Constructing Low-latency Involutory MDS Matrices with Lightweight Circuits

Shun Li, Siwei Sun, Chaoyun Li, Zihao Wei, Lei Hu

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

DCS Center
DATA ASSURANCE & COMMUNICATIONS SECURITY
EST 1980

COSIC

FSE 2019 @ Paris, France

Outlines

1. Background and Motivation
2. Lightweight Involutory MDS matrices
3. Our Construction
4. Low-latency Involutory MDS Matrices
5. Main Results

Outline

1. Background and Motivation
2. Lightweight Involutory MDS matrices
3. Our Construction
4. Low-latency Involutory MDS Matrices
5. Main Results

Diffusion Matrices

The diffusion layers are typically realized with linear operations, expressed as matrices and spreading the internal dependencies as much as possible.

The diffusion property of a diffusion matrix is up to its branch number:

Definition

The branch number $\mathcal{B}_n(A)$ of $A \in \mathbf{M}_{nk}(\mathbb{F}_2)$ is defined as

$$\min_{x \in \mathbb{F}_{2^{nk}} \setminus \{0\}} \{\omega_n(x) + \omega_n(Ax)\}.$$

Diffusion Layer

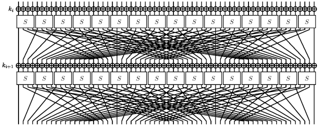
Regular lightweight primitive have following types of diffusion layer:

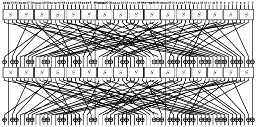
- Bit-level Permutations: PRESENT[A. Bogdanov et al., CHES'07], GIFT[S. Banik et al., CHES'17]

Diffusion Layer

Regular lightweight primitive have following types of diffusion layer:

- Bit-level Permutations: PRESENT[A. Bogdanov et al., CHES'07], GIFT[S. Banik et al., CHES'17]



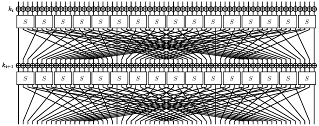


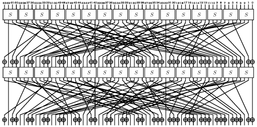
- Bitwise XORs and Rotations: Skinny[C. Beierle et al., CRYPTO'16], CRAFT[C. Beierle et al., FSE'19]

Diffusion Layer

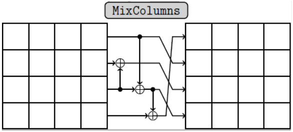
Regular lightweight primitive have following types of diffusion layer:

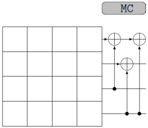
- Bit-level Permutations: PRESENT[A. Bogdanov et al., CHES'07], GIFT[S. Banik et al., CHES'17]





- Bitwise XORs and Rotations: Skinny[C. Beierle et al., CRYPTO'16], CRAFT[C. Beierle et al., FSE'19]





Diffusion Layer

- Maximal Distance Separable (MDS) Matrices: AES

Diffusion Layer

- Maximal Distance Separable (MDS) Matrices: AES
- Almost MDS Matrices: Midori[S. Banik et al., ASIACRYPT'15], QARMA[R. Avanzi, FSE'17]

$$M_{Midori} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

MDS Matrices

Definition

An invertible $nk \times nk$ binary matrix A is MDS over k n-bit words if and only if $\mathcal{B}_n(A) = k + 1$.

Example

The MDS matrix in AES:

$$\begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}$$

Wide Trail Strategy

The wide trail strategy is an approach used to design the round transformations that combine efficiency and resistance against differential and linear cryptanalysis. MDS matrices are in accordance with the strategy, have advantages as diffusion layers in iterative block cipher:

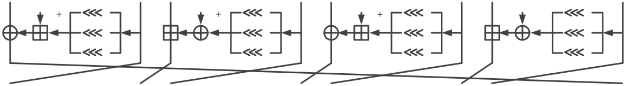
- Relatively small numbers of rounds, low-latency designs.
    - Skinny Bitwise XORs, 128-bit block size and 128-bit tweakey size, number of round is 40.
    - Midori Almost MDS, 128-bit block size and 128-bit key size, number of round is 20.
    - AES MDS, 128-bit block size and 128-bit key size, number of round is 10.
- simple and clear security proofs followed from AES.

Construction

- XOR and Rotations-based: Hight[D. Hong et al., CHES'06]

Construction

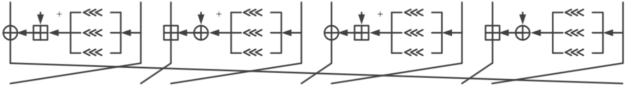
- XOR and Rotations-based: Hight[D. Hong et al., CHES'06]



- Iteration-based: PHOTON hash functions[J. Guo et al., CRYPTO'11]

Construction

- XOR and Rotations-based: Hight[D. Hong et al., CHES'06]



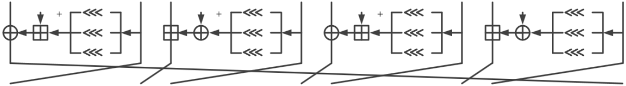
- Iteration-based: PHOTON hash functions[J. Guo et al., CRYPTO'11]

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 1 & 4 \end{pmatrix}^4 = \begin{pmatrix} 1 & 2 & 1 & 4 \\ 4 & 9 & 6 & 17 \\ 17 & 38 & 24 & 66 \\ 66 & 149 & 100 & 11 \end{pmatrix},$$

- Special-type-based: Circulant, Orthogonal, Hadamard, Toeplitz, Cauchy, Involutory matrices

Construction

- XOR and Rotations-based: Hight[D. Hong et al., CHES'06]



- Iteration-based: PHOTON hash functions[J. Guo et al., CRYPTO'11]

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 1 & 4 \end{pmatrix}^4 = \begin{pmatrix} 1 & 2 & 1 & 4 \\ 4 & 9 & 6 & 17 \\ 17 & 38 & 24 & 66 \\ 66 & 149 & 100 & 11 \end{pmatrix},$$

- Special-type-based: Circulant, Orthogonal, Hadamard, Toeplitz, Cauchy, Involutory matrices
- Circuit-search-based: S. Duval and G. Leurent, FSE'19

Involutory Matrices

Definition

An involutory matrix M is a square matrix that is its own inverse. That is, multiplication by matrix M is an involution if and only if $M^2 = I$.

Involutory matrices are preferable in term of hardware implementation, since the same circuit can be reused when the inverse is required.

Involutory MDS Matrices

The advantage of MDS and Involutory makes involutory matrices more preferable, Involutory MDS matrices applied in designs:

- Anubis, [P. Barreto et al., 2000]

Involutory MDS Matrices

The advantage of MDS and Involutory makes involutory matrices more preferable, Involutory MDS matrices applied in designs:

- Anubis, [P. Barreto et al., 2000]

$$\begin{bmatrix} 1 & 2 & 4 & 6 \\ 2 & 1 & 6 & 4 \\ 4 & 6 & 1 & 2 \\ 6 & 4 & 2 & 1 \end{bmatrix}$$

- ICEBERG, [F. Standaert et al., FSE'04]

Involutory MDS Matrices

The advantage of MDS and Involutory makes involutory matrices more preferable, Involutory MDS matrices applied in designs:

- Anubis, [P. Barreto et al., 2000]

$$\begin{bmatrix} 1 & 2 & 4 & 6 \\ 2 & 1 & 6 & 4 \\ 4 & 6 & 1 & 2 \\ 6 & 4 & 2 & 1 \end{bmatrix}$$

- ICEBERG, [F. Standaert et al., FSE'04]
- PRINCE, [J. Borghoff et al., ASIACRYPT'12]

Outline

1 Background and Motivation

2 Lightweight Involutory MDS matrices

3 Our Construction

4 Low-latency Involutory MDS Matrices

5 Main Results

Metrics

We estimate the hardware cost of a linear operation as the number of $\mathbb{F}_2 \times \mathbb{F}_2 \rightarrow \mathbb{F}_2$ `XOR2` gates required in its implementation.

It is NP-hard to obtain the minimum number of `XOR2` gates required:

Theorem (J. Boyar et al.)

For any field $\mathbb{F}$, SHORTEST LINEAR PROGRAM is NP-hard.

Metrics

Only metrics determining the *upper bounds* are available:

Direct XOR Count $A \in \mathbf{M}_{nk}(\mathbb{F}_2)$, $\mathrm{DXC}(A) = \omega(A) - nk$, corresponds to the number of 1s in A.

Global Optimization $A \in \mathbf{M}_{nk}(\mathbb{F}_2)$, its Global Optimization corresponds to a good linear straight-line program, which is based on certain SLP heuristic [BMP13], denoted as $\mathrm{SLP}(A)$.

Metrics

For multiplication by matrix

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

its DXC is 4, while its SLP XOR gates is 3 as following:

$$\begin{aligned} y_2 &= x_2 \\ y_3 &= x_3 \\ t_1 &= x_2 + x_3 \\ t_2 &= x_0 + t_1 \quad [y_0] \\ t_3 &= x_1 + t_1 \quad [y_1] \end{aligned}$$

Previous Work

Sarkar et al. find lightweight 16 × 16 involutory MDS matrix:

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

Its SLP `XOR2` gates is 42.

[Sumanta Sarkar and Habeeb Syed, Lightweight diffusion layer: Importance of toeplitz matrices. FSE'17]

Previous Work

Kranz et al. obtain lightweight 32 × 32 involutory MDS matrix by applying the subfield construction to the former:

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

Its SLP `XOR2` gates is 84.

[Thorsten Kranz, Gregor Leander, Ko Stoffelen, and Friedrich Wiemer. Shorter linear straight-line programs for MDS matrices. FSE'18]

Extracting the Structure

The former two matrices are form of following:

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

We generalize it and try to find lightweight involutory MDS matrices of the following form:

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

One Solution

To keep G involutory, that is $G^2 = I$,
i, j, k, l have to satisfy

$$\begin{cases} A^{2l} + A^{i+j} + A^k = O_8 \\ A^{i+k} + A^j = O_8 \end{cases}$$

Our goal is to find an involutory matrix G, such that $DXC(G)$ is small.
We get a solution which minimizes DXC(G):

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

Outline

1 Background and Motivation

2 Lightweight Involutory MDS matrices

3 Our Construction

4 Low-latency Involutory MDS Matrices

5 Main Results

New Form

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

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

where $A \in \mathrm{GL}(8, \mathbb{F}_2)$ is the companion matrix of $x^8 + x^2 + 1$:

$$\begin{pmatrix} 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \end{pmatrix}$$

Transformation

Without loss of generality, let

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

This operation is benefit for further transformation.

Transformation

With property of involutory, deduce it to

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

and

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

then number of parameters decrease from 12 to 6.

Exhaustive Search

Under the limitation, we inspect all $(\epsilon_{12}, \epsilon_{13}, \epsilon_{14}, r, s, t) \in \mathbb{Z}^6$ satisfying the following conditions:

$$\begin{cases} -8 \leq \epsilon_{1j} \leq 8 \quad \text{for} \quad 2 \leq j \leq 4 \\ 0 \leq r \leq s \leq t \leq 8 \\ 116 \leq DXC(M) \leq 140 \end{cases} .$$

We identify 5550 involutory MDS matrices whose Hamming weights are within the range from 148 to 172.

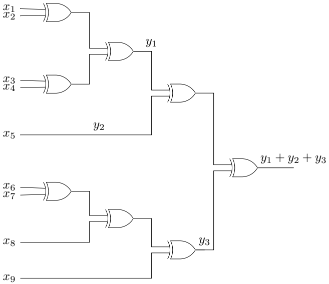
Outline

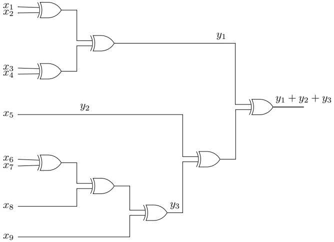
1. Background and Motivation
2. Lightweight Involutory MDS matrices
3. Our Construction
4. Low-latency Involutory MDS Matrices
5. Main Results

Depth of Implementation Circuits

Two implementations of the same summation $y1 + y2 + y3$ with different circuit depths, depth of left is 4 while right is 5.

$$\begin{cases} y_1 & = & x_1 + x_2 + x_3 + x_4 \\ y_2 & = & x_5 \\ y_3 & = & x_6 + x_7 + x_8 + x_9 \end{cases}$$





Depth of Implementation Circuit

Apply Boyar's SLP heuristic algorithm, all matrices we found can be implemented with circuit depth not less than 4.

As AES Mixcolumns can be implemented with depth 3, we wonder if our matrices can be implemented with depth ≤ 3.

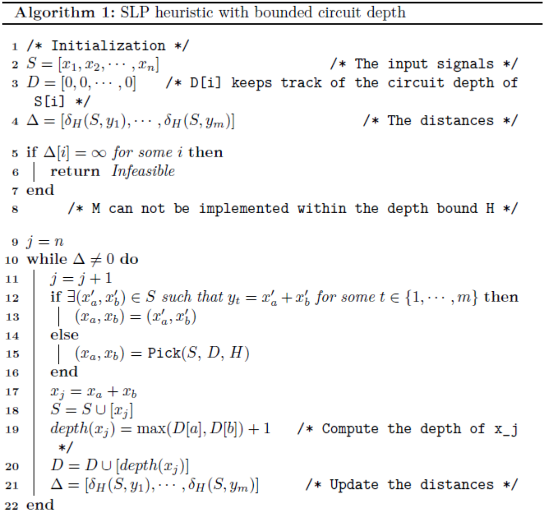
Theorem

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.*

We deduce it by counting the number of 1 in matrix.

Modify Boyar's Algorithm

We try to enhance Boyar's algorithm with depth awareness.



Algorithm 1: SLP heuristic with bounded circuit depth

```
/* Initialization */
$S = [x_1, x_2, \cdots, x_n]$                      /* The input signals */
$D = [0, 0, \cdots, 0]$   /* D[i] keeps track of the circuit depth of
  S[i] */
$\Delta = [\delta_H(S, y_1), \cdots, \delta_H(S, y_m)]$              /* The distances */

if $\Delta[i] = \infty$ for some $i$ then
    return Infeasible
end
      /* M can not be implemented within the depth bound H */

$j = n$
while $\Delta \neq 0$ do
    $j = j + 1$
    if $\exists (x'_a, x'_b) \in S$ such that $y_t = x'_a + x'_b$ for some $t \in \{1, \cdots, m\}$ then
        $(x_a, x_b) = (x'_a, x'_b)$
    else
        $(x_a, x_b) =$ Pick($S$, $D$, $H$)
    end
    $x_j = x_a + x_b$
    $S = S \cup [x_j]$
    $depth(x_j) = \max(D[a], D[b]) + 1$    /* Compute the depth of x_j
      */
    $D = D \cup [depth(x_j)]$
    $\Delta = [\delta_H(S, y_1), \cdots, \delta_H(S, y_m)]$       /* Update the distances */
end
```

Distance Function

Idea

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.

Some Examples

S	sequence of signals
f	linear predicate
$\delta_H(S, f)$	our new distance function
$\delta(S, f)$	Boyar's distance function

If $\delta_H(S, f) = k$, f not only can be obtained by k additions, but also have implementation of k additions within depth H.

Some Examples

Example

$S = [x_1, x_2, x_3, x_4, x_5]$, $f = x_1 + x_2 + x_3 + x_4 + x_5$.
Then $\delta(S, f) = \delta_3(S, f) = 4$ while $\delta_2(S, f) = \infty$,
f can be implemented as $x_6 = x_2 + x_3$, $x_7 = x_4 + x_5$, and
$x_8 = x_6 + x_7$, depth is 2.

$S = [x_1, x_2, x_3, x_4, x_5, x_6 = x_2 + x_4, x_7 = x_3 + x_6]$, $f = x_2 + x_3 + x_4 + x_5$.
Then $\delta(S, f) = 1$ while $\delta_2(S, f) = 2$,
f can be implemented as $x_5 + x_7$, depth is 3,
and f also can be implemented within depth 2 as $x_8 = x_3 + x_5$,
$x_9 = x_6 + x_8$.

Lightweight Involutory MDS Matrices with Depth 3

We apply new algorithm to all matrices we generated, and the lightest one with depth 3 is

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

A is still the companion matrix of $x^8 + x^2 + 1$.
Its `XOR2` gates is 88.

Outline

1 Background and Motivation

2 Lightweight Involutory MDS matrices

3 Our Construction

4 Low-latency Involutory MDS Matrices

5 Main Results

Our Work

MDS Matrix	**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	this
$M_{\mathtt{KLSW}} \in \mathbf{M}_4(\mathbf{M}_2(\mathbb{F}_{2^4}))$	✓	84	4	[KLSW17]
$H \in \mathbf{M}_4(\mathbf{M}_8(\mathbb{F}_2))$	✓	78	4	this
$Q \in \mathbf{M}_4(\mathbf{M}_8(\mathbb{F}_2))$	✓	88 (SLP*)	3	this

All of our code and results are available at `https://github.com/siweisun/involutory_mds`.

In conclusion:

1. Construct a large number of Involutory MDS matrices.
2. Apply Boyar's SLP heuristic algorithm to our matrices, we get lightest involutory MDS matrix
3. Modify Boyar's algorithm and apply new algorithm to all matrices, we get lightest involutory MDS matrix with depth of 3

In conclusion:

1. Construct a large number of Involutory MDS matrices.
2. Apply Boyar's SLP heuristic algorithm to our matrices, we get lightest involutory MDS matrix
3. Modify Boyar's algorithm and apply new algorithm to all matrices, we get lightest involutory MDS matrix with depth of 3

Thank you for your attention!