb{F}_2[x]$ such that $g(A) = 0$, $deg(f) \leq deg(g)$.

Note that a minimal polynomial of $A \in \mathbf{M}_m(\mathbb{F}_2)$ can be reducible.

Definition 4 ([Wan03])**.** Let $f = x^m + a_{m-1}x^{m-1} + \cdots + a_1 x + a_0 \in \mathbb{F}_2[x]$. The companion matrix of f is defined as the $m \times m$ matrix

$$\begin{pmatrix} 0 & & & & a_0 \\ 1 & 0 & & & a_1 \\ & 1 & \ddots & & \vdots \\ & & \ddots & 0 & a_{m-2} \\ & & & 1 & a_{m-1} \end{pmatrix}.$$

It is trivial to verify that the characteristic polynomial of f's companion matrix is f.

Lemma 2 ([BR99, LW16])**.** *Let L be a matrix in $\mathbf{M}_k(\mathbf{M}_n(\mathbb{F}_2))$. Then L is an MDS matrix (with branch number $k + 1$) if and only if all square sub-matrices $G \in \mathbf{M}_t(\mathbf{M}_n(\mathbb{F}_2))$ of L are of full rank for $1 \leq t \leq k$.*

Lemma 2 is employed in this paper to check the MDS property of our candidate lightweight matrices.

3 Metrics

We estimate the hardware cost of a linear operation as the number of $\mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2$ `XOR` gates required in its implementation, where the implementation can be described as a sequence of XOR and assignment operations $x_i \leftarrow x_{a_i} \oplus x_{b_i}$ with $a_i, b_i < i$. But, for a given linear operation, it is NP-hard to obtain the minimum number of `XOR` gates required [BMP08, BMP13], and only metrics determining the *upper bounds* are available. The metrics used in this paper are listed in the following.

Direct XOR Count. Given a matrix $A \in \mathbf{M}_{nk}(\mathbb{F}_2)$, the Direct XOR Count $\mathsf{DXC}(A)$ of A is $\omega(A) - nk$, that is, the number of 1s in the matrix A minus nk. This corresponds to a naive implementation of A, where each row of A is implemented as is. $\mathsf{DXC}(A)$ is essentially the same as the Hamming weight $\omega(A)$ of A up to a constant shift.

Global Optimization. Given a matrix $A \in \mathbf{M}_{nk}(\mathbb{F}_2)$, we can obtain an estimation of its hardware cost by finding a good linear straight-line program corresponding to A with state-of-the-art automatic tools based on certain SLP heuristic [BMP13], and this metric is denoted as $\mathsf{SLP}(A)$. Note that this is so far the most accurate estimation that is practical for 32×32 binary matrices.

In this work, eventually the hardware cost is estimated with Global Optimization. However, before applying the Global Optimization, we first try to construct lighter involutory MDS matrices with fairly low Direct XOR Count (i.e., matrices with low Hamming weights). Finally, we would like to mention that there are other metrics (such as the Sequential XOR Count [JPST17]) in the literature, and we refer the reader to [DL18] for a clear discussion of the comparisons and limitations of different metrics.

Besides the circuit area (measured by the number of `XOR` gates required for an implementation), another important metric of an implementation is the latency, which imposes constraint on the clock frequency at which the circuit can operate. The latency of an implementation can be characterized by its depth.

Definition 5. Let M be an $m \times m$ binary Matrix. Then the function $f_M : x \in \mathbb{F}_2^m \mapsto Mx \in \mathbb{F}_2^m$ can be implemented with a finite number of `XOR` gates. The critical path of such an implementation is defined as the path between an input and output involving the maximum number of `XOR` gates, and the depth of the implementation is the number of `XOR` gates involved in the critical path.

4 Our Constructions

By applying the subfield construction [BNN+10, KPPY14] to the involutory MDS matrix

$$\begin{pmatrix} I_4 & C & C^2 & I_4 \\ C & I_4 & I_4 & C^2 \\ C^3 & C & I_4 & C \\ C & C^3 & C & I_4 \end{pmatrix} \text{ with } C = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

proposed by Sarkar et al. [SS16b], Kranz et al. obtain so far the most lightweight involutory MDS matrix in $\mathbf{M}_4(\mathbf{M}_2(\mathbb{F}_{2^4}))$, whose binary representation is

$$M_{\texttt{KLSW}} = \left(\begin{array}{cc:cc:cc:cc} I_4 & 0 & C & 0 & C^2 & 0 & I_4 & 0 \\ 0 & I_4 & 0 & C & 0 & C^2 & 0 & I_4 \\ \hdashline C & 0 & I_4 & 0 & I_4 & 0 & C^2 & 0 \\ 0 & C & 0 & I_4 & 0 & I_4 & 0 & C^2 \\ \hdashline C^3 & 0 & C & 0 & I_4 & 0 & C & 0 \\ 0 & C^3 & 0 & C & 0 & I_4 & 0 & C \\ \hdashline C & 0 & C^3 & 0 & C & 0 & I_4 & 0 \\ 0 & C & 0 & C^3 & 0 & C & 0 & I_4 \end{array}\right).$$

The involutory MDS matrix $M_{\texttt{KLSW}}$ can be regarded as a matrix in $\mathbf{M}_4(\mathrm{GL}(8,\mathbb{F}_2))$ of the following form

$$\begin{pmatrix} I_8 & A & A^2 & I_8 \\ A & I_8 & I_8 & A^2 \\ A^3 & A & I_8 & A \\ A & A^3 & A & I_8 \end{pmatrix}. \tag{1}$$

Then we can generalize (1) and try to find lightweight involutory MDS matrices of the following form

$$G = \begin{pmatrix} I_8 & A^l & A^i & I_8 \\ A^l & I_8 & I_8 & A^i \\ A^j & A^k & I_8 & A^l \\ A^k & A^j & A^l & I_8 \end{pmatrix}.$$

Observation 1. *The matrix $G \in \mathbf{M}_4(\mathrm{GL}(8,\mathbb{F}_2))$ is involutory if and only if $G^2 = I$ which implies $A^{2l} + A^{i+j} + A^k = O_8$ and $A^{i+k} + A^j = O_8$.*

According to Observation 1, to make G involutory, we have $A^{i+k} + A^j = O_8$ and thus

$$G = \begin{pmatrix} I_8 & A^l & A^i & I_8 \\ A^l & I_8 & I_8 & A^i \\ A^j & A^k & I_8 & A^l \\ A^k & A^j & A^l & I_8 \end{pmatrix} = \begin{pmatrix} I_8 & A^l & A^i & I_8 \\ A^l & I_8 & I_8 & A^i \\ A^{i+k} & A^k & I_8 & A^l \\ A^k & A^{i+k} & A^l & I_8 \end{pmatrix}.$$

First, our goal is to find an involutory matrix G, such that $\mathsf{DXC}(G)$ is small. Since $\mathsf{DXC}(G) = \omega(G) - 32 = 4\omega(A^l) + 2\omega(A^i) + 2\omega(A^k) + 2\omega(A^{i+k}) + 48 - 32$ and heuristically $\omega(A^t)$ increases along with $|t|$ when A is very sparse, we prefer instantiations of i, l, j and k, such that $|i|$, $|l|$, $|j|$ and $|k|$ (the exponents of A appearing in G) are small.

According to [BKL16] (see Table 7 of [BKL16]), $\mathsf{DXC}(A) \geq 2$ if the characteristic polynomial of A is an irreducible polynomial of degree 8. Therefore, we only consider A whose characteristic polynomial is reducible. We find that if we choose

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{2}$$

to be the companion matrix of x^8+x^2+1, whose characteristic polynomial is $(x^4+x+1)^2 = x^8 + x^2 + 1$, then $\mathsf{DXC}(A^{-4}) = 6$, $\mathsf{DXC}(A^{-3}) = 4$, $\mathsf{DXC}(A^{-2}) = 2$, $\mathsf{DXC}(A^{-1}) = 1$,

Table 2: An implementation of G with 80 XOR gates and depth 4, where $(x_0, \cdots, x_{31})$ are input signals, $(y_0, \cdots, y_{31})$ are output signals, and t_i's are intermediate signals.

No	Operation	Depth	No	Operation	Depth	No	Operation	Depth
1	$t_1 = x_0 + x_9$	1	28	$t_{28} = x_{31} + t_{16}$	2	55	$t_{55} = x_4 + t_{38}$	3
2	$t_2 = x_1 + x_8$	1	29	$t_{29} = x_7 + t_{28}$ $[y_7]$	3	56	$t_{56} = t_{40} + t_{55}$ $[y_4]$	4
3	$t_3 = x_2 + t_1$	2	30	$t_{30} = x_7 + x_{19}$	1	57	$t_{57} = x_5 + x_{29}$	1
4	$t_4 = x_{10} + t_2$	2	31	$t_{31} = x_7 + x_{26}$	1	58	$t_{58} = t_6 + t_{57}$ $[y_5]$	2
5	$t_5 = x_3 + x_{30}$	1	32	$t_{32} = x_8 + t_{30}$	2	59	$t_{59} = x_9 + t_{34}$	3
6	$t_6 = x_{11} + x_{22}$	1	33	$t_{33} = x_{29} + t_{32}$ $[y_{29}]$	3	60	$t_{60} = t_{36} + t_{59}$ $[y_9]$	4
7	$t_7 = x_0 + x_{27}$	1	34	$t_{34} = x_{14} + t_{31}$	2	61	$t_{61} = x_{10} + t_7$	2
8	$t_8 = x_6 + x_{18}$	1	35	$t_{35} = x_{20} + t_{34}$ $[y_{20}]$	3	62	$t_{62} = t_8 + t_{61}$ $[y_{10}]$	3
9	$t_9 = x_{15} + t_7$	2	36	$t_{36} = x_{24} + t_{22}$	2	63	$t_{63} = x_{11} + t_{32}$	3
10	$t_{10} = x_{21} + t_9$ $[y_{21}]$	3	37	$t_{37} = x_0 + t_{36}$ $[y_0]$	3	64	$t_{64} = t_{38} + t_{63}$ $[y_{11}]$	4
11	$t_{11} = x_{20} + t_1$	2	38	$t_{38} = x_{28} + t_2$	2	65	$t_{65} = x_{12} + t_{11}$	3
12	$t_{12} = x_{30} + t_{11}$ $[y_{30}]$	3	39	$t_{39} = x_{22} + t_{38}$ $[y_{22}]$	3	66	$t_{66} = t_{13} + t_{65}$ $[y_{12}]$	4
13	$t_{13} = x_{29} + t_3$	3	40	$t_{40} = x_{21} + t_4$	3	67	$t_{67} = x_{13} + x_{21}$	1
14	$t_{14} = x_{23} + t_{13}$ $[y_{23}]$	4	41	$t_{41} = x_{31} + t_{40}$ $[y_{31}]$	4	68	$t_{68} = t_5 + t_{67}$ $[y_{13}]$	2
15	$t_{15} = x_4 + x_{22}$	1	42	$t_{42} = x_{12} + x_{23}$	1	69	$t_{69} = x_{17} + t_{17}$	3
16	$t_{16} = x_{13} + x_{16}$	1	43	$t_{43} = x_{24} + t_{21}$	2	70	$t_{70} = t_{19} + t_{69}$ $[y_{17}]$	4
17	$t_{17} = x_{31} + t_{15}$	2	44	$t_{44} = x_{15} + t_{43}$ $[y_{15}]$	3	71	$t_{71} = x_{18} + t_{43}$	3
18	$t_{18} = x_{14} + t_{17}$ $[y_{14}]$	3	45	$t_{45} = x_{30} + t_{42}$	2	72	$t_{72} = t_{45} + t_{71}$ $[y_{18}]$	4
19	$t_{19} = t_3 + t_6$	3	46	$t_{46} = x_6 + t_{45}$ $[y_6]$	3	73	$t_{73} = x_{19} + t_{26}$	3
20	$t_{20} = x_{24} + t_{19}$ $[y_{24}]$	4	47	$t_{47} = t_4 + t_5$	3	74	$t_{74} = t_{28} + t_{73}$ $[y_{19}]$	4
21	$t_{21} = x_5 + x_{23}$	1	48	$t_{48} = x_{16} + t_{47}$ $[y_{16}]$	4	75	$t_{75} = x_{25} + t_{45}$	3
22	$t_{22} = x_{14} + x_{17}$	1	49	$t_{49} = x_1 + t_{24}$	3	76	$t_{76} = t_{47} + t_{75}$ $[y_{25}]$	4
23	$t_{23} = x_6 + x_{25}$	1	50	$t_{50} = t_{26} + t_{49}$ $[y_1]$	4	77	$t_{77} = x_{26} + t_{15}$	2
24	$t_{24} = x_{15} + t_8$	2	51	$t_{51} = x_2 + t_{32}$	3	78	$t_{78} = t_{16} + t_{77}$ $[y_{26}]$	3
25	$t_{25} = x_{28} + t_{24}$ $[y_{28}]$	3	52	$t_{52} = t_{34} + t_{51}$ $[y_2]$	4	79	$t_{79} = x_{27} + t_{21}$	2
26	$t_{26} = x_{16} + t_{23}$	2	53	$t_{53} = x_3 + t_9$	3	80	$t_{80} = t_{22} + t_{79}$ $[y_{27}]$	3
27	$t_{27} = x_8 + t_{26}$ $[y_8]$	3	54	$t_{54} = t_{11} + t_{53}$ $[y_3]$	4			

$\mathsf{DXC}(A^0) = 0$, $\mathsf{DXC}(A) = 1$, $\mathsf{DXC}(A^2) = 2$, $\mathsf{DXC}(A^3) = 3$, $\mathsf{DXC}(A^4) = 4$, and $A^8 + A^2 + I = 0$ according to Lemma 1.

It is easy to verify that the minimal polynomial of A is also $x^8 + x^2 + 1$ according to Definition 3. Hence $A^8 + A^2 + I = 0$ and thus $A^{8+d} + A^{2+d} + A^d = 0$ for any integer d. Therefore, solving the equation over two sets $\{A^{8+d}, A^{2+d}, A^d\} = \{A^{2l}, A^{2i+k}, A^k\}$, where $A^{2i+k} = A^{i+j}$ according to Observation 1, gives the solutions of l, i, and k such that $A^{2l} + A^{i+j} + A^k = O_8$. We can enumerate all solutions and pick one which minimizes $4|l| + 2|i| + 2|k| + 2|i + j|$. One such possible solution[1] is

$$\begin{cases} d &= -4 \\ l &= 2 \\ k &= -2 \\ i &= -1 \end{cases}$$

which transforms G into

$$G = \begin{pmatrix} I_8 & A^2 & A^{-1} & I_8 \\ A^2 & I_8 & I_8 & A^{-1} \\ A^{-3} & A^{-2} & I_8 & A^2 \\ A^{-2} & A^{-3} & A^2 & I_8 \end{pmatrix}.$$

By applying Boyar's SLP-heuristic algorithm, we obtain an implementation of G with only 80 XOR gates, which breaks the record of 84 XOR gates [KLSW17], and the actual implementation can be found in Table 2

5 More Generalizations

The above result motivates us to consider a more generalized form:

$$M = \begin{pmatrix} A^{\epsilon_{11}} & A^{\epsilon_{12}} & A^{\epsilon_{13}} & A^{\epsilon_{14}} \\ A^{\epsilon_{21}} & A^{\epsilon_{22}} & A^{\epsilon_{23}} & A^{\epsilon_{24}} \\ A^{\epsilon_{31}} & A^{\epsilon_{32}} & A^{\epsilon_{33}} & A^{\epsilon_{34}} \\ A^{\epsilon_{41}} & A^{\epsilon_{42}} & A^{\epsilon_{43}} & A^{\epsilon_{44}} \end{pmatrix} = \begin{pmatrix} I & A^{\epsilon_{12}} & A^{\epsilon_{13}} & A^{\epsilon_{14}} \\ A^{\epsilon_{21}} & I & A^{\epsilon_{23}} & A^{\epsilon_{24}} \\ A^{\epsilon_{31}} & A^{\epsilon_{32}} & I & A^{\epsilon_{34}} \\ A^{\epsilon_{41}} & A^{\epsilon_{42}} & A^{\epsilon_{43}} & I \end{pmatrix}.$$

[1] There are other possible solutions. However, we do not discuss them since all of them will be covered in sebsequent sections.

where $\epsilon_{11} = \epsilon_{22} = \cdots = \epsilon_{44} = 0$, $A \in \mathrm{GL}(8, \mathbb{F}_2)$ is the companion matrix of $x^8 + x^2 + 1$ shown in Equation (2), and ϵ_{ij} are integers for $1 \le i, j \le 4$. Without loss of generality, let

$$\begin{cases} A^{\epsilon_{42}} = A^{r+\epsilon_{13}} \\ A^{\epsilon_{43}} = A^{s+\epsilon_{12}} \\ A^{\epsilon_{24}} = A^{t+\epsilon_{13}} \end{cases}.$$

Since M is involutory and thus $A^2 = I$, we can deduce that

$$M = \begin{pmatrix} I & A^{\epsilon_{12}} & A^{\epsilon_{13}} & A^{\epsilon_{14}} \\ A^{\epsilon_{12}+s+t} & I & A^{\epsilon_{14}+s} & A^{\epsilon_{13}+t} \\ A^{\epsilon_{13}+r+t} & A^{\epsilon_{14}+r} & I & A^{\epsilon_{12}+t} \\ A^{\epsilon_{14}+r+s} & A^{\epsilon_{13}+r} & A^{\epsilon_{12}+s} & I \end{pmatrix} \tag{3}$$

and

$$(I, A^{\epsilon_{12}}, A^{\epsilon_{13}}, A^{\epsilon_{14}}) \begin{pmatrix} A^{\epsilon_{11}} \\ A^{\epsilon_{12}+s+t} \\ A^{\epsilon_{13}+r+t} \\ A^{\epsilon_{14}+r+s} \end{pmatrix} = I,$$

which implies

$$A^{2\epsilon_{12}-r} + A^{2\epsilon_{13}-s} + A^{2\epsilon_{14}-t} = 0. \tag{4}$$

According to Equation (3), the matrix M can be completely determined by the parameters $\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s$ and t. Therefore, we inspect all $(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t) \in \mathbb{Z}^6$ satisfying the following conditions[2]

$$\begin{cases} -8 \le \epsilon_{1j} \le 8 \ \text{ for } \ 1 \le j \le 4 \\ 0 \le r \le s \le t \le 8 \\ A^{2\epsilon_{12}-r} + A^{2\epsilon_{13}-s} + A^{2\epsilon_{14}-t} = 0 \end{cases}. \tag{5}$$

Finally, we identify 5550 involutory MDS matrices whose Hamming weights are within the range from 148 to 172. We apply Boyar's SLP-heuristic algorithm to all these matrices to obtain their lightweight implementations and the results are summarized in Table 3.

The above approach produces many equivalent matrices. For instance, let

$$M = \begin{pmatrix} I & A^{\epsilon_{12}} & A^{\epsilon_{13}} & A^{\epsilon_{14}} \\ A^{\epsilon_{12}+s+t} & I & A^{\epsilon_{14}+s} & A^{\epsilon_{13}+t} \\ A^{\epsilon_{13}+r+t} & A^{\epsilon_{14}+r} & I & A^{\epsilon_{12}+t} \\ A^{\epsilon_{14}+r+s} & A^{\epsilon_{13}+r} & A^{\epsilon_{12}+s} & I \end{pmatrix},$$

which is parameterized by $(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$. If we exchange the second row and third row, and then exchange the second and third column, we obtain

$$\tilde{M} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \end{pmatrix}^T M \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \end{pmatrix} = \begin{pmatrix} I & A^{\epsilon_{13}} & A^{\epsilon_{12}} & A^{\epsilon_{14}} \\ A^{\epsilon_{13}+r+t} & I & A^{\epsilon_{14}+r} & A^{\epsilon_{12}+t} \\ A^{\epsilon_{12}+s+t} & A^{\epsilon_{14}+s} & I & A^{\epsilon_{13}+t} \\ A^{\epsilon_{14}+r+s} & A^{\epsilon_{12}+s} & A^{\epsilon_{13}+r} & I \end{pmatrix},$$

corresponding to the parameter $(\epsilon_{13}, \epsilon_{12}, \epsilon_{14}, s, r, t)$. Obviously, $\tilde{M}$ is an involutory MDS matrix if and only if M is involutory and MDS. In addition, from any implementation of M, we can derive an implementation of $\tilde{M}$ with the same circuit size and depth. Hence,

Table 3: A summary of the result. The first row means that we identify a set of 18 matrices whose Hamming weight and DXC are 148 and 116 respectively. The maximal and minimal XOR gate counts of these matrices after applying Boyar's SLP heuristic are 80, and the minimum circuit depth is 4.

$\omega(A)$	#Matrices	DXC(A)	min SLP(A)	max SLP(A)	min depth(A)
148	18	116	80	80	4
149	48	117	80	80	4
150	72	118	80	83	4
151	48	119	83	84	4
152	60	120	83	87	4
153	72	121	80	84	4
154	84	122	80	86	4
155	24	123	86	87	5
156	72	124	86	87	4
157	96	125	82	84	5
158	156	126	80	90	4
159	0	–	–	–	–
160	210	128	78	90	4
161	144	129	79	84	4
162	204	130	79	89	4
163	192	131	79	91	5
164	300	132	78	93	4
165	312	133	79	88	5
166	324	134	80	93	4
167	336	135	80	94	5
168	600	136	78	99	4
169	384	137	79	97	4
170	504	138	80	98	4
171	528	139	81	99	4
172	762	140	79	102	4

the parameters $(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$, and $(\epsilon_{13}, \epsilon_{12}, \epsilon_{14}, s, r, t)$ are equivalent. We list all equivalent parameters in Table 4.

Every entry in the rightmost column of Table 4 is the cycle notation of a permutation π over $\{1, 2, 3, 4\}$. The parameter in the same row is obtained by permute the columns and rows of

$$M = \begin{pmatrix} I & A^{\epsilon_{12}} & A^{\epsilon_{13}} & A^{\epsilon_{14}} \\ A^{\epsilon_{12}+s+t} & I & A^{\epsilon_{14}+s} & A^{\epsilon_{13}+t} \\ A^{\epsilon_{13}+r+t} & A^{\epsilon_{14}+r} & I & A^{\epsilon_{12}+t} \\ A^{\epsilon_{14}+r+s} & A^{\epsilon_{13}+r} & A^{\epsilon_{12}+s} & I \end{pmatrix}$$

according to π. Taking the 4th row for example, we have $\pi = (2, 4, 3)$, and the transformation is performed as follows

$$\begin{pmatrix} I_8 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_8 \\ 0 & I_8 & 0 & 0 \\ 0 & 0 & I_8 & 0 \end{pmatrix}^T \begin{pmatrix} I & A^{\epsilon_{12}} & A^{\epsilon_{13}} & A^{\epsilon_{14}} \\ A^{\epsilon_{12}+s+t} & I & A^{\epsilon_{14}+s} & A^{\epsilon_{13}+t} \\ A^{\epsilon_{13}+r+t} & A^{\epsilon_{14}+r} & I & A^{\epsilon_{12}+t} \\ A^{\epsilon_{14}+r+s} & A^{\epsilon_{13}+r} & A^{\epsilon_{12}+s} & I \end{pmatrix} \begin{pmatrix} I_8 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_8 \\ 0 & I_8 & 0 & 0 \\ 0 & 0 & I_8 & 0 \end{pmatrix} = \begin{pmatrix} I & A^{\epsilon_{13}} & A^{\epsilon_{14}} & A^{\epsilon_{12}} \\ A^{\epsilon_{13}+r+t} & I & A^{\epsilon_{12}+t} & A^{\epsilon_{14}+r} \\ A^{\epsilon_{14}+r+s} & A^{\epsilon_{12}+s} & I & A^{\epsilon_{13}+r} \\ A^{\epsilon_{12}+s+t} & A^{\epsilon_{14}+s} & A^{\epsilon_{13}+t} & I \end{pmatrix},$$

from which we can see that $(\epsilon_{13}, \epsilon_{14}, \epsilon_{12}, s, t, r)$ and $(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$ are equivalent. However, such equivalences are not visible to Boyar's tool [BMP13] due to its heuristic nature, where the orders of the rows and columns do matter. That is, Boyar's tool may

[2]These conditions can be relaxed to find potentially better matrices.

Table 4: A list of equivalent parameters, where the Transformation column corresponds to certain column and row permutations explained in the following.

No.	Parameter	Transformation
1	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	–
2	$(\epsilon_{12}, \epsilon_{14}, \epsilon_{13}, r, t, s)$	(3, 4)
3	$(\epsilon_{13}, \epsilon_{12}, \epsilon_{14}, s, r, t)$	(2,3)
4	$(\epsilon_{13}, \epsilon_{14}, \epsilon_{12}, s, t, r)$	(2,4,3)
5	$(\epsilon_{14}, \epsilon_{12}, \epsilon_{13}, t, r, s)$	(2,3,4)
6	$(\epsilon_{14}, \epsilon_{13}, \epsilon_{12}, t, s, r)$	(2,4)
7	$(\epsilon_{12}+s+t, \epsilon_{13}+t, \epsilon_{14}+s, r-s, -t)$	(1,2)(3,4)
8	$(\epsilon_{12}+s+t, \epsilon_{14}+s, \epsilon_{13}+t, r, -t, -s)$	(1,2)
9	$(\epsilon_{13}+t, \epsilon_{12}+s+t, \epsilon_{14}+s, -s, r, -t)$	(1,3,4,2)
10	$(\epsilon_{13}+t, \epsilon_{14}+s, \epsilon_{12}+s+t, -s, -t, r)$	(1,4,2)
11	$(\epsilon_{14}+s, \epsilon_{12}+s+t, \epsilon_{13}+t, -t, r, -s)$	(1,3,2)
12	$(\epsilon_{14}+s, \epsilon_{13}+t, \epsilon_{12}+s+t, -t, -s, r)$	(1,4,3,2)
13	$(\epsilon_{12}+t, \epsilon_{13}+r+t, \epsilon_{14}+r, -r, s, -t)$	(1,3)(2,4)
14	$(\epsilon_{12}+t, \epsilon_{14}+r, \epsilon_{13}+r+t, -r, -t, s)$	(1,4,2,3)
15	$(\epsilon_{13}+r+t, \epsilon_{12}+t, \epsilon_{14}+r, s, -r, -t)$	(1,2,4,3)
16	$(\epsilon_{13}+r+t, \epsilon_{14}+r, \epsilon_{12}+t, s, -t, -r)$	(1,2,3)
17	$(\epsilon_{14}+r, \epsilon_{12}+t, \epsilon_{13}+r+t, -t, -r, s)$	(1,4,3)
18	$(\epsilon_{14}+r, \epsilon_{13}+r+t, \epsilon_{12}+t, -t, s, -r)$	(1,3)
19	$(\epsilon_{12}+s, \epsilon_{13}+r, \epsilon_{14}+r+s, -r, -s, t)$	(1,4)(2,3)
20	$(\epsilon_{12}+s, \epsilon_{14}+r+s, \epsilon_{13}+r, -r, t, -s)$	(1,3,2,4)
21	$(\epsilon_{13}+r, \epsilon_{12}+s, \epsilon_{14}+r+s, -s, -r, t)$	(1,4)
22	$(\epsilon_{13}+r, \epsilon_{14}+r+s, \epsilon_{12}+s, -s, t, -r)$	(1,3,4)
23	$(\epsilon_{14}+r+s, \epsilon_{12}+s, \epsilon_{13}+r, t, -r, -s)$	(1,2,4)
24	$(\epsilon_{14}+r+s, \epsilon_{13}+r, \epsilon_{12}+s, t, -s, -r)$	(1,2,3,4)

output circuits with different sizes and depths for two equivalent matrices. Therefore, in our experiment, we still need to search through all matrices we generated, and pick the ones with better implementations. One of the optimal matrices we find is

$$H = \begin{pmatrix} I_8 & I_8 & I_8 & A^4 \\ A^4 & I_8 & A^6 & A^2 \\ A^2 & A^4 & I_8 & A^2 \\ A^6 & I_8 & A^2 & I_8 \end{pmatrix}$$

corresponding to the parameter $(0, 0, 4, 0, 2, 2)$, where A is the companion matrix of $x^8 + x^2 + 1$ shown in Equation (2). The actual implementation of H is given in Table 5.

6 Searching for Low-latency Involutory MDS Matrices

In the previous section, we identify an involutory MDS Matrix which can be implemented with 78 `XOR` gates whose circuit depth is 4. Although this matrix is good with respect to lightweightness, we find that it is inferior to AES MixColumns operation in terms of latency. The lightest implementation (97 `XOR` gates) of the AES MixColumns operation is of depth 8, and if we increase the number of `XOR` gates, the AES MixColumns can be implemented with depth 3. In the following, we show that depth 3 is optimal.

Theorem 1. *The circuit depth of an MDS matrix $A \in \mathbf{M}_4(\mathrm{GL}(8, \mathbb{F}_2))$ with branch number 5 is at least 3.*

Table 5: An implementation of H, corresponding to parameter $(0, 0, 4, 0, 2, 2)$, with 78 `XOR` gates and depth 4, where $(x_0, \cdots, x_{31})$ are input signals, $(y_0, \cdots, y_{31})$ are output signals, and t_i's are intermediate signals.

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

Proof. Let

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} \\ A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} \\ A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} \\ A_{4,1} & A_{4,2} & A_{4,3} & A_{4,4} \end{pmatrix} \quad \text{with} \quad A_{i,j} \in \mathrm{GL}(8, \mathbb{F}_2) \tag{6}$$

be an MDS matrix with branch number 5 whose circuit depth is 2, which implies that each of the $4 \times 8 = 32$ rows of A contains at most four 1's. Then the Hamming weight of each row of the 8×8 submatrix $A_{i,j}$ is 1. Otherwise, there is one row of some submatrix $A_{i,j}$ whose Hamming weight is 0, which contradicts our assumption that A is MDS (see Lemma 2). Moreover, each column of $A_{i,j}$ contains only one 1. Otherwise we can identify two linearly dependent rows, which is a contradiction to the MDS property. Therefore, $A_{i,j}$ is a permutation matrix. Now let us consider the submatrix

$$A' = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix}.$$

The Hamming weights of each row and each column of A' is 2. Thus, the sum of the $2 \times 8 = 16$ rows of A' is a zero vector, meaning that A' is not invertible. This is a contradiction to the MDS property of A. □

Therefore, our goal is to find lightweight involutory matrices whose circuit depth is 3. Hopefully, we can identify one that is lighter than the MixColumns operation of AES, which does not enjoy the involutory property. For a given 32×32 matrix, Boyar's SLP-heuristic algorithm [BMP13] is virtually the best tool available for finding its lightweight implementation. However, Boyar's algorithm aims at minimizing the number of `XOR` gates of an implementation regardless of its circuit depth, which is not applicable in our scenario.

Given a set of input signals and a set of linear predicates represented as a binary matrix, Boyar's algorithm repeatedly picks two signals according to some rules, adds them together as a new signal, and puts this new signal into the signal set. Intuitively, after each iteration the signal set becomes "closer" to the set of linear predicates according to a notion of *distance*. The algorithm stops executing if and only if the distance becomes 0, that is, the set of signals compute the set of linear predicates.

In the following, we enhance Boyar's algorithm with circuit depth awareness. Basically, we modify Boyar's algorithm by only picking signals which are not going to exceed a

specified depth bound, and defining a new notion of distance which takes the circuit depth into account. The details are presented in Algorithm 1, where the subroutine `Pick()` picks two elements from the current signal set S such that when the exclusive-or of these two elements are put into the signal sets S, the sum of the values in the new distance vector Δ is minimized among all possible choices of the selected two elements, and ties will be resolved by maximizing the Euclidean norm of Δ. This strategy is exactly the same as Boyar's method [BMP13], except that the distances in Δ are computed according to our new definition presented in the following.

Algorithm 1: SLP heuristic with bounded circuit depth

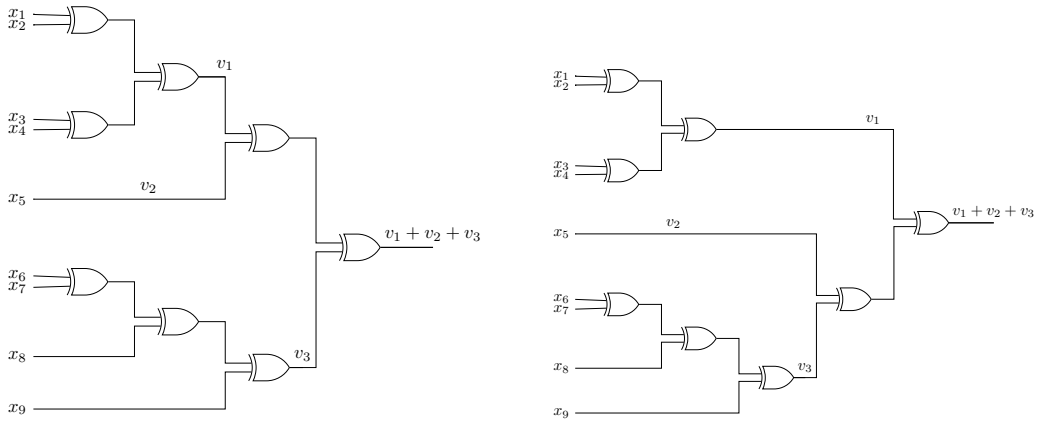
Input: An $m \times n$ binary matrix M representing m linear predicates in n variables, i.e., $(y_1, \cdots, y_m) = M(x_1, \cdots, x_n)^T$, and a positive integer H

Output: $S = [x_1, x_2, \cdots, x_n, x_{n+1}, x_{n+2}, \cdots, x_{n+l}]$ such that $d(x_j) \leq H$ for all j, and for any y_k with $1 \leq k \leq m$, y_k can be computed by one element in S_l, where $x_{n+j} = x_a + x_b$, $x_a, x_b \in \{x_1, \cdots, x_{n+j-1}\}$ for $j \geq 1$.

```
/* Initialization */
S ← [x_1, x_2, ··· , x_n]                                  /* The input signals */
D ← [0, 0, ··· , 0]    /* D[i] keeps track of the circuit depth of S[i] */
Δ ← [δ_H(S, y_1), ··· , δ_H(S, y_m)]                        /* The distances */

if Δ[i] = ∞ for some i then
    return Infeasible
end
                /* M can not be implemented within the depth bound H */

j ← n
while Δ ≠ 0 do
    j ← j + 1
    if ∃(x'_a, x'_b) ∈ S such that y_t = x'_a + x'_b for some t ∈ {1, ··· , m} then
        (x_a, x_b) ← (x'_a, x'_b)
    else
        (x_a, x_b) ← Pick(S, D, H)
    end
    x_j ← x_a + x_b
    S ← S ∪ [x_j]
    depth(x_j) ← max(D[a], D[b]) + 1          /* Compute the depth of x_j */
    D ← D ∪ [depth(x_j)]
    Δ ← [δ_H(S, y_1), ··· , δ_H(S, y_m)]                /* Update the distances */
end

return S
```

Let S be a sequence of signals. For any linear predicate f, we define $\delta_H(S, f)$ as the minimum number of additions (`XOR` gates) required to implement f with input signals from S, such that the depth of the implementation is not greater than H. We call $\delta_H(S, f)$ the *H-Distance* from S to f. Note that our notion of distance is different from Boyar's in that if $\delta_H(S, f) = k$, we not only require that f can be obtained by k additions, but also that there exits an implementation of k additions within depth H. If f can not be implemented within depth H, we have $\delta_H(S, f) = \infty$. In what follows, we use $\delta(S, f)$ to denote the distance defined in Boyar's work [BMP13], where the circuit depth is not considered.

Example 1. Let $S = [x_1, x_2, x_3, x_4, x_5]$, and $f = x_2 + x_3 + x_4 + x_5$. Then $\delta(S, f) = \delta_2(S, f) = 3$, and f can be implemented as $x_6 = x_2 + x_3$, $x_7 = x_4 + x_5$, and $x_8 = x_6 + x_7$,



(a) Implementation I (depth 4)

(b) Implementation II (depth 5)

Figure 1: Two implementations of the same summation $v_1 + v_2 + v_3$ with different circuit depths, where the depths of v_1, v_2 and v_3 are 2, 0, and 3 respectively.

where x_8 computes f, whose depth is 2.

Example 2. Let $S = [x_1, x_2, x_3, x_4, x_5, x_6 = x_2 + x_4, x_7 = x_3 + x_6]$ (note that the depths of x_6 and x_7 are 1 and 2 respectively), and $f = x_2 + x_3 + x_4 + x_5$. Then $\delta(S, f) = 1$, and f can be implemented as $x_5 + x_7$, whose depth is 3, while $\delta_2(S, f) = 2$, and f can be implemented within depth 2 as $x_8 = x_3 + x_5$, $x_9 = x_6 + x_8$, where x_9 computes f.

Example 3. Let $S = [x_1, x_2, x_3, x_4, x_5]$, and $f = x_1 + x_2 + x_3 + x_4 + x_5$. Then it is easy to check that $\delta(S, f) = 4$, and $\delta_2(S, f) = \infty$.

In Algorithm 1, we need a method to compute the minimal circuit depth of $v_1 + \cdots + v_k$, where the depths of v_i's are known. Note that there are many different ways of implementing $v_1 + \cdots + v_k$ which lead to different circuit depths as illustrated in Fig. 1. To deal with this, we prove the following theorem.

Theorem 2. *Let $\{v_1, v_2, \cdots, v_n\}$ be a set of signals with depth$(v_i) = d_i$, then the lower bound of the depth of the circuit implementing $z = v_1 + \cdots + v_n$ is $\lceil \log_2 \sum_{i=1}^{n} 2^{d_i} \rceil$. Moreover, there is always a circuit implementing z with depth $\lceil \log_2 \sum_{i=1}^{n} 2^{d_i} \rceil$, i.e., the lower bound is always achievable.*

Proof. We prove by induction on k, the number of terms in the summation. For $n = 1$ and $n = 2$, Theorem 2 holds obviously. Assuming that it holds for all $k < n$, we show in the following that it also holds for $k = n$.

Without loss of generality, any implementation of $z = v_1 + \cdots + v_n$ is of the form $z = z_a + z_b$, where $z_a = v_{i_1} + \cdots + v_{i_q}$, $z_b = v_{j_1} + \cdots + v_{j_{n-q}}$, and

$$\{v_{i_1}, \cdots, v_{i_q}\} \cup \{v_{j_1}, \cdots, v_{j_{n-q}}\} = \{v_1, v_2, \cdots, v_n\}.$$

Then $\text{depth}(z) = \max\{\text{depth}(z_a), \text{depth}(z_b)\} + 1$. According to the induction hypothesis, we have

$$\text{depth}(z_a) \geq \lceil \log_2 \sum_{t=1}^{q} 2^{d_{i_t}} \rceil,$$

$$\text{depth}(z_b) \geq \lceil \log_2 \sum_{t=1}^{n-q} 2^{d_{j_t}} \rceil.$$

Therefore, we can obtain that

$$\begin{aligned}\mathrm{depth}(z) &\geq \max\{\lceil \log_2 \sum_{t=1}^{q} 2^{d_{i_t}} \rceil, \lceil \log_2 \sum_{t=1}^{n-q} 2^{d_{j_t}} \rceil\} + 1 \\ &\geq \max\{1 + \lceil \log_2 \sum_{t=1}^{q} 2^{d_{i_t}} \rceil, 1 + \lceil \log_2 \sum_{t=1}^{n-q} 2^{d_{j_t}} \rceil\} \\ \geq \max\{\lceil \log_2 2 \sum_{t=1}^{q} 2^{d_{i_t}} \rceil, \lceil \log_2 2 \sum_{t=1}^{n-q} 2^{d_{j_t}} \rceil\} &\geq \lceil \log_2 \sum_{i=1}^{n} 2^{d_i} \rceil.\end{aligned}$$

Next, we show that the lower bound is achievable. First, we sort the set $\{v_1, \cdots, v_n\}$ of signals with non-decreasing depths. Then, we remove the leftmost two signals with the same depth, and insert the signal of their sum into the depth-ordered list. Without loss of generality, we assume that $\{v_1, \cdots, v_n\}$ is already in order, and $\mathrm{depth}(v_1) = \mathrm{depth}(v_2)$. After we update the set according to the above rule, we have a new set of signals $\{v_1 + v_2, v_3, \cdots, v_n\}$. Note that such operation preserves the sum $\sum_x 2^{\mathrm{depth}(x)}$, that is

$$2^{\mathrm{depth}(v_1)} + \cdots + 2^{\mathrm{depth}(v_n)} = 2^{\mathrm{depth}(v_1+v_2)} + 2^{\mathrm{depth}(v_3)} + \cdots + 2^{\mathrm{depth}(v_n)}.$$

We repeat the above operations until we obtain a set of signals $\{z_1, \cdots, z_m\}$ with $\mathrm{depth}(z_i) = q_i$ such that $q_1 < q_2 < \cdots < q_m$. Now, we are ready to give the implementation achieving the lower bound. First, if $m > 1$, we add z_1 and z_2 and obtain $z_{m+1} = z_1 + z_2$ whose depth $\mathrm{depth}(z_{m+1}) = q_2 + 1$; Then we add z_{m+1} and z_3 and obtain z_{m+2} whose depth $\mathrm{depth}(z_{m+2}) = q_3 + 1$; $\cdots$; Finally, we add z_{2m-2} and z_m and obtain z which implements $v_1 + \cdots + v_n$ whose depth $\mathrm{depth}(z) = q_m + 1$. Since $2^{q_m+1} > 2^{q_1} + \cdots + 2^{q_m} = 2^{\mathrm{depth}(v_1)} + \cdots + 2^{\mathrm{depth}(v_n)} > 2^{q_m}$, we can derive that $q_m + 1 = \lceil \log_2 \sum_{i=1}^{n} 2^{d_i} \rceil$.

If $m = 1$, $\mathrm{depth}(z) = q_1$, and $2^{\mathrm{depth}(v_1)} + \cdots + 2^{\mathrm{depth}(v_n)}$ is exactly a power of 2. In this case, we have $q_1 = \log_2 \sum_{i=1}^{n} 2^{d_i}$ □

In our algorithm, initially S is the sequence of all input signals. We maintain a list Δ to track the *H-distances* of the output signals from S. At the same time, we keep a list D such that $D[i]$ is the circuit depth of $S[i]$. At each iteration, we pick two different elements from S with `Pick`(S, D, H). Basically, we create a new element for S whose circuit depth is not greater than H by adding the two elements returned by `Pick`() which minimizes the sum of the new *H-distances*, where ties are resolved by maximizing the Euclidean norm of the new Δ. This strategy is the same as Boyar's SLP heuristic, and we refer the reader to [BMP13] for more information. Our algorithm is best illustrated by running through a toy example.

Example 4. Let the set of input signals be $\{x_1, x_2, x_3, x_4, x_5\}$, and

$$\begin{cases} y_1 = x_1 + x_2 + x_3 \\ y_2 = x_2 + x_4 + x_5 \\ y_3 = x_1 + x_3 + x_4 + x_5 \\ y_4 = x_2 + x_3 + x_4 \\ y_5 = x_1 + x_2 + x_4 \\ y_6 = x_2 + x_3 + x_4 + x_5 \end{cases}, \text{ which can be represented as } \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{7}$$

We execute the Algorithm 1 with $H = 2$.

Step 0. $S_0 = [x_1, x_2, x_3, x_4, x_5]$, $D_0 = [0, 0, 0, 0, 0]$, and $\Delta_0 = [2, 2, 3, 2, 2, 3]$.

Step 1. $S_1 = S_0 \cup [x_6 = x_2 + x_4] = [x_1, x_2, x_3, x_4, x_5, x_6 = x_2 + x_4]$, $D_1 = [0, 0, 0, 0, 0, 1]$, and $\Delta_1 = [2, 1, 3, 1, 1, 2]$.

Step 2. $S_2 = S_1 \cup [x_7 = x_5 + x_6] = [x_1, x_2, x_3, x_4, x_5, x_6 = x_2 + x_4, x_7 = x_5 + x_6]$, $D_2 = [0, 0, 0, 0, 0, 1, 2]$, and $\Delta_2 = [2, 0, 3, 1, 1, 2]$, where x_7 computes $x_2 + x_5 + x_4$.

Step 3. $S_3 = S_2 \cup [x_8 = x_3 + x_6] = [x_1, x_2, x_3, x_4, x_5, x_6 = x_2 + x_4, x_7 = x_5 + x_6, x_8 = x_3 + x_6]$, $D_3 = [0, 0, 0, 0, 0, 1, 2, 2]$, and $\Delta_3 = [2, 0, 3, 0, 1, 2]$, where x_8 computes $x_2 + x_3 + x_4$.

Step 4. $S_4 = S_3 \cup [x_9 = x_1 + x_6] = [x_1, x_2, x_3, x_4, x_5, x_6 = x_2 + x_4, x_7 = x_5 + x_6, x_8 = x_3 + x_6, x_9 = x_1 + x_6]$, $D_4 = [0, 0, 0, 0, 0, 1, 2, 2, 2]$, and $\Delta_4 = [2, 0, 3, 0, 0, 2]$, where x_9 computes $x_1 + x_2 + x_4$.

Step 5. $S_5 = S_4 \cup [x_{10} = x_1 + x_3] = [x_1, x_2, x_3, x_4, x_5, x_6 = x_2 + x_4, x_7 = x_5 + x_6, x_8 = x_3 + x_6, x_9 = x_1 + x_6, x_{10} = x_1 + x_3]$, $D_5 = [0, 0, 0, 0, 0, 1, 2, 2, 2, 1]$, and $\Delta_5 = [1, 0, 2, 0, 0, 2]$.

Step 6. $S_6 = S_5 \cup [x_{11} = x_2 + x_{10}] = [x_1, x_2, x_3, x_4, x_5, x_6 = x_2 + x_4, x_7 = x_5 + x_6, x_8 = x_3 + x_6, x_9 = x_1 + x_6, x_{10} = x_1 + x_3, x_{11} = x_2 + x_{10}]$, $D_6 = [0, 0, 0, 0, 0, 1, 2, 2, 2, 1, 2]$, and $\Delta_6 = [0, 0, 2, 0, 0, 2]$, where x_{11} computes $x_1 + x_2 + x_3$.

Step 7. $S_7 = S_6 \cup [x_{12} = x_3 + x_5] = [x_1, x_2, x_3, x_4, x_5, x_6 = x_2 + x_4, x_7 = x_5 + x_6, x_8 = x_3 + x_6, x_9 = x_1 + x_6, x_{10} = x_1 + x_3, x_{11} = x_2 + x_{10}, x_{12} = x_3 + x_5]$, $D_7 = [0, 0, 0, 0, 0, 1, 2, 2, 2, 1, 2, 1]$, and $\Delta_7 = [0, 0, 2, 0, 0, 1]$.

Step 8. $S_8 = S_7 \cup [x_{13} = x_6 + x_{12}] = [x_1, x_2, x_3, x_4, x_5, x_6 = x_2 + x_4, x_7 = x_5 + x_6, x_8 = x_3 + x_6, x_9 = x_1 + x_6, x_{10} = x_1 + x_3, x_{11} = x_2 + x_{10}, x_{12} = x_3 + x_5, x_{13} = x_6 + x_{12}]$, $D_8 = [0, 0, 0, 0, 0, 1, 2, 2, 2, 1, 2, 1, 2]$, and $\Delta_8 = [0, 0, 2, 0, 0, 0]$, where x_{13} computes $x_2 + x_3 + x_4 + x_5$.

Step 9. $S_9 = S_8 \cup [x_{14} = x_1 + x_4] = [x_1, x_2, x_3, x_4, x_5, x_6 = x_2 + x_4, x_7 = x_5 + x_6, x_8 = x_3 + x_6, x_9 = x_1 + x_6, x_{10} = x_1 + x_3, x_{11} = x_2 + x_{10}, x_{12} = x_3 + x_5, x_{13} = x_6 + x_{12}, x_{14} = x_1 + x_4]$, $D_9 = [0, 0, 0, 0, 0, 1, 2, 2, 2, 1, 2, 1, 2, 1]$, and $\Delta_9 = [0, 0, 1, 0, 0, 0]$.

Step 10. $S_{10} = S_9 \cup [x_{15} = x_{12} + x_{14}] = [x_1, x_2, x_3, x_4, x_5, x_6 = x_2 + x_4, x_7 = x_5 + x_6, x_8 = x_3 + x_6, x_9 = x_1 + x_6, x_{10} = x_1 + x_3, x_{11} = x_2 + x_{10}, x_{12} = x_3 + x_5, x_{13} = x_6 + x_{12}, x_{14} = x_1 + x_4, x_{15} = x_{12} + x_{14}]$, $D_{10} = [0, 0, 0, 0, 0, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2]$, and $\Delta_{10} = [0, 0, 0, 0, 0, 0]$, where x_{15} computes $x_1 + x_3 + x_4 + x_5$.

We apply this algorithm to all matrices we generated in Sect. 5, and the lightest one achieving the lower bound of the circuit depth (i.e., 3) we find is Q,

$$Q = \begin{pmatrix} I_8 & I_8 & A^{-2} & A^{-2} \\ A^{10} & I_8 & A^2 & A^4 \\ A^6 & I_8 & I_8 & A^6 \\ A^4 & I_8 & A^4 & I_8 \end{pmatrix}$$

corresponding to the parameter $(0, -2, -2, 2, 4, 6)$, where A the companion matrix of $x^8 + x^2 + 1$ shown in Equation (2). The actual implementation of Q is given in Table 6.

Remark. In Sects. 4-6, we only show the best matrices we find. We present a summary of all other results we obtained in Supplementary materials A and B, where we only show the parameter resulting in better circuit when equivalences are encountered. Moreover, The raw data and source code are also submitted as supplementary material along the paper.

7 Conclusion

In this work, we find so far the lightest 32×32 involutory MDS matrices whose branch number is 5 by searching through a large set of matrices whose entries are the powers of the companion matrix of $x^8 + x^2 + 1$. Moreover, we enhance Boyar's SLP heuristic with

Table 6: An implementation of Q, corresponding to parameter $(0, -2, -2, 2, 4, 6)$, with 88 `XOR` gates and depth 3, where $(x_0, \cdots, x_{31})$ are input signals, $(y_0, \cdots, y_{31})$ are output signals, and t_i's are intermediate signals.

No	Operation	Depth	No	Operation	Depth	No	Operation	Depth
1	$t_1 = x_4 + x_{20}$	1	31	$t_{31} = x_5 + x_{23}$	1	61	$t_{61} = x_{14} + x_{26}$	1
2	$t_2 = x_5 + x_{21}$	1	32	$t_{32} = t_{14} + t_{31}$ $[y_5]$	2	62	$t_{62} = t_{25} + t_{61}$ $[y_{14}]$	3
3	$t_3 = x_6 + x_{22}$	1	33	$t_{33} = x_6 + x_{16}$	1	63	$t_{63} = x_{14} + x_{30}$	1
4	$t_4 = x_7 + x_{23}$	1	34	$t_{34} = t_{19} + t_{33}$ $[y_6]$	2	64	$t_{64} = t_{21} + t_{63}$ $[y_{30}]$	2
5	$t_5 = x_2 + x_{26}$	1	35	$t_{35} = x_{22} + x_{30}$	1	65	$t_{65} = x_{15} + x_{27}$	1
6	$t_6 = x_3 + x_{27}$	1	36	$t_{36} = t_9 + t_{35}$	2	66	$t_{66} = t_{27} + t_{65}$ $[y_{15}]$	3
7	$t_7 = x_4 + x_{28}$	1	37	$t_{37} = t_8 + t_{36}$ $[y_{10}]$	3	67	$t_{67} = x_{15} + x_{31}$	1
8	$t_8 = x_{10} + t_7$	2	38	$t_{38} = t_{34} + t_{36}$ $[y_{22}]$	3	68	$t_{68} = t_{23} + t_{67}$ $[y_{31}]$	2
9	$t_9 = x_0 + x_{16}$	1	39	$t_{39} = x_7 + x_{17}$	1	69	$t_{69} = x_{18} + t_5$	2
10	$t_{10} = x_5 + x_{29}$	1	40	$t_{40} = t_{20} + t_{39}$ $[y_7]$	2	70	$t_{70} = t_8 + t_{69}$ $[y_{18}]$	3
11	$t_{11} = x_{11} + t_{10}$	2	41	$t_{41} = x_{23} + x_{31}$	1	71	$t_{71} = x_{19} + t_6$	2
12	$t_{12} = x_1 + x_{17}$	1	42	$t_{42} = t_{12} + t_{41}$	2	72	$t_{72} = t_{11} + t_{71}$ $[y_{19}]$	3
13	$t_{13} = x_{12} + x_{30}$	1	43	$t_{43} = t_{11} + t_{42}$ $[y_{11}]$	3	73	$t_{73} = x_{20} + t_7$	2
14	$t_{14} = x_{13} + x_{31}$	1	44	$t_{44} = t_{40} + t_{42}$ $[y_{23}]$	3	74	$t_{74} = t_{22} + t_{73}$ $[y_{20}]$	3
15	$t_{15} = x_8 + x_{24}$	1	45	$t_{45} = x_8 + x_{16}$	1	75	$t_{75} = x_{21} + t_{10}$	2
16	$t_{16} = t_1 + t_{15}$ $[y_{24}]$	2	46	$t_{46} = t_5 + t_{45}$ $[y_{16}]$	2	76	$t_{76} = t_{24} + t_{75}$ $[y_{21}]$	3
17	$t_{17} = x_9 + x_{25}$	1	47	$t_{47} = x_0 + x_{24}$	1	77	$t_{77} = x_6 + x_{28}$	1
18	$t_{18} = t_2 + t_{17}$ $[y_{25}]$	2	48	$t_{48} = t_{21} + t_{47}$	2	78	$t_{78} = t_{13} + t_{77}$	2
19	$t_{19} = x_{14} + x_{24}$	1	49	$t_{49} = t_{46} + t_{48}$ $[y_0]$	3	79	$t_{79} = t_{36} + t_{78}$ $[y_{28}]$	3
20	$t_{20} = x_{15} + x_{25}$	1	50	$t_{50} = t_{22} + t_{48}$ $[y_{12}]$	3	80	$t_{80} = x_7 + x_{29}$	1
21	$t_{21} = x_2 + x_{18}$	1	51	$t_{51} = x_8 + t_3$	2	81	$t_{81} = t_{14} + t_{80}$	2
22	$t_{22} = x_6 + t_{13}$	2	52	$t_{52} = t_7 + t_{51}$ $[y_8]$	3	82	$t_{82} = t_{42} + t_{81}$ $[y_{29}]$	3
23	$t_{23} = x_3 + x_{19}$	1	53	$t_{53} = x_9 + x_{17}$	1	83	$t_{83} = x_{10} + x_{26}$	1
24	$t_{24} = x_7 + t_{14}$	2	54	$t_{54} = t_6 + t_{53}$ $[y_{17}]$	2	84	$t_{84} = t_1 + t_3$	2
25	$t_{25} = x_2 + t_1$	2	55	$t_{55} = x_1 + x_{25}$	1	85	$t_{85} = t_{83} + t_{84}$ $[y_{26}]$	3
26	$t_{26} = t_8 + t_{25}$ $[y_2]$	3	56	$t_{56} = t_{23} + t_{55}$	2	86	$t_{86} = x_{11} + x_{27}$	1
27	$t_{27} = x_3 + t_2$	2	57	$t_{57} = t_{54} + t_{56}$ $[y_1]$	3	87	$t_{87} = t_2 + t_4$	2
28	$t_{28} = t_{11} + t_{27}$ $[y_3]$	3	58	$t_{58} = t_{24} + t_{56}$ $[y_{13}]$	3	88	$t_{88} = t_{86} + t_{87}$ $[y_{27}]$	3
29	$t_{29} = x_4 + x_{22}$	1	59	$t_{59} = x_9 + t_4$	2			
30	$t_{30} = t_{13} + t_{29}$ $[y_4]$	2	60	$t_{60} = t_{10} + t_{59}$ $[y_9]$	3			

circuit depth awareness, which enables us to identify so far the lightest 32×32 involutory MDS matrix whose circuit depth is 3, achieving the provable lower bound for a 32×32 MDS matrix. Along the way, we present a formula, which is of independent interest, for computing the minimum achievable depth of a circuit implementing the summation of a set of signals with given depths. The results of this work can be potentially applied in the design of lightweight and low-latency symmetric-key primitives.

Acknowledgments

The authors thank the anonymous reviewers for many helpful comments. The work is supported by the National Key R&D Program of China (Grant No. 2018YFB0804402), the Chinese Major Program of National Cryptography Development Foundation (Grant No. MMJJ20180102), the National Natural Science Foundation of China (61732021, 61802400, 61772519, 61802399), and the Youth Innovation Promotion Association of Chinese Academy of Sciences. Chaoyun Li is supported by the Research Council KU Leuven: C16/15/058, OT/13/071, and by European Union's Horizon 2020 research and innovation programme under grant agreement No. H2020-MSCA-ITN-2014-643161 ECRYPT-NET.

References

[AF14] Daniel Augot and Matthieu Finiasz. Direct construction of recursive MDS diffusion layers using shortened BCH codes. In *FSE'14*, pages 3–17, 2014.

[Ava17] Roberto Avanzi. The QARMA block cipher family. almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency S-Boxes. *IACR Trans. Symmetric Cryptol.*, 2017(1):4–44, 2017.

[BBI+15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In *ASIACRYPT'15*, pages 411–436, 2015.

[BCG+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In *ASIACRYPT'12*, pages 208–225, 2012.

[Ber13] Thierry P. Berger. Construction of recursive MDS diffusion layers from gabidulin codes. In *INDOCRYPT'13*, pages 274–285, 2013.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In *CRYPTO'16*, pages 123–153, 2016.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In *CHES'07*, pages 450–466, 2007.

[BKL16] Christof Beierle, Thorsten Kranz, and Gregor Leander. Lightweight multiplication in gf(2^n) with applications to MDS matrices. In *CRYPTO'16*, pages 625–653, 2016.

[BMP08] Joan Boyar, Philip Matthews, and René Peralta. On the shortest linear straight-line program for computing linear forms. In *MFCS'08*, pages 168–179, 2008.

[BMP13] Joan Boyar, Philip Matthews, and René Peralta. Logic minimization techniques with applications to cryptology. *J. Cryptology*, 26(2):280–312, 2013.

[BNN+10] Paulo S. L. M. Barreto, Ventzislav Nikov, Svetla Nikova, Vincent Rijmen, and Elmar Tischhauser. Whirlwind: a new cryptographic hash function. *Des. Codes Cryptography*, 56(2-3):141–162, 2010.

[BR99] Mario Blaum and Ron M. Roth. On lowest density MDS codes. *IEEE Trans. Information Theory*, 45(1):46–59, 1999.

[BR00] Paulo Sérgio L.M. Barreto and Vincent Rijmen. The Anubis block cipher, 2000. Submission to the NESSIE project.

[CLM16] Victor Cauchois, Pierre Loidreau, and Nabil Merkiche. Direct construction of quasi-involutory recursive-like MDS matrices from 2-cyclic codes. *IACR Trans. Symmetric Cryptol.*, 2016(2):80–98, 2016.

[Con14] Keith Conrad. The minimal polynomial and some applications. http://www.math.uconn.edu/~kconrad/blurbs/linmultialg/minpolyandappns.pdf, 2014.

[DF04] David S. Dummit and Richard M. Foote. *Abstract algebra*, volume 3. Wiley Hoboken, 2004.

[DL18] Sébastien Duval and Gaëtan Leurent. MDS matrices with lightweight circuits. *IACR Trans. Symmetric Cryptol.*, 2018(2):48–78, 2018.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[GLWL16] Zhiyuan Guo, Renzhang Liu, Wenling Wu, and Dongdai Lin. Direct construction of lightweight rotational-xor MDS diffusion layers. *IACR Cryptology ePrint Archive*, 2016:1036, 2016.

[GPP11] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. In *CRYPTO'11*, pages 222–239, 2011.

[GPPR11] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In *CHES'11*, pages 326–341, 2011.

[GPV17] Kishan Chand Gupta, Sumit Kumar Pandey, and Ayineedi Venkateswarlu. Towards a general construction of recursive MDS diffusion layers. *Des. Codes Cryptography*, 82(1-2):179–195, 2017.

[JPST17] Jérémy Jean, Thomas Peyrin, Siang Meng Sim, and Jade Tourteaux. Optimizing implementations of lightweight building blocks. *IACR Trans. Symmetric Cryptol.*, 2017(4):130–168, 2017.

[KLSW17] Thorsten Kranz, Gregor Leander, Ko Stoffelen, and Friedrich Wiemer. Shorter linear straight-line programs for MDS matrices. *IACR Trans. Symmetric Cryptol.*, 2017(4):188–211, 2017.

[KPPY14] Khoongming Khoo, Thomas Peyrin, Axel York Poschmann, and Huihui Yap. FOAM: Searching for hardware-optimal SPN structures and components with a fair comparison. In *CHES'14*, pages 433–450, 2014.

[LS16] Meicheng Liu and Siang Meng Sim. Lightweight MDS generalized circulant matrices. In *FSE'16*, volume 9783 of *LNCS*, pages 101–120. Springer, 2016.

[LW16] Yongqiang Li and Mingsheng Wang. On the construction of lightweight circulant involutory MDS matrices. In *FSE'16*, pages 121–139, 2016.

[LW17] Chaoyun Li and Qingju Wang. Design of lightweight linear diffusion layers from near-MDS matrices. *IACR Trans. Symmetric Cryptol.*, 2017(1):129–155, 2017.

[SKOP15] Siang Meng Sim, Khoongming Khoo, Frédérique E. Oggier, and Thomas Peyrin. Lightweight MDS involution matrices. In *FSE'15*, pages 471–493, 2015.

[SPR+04] François-Xavier Standaert, Gilles Piret, Gaël Rouvroy, Jean-Jacques Quisquater, and Jean-Didier Legat. ICEBERG: An involutional cipher efficient for block encryption in reconfigurable hardware. In *FSE'04*, pages 279–299, 2004.

[SS16a] Sumanta Sarkar and Siang Meng Sim. A deeper understanding of the XOR count distribution in the context of lightweight cryptography. In *AFRICACRYPT'16*, pages 167–182, 2016.

[SS16b] Sumanta Sarkar and Habeeb Syed. Lightweight diffusion layer: Importance of toeplitz matrices. *IACR Trans. Symmetric Cryptol.*, 2016(1):95–113, 2016.

[SS17] Sumanta Sarkar and Habeeb Syed. Analysis of toeplitz MDS matrices. In *ACISP'17*, pages 3–18, 2017.

[TTKS18] Dylan Toh, Jacob Teo, Khoongming Khoo, and Siang Meng Sim. Lightweight MDS serial-type matrices with minimal fixed XOR count. In *AFRICACRYPT'18*, pages 51–71, 2018.

[Wan03] Zhexian Wan. *Lectures on finite fields and Galois rings*. World Scientific Publishing Company, 2003.

[WWW12] Shengbao Wu, Mingsheng Wang, and Wenling Wu. Recursive diffusion layers for (lightweight) block ciphers and hash functions. In *SAC'12*, pages 355–371, 2012.

[ZWS18] Lijing Zhou, Licheng Wang, and Yiru Sun. On efficient constructions of lightweight MDS matrices. *IACR Trans. Symmetric Cryptol.*, 2018(1):180–200, 2018.

A A List of Involutory MDS Matrices

$\omega(A) = 148$, $\mathsf{DXC}(A) = 116$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-2, -1, 2, 0, 0, 0)$	80	4
2	$(-2, 1, -2, 0, 0, 2)$	80	4

$\omega(A) = 149$, $\mathsf{DXC}(A) = 117$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-3, -2, 1, 1, 1, 1)$	80	4
2	$(-1, 0, 3, -1, -1, -1)$	80	4

$\omega(A) = 150$, $\mathsf{DXC}(A) = 118$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-3, -2, 2, 0, 0, 2)$	80	4
2	$(-3, 1, -1, 0, 0, 2)$	80	4
3	$(-4, -2, 1, 0, 2, 2)$	80	4
4	$(0, -3, -2, 0, 2, 2)$	83	4

$\omega(A) = 151$, $\mathsf{DXC}(A) = 119$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-4, 0, -2, 1, 1, 3)$	83	4
2	$(0, 4, 0, -1, -1, -3)$	83	5

$\omega(A) = 152$, $\mathsf{DXC}(A) = 120$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-4, 0, -1, 0, 0, 4)$	86	5
2	$(-3, 0, -3, 1, 1, 3)$	83	4
3	$(1, 4, -1, -1, -1, -3)$	83	4

$\omega(A) = 153$, $\mathsf{DXC}(A) = 121$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-3, -3, 1, 0, 2, 2)$	80	5
2	$(-4, -3, 0, 2, 2, 2)$	83	4
3	$(0, 1, 4, -2, -2, -2)$	83	5

$\omega(A) = 154$, $\mathsf{DXC}(A) = 122$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-3, 0, -2, 0, 0, 4)$	86	4
2	$(-1, -4, -2, 0, 2, 4)$	86	4
3	$(-4, -3, 1, 1, 1, 3)$	80	5
4	$(0, 1, 3, -1, -1, -3)$	80	5

$\omega(A) = 155$, $\mathsf{DXC}(A) = 123$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-5, 0, -2, 0, 2, 4)$	86	5

$\omega(A) = 156$, $\mathsf{DXC}(A) = 124$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-4, 0, -3, 0, 2, 4)$	86	5
2	$(-1, -4, -3, 1, 3, 3)$	86	4
3	$(5, 0, 1, -1, -3, -3)$	86	4

$\omega(A) = 157$, $\mathsf{DXC}(A) = 125$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-5, -3, 0, 1, 3, 3)$	83	5
2	$(1, 1, 4, -1, -3, -3)$	82	5
3	$(-4, -4, 0, 1, 3, 3)$	83	5
4	$(2, 0, 4, -1, -3, -3)$	83	5

$\omega(A) = 158$, $\mathsf{DXC}(A) = 126$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-4, -3, 2, 0, 0, 4)$	80	5
2	$(-4, -4, 1, 0, 2, 4)$	80	5
3	$(-1, -3, -3, 0, 2, 4)$	86	4
4	$(-5, -1, -2, 1, 1, 5)$	89	5
5	$(1, 5, 0, -1, -1, -5)$	89	5
6	$(-4, -1, -4, 2, 2, 4)$	86	4
7	$(2, 5, 0, -2, -2, -4)$	85	6

$\omega(A) = 160$, $\mathsf{DXC}(A) = 128$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(1, 2, 5, 0, 0, 0)$	82	5
2	$(0, 1, 5, 0, 0, 2)$	80	5
3	$(0, 4, 2, 0, 0, 2)$	80	5
4	$(1, 4, 1, 0, 0, 2)$	78	4
5	$(0, 0, 4, 0, 2, 2)$	78	4
6	$(0, 1, 4, 1, 1, 1)$	79	5
7	$(2, 3, 6, -1, -1, -1)$	79	5
8	$(-4, -1, -3, 1, 1, 5)$	89	5
9	$(2, 5, -1, -1, -1, -5)$	89	4
10	$(-5, -1, -3, 2, 2, 4)$	86	5
11	$(1, 5, 1, -2, -2, -4)$	85	5

$\omega(A) = 161$, $\mathsf{DXC}(A) = 129$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(3, 0, 1, 0, 2, 2)$	80	5
2	$(-5, -3, 1, 0, 2, 4)$	80	6
3	$(0, 3, 0, 1, 1, 3)$	79	7
4	$(4, 7, 2, -1, -1, -3)$	79	4
5	$(-5, -4, 0, 2, 2, 4)$	83	5
6	$(1, 2, 4, -2, -2, -4)$	82	5

$\omega(A) = 162$, $\mathsf{DXC}(A) = 130$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(0, 3, 1, 0, 0, 4)$	80	7
2	$(-1, 1, 4, 0, 2, 2)$	80	5
3	$(2, 0, 0, 0, 2, 4)$	80	4
4	$(-5, -4, 0, 0, 4, 4)$	85	6
5	$(-2, -4, -3, 0, 4, 4)$	88	4
6	$(-1, 0, 4, 1, 1, 3)$	79	5
7	$(3, 4, 6, -1, -1, -3)$	79	5
8	$(-5, -4, -1, 3, 3, 3)$	86	5
9	$(1, 2, 5, -3, -3, -3)$	85	5

$\omega(A) = 163$, $\mathsf{DXC}(A) = 131$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-1, 3, 1, 1, 1, 3)$	81	5
2	$(3, 7, 3, -1, -1, -3)$	81	5
3	$(-5, -1, -4, 1, 3, 5)$	88	6
4	$(3, 5, 0, -1, -3, -5)$	88	6
5	$(-2, -5, -3, 1, 3, 5)$	88	5
6	$(6, 1, 1, -1, -3, -5)$	88	5
7	$(-1, 0, 3, 2, 2, 2)$	79	5
8	$(3, 4, 7, -2, -2, -2)$	80	5

$\omega(A) = 164$, $\mathsf{DXC}(A) = 132$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	depth(A)
1	$(-1, 0, 5, 0, 0, 4)$	81	7
2	$(-5, -1, -1, 0, 0, 6)$	92	4
3	$(-4, -1, -2, 0, 0, 6)$	92	4
4	$(-2, 0, 4, 0, 2, 4)$	80	5
5	$(-1, 3, 0, 0, 2, 4)$	78	5
6	$(-6, -1, -2, 0, 2, 6)$	91	5
7	$(-2, -5, -2, 0, 2, 6)$	91	4
8	$(-6, -3, 0, 0, 4, 4)$	84	6
9	$(2, -1, 0, 1, 3, 3)$	80	7
10	$(8, 3, 4, -1, -3, -3)$	81	5
11	$(-6, -1, -3, 1, 3, 5)$	88	6
12	$(2, 5, 1, -1, -3, -5)$	88	5
13	$(-2, -4, -4, 1, 3, 5)$	88	4
14	$(6, 2, 0, -1, -3, -5)$	87	6

$\omega(A) = 165$, $\mathsf{DXC}(A) = 133$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	depth(A)
1	$(1, -1, 0, 0, 4, 4)$	81	5
2	$(-5, -4, 1, 1, 1, 5)$	80	6
3	$(1, 2, 3, -1, -1, -5)$	80	6
4	$(-1, 2, 0, 1, 1, 5)$	79	7
5	$(5, 8, 2, -1, -1, -5)$	79	5
6	$(-2, 0, 3, 1, 3, 3)$	80	5
7	$(4, 4, 7, -1, -3, -3)$	81	5
8	$(-1, -1, 3, 1, 3, 3)$	79	5
9	$(5, 3, 7, -1, -3, -3)$	81	6
10	$(1, -1, -1, 1, 3, 5)$	79	7
11	$(9, 5, 3, -1, -3, -5)$	82	6
12	$(-2, -5, -4, 2, 4, 4)$	88	5
13	$(6, 1, 2, -2, -4, -4)$	87	6

$\omega(A) = 166$, $\mathsf{DXC}(A) = 134$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	depth(A)
1	$(-1, 3, 2, 0, 0, 4)$	82	7
2	$(-2, 3, 1, 0, 2, 4)$	81	5
3	$(-1, -1, 4, 0, 2, 4)$	81	5
4	$(2, -1, 1, 0, 2, 4)$	81	5
5	$(-2, -4, -3, 0, 2, 6)$	91	5
6	$(1, -1, 0, 0, 2, 6)$	83	5
7	$(-5, -5, 0, 1, 3, 5)$	83	6
8	$(3, 1, 4, -1, -3, -5)$	82	6
9	$(-2, 2, 0, 2, 2, 4)$	80	5
10	$(4, 8, 4, -2, -2, -4)$	82	5
11	$(-1, 2, -1, 2, 2, 4)$	80	5
12	$(5, 8, 3, -2, -2, -4)$	82	7
13	$(-6, -4, -1, 2, 4, 4)$	85	6
14	$(2, 2, 5, -2, -4, -4)$	84	5

$\omega(A) = 167$, $\mathsf{DXC}(A) = 135$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(5, -2, 0, 0, 2, 2)$	90	5
2	$(-5, -1, -3, 0, 2, 6)$	91	5
3	$(1, -2, 0, 1, 3, 5)$	82	5
4	$(9, 4, 4, -1, -3, -5)$	82	5
5	$(-2, -1, 3, 2, 2, 4)$	80	5
6	$(4, 5, 7, -2, -2, -4)$	83	6
7	$(-6, -2, -3, 2, 2, 6)$	91	6
8	$(2, 6, 1, -2, -2, -6)$	90	5
9	$(-5, -2, -4, 2, 2, 6)$	91	5
10	$(3, 6, 0, -2, -2, -6)$	89	6
11	$(-5, -5, -1, 2, 4, 4)$	86	6
12	$(3, 1, 5, -2, -4, -4)$	86	7
13	$(1, -2, -1, 2, 4, 4)$	80	5
14	$(9, 4, 5, -2, -4, -4)$	83	6

$\omega(A) = 168$, $\mathsf{DXC}(A) = 136$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-3, -1, 5, 0, 0, 0)$	97	5
2	$(-1, 1, 7, 0, 0, 0)$	91	5
3	$(-4, -2, 5, 0, 0, 2)$	98	6
4	$(-4, 4, -1, 0, 0, 2)$	98	5
5	$(-2, 0, 7, 0, 0, 2)$	90	5
6	$(-2, 4, -3, 0, 0, 2)$	94	4
7	$(-2, 6, 1, 0, 0, 2)$	92	6
8	$(0, 6, -1, 0, 0, 2)$	90	4
9	$(-1, 2, 1, 0, 0, 6)$	79	5
10	$(-2, 2, 0, 0, 2, 6)$	78	5
11	$(0, -2, 0, 0, 4, 6)$	82	5
12	$(0, -1, -1, 0, 4, 6)$	81	6
13	$(-4, -2, 4, 1, 1, 1)$	96	5
14	$(-2, 0, 6, -1, -1, -1)$	97	5
15	$(-2, 0, 6, 1, 1, 1)$	90	5
16	$(0, 2, 8, -1, -1, -1)$	90	6
17	$(-2, -1, 4, 1, 1, 5)$	82	5
18	$(4, 5, 6, -1, -1, -5)$	81	5
19	$(-6, -4, 0, 1, 3, 5)$	82	6
20	$(2, 2, 4, -1, -3, -5)$	82	6
21	$(0, -2, -1, 1, 5, 5)$	81	5
22	$(10, 4, 5, -1, -5, -5)$	83	6
23	$(-2, 1, -1, 2, 2, 6)$	79	5
24	$(6, 9, 3, -2, -2, -6)$	81	6
25	$(0, -2, -2, 2, 4, 6)$	78	4
26	$(10, 6, 4, -2, -4, -6)$	84	6
27	$(-2, -1, 2, 3, 3, 3)$	80	5
28	$(4, 5, 8, -3, -3, -3)$	83	6
29	$(-5, -2, -5, 3, 3, 5)$	88	6
30	$(3, 6, 1, -3, -3, -5)$	88	6

$\omega(A) = 169$, $\mathsf{DXC}(A) = 137$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(3, -4, -2, 0, 2, 2)$	96	6
2	$(-3, 0, 3, 0, 4, 4)$	83	7
3	$(-1, 5, -2, 1, 1, 3)$	90	4
4	$(3, 9, 0, -1, -1, -3)$	92	5
5	$(-2, 2, 1, 1, 1, 5)$	83	7
6	$(4, 8, 3, -1, -1, -5)$	83	7
7	$(-2, 1, 0, 1, 1, 7)$	83	5
8	$(6, 9, 2, -1, -1, -7)$	82	5
9	$(-2, 2, -1, 1, 3, 5)$	79	5
10	$(6, 8, 3, -1, -3, -5)$	82	6
11	$(0, -2, -1, 1, 3, 7)$	82	5
12	$(10, 6, 3, -1, -3, -7)$	83	6
13	$(-6, -2, -4, 3, 3, 5)$	87	6
14	$(2, 6, 2, -3, -3, -5)$	87	5
15	$(-2, 1, -2, 3, 3, 5)$	79	5
16	$(6, 9, 4, -3, -3, -5)$	84	6

$\omega(A) = 170$, $\mathsf{DXC}(A) = 138$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-1, 5, -1, 0, 0, 4)$	92	5
2	$(-2, 1, 1, 0, 0, 8)$	86	5
3	$(-5, -2, 4, 0, 2, 2)$	96	5
4	$(-1, -2, 6, 0, 2, 2)$	92	7
5	$(-2, 5, -2, 0, 2, 4)$	93	5
6	$(4, -1, -2, 0, 2, 4)$	93	5
7	$(1, -2, 1, 0, 2, 6)$	83	7
8	$(0, -2, 0, 0, 2, 8)$	86	5
9	$(-2, -1, 3, 0, 4, 4)$	82	6
10	$(-6, -1, -4, 0, 4, 6)$	90	6
11	$(-3, 5, 0, 1, 1, 3)$	93	6
12	$(1, 9, 2, -1, -1, -3)$	92	6
13	$(-6, -2, -2, 1, 1, 7)$	94	4
14	$(2, 6, 0, -1, -1, -7)$	93	4
15	$(-3, -1, 3, 1, 3, 5)$	81	5
16	$(5, 5, 7, -1, -3, -5)$	81	5
17	$(-3, 2, 0, 1, 3, 5)$	83	6
18	$(5, 8, 4, -1, -3, -5)$	83	6
19	$(-2, -2, 2, 2, 4, 4)$	80	5
20	$(6, 4, 8, -2, -4, -4)$	84	6
21	$(-3, 1, -1, 3, 3, 5)$	82	5
22	$(5, 9, 5, -3, -3, -5)$	82	5

$\omega(A) = 171$, $\mathsf{DXC}(A) = 139$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(-3, -4, 4, 0, 2, 2)$	96	7
2	$(-3, 0, 6, 0, 2, 2)$	91	6
3	$(-5, 3, -2, 1, 1, 3)$	98	6
4	$(-1, 7, 0, -1, -1, -3)$	98	7
5	$(-2, 4, -2, 1, 1, 5)$	92	5
6	$(4, 10, 0, -1, -1, -5)$	92	6
7	$(-5, -2, -3, 1, 1, 7)$	93	5
8	$(3, 6, -1, -1, -1, -7)$	93	5
9	$(4, -3, -1, 1, 3, 3)$	92	5
10	$(10, 1, 3, -1, -3, -3)$	94	5
11	$(-2, -2, 3, 1, 3, 5)$	82	5
12	$(6, 4, 7, -1, -3, -5)$	84	6
13	$(-3, 1, 0, 2, 2, 6)$	84	6
14	$(5, 9, 4, -2, -2, -6)$	81	5
15	$(-3, -1, 2, 2, 4, 4)$	81	5
16	$(5, 5, 8, -2, -4, -4)$	82	5
17	$(0, -3, -1, 2, 4, 6)$	83	5
18	$(10, 5, 5, -2, -4, -6)$	82	5
19	$(-6, -5, -1, 3, 3, 5)$	85	7
20	$(2, 3, 5, -3, -3, -5)$	84	6
21	$(0, -3, -2, 3, 5, 5)$	81	6
22	$(10, 5, 6, -3, -5, -5)$	83	6

$\omega(A) = 172$, $\mathsf{DXC}(A) = 140$			
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}(A)$	$\mathrm{depth}(A)$
1	$(2, 3, 6, 0, 0, 0)$	84	5
2	$(1, 2, 6, 0, 0, 2)$	84	5
3	$(-3, 5, 1, 0, 0, 4)$	96	8
4	$(0, 1, 6, 0, 0, 4)$	87	5
5	$(-5, -4, 2, 0, 0, 6)$	80	6
6	$(-2, -1, 5, 0, 0, 6)$	83	6
7	$(-2, 2, 2, 0, 0, 6)$	84	7
8	$(-2, 4, -1, 0, 0, 6)$	94	5
9	$(-4, 5, 0, 0, 2, 4)$	94	5
10	$(4, -3, 0, 0, 2, 4)$	96	6
11	$(-6, -4, 1, 0, 2, 6)$	82	6
12	$(-2, -2, 4, 0, 2, 6)$	82	5
13	$(3, -2, -2, 0, 2, 6)$	95	5
14	$(-3, 1, 0, 0, 2, 8)$	79	5
15	$(2, 0, 1, 0, 4, 4)$	86	4
16	$(-7, -1, -3, 0, 4, 6)$	89	7
17	$(-3, -4, -4, 0, 4, 6)$	90	5
18	$(1, 0, 0, 0, 4, 6)$	88	5
19	$(-3, 3, -4, 1, 1, 3)$	94	5
20	$(1, 7, -2, -1, -1, -3)$	94	6
21	$(-7, -2, -3, 1, 3, 7)$	94	6
22	$(3, 6, 1, -1, -3, -7)$	92	6
23	$(0, -3, 0, 1, 3, 7)$	85	6
24	$(10, 5, 4, -1, -3, -7)$	81	5
25	$(-7, -4, -1, 1, 5, 5)$	86	7
26	$(3, 2, 5, -1, -5, -5)$	86	7
27	$(-6, -5, -1, 1, 5, 5)$	87	8
28	$(4, 1, 5, -1, -5, -5)$	86	6
29	$(-3, -1, 5, 2, 2, 2)$	92	6
30	$(1, 3, 9, -2, -2, -2)$	93	5
31	$(-7, -2, -4, 2, 4, 6)$	90	6
32	$(3, 6, 2, -2, -4, -6)$	89	6
33	$(-6, -5, -2, 4, 4, 4)$	87	7
34	$(2, 3, 6, -4, -4, -4)$	87	6
35	$(-3, -2, 1, 4, 4, 4)$	82	5
36	$(5, 6, 9, -4, -4, -4)$	83	6

B A List of Involutory MDS Matrices with Depth 3

$\omega(A) = 148$, $\mathsf{DXC}(A) = 116$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-2, -1, 2, 0, 0, 0)$	90
2	$(-2, 1, -2, 0, 0, 2)$	90

$\omega(A) = 149$, $\mathsf{DXC}(A) = 117$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-3, -2, 1, 1, 1, 1)$	90
2	$(-1, 0, 3, -1, -1, -1)$	90

$\omega(A) = 150$, $\mathsf{DXC}(A) = 118$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-3, -2, 2, 0, 0, 2)$	91
2	$(-3, 1, -1, 0, 0, 2)$	90
3	$(-4, -2, 1, 0, 2, 2)$	90
4	$(0, -3, -2, 0, 2, 2)$	93

$\omega(A) = 151$, $\mathsf{DXC}(A) = 119$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-4, 0, -2, 1, 1, 3)$	94
2	$(0, 4, 0, -1, -1, -3)$	94

$\omega(A) = 152$, $\mathsf{DXC}(A) = 120$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-4, 0, -1, 0, 0, 4)$	96
2	$(-3, 0, -3, 1, 1, 3)$	93
3	$(1, 4, -1, -1, -1, -3)$	94

$\omega(A) = 153$, $\mathsf{DXC}(A) = 121$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-3, -3, 1, 0, 2, 2)$	93
2	$(-4, -3, 0, 2, 2, 2)$	94
3	$(0, 1, 4, -2, -2, -2)$	95

$\omega(A) = 154$, $\mathsf{DXC}(A) = 122$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-3, 0, -2, 0, 0, 4)$	95
2	$(-1, -4, -2, 0, 2, 4)$	95
3	$(-4, -3, 1, 1, 1, 3)$	94
4	$(0, 1, 3, -1, -1, -3)$	93

$\omega(A) = 155$, $\mathsf{DXC}(A) = 123$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-5, 0, -2, 0, 2, 4)$	96

$\omega(A) = 156$, $\mathsf{DXC}(A) = 124$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-4, 0, -3, 0, 2, 4)$	97
3	$(5, 0, 1, -1, -3, -3)$	96

$\omega(A) = 157$, $\mathsf{DXC}(A) = 125$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
2	$(1, 1, 4, -1, -3, -3)$	95
3	$(-4, -4, 0, 1, 3, 3)$	96
4	$(2, 0, 4, -1, -3, -3)$	97

$\omega(A) = 158$, $\mathsf{DXC}(A) = 126$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-4, -3, 2, 0, 0, 4)$	97
2	$(-4, -4, 1, 0, 2, 4)$	96
5	$(1, 5, 0, -1, -1, -5)$	97
7	$(2, 5, 0, -2, -2, -4)$	97

$\omega(A) = 160$, $\mathsf{DXC}(A) = 128$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(1, 2, 5, 0, 0, 0)$	94
2	$(0, 1, 5, 0, 0, 2)$	93
3	$(0, 4, 2, 0, 0, 2)$	94
4	$(1, 4, 1, 0, 0, 2)$	93
5	$(0, 0, 4, 0, 2, 2)$	92
6	$(0, 1, 4, 1, 1, 1)$	93
7	$(2, 3, 6, -1, -1, -1)$	93
9	$(2, 5, -1, -1, -1, -5)$	98
11	$(1, 5, 1, -2, -2, -4)$	97

$\omega(A) = 161$, $\mathsf{DXC}(A) = 129$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(3, 0, 1, 0, 2, 2)$	93
3	$(0, 3, 0, 1, 1, 3)$	92
4	$(4, 7, 2, -1, -1, -3)$	92
6	$(1, 2, 4, -2, -2, -4)$	98

$\omega(A) = 162$, $\mathsf{DXC}(A) = 130$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(0, 3, 1, 0, 0, 4)$	92
2	$(-1, 1, 4, 0, 2, 2)$	93
3	$(2, 0, 0, 0, 2, 4)$	92
6	$(-1, 0, 4, 1, 1, 3)$	92
7	$(3, 4, 6, -1, -1, -3)$	93
9	$(1, 2, 5, -3, -3, -3)$	98

$\omega(A) = 163$, $\mathsf{DXC}(A) = 131$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-1, 3, 1, 1, 1, 3)$	94
2	$(3, 7, 3, -1, -1, -3)$	93
6	$(6, 1, 1, -1, -3, -5)$	96
7	$(-1, 0, 3, 2, 2, 2)$	94
8	$(3, 4, 7, -2, -2, -2)$	94

$\omega(A) = 164$, $\mathsf{DXC}(A) = 132$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-1, 0, 5, 0, 0, 4)$	93
3	$(-4, -1, -2, 0, 0, 6)$	99
4	$(-2, 0, 4, 0, 2, 4)$	92
5	$(-1, 3, 0, 0, 2, 4)$	92
9	$(2, -1, 0, 1, 3, 3)$	93
10	$(8, 3, 4, -1, -3, -3)$	92
12	$(2, 5, 1, -1, -3, -5)$	100

$\omega(A) = 165$, $\mathsf{DXC}(A) = 133$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(1, -1, 0, 0, 4, 4)$	93
4	$(-1, 2, 0, 1, 1, 5)$	92
5	$(5, 8, 2, -1, -1, -5)$	92
6	$(-2, 0, 3, 1, 3, 3)$	94
7	$(4, 4, 7, -1, -3, -3)$	92
8	$(-1, -1, 3, 1, 3, 3)$	90
9	$(5, 3, 7, -1, -3, -3)$	93
10	$(1, -1, -1, 1, 3, 5)$	90
11	$(9, 5, 3, -1, -3, -5)$	95
13	$(6, 1, 2, -2, -4, -4)$	97

$\omega(A) = 166$, $\mathsf{DXC}(A) = 134$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-1, 3, 2, 0, 0, 4)$	95
2	$(-2, 3, 1, 0, 2, 4)$	93
3	$(-1, -1, 4, 0, 2, 4)$	93
4	$(2, -1, 1, 0, 2, 4)$	94
6	$(1, -1, 0, 0, 2, 6)$	93
9	$(-2, 2, 0, 2, 2, 4)$	94
10	$(4, 8, 4, -2, -2, -4)$	92
11	$(-1, 2, -1, 2, 2, 4)$	90
12	$(5, 8, 3, -2, -2, -4)$	93
14	$(2, 2, 5, -2, -4, -4)$	100

$\omega(A) = 167$, $\mathsf{DXC}(A) = 135$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(5, -2, 0, 0, 2, 2)$	97
3	$(1, -2, 0, 1, 3, 5)$	94
4	$(9, 4, 4, -1, -3, -5)$	93
5	$(-2, -1, 3, 2, 2, 4)$	90
6	$(4, 5, 7, -2, -2, -4)$	93
8	$(2, 6, 1, -2, -2, -6)$	100
10	$(3, 6, 0, -2, -2, -6)$	98
13	$(1, -2, -1, 2, 4, 4)$	91
14	$(9, 4, 5, -2, -4, -4)$	95

$\omega(A) = 168$, $\mathsf{DXC}(A) = 136$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-3, -1, 5, 0, 0, 0)$	99
2	$(-1, 1, 7, 0, 0, 0)$	98
3	$(-4, -2, 5, 0, 0, 2)$	101
4	$(-4, 4, -1, 0, 0, 2)$	102
5	$(-2, 0, 7, 0, 0, 2)$	96
6	$(-2, 4, -3, 0, 0, 2)$	100
7	$(-2, 6, 1, 0, 0, 2)$	97
8	$(0, 6, -1, 0, 0, 2)$	96
9	$(-1, 2, 1, 0, 0, 6)$	93
10	$(-2, 2, 0, 0, 2, 6)$	92
11	$(0, -2, 0, 0, 4, 6)$	94
12	$(0, -1, -1, 0, 4, 6)$	93
13	$(-4, -2, 4, 1, 1, 1)$	101
14	$(-2, 0, 6, -1, -1, -1)$	98
15	$(-2, 0, 6, 1, 1, 1)$	96
16	$(0, 2, 8, -1, -1, -1)$	98
17	$(-2, -1, 4, 1, 1, 5)$	93
18	$(4, 5, 6, -1, -1, -5)$	93
21	$(0, -2, -1, 1, 5, 5)$	91
22	$(10, 4, 5, -1, -5, -5)$	96
23	$(-2, 1, -1, 2, 2, 6)$	90
24	$(6, 9, 3, -2, -2, -6)$	95
25	$(0, -2, -2, 2, 4, 6)$	88
26	$(10, 6, 4, -2, -4, -6)$	98
27	$(-2, -1, 2, 3, 3, 3)$	91
28	$(4, 5, 8, -3, -3, -3)$	93
30	$(3, 6, 1, -3, -3, -5)$	99

$\omega(A) = 169$, $\mathsf{DXC}(A) = 137$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(3, -4, -2, 0, 2, 2)$	101
2	$(-3, 0, 3, 0, 4, 4)$	92
3	$(-1, 5, -2, 1, 1, 3)$	98
4	$(3, 9, 0, -1, -1, -3)$	97
5	$(-2, 2, 1, 1, 1, 5)$	95
6	$(4, 8, 3, -1, -1, -5)$	94
7	$(-2, 1, 0, 1, 1, 7)$	93
8	$(6, 9, 2, -1, -1, -7)$	93
9	$(-2, 2, -1, 1, 3, 5)$	90
10	$(6, 8, 3, -1, -3, -5)$	93
11	$(0, -2, -1, 1, 3, 7)$	91
12	$(10, 6, 3, -1, -3, -7)$	96
14	$(2, 6, 2, -3, -3, -5)$	100
15	$(-2, 1, -2, 3, 3, 5)$	88
16	$(6, 9, 4, -3, -3, -5)$	96

$\omega(A) = 170$, $\mathsf{DXC}(A) = 138$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-1, 5, -1, 0, 0, 4)$	97
2	$(-2, 1, 1, 0, 0, 8)$	94
3	$(-5, -2, 4, 0, 2, 2)$	102
4	$(-1, -2, 6, 0, 2, 2)$	97
5	$(-2, 5, -2, 0, 2, 4)$	97
6	$(4, -1, -2, 0, 2, 4)$	98
7	$(1, -2, 1, 0, 2, 6)$	95
8	$(0, -2, 0, 0, 2, 8)$	94
9	$(-2, -1, 3, 0, 4, 4)$	91
11	$(-3, 5, 0, 1, 1, 3)$	98
12	$(1, 9, 2, -1, -1, -3)$	99
15	$(-3, -1, 3, 1, 3, 5)$	94
16	$(5, 5, 7, -1, -3, -5)$	92
17	$(-3, 2, 0, 1, 3, 5)$	93
18	$(5, 8, 4, -1, -3, -5)$	92
19	$(-2, -2, 2, 2, 4, 4)$	88
20	$(6, 4, 8, -2, -4, -4)$	94
21	$(-3, 1, -1, 3, 3, 5)$	92
22	$(5, 9, 5, -3, -3, -5)$	93

$\omega(A) = 171$, $\mathsf{DXC}(A) = 139$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(-3, -4, 4, 0, 2, 2)$	102
2	$(-3, 0, 6, 0, 2, 2)$	98
3	$(-5, 3, -2, 1, 1, 3)$	105
4	$(-1, 7, 0, -1, -1, -3)$	106
5	$(-2, 4, -2, 1, 1, 5)$	97
6	$(4, 10, 0, -1, -1, -5)$	99
9	$(4, -3, -1, 1, 3, 3)$	97
10	$(10, 1, 3, -1, -3, -3)$	99
11	$(-2, -2, 3, 1, 3, 5)$	90
12	$(6, 4, 7, -1, -3, -5)$	93
13	$(-3, 1, 0, 2, 2, 6)$	94
14	$(5, 9, 4, -2, -2, -6)$	96
15	$(-3, -1, 2, 2, 4, 4)$	96
16	$(5, 5, 8, -2, -4, -4)$	92
17	$(0, -3, -1, 2, 4, 6)$	92
18	$(10, 5, 5, -2, -4, -6)$	94
21	$(0, -3, -2, 3, 5, 5)$	89
22	$(10, 5, 6, -3, -5, -5)$	96

$\omega(A) = 172$, $\mathsf{DXC}(A) = 140$, $\mathrm{depth}(A) = 3$		
No.	$(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t)$	$\mathsf{SLP}^*(A)$
1	$(2, 3, 6, 0, 0, 0)$	98
2	$(1, 2, 6, 0, 0, 2)$	100
3	$(-3, 5, 1, 0, 0, 4)$	104
4	$(0, 1, 6, 0, 0, 4)$	98
6	$(-2, -1, 5, 0, 0, 6)$	94
7	$(-2, 2, 2, 0, 0, 6)$	96
8	$(-2, 4, -1, 0, 0, 6)$	99
9	$(-4, 5, 0, 0, 2, 4)$	103
10	$(4, -3, 0, 0, 2, 4)$	102
12	$(-2, -2, 4, 0, 2, 6)$	92
13	$(3, -2, -2, 0, 2, 6)$	101
14	$(-3, 1, 0, 0, 2, 8)$	92
15	$(2, 0, 1, 0, 4, 4)$	102
18	$(1, 0, 0, 0, 4, 6)$	103
19	$(-3, 3, -4, 1, 1, 3)$	103
20	$(1, 7, -2, -1, -1, -3)$	103
22	$(3, 6, 1, -1, -3, -7)$	103
23	$(0, -3, 0, 1, 3, 7)$	93
24	$(10, 5, 4, -1, -3, -7)$	96
26	$(3, 2, 5, -1, -5, -5)$	101
29	$(-3, -1, 5, 2, 2, 2)$	97
30	$(1, 3, 9, -2, -2, -2)$	100
32	$(3, 6, 2, -2, -4, -6)$	103
34	$(2, 3, 6, -4, -4, -4)$	102
35	$(-3, -2, 1, 4, 4, 4)$	89
36	$(5, 6, 9, -4, -4, -4)$	94