Radboud University

Column Parity Mixers

Joan Daemen[1,2]

partially based on joint work with Ko Stoffelen[1] and Gilles Van Assche[2]

Mathematical Methods for Cryptography Workshop 2017

[1]Radboud University [2]STMicroelectronics

Outline

Some context

Column parity mixers

Algebraic properties

Diffusion properties

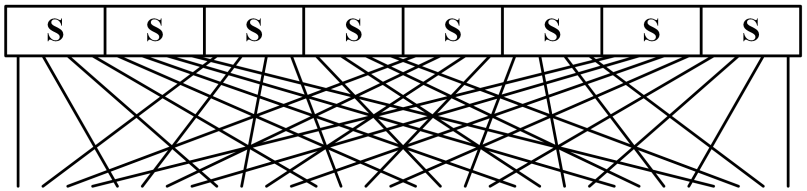
Showcase 1: a permutation using a CPM (with Ko Stoffelen)

Showcase 2: efficient inverse of θ of Keccak-p (with Gilles Van Assche)

Outline

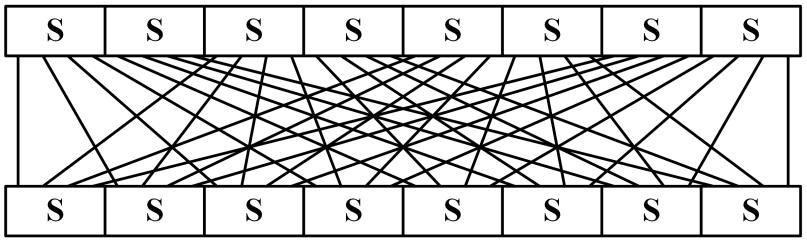
Product cipher [Claude Shannon, 1949] and SPN

Confusion and Diffusion



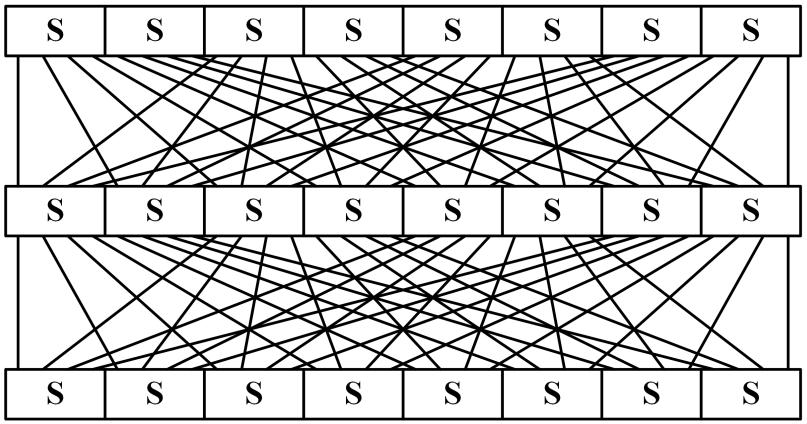
Product cipher [Claude Shannon, 1949] and SPN

Confusion and Diffusion



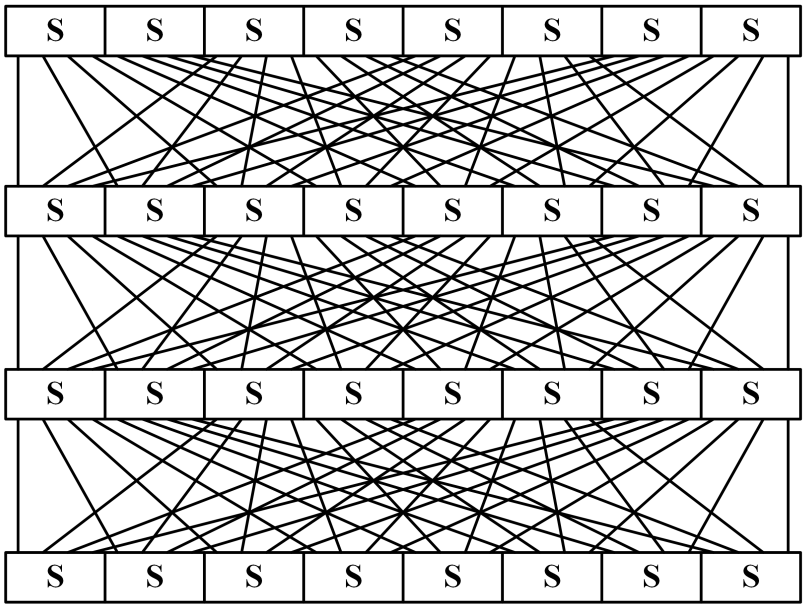
Product cipher [Claude Shannon, 1949] and SPN

Confusion and Diffusion



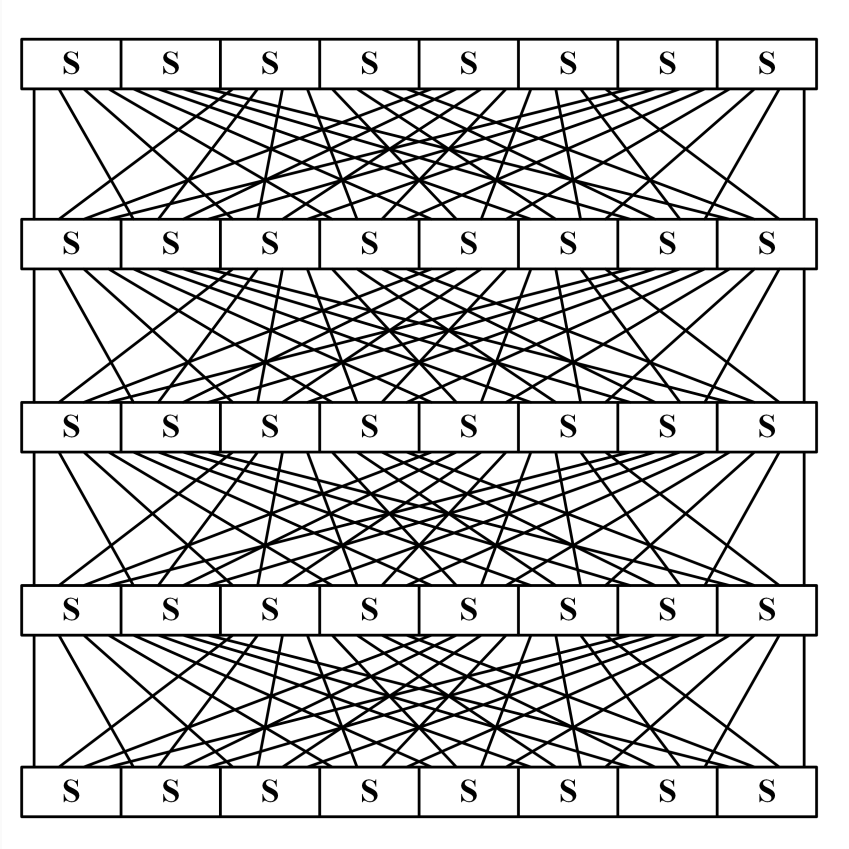
Product cipher [Claude Shannon, 1949] and SPN

Confusion and Diffusion



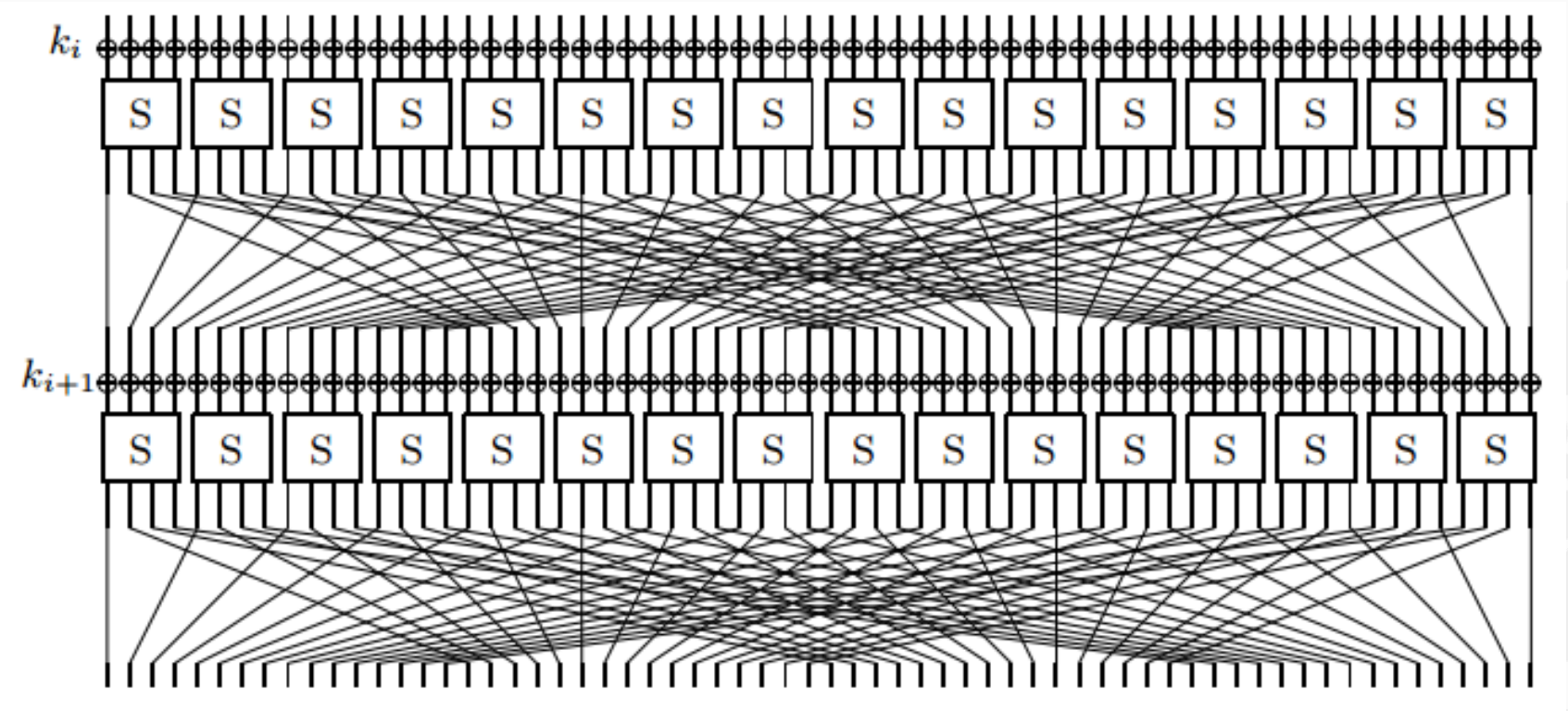
Product cipher [Claude Shannon, 1949] and SPN

Confusion and Diffusion

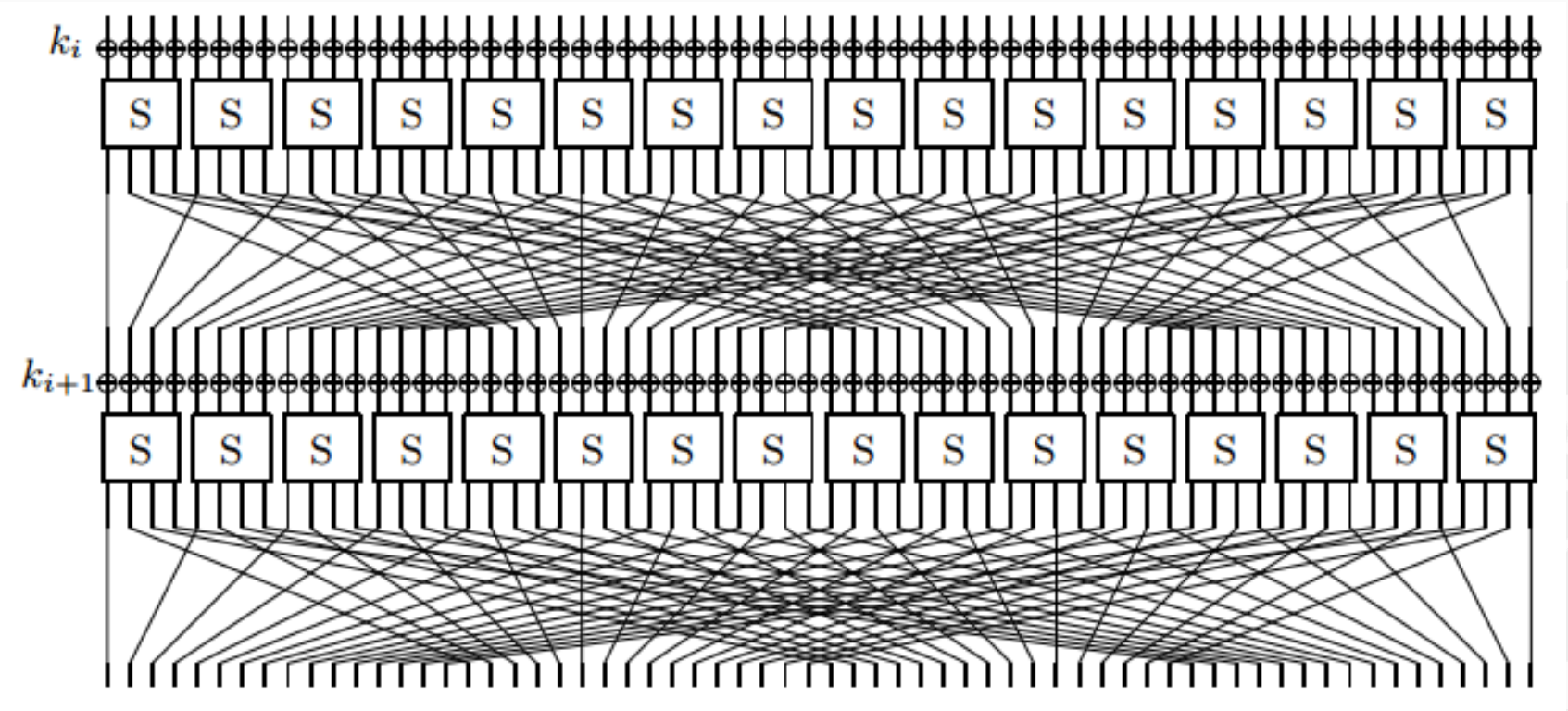


Substitution-Permutation Network

Modern example of SPN

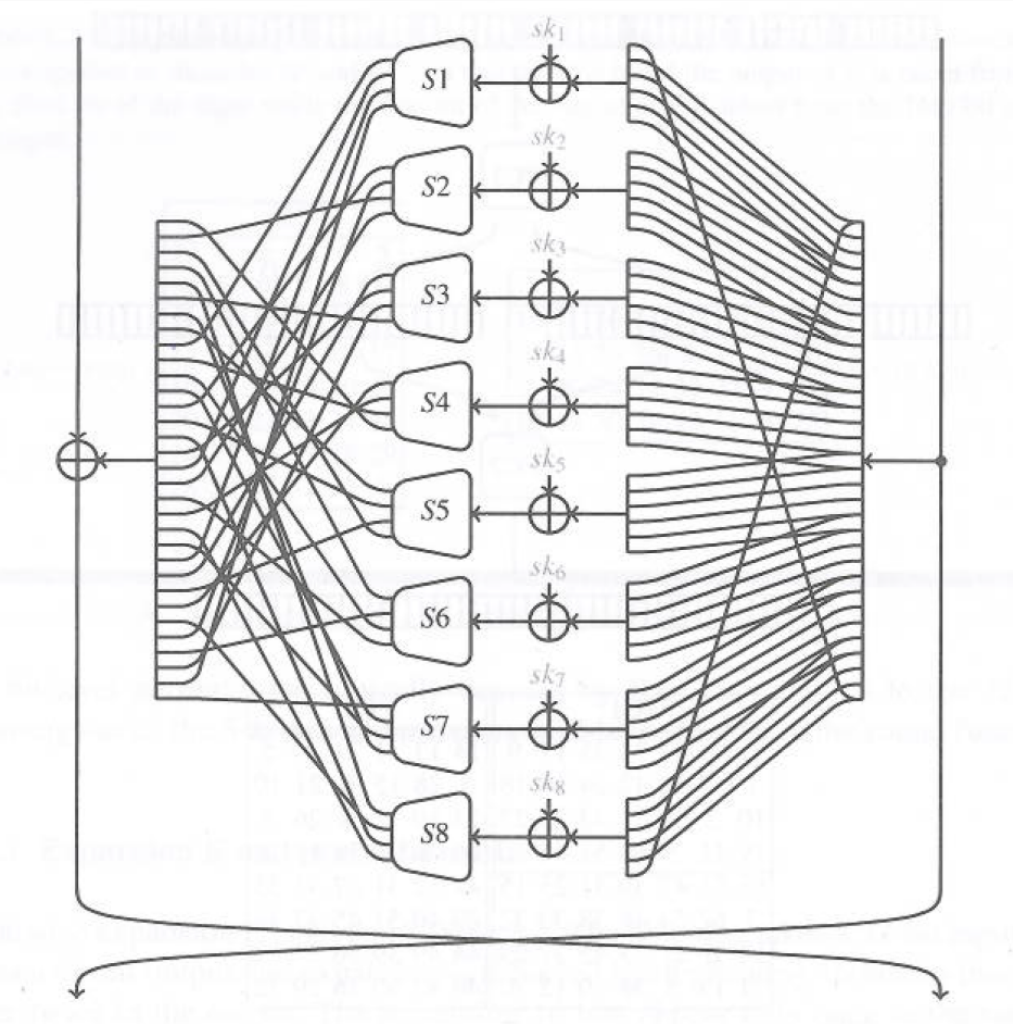


Modern example of SPN

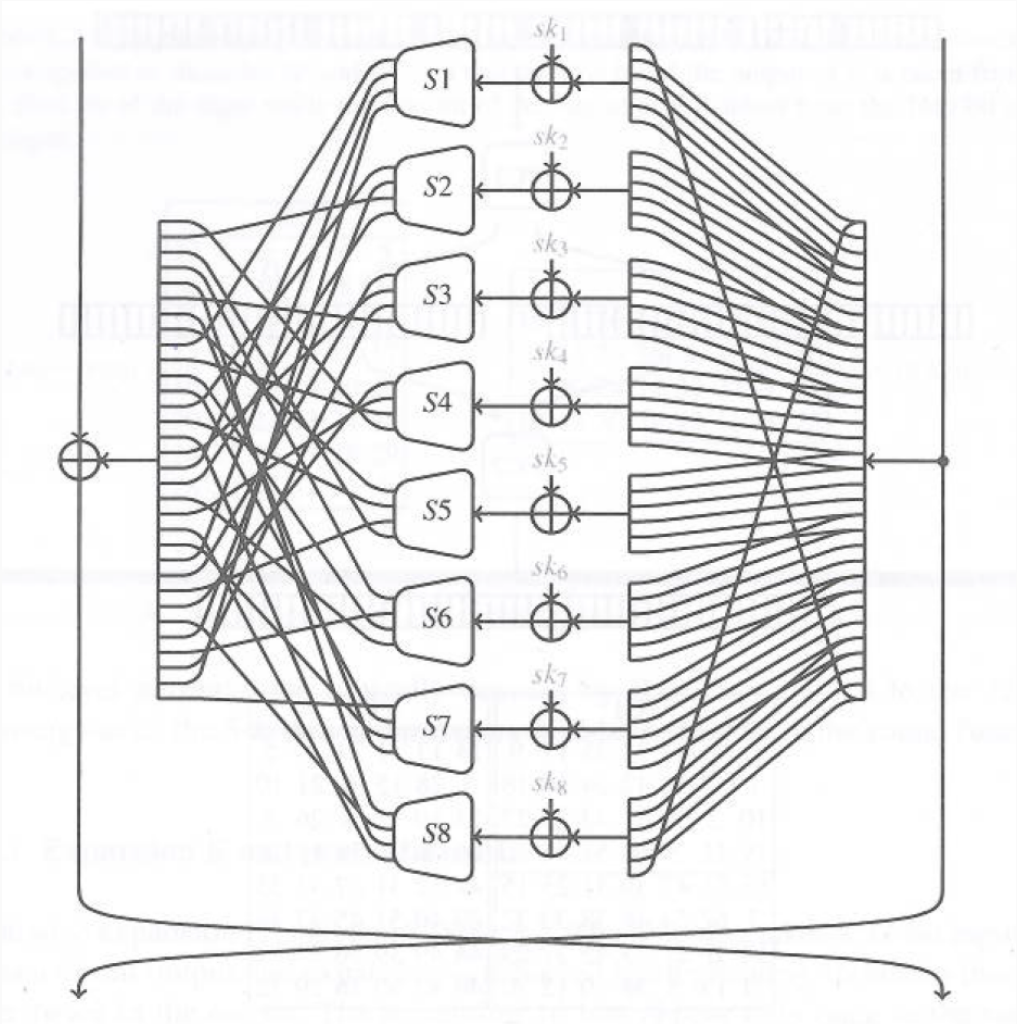


Present [Bogdanov, Knudsen, Leander, Paar, Poschmann, Robshaw, Seurin, Vikkelsoe, CHES '07]

Round function of another famous block cipher

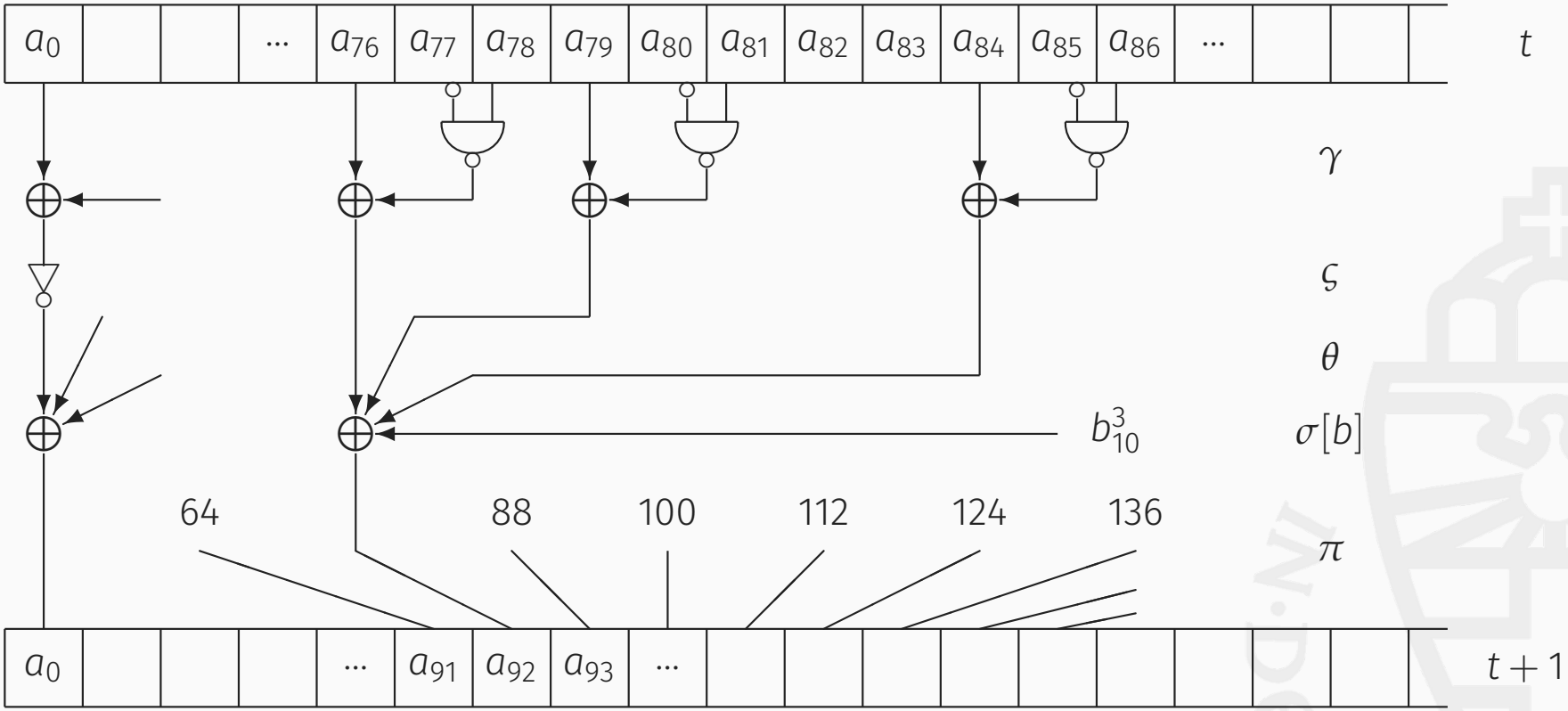


Round function of another famous block cipher

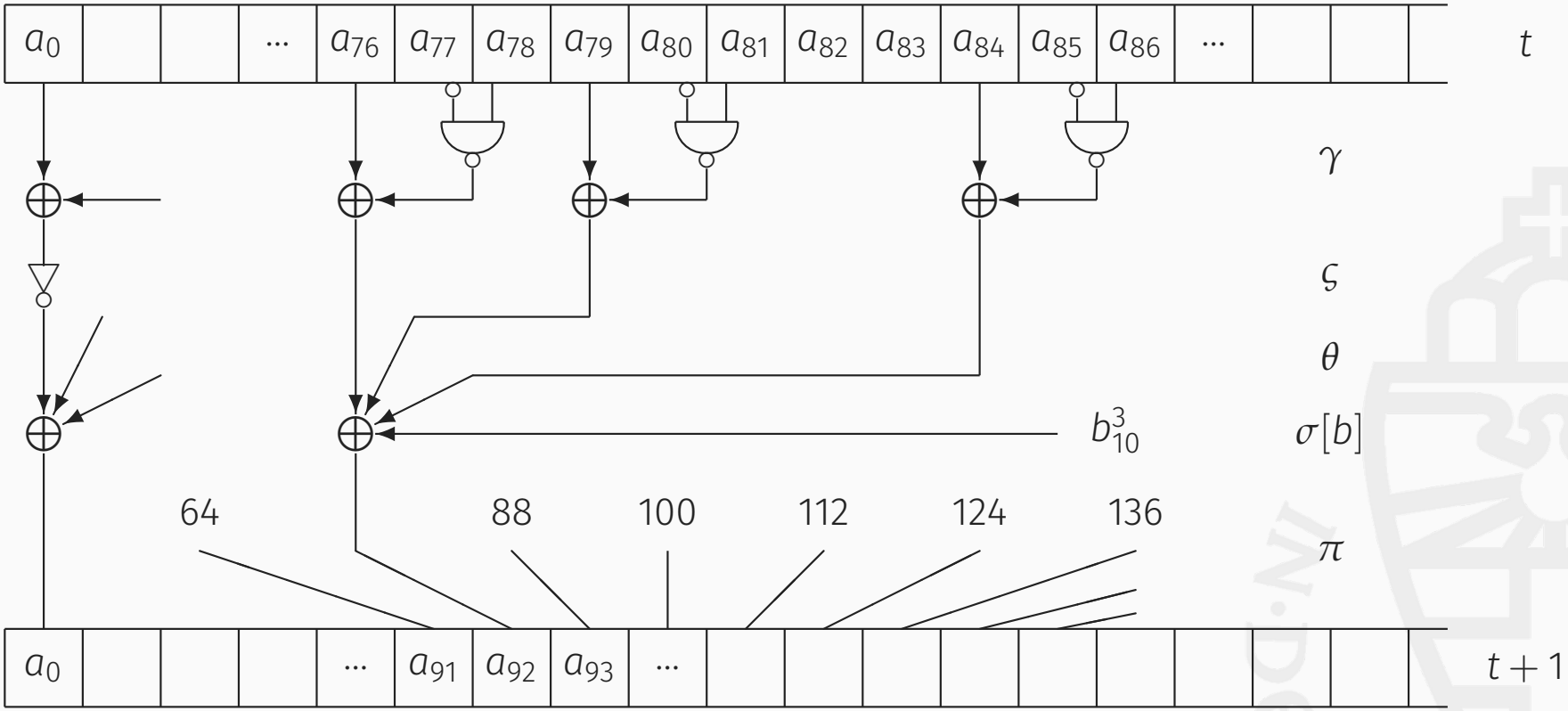


DES [FIPS 46, '77]

A round function with a dedicated mixing layer

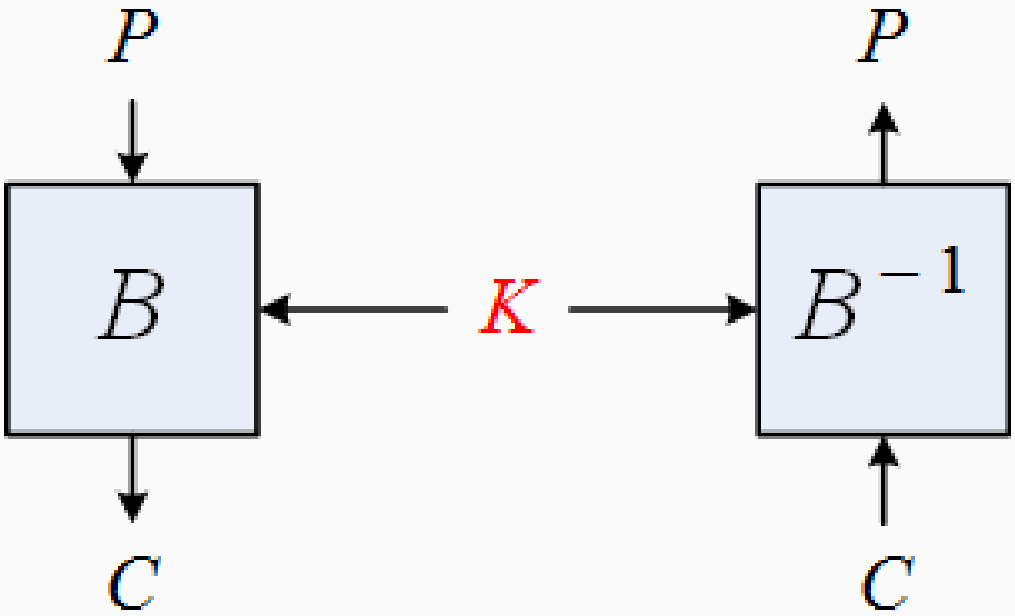


A round function with a dedicated mixing layer



Subterranean [JDA '92]

Block cipher functionality



The block cipher 3-Way [JDA, '93]

- Alternate round key addition with round function
- Round function with 4 steps:

  θ mixing
  π_1 first transposition
  γ nonlinearity (variant of χ on 3-bit units)
  π_2 second transposition

- Goal is that inverse and forward can use same hardware

The mixing layer θ in 3-Way

$$\mathrm{M}_\theta = \begin{pmatrix}
1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & 1 & 1 \\
1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & 1 \\
1 & 1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 \\
1 & 1 & 1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot \\
\cdot & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 \\
1 & \cdot & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 \\
1 & 1 & \cdot & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot \\
\cdot & 1 & 1 & \cdot & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot & \cdot \\
\cdot & \cdot & 1 & 1 & \cdot & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot \\
\cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & 1 & 1 & 1 & \cdot & 1 \\
1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & 1 & 1 & 1 & \cdot \\
\cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & 1 & 1 & 1
\end{pmatrix}$$

The mixing layer θ in 3-Way

$$\mathrm{M}_\theta = \begin{pmatrix}
1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & 1 & 1 \\
1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & 1 \\
1 & 1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 \\
1 & 1 & 1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot \\
\cdot & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 \\
1 & \cdot & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 \\
1 & 1 & \cdot & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot & \cdot & \cdot \\
\cdot & 1 & 1 & \cdot & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot & \cdot \\
\cdot & \cdot & 1 & 1 & \cdot & 1 & 1 & 1 & 1 & \cdot & 1 & \cdot \\
\cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & 1 & 1 & 1 & \cdot & 1 \\
1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & 1 & 1 & 1 & \cdot \\
\cdot & 1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & 1 & 1 & 1
\end{pmatrix}$$

Turns out to be orthogonal: $\mathrm{M}_\theta^{-1} = \mathrm{M}_\theta{}^\mathsf{T}$

Diffusion properties of 3-Way mixing layer

$\lvert x\rvert \setminus \lvert M_\theta x\rvert$	1	2	3	4	5	6	7	8	9	10	11
1	-	-	-	-	-	-	12	-	-	-	-
2	-	-	-	-	-	60	-	-	-	6	-
3	-	-	-	-	180	-	-	-	40	-	-
4	-	-	-	255	-	-	-	240	-	-	-
5	-	-	180	-	-	-	600	-	-	-	12
6	-	60	-	-	-	804	-	-	-	60	-
7	12	-	-	-	600	-	-	-	180	-	-
8	-	-	-	240	-	-	-	255	-	-	-
9	-	-	40	-	-	-	180	-	-	-	-
10	-	6	-	-	-	60	-	-	-	-	-
11	-	-	-	-	12	-	-	-	-	-	-

Diffusion properties of 3-Way mixing layer

$\lvert x\rvert \setminus \lvert M_\theta x\rvert$	1	2	3	4	5	6	7	8	9	10	11
1	-	-	-	-	-	-	12	-	-	-	-
2	-	-	-	-	-	60	-	-	-	6	-
3	-	-	-	-	180	-	-	-	40	-	-
4	-	-	-	255	-	-	-	240	-	-	-
5	-	-	180	-	-	-	600	-	-	-	12
6	-	60	-	-	-	804	-	-	-	60	-
7	12	-	-	-	600	-	-	-	180	-	-
8	-	-	-	240	-	-	-	255	-	-	-
9	-	-	40	-	-	-	180	-	-	-	-
10	-	6	-	-	-	60	-	-	-	-	-
11	-	-	-	-	12	-	-	-	-	-	-

- $\min(|x| + |M_\theta x|)$: *branch number* $\mathcal{B}$ [JDA, 1993]

Diffusion properties of 3-Way mixing layer

$\lvert x\rvert \backslash \lvert M_\theta x\rvert$	1	2	3	4	5	6	7	8	9	10	11
1	-	-	-	-	-	-	12	-	-	-	-
2	-	-	-	-	-	60	-	-	-	6	-
3	-	-	-	-	180	-	-	-	40	-	-
4	-	-	-	255	-	-	-	240	-	-	-
5	-	-	180	-	-	-	600	-	-	-	12
6	-	60	-	-	-	804	-	-	-	60	-
7	12	-	-	-	600	-	-	-	180	-	-
8	-	-	-	240	-	-	-	255	-	-	-
9	-	-	40	-	-	-	180	-	-	-	-
10	-	6	-	-	-	60	-	-	-	-	-
11	-	-	-	-	12	-	-	-	-	-	-

- $\min(|x| + |M_\theta x|)$: *branch number* $\mathcal{B}$ [JDA, 1993]
- link with error-correcting codes [JDA, 1995]:
  - $(x, \mathrm{M}_\theta x)$ as codeword for x: minimum distance $d = \mathcal{B}$

Diffusion properties of 3-Way mixing layer

$\lvert x\rvert \backslash \lvert M_\theta x\rvert$	1	2	3	4	5	6	7	8	9	10	11
1	-	-	-	-	-	-	12	-	-	-	-
2	-	-	-	-	-	60	-	-	-	6	-
3	-	-	-	-	180	-	-	-	40	-	-
4	-	-	-	255	-	-	-	240	-	-	-
5	-	-	180	-	-	-	600	-	-	-	12
6	-	60	-	-	-	804	-	-	-	60	-
7	12	-	-	-	600	-	-	-	180	-	-
8	-	-	-	240	-	-	-	255	-	-	-
9	-	-	40	-	-	-	180	-	-	-	-
10	-	6	-	-	-	60	-	-	-	-	-
11	-	-	-	-	12	-	-	-	-	-	-

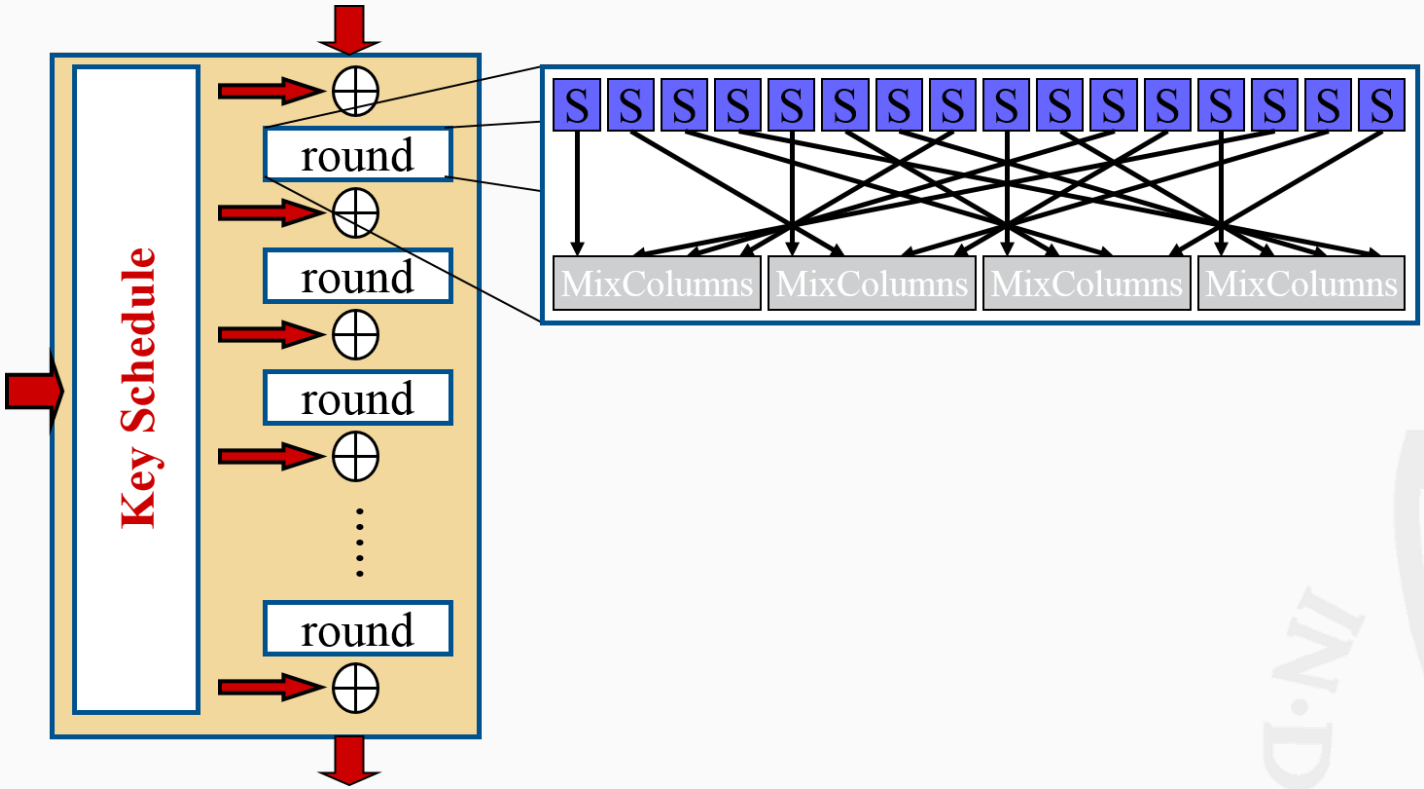
- $\min(|x| + |M_\theta x|)$: *branch number* $\mathcal{B}$ [JDA, 1993]
- link with error-correcting codes [JDA, 1995]:
  - $(x, \mathrm{M}_\theta x)$ as codeword for x: minimum distance $d = \mathcal{B}$
  - $[24, 12, 8]$-code:

Diffusion properties of 3-Way mixing layer

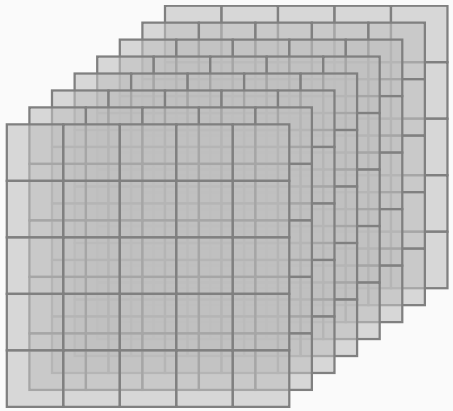
$\lvert x\rvert \backslash \lvert M_\theta x\rvert$	1	2	3	4	5	6	7	8	9	10	11
1	-	-	-	-	-	-	12	-	-	-	-
2	-	-	-	-	-	60	-	-	-	6	-
3	-	-	-	-	180	-	-	-	40	-	-
4	-	-	-	255	-	-	-	240	-	-	-
5	-	-	180	-	-	-	600	-	-	-	12
6	-	60	-	-	-	804	-	-	-	60	-
7	12	-	-	-	600	-	-	-	180	-	-
8	-	-	-	240	-	-	-	255	-	-	-
9	-	-	40	-	-	-	180	-	-	-	-
10	-	6	-	-	-	60	-	-	-	-	-
11	-	-	-	-	12	-	-	-	-	-	-

- $\min(|x| + |M_\theta x|)$: *branch number* $\mathcal{B}$ [JDA, 1993]
- link with error-correcting codes [JDA, 1995]:
  - $(x, \mathrm{M}_\theta x)$ as codeword for x: minimum distance $d = \mathcal{B}$
  - $[24, 12, 8]$-code: extended binary Golay code

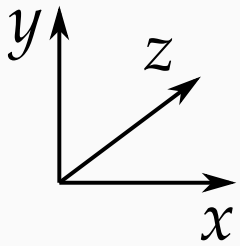
Rijndael [Daemen-Rijmen '98]



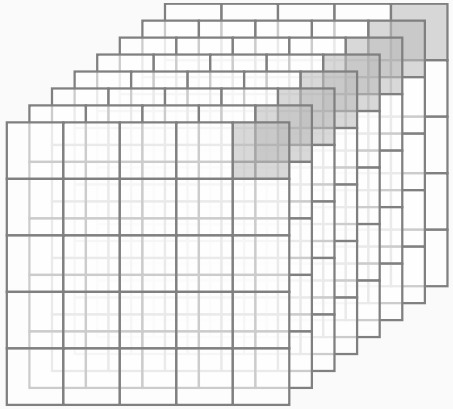
- bytes rather than bits
- MixColumns has $\mathcal{B} = 5$: maximum-distance separable (MDS) code
- Great! ...but
  - scales badly to widths required in hashing or permutations
  - SubBytes 8-bit S-box is heavy in HW or bit-sliced SW



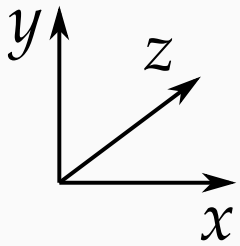
state



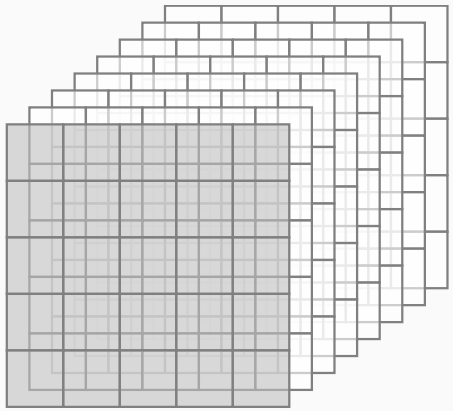
- 5×5 lanes, each containing 2^ℓ bits (1, 2, 4, 8, 16, 32 or 64)
- (5×5)-bit slices, 2^ℓ of them



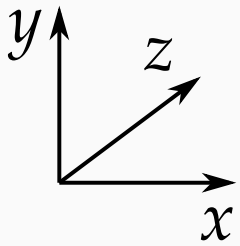
lane



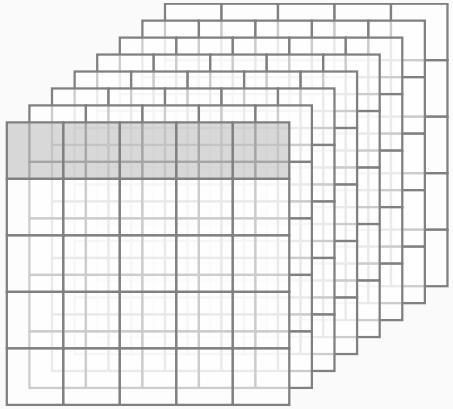
- 5×5 lanes, each containing 2^ℓ bits (1, 2, 4, 8, 16, 32 or 64)
- (5×5)-bit slices, 2^ℓ of them

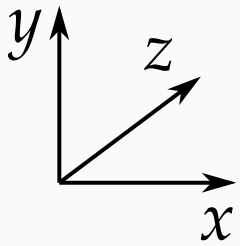


slice

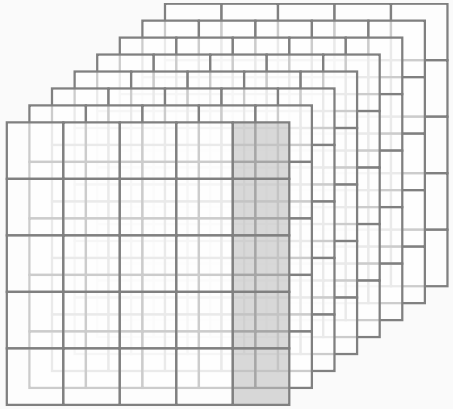


- 5×5 lanes, each containing 2^ℓ bits (1, 2, 4, 8, 16, 32 or 64)
- (5×5)-bit slices, 2^ℓ of them

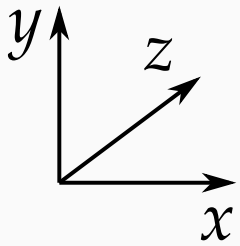




- 5×5 lanes, each containing 2^ℓ bits (1, 2, 4, 8, 16, 32 or 64)
- (5×5)-bit slices, 2^ℓ of them

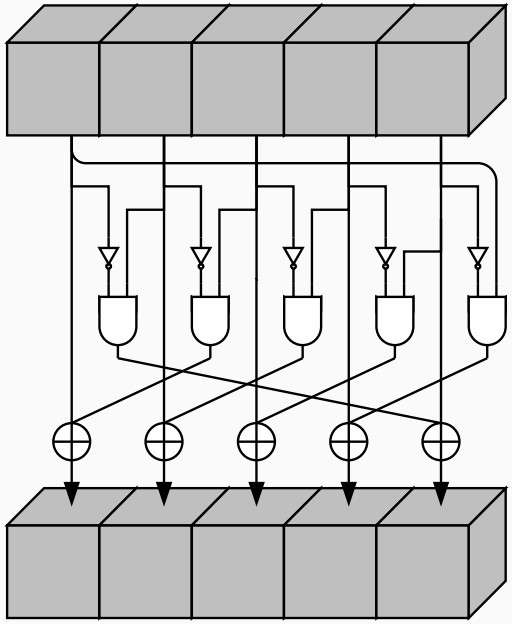


column

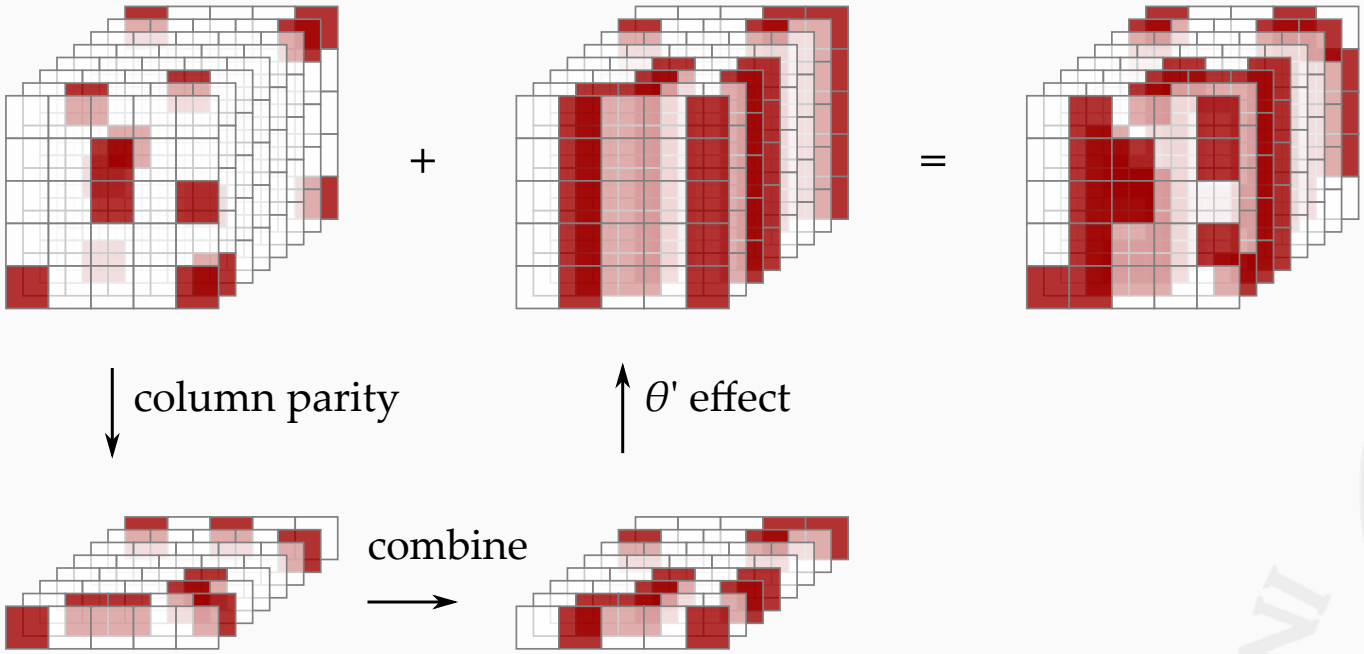


- 5×5 lanes, each containing 2^ℓ bits (1, 2, 4, 8, 16, 32 or 64)
- (5×5)-bit slices, 2^ℓ of them

Nonlinear step χ operates on rows

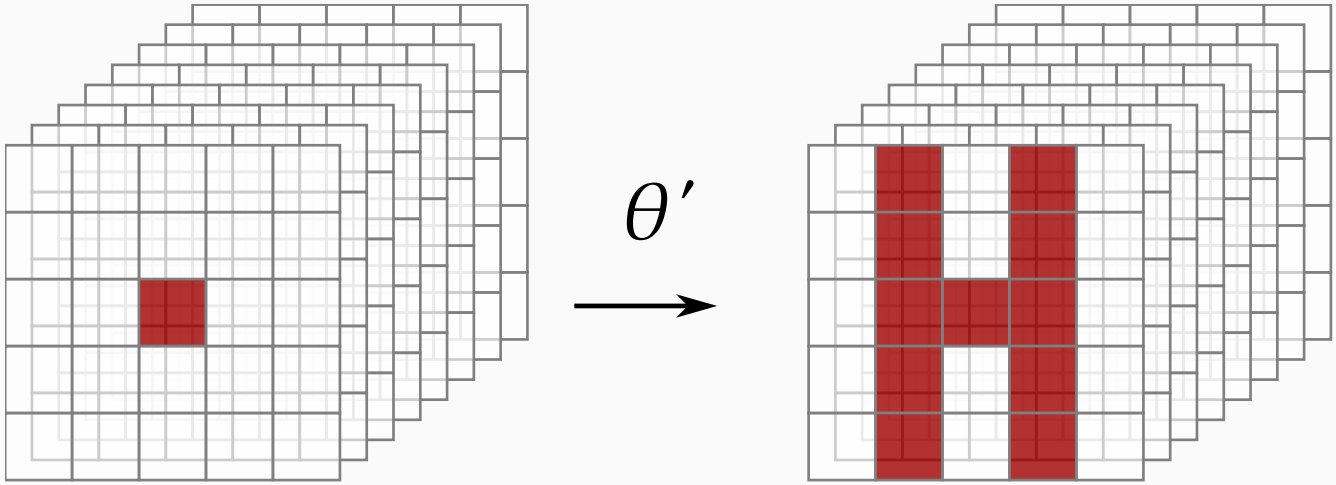


θ', a simplified version of the mixing layer

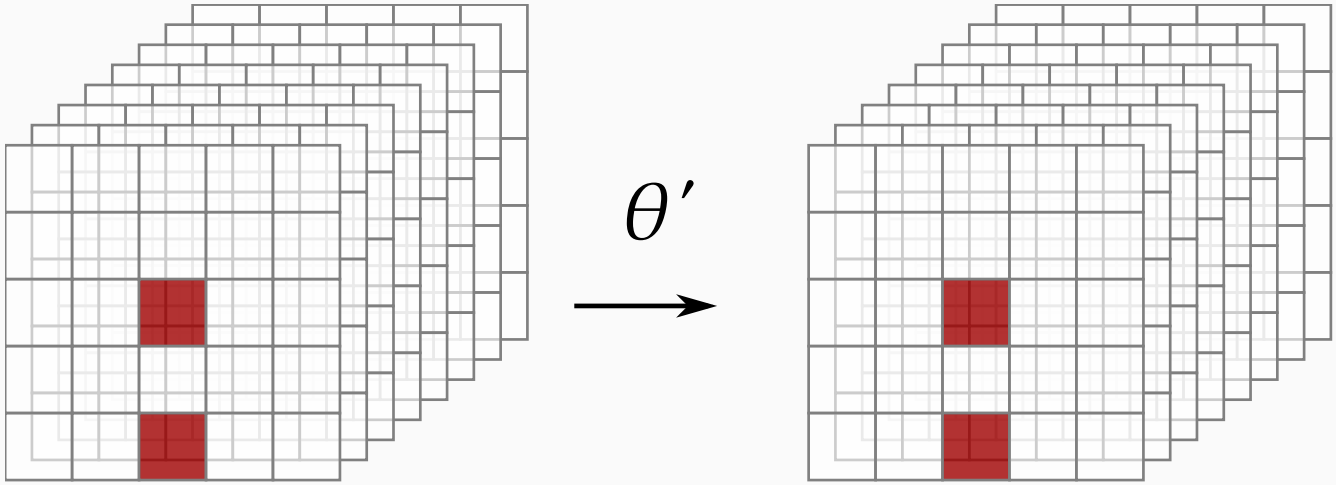


- Add to each cell parity of neighboring columns:
  $b_{x,y,z} = a_{x,y,z} \oplus p_{x-1,z} \oplus p_{x+1,z}$
- Cost: 2 XORs per bit

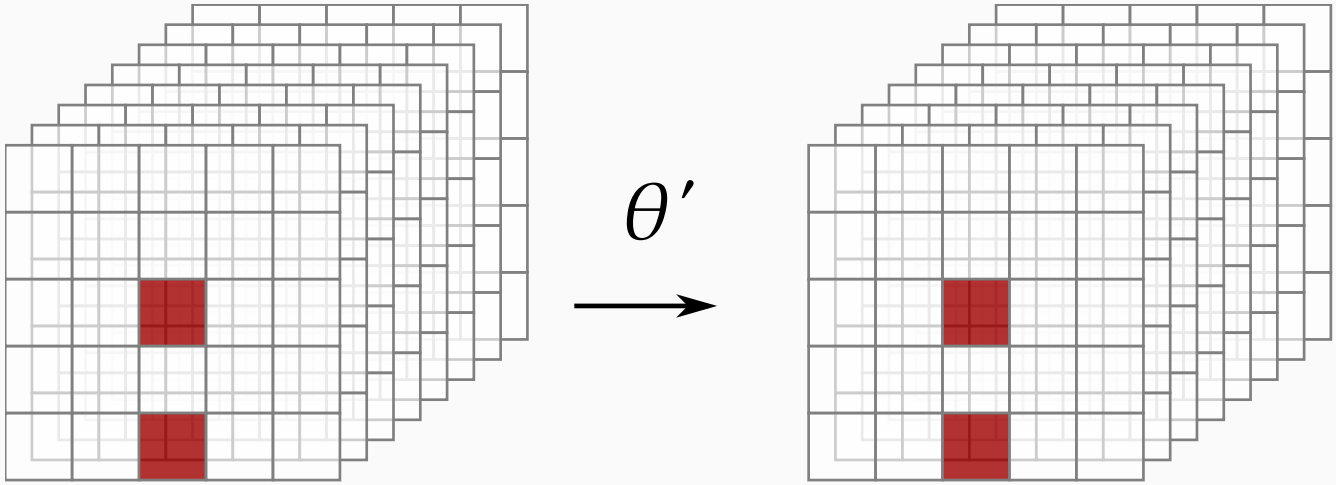
Propagation of single-bit difference



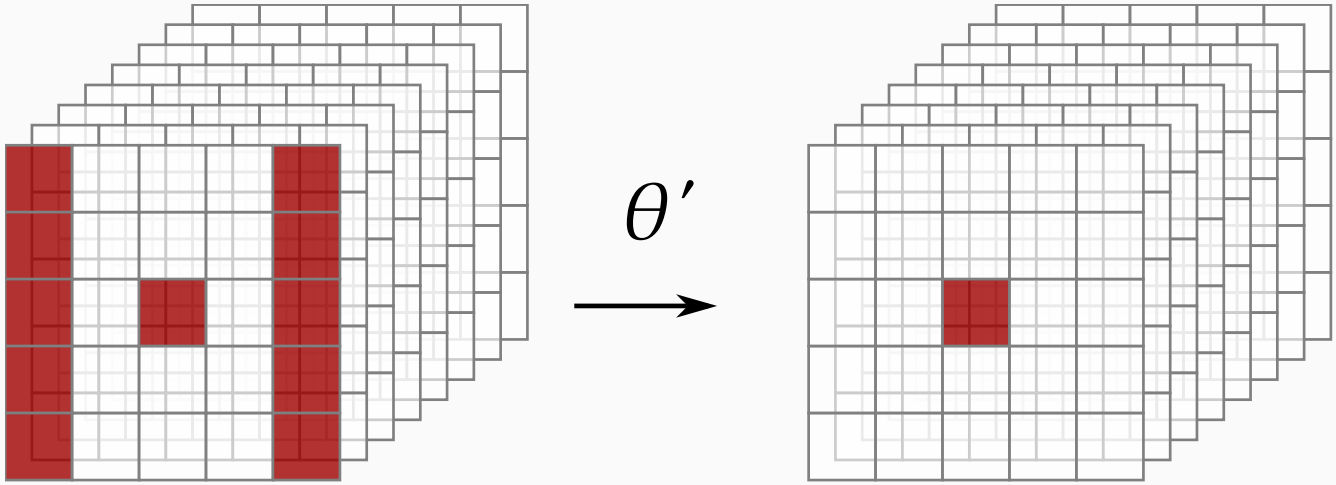
Propagation of particular 2-bit difference (kernel)



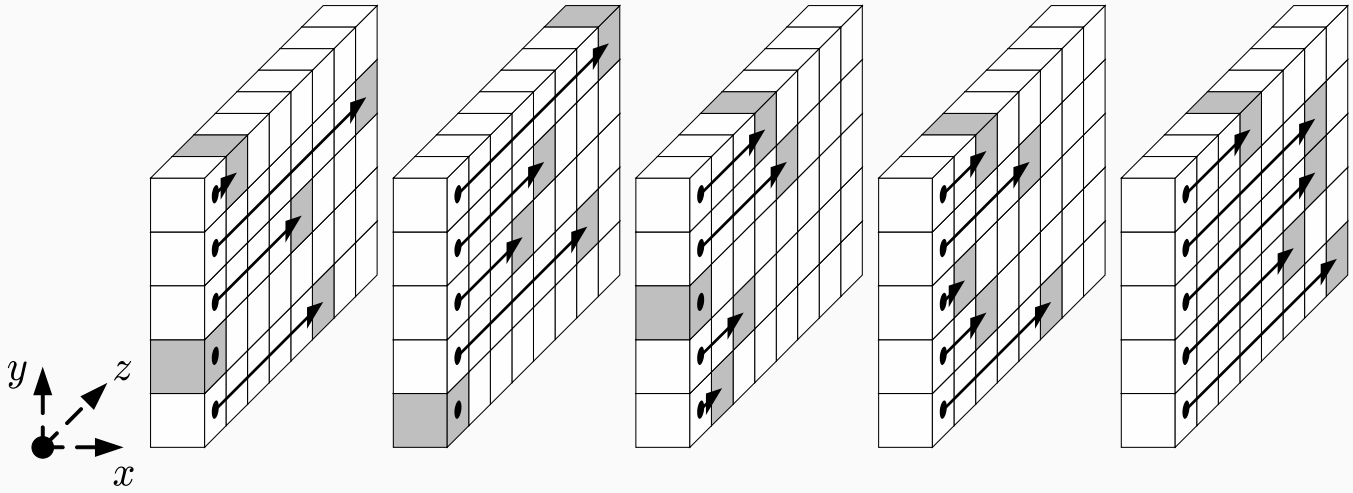
Propagation of particular 2-bit difference (kernel)



Propagation of a single-bit difference through the inverse

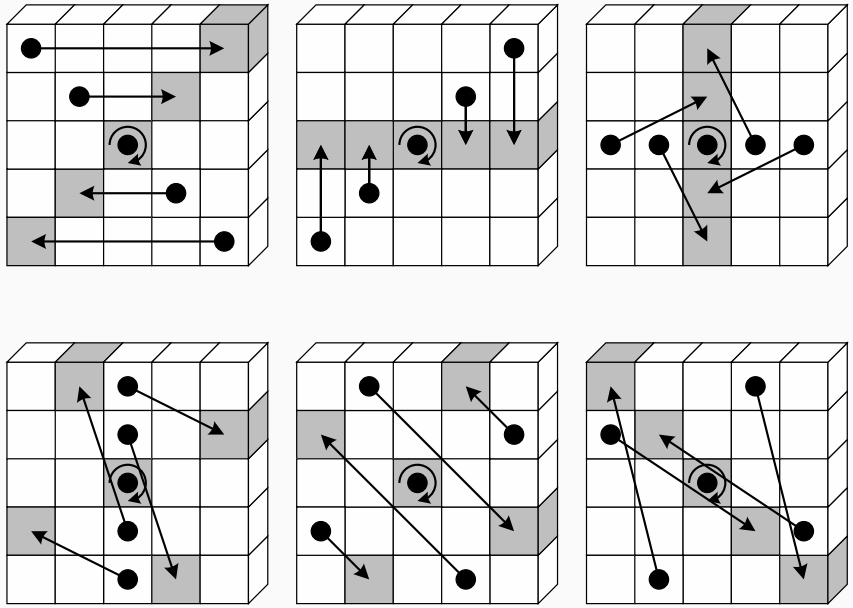


Transposition step ρ



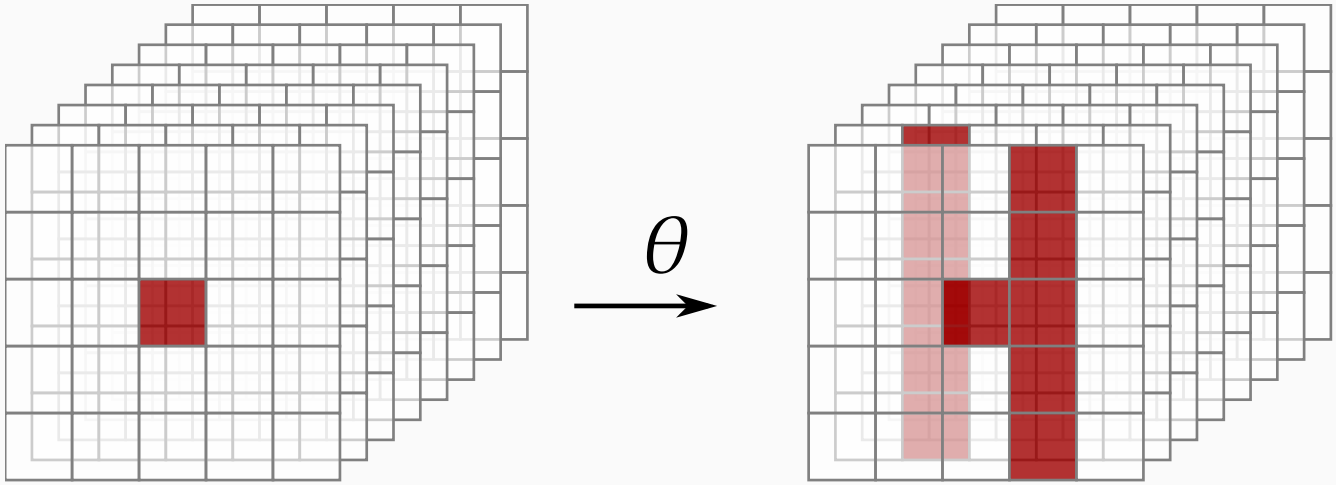
For diffusion between the slices

Transposition step π

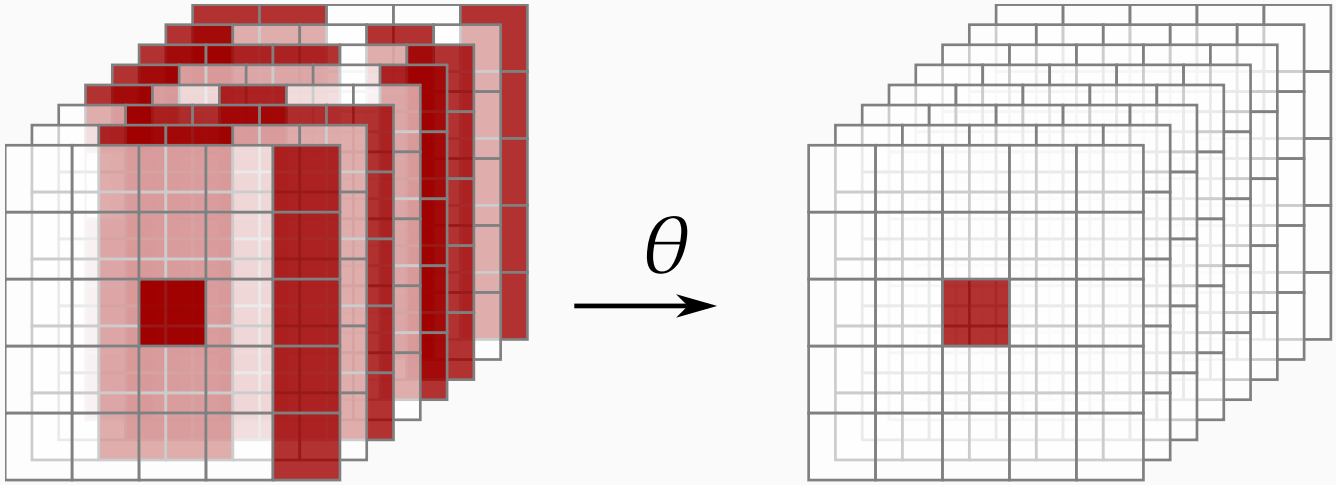


$$a_{x,y} \leftarrow a_{x',y'} \text{ with } \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Tweaking θ' to θ



Inverse of θ



Density of inverse frustrates cryptanalysts

Outline

Column parity mixers: the state

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix} = \text{e.g.} \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Column parity mixers: the state

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix} = \text{e.g.} \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Matrix A with m rows and n columns

Computing the column parity

$$\begin{pmatrix} p_0 & p_1 & p_2 & p_3 & p_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}$$

Computing the column parity

$$\begin{pmatrix} p_0 & p_1 & p_2 & p_3 & p_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}$$

$$P = \mathbf{1}_m^{\mathsf{T}} A$$

Folding the parity

$$\begin{pmatrix} e_0 & e_1 & e_2 & e_3 & e_4 \end{pmatrix} = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 & p_4 \end{pmatrix} \begin{pmatrix} z_{00} & z_{01} & z_{02} & z_{03} & z_{04} \\ z_{10} & z_{11} & z_{12} & z_{13} & z_{14} \\ z_{20} & z_{21} & z_{22} & z_{23} & z_{24} \\ z_{30} & z_{31} & z_{32} & z_{33} & z_{34} \\ z_{40} & z_{41} & z_{42} & z_{43} & z_{44} \end{pmatrix}$$

Folding the parity

$$\begin{pmatrix} e_0 & e_1 & e_2 & e_3 & e_4 \end{pmatrix} = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 & p_4 \end{pmatrix} \begin{pmatrix} z_{00} & z_{01} & z_{02} & z_{03} & z_{04} \\ z_{10} & z_{11} & z_{12} & z_{13} & z_{14} \\ z_{20} & z_{21} & z_{22} & z_{23} & z_{24} \\ z_{30} & z_{31} & z_{32} & z_{33} & z_{34} \\ z_{40} & z_{41} & z_{42} & z_{43} & z_{44} \end{pmatrix}$$

$$e = PZ$$

Stretching the effect

$$\begin{pmatrix} e_0 & e_1 & e_2 & e_3 & e_4 \\ e_0 & e_1 & e_2 & e_3 & e_4 \\ e_0 & e_1 & e_2 & e_3 & e_4 \\ e_0 & e_1 & e_2 & e_3 & e_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} e_0 & e_1 & e_2 & e_3 & e_4 \end{pmatrix}$$

Stretching the effect

$$\begin{pmatrix} e_0 & e_1 & e_2 & e_3 & e_4 \\ e_0 & e_1 & e_2 & e_3 & e_4 \\ e_0 & e_1 & e_2 & e_3 & e_4 \\ e_0 & e_1 & e_2 & e_3 & e_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} e_0 & e_1 & e_2 & e_3 & e_4 \end{pmatrix}$$

$$E = \mathbf{1}_m e$$

Putting it together

Compute the column parity

$$P = \mathbf{1}_m^{\mathsf{T}} A$$

Putting it together

Compute the column parity

$$P = \mathbf{1}_m^{\mathsf{T}} A$$

Compute the θ-effect by applying the folding matrix Z

$$e = PZ$$

Putting it together

Compute the column parity

$$P = \mathbf{1}_m^{\mathsf{T}} A$$

Compute the θ-effect by applying the folding matrix Z

$$e = PZ$$

Stretch the θ-effect

$$E = \mathbf{1}_m PZ = \mathbf{1}_m \mathbf{1}_m^{\mathsf{T}} AZ = \mathbf{1}_m^m AZ$$

Putting it together

Compute the column parity

$$P = \mathbf{1}_m^{\mathsf{T}} A$$

Compute the θ-effect by applying the folding matrix Z

$$e = PZ$$

Stretch the θ-effect

$$E = \mathbf{1}_m PZ = \mathbf{1}_m \mathbf{1}_m^{\mathsf{T}} AZ = \mathbf{1}_m^m AZ$$

Add the θ-effect to the state

$$\theta(A) = A + E$$

Putting it together

Compute the column parity

$$P = \mathbf{1}_m^{\mathsf{T}} A$$

Compute the θ-effect by applying the folding matrix Z

$$e = PZ$$

Stretch the θ-effect

$$E = \mathbf{1}_m PZ = \mathbf{1}_m \mathbf{1}_m^{\mathsf{T}} AZ = \mathbf{1}_m^m AZ$$

Add the θ-effect to the state

$$\theta(A) = A + E$$

Working out gives:

$$\theta(A) = A + \mathbf{1}_m^m AZ$$

Definition

A column parity mixer is defined by

m : number of rows of the state

n : number of columns of the state

Z : $n \times n$ parity-folding matrix

$$\theta(A) = A + \mathbf{1}_m^m AZ$$

Outline

Composing CPMs

What does $\theta'' = \theta' \circ \theta$ look like?

$$\begin{aligned}
\theta'(\theta(A)) &= \theta'(A + \mathbf{1}_m^m AZ) \\
&= A + \mathbf{1}_m^m AZ + \mathbf{1}_m^m (A + \mathbf{1}_m^m AZ) Z' \\
&= A + \mathbf{1}_m^m AZ + \mathbf{1}_m^m AZ' + (\mathbf{1}_m^m)^2 AZZ'
\end{aligned}$$

Composing CPMs

What does $\theta'' = \theta' \circ \theta$ look like?

$$\begin{aligned}\theta'(\theta(A)) &= \theta'(A + \mathbf{1}_m^m AZ) \\ &= A + \mathbf{1}_m^m AZ + \mathbf{1}_m^m (A + \mathbf{1}_m^m AZ) Z' \\ &= A + \mathbf{1}_m^m AZ + \mathbf{1}_m^m AZ' + (\mathbf{1}_m^m)^2 AZZ'\end{aligned}$$

For even m : $(\mathbf{1}_m^m)^2 = \mathbf{0}$

For odd m : $(\mathbf{1}_m^m)^2 = \mathbf{1}_m^m$

Algebraic properties: case of even m

$$\begin{aligned}\theta'(\theta(A)) &= A + 1_m^m AZ + 1_m^m AZ' \\ &= A + 1_m^m A(Z + Z')\end{aligned}$$

Algebraic properties: case of even m

$$\begin{aligned}\theta'(\theta(A)) &= A + 1_m^m AZ + 1_m^m AZ' \\ &= A + 1_m^m A(Z + Z')\end{aligned}$$

$\theta' \circ \theta$ is CPM with $Z'' = Z' + Z$

- group isomorphic to $(\mathbb{Z}_2^{n^2}, +)$
- any CPM is an involution
- any CPM is invertible
- commutativity: $\theta' \circ \theta = \theta \circ \theta'$

Algebraic properties: case of odd m

$$\begin{aligned}\theta'(\theta(A)) &= A + 1_m^m AZ + 1_m^m AZ' + 1_m^m AZZ' \\ &= A + 1_m^m A\left((Z + \mathsf{I})(Z' + \mathsf{I}) + \mathsf{I}\right)\end{aligned}$$

Algebraic properties: case of odd m

$$\begin{aligned}\theta'(\theta(A)) &= A + 1_m^m AZ + 1_m^m AZ' + 1_m^m AZZ' \\ &= A + 1_m^m A\left((Z + \mathsf{I})(Z' + \mathsf{I}) + \mathsf{I}\right)\end{aligned}$$

Algebraic properties: case of odd m

$$\begin{aligned}\theta'(\theta(A)) &= A + 1_m^m AZ + 1_m^m AZ' + 1_m^m AZZ' \\ &= A + 1_m^m A\left((Z+\mathsf{I})(Z'+\mathsf{I})+\mathsf{I}\right)\end{aligned}$$

$\theta' \circ \theta$ is CPM with $Z'' + \mathsf{I} = (Z+\mathsf{I})(Z'+\mathsf{I})$

- Group isomorphic to $GL(n,2)$
- CPM is invertible iff $Z+\mathsf{I}$ is, and inverse has $Z'+\mathsf{I} = (Z+\mathsf{I})^{-1}$
- Non-commutative

Difference propagation

Input difference: $A' = A + A^*$, output difference $B' = \theta(A) + \theta(A^*)$

Difference propagation

Input difference: $A' = A + A^*$, output difference $B' = \theta(A) + \theta(A^*)$

Thanks to linearity:

$$B' = A' + \mathbf{1}_m^m A' Z$$

Linear cryptanalysis: linear functions and mask propagation

With (vector) state a, linear function defined as *mask* w: $w^T a$

state a:	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}
mask w:	0	1	0	0	1	1	0	0	1	0	0	0

$w^T x$: $a_1 + a_4 + a_5 + a_8$

Linear cryptanalysis: linear functions and mask propagation

With (vector) state a, linear function defined as *mask* w: $w^\mathsf{T} a$

state a:	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}
mask w:	0	1	0	0	1	1	0	0	1	0	0	0

$w^\mathsf{T} x$: $a_1 + a_4 + a_5 + a_8$

Mask propagation through linear mapping with matrix M:

Let $b = Ma$. Then given v, what is u such that $u^\mathsf{T} a = v^\mathsf{T} b$?

Linear cryptanalysis: linear functions and mask propagation

With (vector) state a, linear function defined as *mask* w: $w^\mathsf{T} a$

state a:	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}
mask w:	0	1	0	0	1	1	0	0	1	0	0	0

$w^\mathsf{T} x$: $\quad a_1 \quad + \quad a_4 + a_5 \quad + \quad a_8$

Mask propagation through linear mapping with matrix M:

Let $b = Ma$. Then given v, what is u such that $u^\mathsf{T} a = v^\mathsf{T} b$?

$$v^\mathsf{T} b = v^\mathsf{T} M a = (M^\mathsf{T} v)^\mathsf{T} a$$

v at output of M propagates to $u = M^\mathsf{T} v$ at input

Mask propagation in CPMs

Representation of linear functions:

- Mask: $m \times n$ binary matrix V
- Linear function of A: $\sum_{ij} v_{ij} a_{ij}$

Mask propagation in CPMs

Representation of linear functions:

- Mask: $m \times n$ binary matrix V
- Linear function of A: $\sum_{ij} v_{ij} a_{ij}$

More usable expression: $\mathrm{tr}(V^\mathsf{T} A)$, with $\mathrm{tr}(M) = \sum_i m_{ii}$

Mask propagation in CPMs

Representation of linear functions:

- Mask: $m \times n$ binary matrix V
- Linear function of A: $\sum_{ij} v_{ij} a_{ij}$

More usable expression: $\mathrm{tr}(V^\mathsf{T} A)$, with $\mathrm{tr}(M) = \sum_i m_{ii}$

$$
\begin{aligned}
\mathrm{tr}(U^\mathsf{T} A) &= \mathrm{tr}(V^\mathsf{T} B) \\
&= \mathrm{tr}(V^\mathsf{T}(A + \mathbf{1}_m^m A Z)) \\
&= \mathrm{tr}(V^\mathsf{T} A + V^\mathsf{T} \mathbf{1}_m^m A Z)) \\
&= \mathrm{tr}(V^\mathsf{T} A + (\mathbf{1}_m^m V)^\mathsf{T} A Z)) \\
&= \mathrm{tr}(V^\mathsf{T} A + Z(\mathbf{1}_m^m V)^\mathsf{T} A)) \\
&= \mathrm{tr}((V + \mathbf{1}_m^m V Z^\mathsf{T})^\mathsf{T} A)\,.
\end{aligned}
$$

V at output of M propagates to $U = V + \mathbf{1}_m^m V Z^\mathsf{T}$ at input

Interesting special case: circulant Z

Polynomial representation of parity:

$$\begin{pmatrix} p_0 & p_1 & p_2 & p_3 & p_4 \end{pmatrix} \longleftrightarrow p_0 + p_1 x + p_2 x^2 + p_3 x^3 + p_4 x^4 = p(x)$$

Interesting special case: circulant Z

Polynomial representation of parity:

$$\begin{pmatrix} p_0 & p_1 & p_2 & p_3 & p_4 \end{pmatrix} \longleftrightarrow p_0 + p_1 x + p_2 x^2 + p_3 x^3 + p_4 x^4 = p(x)$$

Computation of e:

$$e(x) = z(x)p(x) \bmod 1 + x^n$$

Implications for even number of rows m

- composition of circulant CPMs is commutative: $\theta \circ \theta' = \theta' \circ \theta$
- CPM can only be invertible if $z(x)$ has even number of terms

Outline

Vector spaces and affine spaces

- State space $\mathcal{A}$ has dimension mn
- Parity space $\mathcal{P}$ has dimension n
- (Column-parity) kernel: subspace of $\mathcal{A}$ with zero parity
  - dimension $(m-1)n$
- Set of states with given parity p: affine space with dimension $(m-1)n$
- Parities partition state space

The column-parity kernel

- For states in the kernel, θ is the identity
- Lightest states have Hamming weight 2
  - two active bits in same column: *orbital*
- Any state in kernel can be expressed as set of orbitals
- Branch number $\mathcal{B} = 4$
- Number of states with $|A| + |\theta(A)| = 4$
  - Absolute: $n\binom{m}{2}$
  - Per statebit: $(m-1)/2$
- Number of states with $|A| + |\theta(A)| = 8$
  - Absolute: $\binom{n}{2}\binom{m}{2}^2 + \ldots$
  - Per statebit: $(n-1)m(m-1)^2/8$

Class of states with given parity P

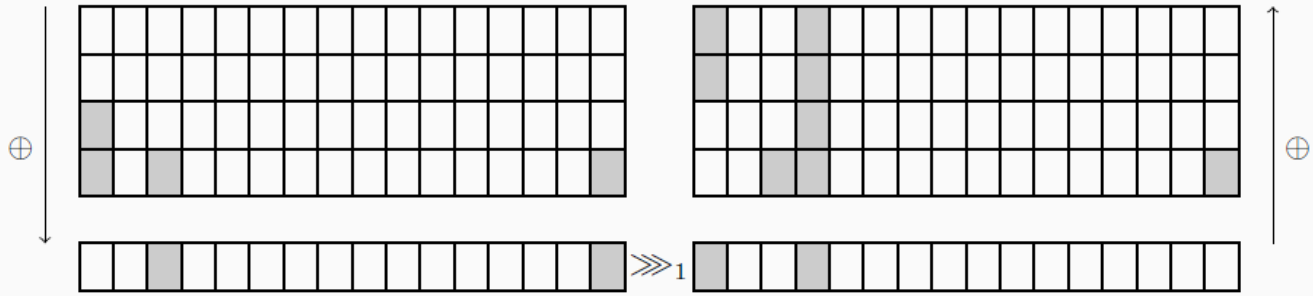
- We are interested in $\min_{\mathbf{1}_m^\intercal A = P} |A| + |\theta(A)|$
- θ consists of adding $E = \mathbf{1}_m^m PZ$
- We partition columns x of A in 3 classes
  - $E_x = 1$: affected
  - $E_x = 0, P_x = 1$: unaffected odd
  - $E_x = 0, P_x = 0$: unaffected even
- Rule: removal of orbital from A does not change its parity P
  - affected column contributes m bits
  - unaffected odd column contributes at least 2 bits
  - unaffected even column may have 0 bits
- Branch number given the parity-class: $\mathcal{B}(P)$
  - value: $2y + m\alpha$
  - number of states with this weight: $m^y 2^{(m-1)\alpha}$
  - called *parity-bare* states
- Z can be chosen such that α can only be low if y is high

Outline

Design approach

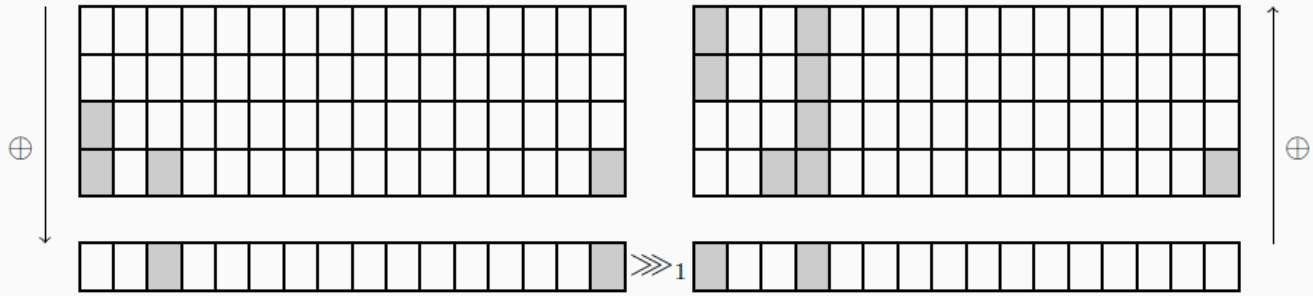
- Goals
  - efficient using only (cyclic) shifts and bitwise Boolean instructions
  - efficient in hardware
  - simple and symmetric
- Structure: two-dimensional state of 4-bit nibbles
- Round function:
  - non-linearity: 4-bit S-box operating on nibbles
  - mixing: column-parity mixer
  - transposition: Shift of rows and permutation of rows
  - addition of round constant

Mixing layer

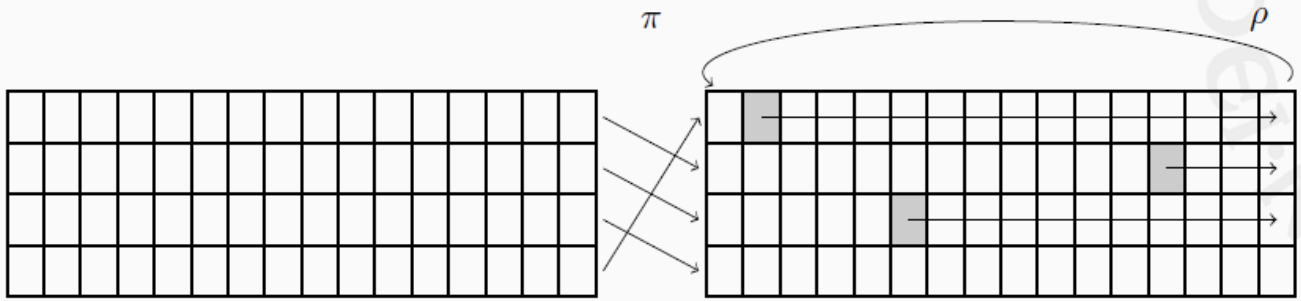


The 256-bit permutation Mixifer

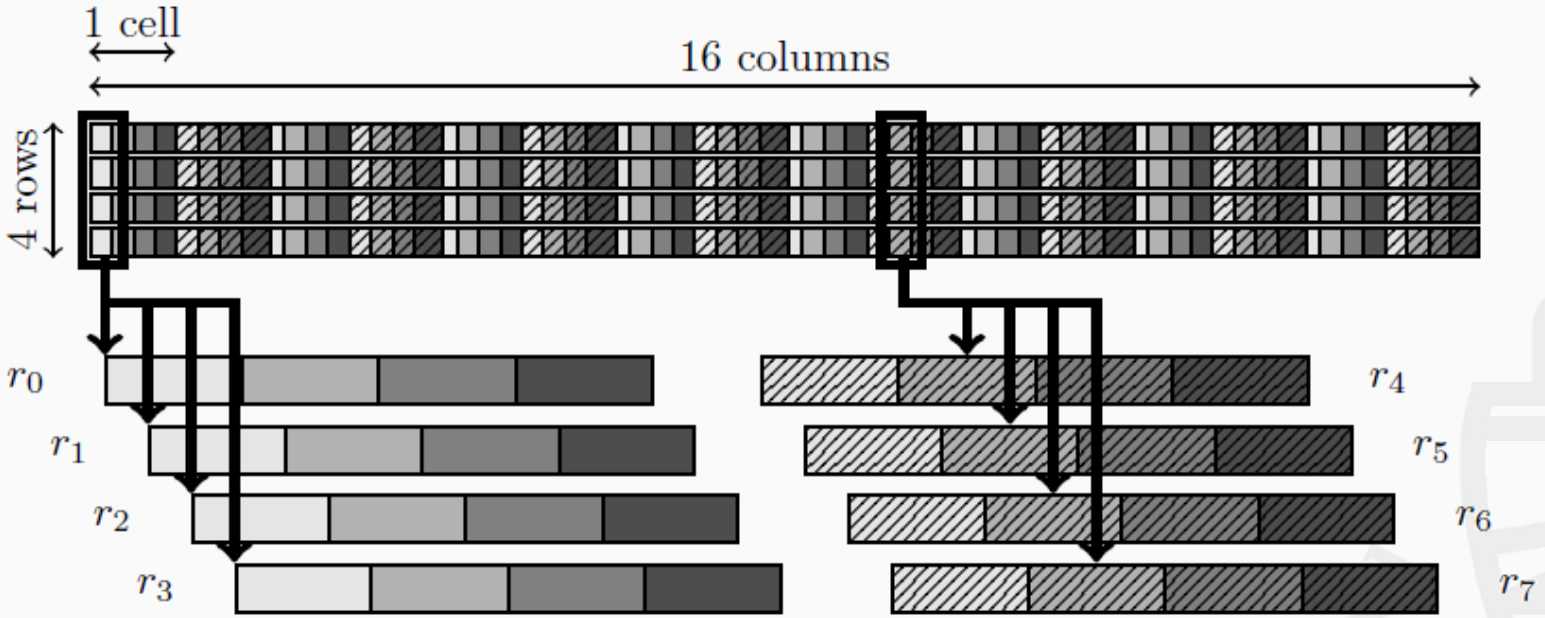
Mixing layer



Transposition layer



Mapping Mixifer state to 8 32-bit words



Requires S-box to be rotation-invariant for efficiency

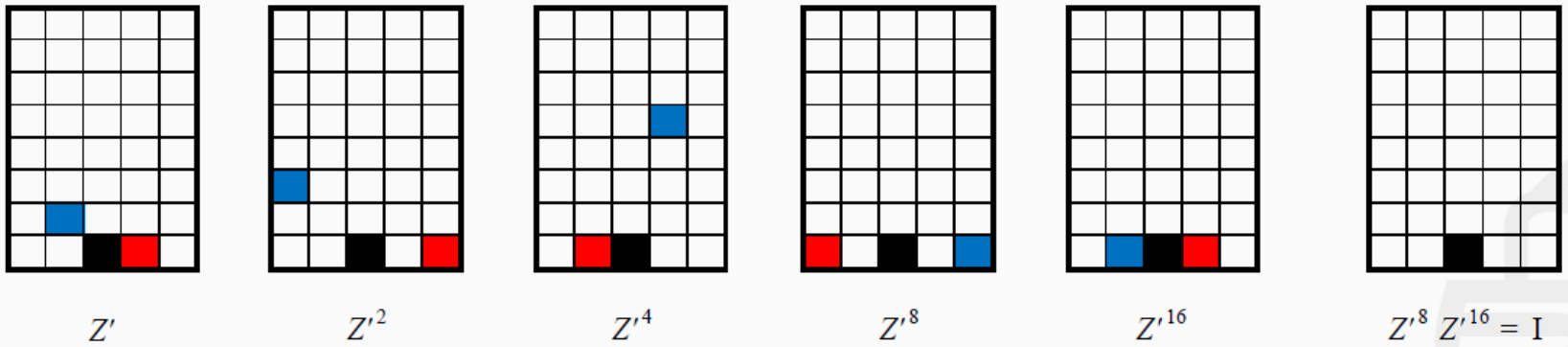
Mixifer vs. AES

		AES	Mixifer
Performance	cycles/byte (1 round)	bit-sliced	
	on ARM cortex M4	10	2.3
Differential trails	# rounds	4	5
	# active S-boxes	25	48
	$-\log$ max DP S-box	6	2
	weight/round	37.5	19
Linear trails	# rounds	4	5
	# active S-boxes	25	43
	$-\log$ max LP S-box	6	2
	weight/round	37.5	17

Outline

Inverse of θ of KECCAK-p (reverse-engineered from code of Mike Hamburg)

Example for KECCAK-$p[200, n_r] : \theta^{-1}(A) = A + \mathbf{1}_5^5 A(Z'^{-1} + \mathrm{I})$ with $Z' = Z + \mathrm{I}$



$$Z'^{-1} = Z'^{23} = Z' \times Z'^2 \times Z'^4 \times Z'^{16}$$

Algorithm:

(1) $P \leftarrow \mathbf{1}_5^{\mathsf{T}} A$
(2) $e \leftarrow PZ'$; $e \leftarrow eZ'^2$; $e \leftarrow eZ'^4$; $e \leftarrow eZ'16$
(3) $e \leftarrow e + P$
(4) $A \leftarrow A + \mathbf{1}_5 e$

Cost per bit: 3.6 XORs

For KECCAK-$p[1600, n_r]$ order of Z' is 191 and cost per bit is 4.4 XORs

Thanks for your attention!

Questions?