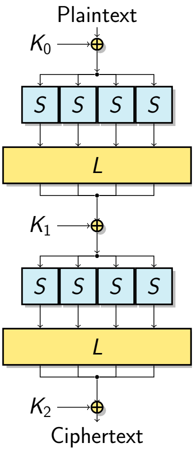
MDS Matrices with Lightweight Circuits

Sébastien Duval

Gaëtan Leurent

March 26, 2019

SPN Ciphers

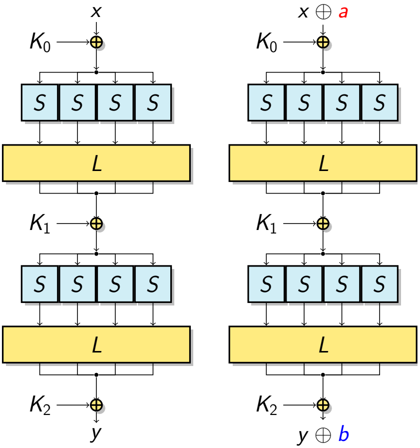


Shannon's criteria

1. Diffusion
   - Every bit of plaintext and key must affect every bit of the output
   - We usually use linear functions
2. Confusion
   - Relation between plaintext and ciphertext must be intractable
   - Requires non-linear operations
   - Often implemented with tables: S-Boxes

Example: Rijndael/AES [Daemen Rijmen 1998]

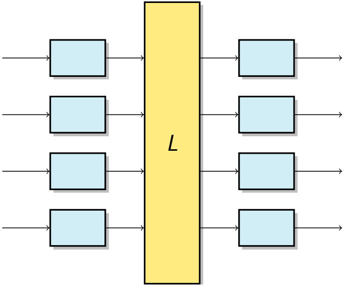
Block Cipher Security Analysis



Differential Attacks [Biham Shamir 91]

- Attacker exploits (a,b) such that
$$E_K(x) \oplus E_K(x \oplus a) = b$$
with high probability
- Maximum of the probability over all (a, b) bounded by
$$\left(\frac{\delta(S)}{2^n}\right)^{\mathcal{B}_d(L)-1}$$

MDS Matrices



L linear permutation on k words of n bits.

Differential Branch Number

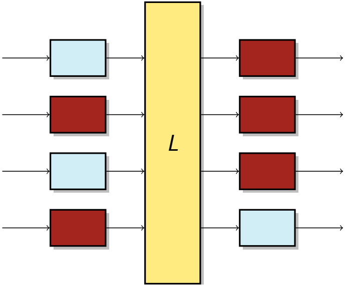
$$\mathcal{B}_d(L) = \min_{x \neq 0}\{w(x) + w(L(x))\}$$

where $w(x)$ is the number of non-zero n-bits words in x.

Linear Branch Number

$$\mathcal{B}_l(L) = \min_{x \neq 0}\{w(x) + w(L^\top(x))\}$$

MDS Matrices



L linear permutation on k words of n bits.

Differential Branch Number

$$\mathcal{B}_d(L) = \min_{x \neq 0}\{w(x) + w(L(x))\}$$

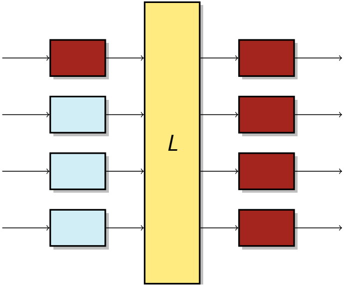
where $w(x)$ is the number of non-zero n-bits words in x.

Linear Branch Number

$$\mathcal{B}_l(L) = \min_{x \neq 0}\{w(x) + w(L^\top(x))\}$$

Maximum branch number : $k+1$
Equivalent to MDS codes.

MDS Matrices



L linear permutation on k words of n bits.

Differential Branch Number

$$\mathcal{B}_d(L) = \min_{x \neq 0}\{w(x) + w(L(x))\}$$

where $w(x)$ is the number of non-zero n-bits words in x.

Linear Branch Number

$$\mathcal{B}_l(L) = \min_{x \neq 0}\{w(x) + w(L^\top(x))\}$$

Maximum branch number : $k+1$
Equivalent to MDS codes.

Matrices and Characterisation

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

AES MixColumns

Usually on finite fields:
x a primitive element of $\mathbb{F}_2^n$
Coeffs. $\in \mathbb{F}_2[x]/P$, with P a primitive polynomial
$2 \leftrightarrow x$
$3 \leftrightarrow x + 1$

Characterisation

L is MDS iff its minors are non-zero

Previous Works

Recursive Matrices [Guo *et al.* 2011]

A lightweight matrix
A^i MDS
Implement A, then iterate A i times.

Optimizing Coefficients

- Structured matrices: restrict to a small subspace with many MDS matrices
- More general than finite fields: inputs are binary vectors, matrix coeffs. are $n \times n$ matrices.
  $\Rightarrow$ less costly operations than multiplication in a finite field

Cost Evaluation

"Real cost"

Number of operations of the best implementation.

Xor count (naive cost)

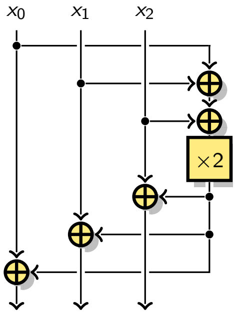
Hamming weight of the binary matrix. Cannot reuse intermediate values.

Intermediate values

- *Local optimisation*: LIGHTER [Jean *et al.* 2017]
  cost of matrix multiplication = number of XORs + cost of the mult. by each coefficent.
- *Global optimisation*:
  - Hardware synthesis: straight line programs [Kranz *et al.* 2018]. Heuristics to implement binary matrices.
  - Our approach: Number of operations of the best implementation using operations on words.

Metrics Comparison

$$\begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & 2 \\ 2 & 2 & 3 \end{bmatrix}$$



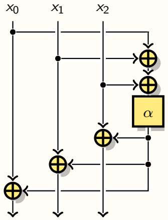
Xor Count: $\begin{cases} 6 \text{ mult. by } 2 \\ 3 \text{ mult. by } 3 \\ 6 \text{ XORS} \end{cases}$

Our approach: $\begin{cases} 1 \text{ mult. by } 2 \\ 5 \text{ XORS} \end{cases}$

Formal Matrices

Formal matrices

- Optimise in 2 steps:
  1. Find $M(\alpha)$ for α an undefined linear mapping.
  2. Instantiate with the best choice of α
- Not necessarily a finite field.
- Then coeffs. are polynomials in α.

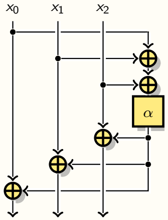


$$\begin{bmatrix} \alpha+1 & \alpha & \alpha \\ \alpha & \alpha+1 & \alpha \\ \alpha & \alpha & \alpha+1 \end{bmatrix}$$

Formal Matrices

Formal matrices

- Optimise in 2 steps:
  1. Find $M(\alpha)$ for α an undefined linear mapping.
  2. Instantiate with the best choice of α
- Not necessarily a finite field.
- Then coeffs. are polynomials in α.



$$\begin{bmatrix} \alpha+1 & \alpha & \alpha \\ \alpha & \alpha+1 & \alpha \\ \alpha & \alpha & \alpha+1 \end{bmatrix}$$

Characterisation of formally MDS matrices

- Objective: find $M(\alpha)$ *s.t.* $\exists A$, $M(A)$ MDS.
- If a minor of $M(\alpha)$ is null, then impossible.
- Otherwise, there always exists an A.

Characterisation possible on $M(\alpha)$.

Search Space

Search over circuits

Search Space

Operations:

- word-wise XOR
- α (generalization of a multiplication)
- Copy

Note: Only word-wise operations.

r registers:

one register per word (3 for 3×3)

$+$ (at least) one more register $\rightarrow$ more complex operations

Implementation: Main Idea

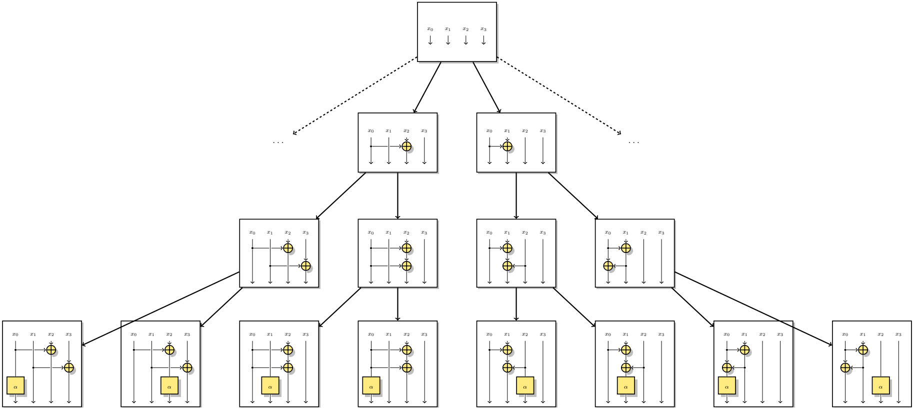
Tree-based Dijkstra search

- Node = matrix = sequence of operations
- Lightest circuit = shortest path to MDS matrix
- When we spawn a node, we test if it is MDS

Search results

- $k = 3$ fast (seconds)
- $k = 4$ long (hours)
- $k = 5$ out of reach
- Collection of MDS matrices with trade-off between cost and depth (latency).

Scheme of the Search



Optimization: A^*

A^*

Idea of A^*

- Guided Dijkstra
- weight $=$ weight from origin $+$ estimated weight to objective

Optimization: A^*

A^*

Idea of A^*

- Guided Dijkstra
- weight $=$ weight from origin $+$ estimated weight to objective

Our estimate:

Optimization: A^*

A^*

Idea of A^*

- Guided Dijkstra
- weight = weight from origin + estimated weight to objective

Our estimate:

- Heuristic
- How far from MDS ?

Optimization: A^*

A^*

Idea of A^*

- Guided Dijkstra
- weight $=$ weight from origin $+$ estimated weight to objective

Our estimate:

- Heuristic
- How far from MDS ?
- Column with a 0: cannot be part of MDS matrix

Optimization: A^*

A^*

Idea of A^*

- Guided Dijkstra
- weight = weight from origin + estimated weight to objective

Our estimate:

- Heuristic
- How far from MDS ?
- Column with a 0: cannot be part of MDS matrix
- Linearly dependent columns: not part of MDS matrix

Optimization: A^*

A^*

Idea of A^*

- Guided Dijkstra
- weight = weight from origin + estimated weight to objective

Our estimate:

- Heuristic
- How far from MDS ?
- Column with a 0: cannot be part of MDS matrix
- Linearly dependent columns: not part of MDS matrix
- Estimate: m = rank of the matrix (without columns containing 0)
- Need at least $k - m$ word-wise XORs to MDS

Result: much faster

Methodology of the Instantiation

The Idea

1. Input: Formal matrix $M(\alpha)$ MDS
2. Output: $M(A)$ MDS, with A a linear mapping (the lightest we can find)

Characterisation of MDS Instantiations

MDS Test

- Intuitive approach:
  - Choose A a linear mapping
  - Evaluate $M(A)$
  - See if all minors are non-singular

Characterisation of MDS Instantiations

MDS Test

- Intuitive approach:
  - Choose A a linear mapping
  - Evaluate $M(A)$
  - See if all minors are non-singular
- We can start by computing the minors:
  - Let I, J subsets of the lines and columns
  - Define $m_{I,J} = \det_{\mathbb{F}_2[\alpha]}(M_{|I,J})$
  - $M(A)$ is MDS iff all $m_{I,J}(A)$ are non-singular

Characterisation of MDS Instantiations

MDS Test

- Intuitive approach:
  - Choose A a linear mapping
  - Evaluate $M(A)$
  - See if all minors are non-singular
- We can start by computing the minors:
  - Let I, J subsets of the lines and columns
  - Define $m_{I,J} = \det_{\mathbb{F}_2[\alpha]}(M_{|I,J})$
  - $M(A)$ is MDS iff all $m_{I,J}(A)$ are non-singular
- With the minimal polynomial
  - Let μ_A the minimal polynomial of A
  - $M(A)$ is MDS iff $\forall(I, J)$, $\gcd(\mu_A, m_{I,J}) = 1$

Multiplications in a Finite Field

We want A s.t. $\forall (I, J)$, $\gcd(\mu_A, m_{I,J}) = 1$

Multiplications in a Finite Field

We want A s.t. $\forall(I, J)$, $\gcd(\mu_A, m_{I,J}) = 1$

Easy Way to Instantiate: Multiplications

- $d > \max_{I,J}\{deg(m_{I,J})\}$

Multiplications in a Finite Field

We want A s.t. $\forall(I, J)$, $\gcd(\mu_A, m_{I,J}) = 1$

Easy Way to Instantiate: Multiplications

- $d > \max_{I,J}\{deg(m_{I,J})\}$
- Choose π an irreducible polynomial of degree d

Multiplications in a Finite Field

We want A s.t. $\forall(I, J)$, $\gcd(\mu_A, m_{I,J}) = 1$

Easy Way to Instantiate: Multiplications

- $d > \max_{I,J}\{deg(m_{I,J})\}$
- Choose π an irreducible polynomial of degree d
- π is relatively prime with all $m_{I,J}$

Multiplications in a Finite Field

We want A s.t. $\forall (I, J)$, $\gcd(\mu_A, m_{I,J}) = 1$

Easy Way to Instantiate: Multiplications

- $d > \max_{I,J}\{deg(m_{I,J})\}$
- Choose π an irreducible polynomial of degree d
- π is relatively prime with all $m_{I,J}$
- Take $A =$ companion matrix of π

Multiplications in a Finite Field

We want A s.t. $\forall(I, J)$, $\gcd(\mu_A, m_{I,J}) = 1$

Easy Way to Instantiate: Multiplications

- $d > \max_{I,J}\{deg(m_{I,J})\}$
- Choose π an irreducible polynomial of degree d
- π is relatively prime with all $m_{I,J}$
- Take $A =$ companion matrix of π
- A corresponds to a finite field multiplication

Multiplications in a Finite Field

We want A s.t. $\forall(I, J)$, $\gcd(\mu_A, m_{I,J}) = 1$

Easy Way to Instantiate: Multiplications

- $d > \max_{I,J}\{deg(m_{I,J})\}$
- Choose π an irreducible polynomial of degree d
- π is relatively prime with all $m_{I,J}$
- Take $A =$ companion matrix of π
- A corresponds to a finite field multiplication

Low Cost Instantiation

- Pick π with few coefficients: a trinomial requires 1 rotation + 1 binary xor

Concrete Choices of A

We need to fix the size

Branches of size 4 bits ($\mathbb{F}_2^4$)

$$A_4 = \begin{bmatrix} . & 1 & . & . \\ . & . & 1 & . \\ . & . & . & 1 \\ 1 & 1 & . & . \end{bmatrix}$$

(companion matrix of $X^4 + X + 1$ (irreducible))

$$A_4^{-1} = \begin{bmatrix} 1 & . & . & 1 \\ 1 & . & . & . \\ . & 1 & . & . \\ . & . & 1 & . \end{bmatrix}$$

(minimal polynomial is $X^4 + X^3 + 1$)

Branches of size 8 bits ($\mathbb{F}_2^8$)

$$A_8 = \begin{bmatrix} . & 1 & . & . & . & . & . & . \\ . & . & 1 & . & . & . & . & . \\ . & . & . & 1 & . & . & . & . \\ . & . & . & . & 1 & . & . & . \\ . & . & . & . & . & 1 & . & . \\ . & . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & . & 1 \\ 1 & . & 1 & . & . & . & . & . \end{bmatrix}$$

(companion matrix of
$X^8 + X^2 + 1 = (X^4 + X + 1)^2$)

$$A_8^{-1} = \begin{bmatrix} . & 1 & . & . & . & . & . & 1 \\ 1 & . & . & . & . & . & . & . \\ . & 1 & . & . & . & . & . & . \\ . & . & 1 & . & . & . & . & . \\ . & . & . & 1 & . & . & . & . \\ . & . & . & . & 1 & . & . & . \\ . & . & . & . & . & 1 & . & . \\ . & . & . & . & . & . & 1 & . \end{bmatrix}$$

(minimal polynomial is $X^8 + X^6 + 1$)

Comparison With Existing MDS Matrices

Size	Ring	Matrix	Cost: Naive	Cost: Best	Cost: Depth	Ref
$M_4(M_8(\mathbb{F}_2))$	$GL(8,\mathbb{F}_2)$	Circulant	106			(Li Wang 2016)
	$GL(8,\mathbb{F}_2)$	Hadamard		72	6	(Kranz *et al.* 2018)
	$\mathbb{F}_2[\alpha]$	$M_{4,6}^{8,3}$		67	5	$\alpha = A_8$ or A_8^{-1}
	$\mathbb{F}_2[\alpha]$	$M_{4,4}^{8,4}$		69	4	$\alpha = A_8$
	$\mathbb{F}_2[\alpha]$	$M_{4,3}^{9,5}$		77	3	$\alpha = A_8$ or A_8^{-1}
$M_4(M_4(\mathbb{F}_2))$	$GF(2^4)$	$M_{4,n,4}$	58	58	3	(Jean Peyrin Sim 2017)
	$GF(2^4)$	Toeplitz	58	58	3	(Sarkar Syed 2016)
	$GL(4,\mathbb{F}_2)$	Subfield		36	6	(Kranz *et al.* 2018)
	$\mathbb{F}_2[\alpha]$	$M_{4,6}^{8,3}$		35	5	$\alpha = A_4$ or A_4^{-1}
	$\mathbb{F}_2[\alpha]$	$M_{4,4}^{8,4}$		37	4	$\alpha = A_4$
	$\mathbb{F}_2[\alpha]$	$M_{4,3}^{9,5}$		41	3	$\alpha = A_4$ or A_4^{-1}