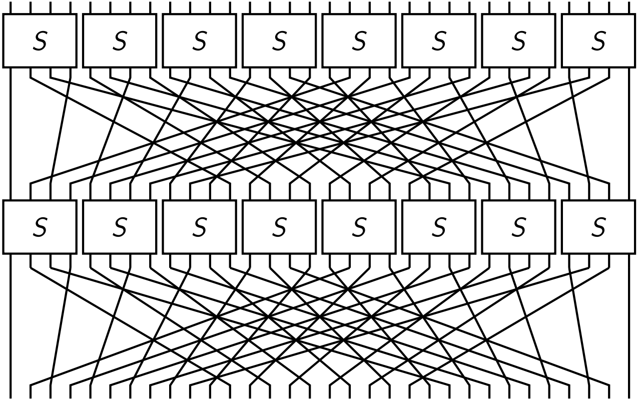
Column Parity Mixers

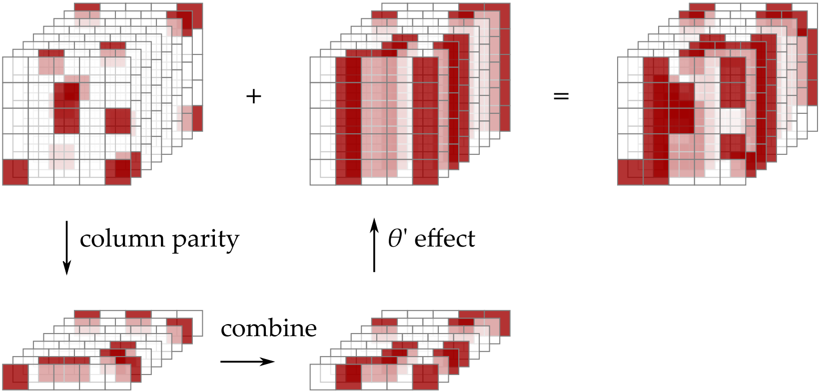
Ko Stoffelen and Joan Daemen

iCIS | Digital Security
Radboud University

Diffusion



Diffusion in Keccak-f



Only 2 XORs/bit + good bounds on differential trails [MDA17]

Column parity mixers

For an $m \times n$ matrix A over $\mathbb{F}_2^{\ell}$:

$$\theta(A) = A + \quad f(A)$$

$$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \end{pmatrix}$$

Column parity mixers

For an $m \times n$ matrix A over $\mathbb{F}_2^\ell$:

$$\theta(A) = A + \quad \mathbf{1}_m^\mathsf{T} A$$

$$\underbrace{\begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \end{pmatrix}}_{1 \times n \text{ column parity}}$$

Column parity mixers

For an $m \times n$ matrix A over $\mathbb{F}_2^\ell$:

$$\theta(A) = A + \quad \mathbf{1}_m^\mathsf{T} A Z$$

$$\underbrace{\underbrace{\begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \end{pmatrix}}_{1 \times n \text{ column parity}} \underbrace{\begin{pmatrix} z_{0,0} & z_{0,1} & z_{0,2} & z_{0,3} \\ z_{1,0} & z_{1,1} & z_{1,2} & z_{1,3} \\ z_{2,0} & z_{2,1} & z_{2,2} & z_{2,3} \\ z_{3,0} & z_{3,1} & z_{3,2} & z_{3,3} \end{pmatrix}}_{n \times n \text{ parity-folding matrix}}}_{1 \times n \ \theta\text{-effect}}$$

Column parity mixers

For an $m \times n$ matrix A over $\mathbb{F}_2^\ell$:

$$\theta(A) = A + \mathbf{1}_m \mathbf{1}_m^\mathsf{T} A Z$$

$$\underbrace{\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \underbrace{\underbrace{\begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \end{pmatrix}}_{1 \times n \text{ column parity}} \underbrace{\begin{pmatrix} z_{0,0} & z_{0,1} & z_{0,2} & z_{0,3} \\ z_{1,0} & z_{1,1} & z_{1,2} & z_{1,3} \\ z_{2,0} & z_{2,1} & z_{2,2} & z_{2,3} \\ z_{3,0} & z_{3,1} & z_{3,2} & z_{3,3} \end{pmatrix}}_{n \times n \text{ parity-folding matrix}}}_{1 \times n\ \theta\text{-effect}}}_{m \times n \text{ expanded } \theta\text{-effect}}$$

Column parity mixers

For an $m \times n$ matrix A over $\mathbb{F}_2^\ell$:

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

$$\underbrace{\begin{pmatrix}1\\1\\1\end{pmatrix} \underbrace{\underbrace{\begin{pmatrix}1 & 1 & 1\end{pmatrix} \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \end{pmatrix}}_{1\times n \text{ column parity}} \underbrace{\begin{pmatrix} z_{0,0} & z_{0,1} & z_{0,2} & z_{0,3} \\ z_{1,0} & z_{1,1} & z_{1,2} & z_{1,3} \\ z_{2,0} & z_{2,1} & z_{2,2} & z_{2,3} \\ z_{3,0} & z_{3,1} & z_{3,2} & z_{3,3} \end{pmatrix}}_{n\times n \text{ parity-folding matrix}}}_{1\times n\ \theta\text{-effect}}}_{m\times n \text{ expanded } \theta\text{-effect}}$$

Column parity mixers

For an $m \times n$ matrix A over $\mathbb{F}_2^{\ell}$:

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

$$\underbrace{\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \underbrace{\underbrace{\begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \end{pmatrix}}_{1 \times n \text{ column parity}} \underbrace{\begin{pmatrix} z_{0,0} & z_{0,1} & z_{0,2} & z_{0,3} \\ z_{1,0} & z_{1,1} & z_{1,2} & z_{1,3} \\ z_{2,0} & z_{2,1} & z_{2,2} & z_{2,3} \\ z_{3,0} & z_{3,1} & z_{3,2} & z_{3,3} \end{pmatrix}}_{n \times n \text{ parity-folding matrix}}}_{1 \times n\ \theta\text{-effect}}}_{m \times n \text{ expanded } \theta\text{-effect}}$$

θ fully defined by m, n and Z

Special case: circulant Z

$$\begin{pmatrix} z_0 & z_1 & z_2 & z_3 \\ z_1 & z_2 & z_3 & z_0 \\ z_2 & z_3 & z_0 & z_1 \\ z_3 & z_0 & z_1 & z_2 \end{pmatrix}$$

Special case: circulant Z

$$\begin{pmatrix} z_0 & z_1 & z_2 & z_3 \\ z_1 & z_2 & z_3 & z_0 \\ z_2 & z_3 & z_0 & z_1 \\ z_3 & z_0 & z_1 & z_2 \end{pmatrix}$$

$$z(x) = z_0 + z_1 x + z_2 x^2 + z_3 x^3$$

Special case: circulant Z

$$\begin{pmatrix} z_0 & z_1 & z_2 & z_3 \\ z_1 & z_2 & z_3 & z_0 \\ z_2 & z_3 & z_0 & z_1 \\ z_3 & z_0 & z_1 & z_2 \end{pmatrix}$$

$$z(x) = z_0 + z_1 x + z_2 x^2 + z_3 x^3$$

θ-effect: $z(x)p(x) \bmod 1 + x^n$

Special case: circulant Z

$$\begin{pmatrix} z_0 & z_1 & z_2 & z_3 \\ z_1 & z_2 & z_3 & z_0 \\ z_2 & z_3 & z_0 & z_1 \\ z_3 & z_0 & z_1 & z_2 \end{pmatrix}$$

$$z(x) = z_0 + z_1 x + z_2 x^2 + z_3 x^3$$

θ-effect: $z(x)p(x) \bmod 1 + x^n$

$$\theta(a(x,y)) = a(x,y) + \frac{1+y^m}{1+y} z(x) a(x,y) \bmod (1+x^n)(1+y^m)$$

Algebraic properties

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

Algebraic properties

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

- If m even, $(\mathbf{1}_m^m)^2 = \mathbf{0}_m^m$:
  - $\theta'(\theta(A)) = A + \mathbf{1}_m^m A(Z + Z')$
  - Group isomorphic to $\left(\mathbb{Z}_2^{n^2}, +\right)$
  - CPM is invertible, involution, commutative

Algebraic properties

$$\theta'(\theta(A)) = \theta'(A + \mathbf{1}_m^m AZ)$$
$$= A + \mathbf{1}_m^m AZ + \mathbf{1}_m^m AZ' + (\mathbf{1}_m^m)^2 AZZ'$$

- If m even, $(\mathbf{1}_m^m)^2 = \mathbf{0}_m^m$:
  - $\theta'(\theta(A)) = A + \mathbf{1}_m^m A(Z + Z')$
  - Group isomorphic to $\left(\mathbb{Z}_2^{n^2}, +\right)$
  - CPM is invertible, involution, commutative
- If m odd, $(\mathbf{1}_m^m)^2 = \mathbf{1}_m^m$:
  - $\theta'(\theta(A)) = A + \mathbf{1}_m^m A\left((Z + \mathbf{I})(Z' + \mathbf{I}) + \mathbf{I}\right)$
  - Group isomorphic to $GL(n, 2)$
  - CPM is invertible iff $Z + \mathbf{I}$ is, non-commutative

Propagation properties

- Differences

  $A_\Delta = A + A'$ at the input

  $\Rightarrow B_\Delta = \theta(A) + \theta(A') = \theta(A_\Delta)$ at the output

Propagation properties

- Differences

  $A_\Delta = A + A'$ at the input

  $\Rightarrow B_\Delta = \theta(A) + \theta(A') = \theta(A_\Delta)$ at the output

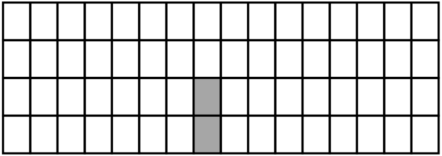
- Linear masks

  V at the output

  $\Rightarrow U = V + \mathbf{1}_m^m V Z^\mathsf{T}$ at the input

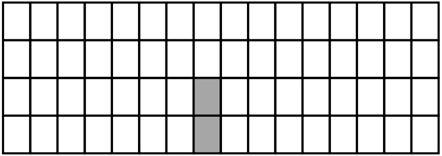
Diffusion with CPMs

- How about a state like this?



Diffusion with CPMs

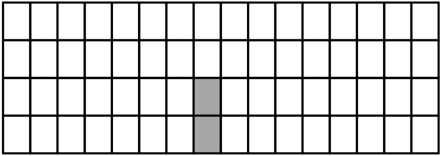
- How about a state like this?



- *Orbital*: pair of active bits in the same column

Diffusion with CPMs

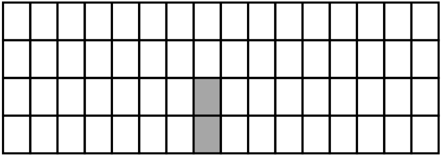
- How about a state like this?



- *Orbital*: pair of active bits in the same column
- θ is identity for states in the kernel

Diffusion with CPMs

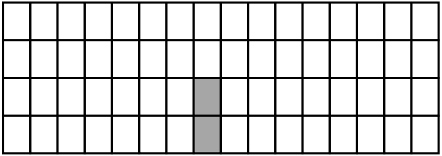
- How about a state like this?



- *Orbital*: pair of active bits in the same column
- θ is identity for states in the kernel
- States in the kernel can be expressed as a set of orbitals

Diffusion with CPMs

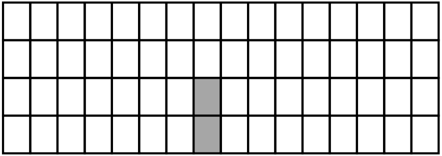
- How about a state like this?



- *Orbital*: pair of active bits in the same column
- θ is identity for states in the kernel
- States in the kernel can be expressed as a set of orbitals
- Branch number 4

Diffusion with CPMs

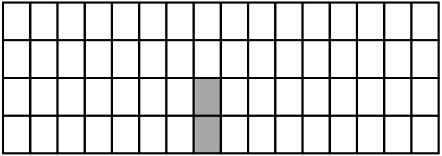
- How about a state like this?



- *Orbital*: pair of active bits in the same column
- θ is identity for states in the kernel
- States in the kernel can be expressed as a set of orbitals
- Branch number 4
- Requires transposition layer

Diffusion with CPMs

- How about a state like this?



- *Orbital*: pair of active bits in the same column
- θ is identity for states in the kernel
- States in the kernel can be expressed as a set of orbitals
- Branch number 4
- Requires transposition layer
- Single-bit difference propagates to $1 + |Z|\, m$ bits

CPMs vs. (near-)MDS matrices

Cipher	Type	XORs/bit	Branch no.
AES	MDS	3.03	5
Joltik	MDS	3	5
PHOTON	MDS	5†	7
Prøst	MDS	4.5†	5
Midori	Not MDS‡	1.5	4
Minalpher	Not MDS‡	1.5	4
Prince	Not MDS	1.5	4
SKINNY	Not MDS	0.75	2
Keccak-f	CPM	2	4
Circulant CPM	CPM	$2 + \frac{\lvert z(x)\rvert - 2}{m}$ *	4

* XORs/bit $\in [2 - 1/m, 2 + (n-2)/m]$

† Unknown whether it can be computed with less XORs

‡ Can also be considered to be a CPM!

CPM example

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

CPM example

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

$$\Leftrightarrow$$

$$m = 2, Z = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Building a permutation with a CPM

1. Determine design goals

Building a permutation with a CPM

1. Determine design goals
2. Pick m, n, and cell width

Building a permutation with a CPM

1. Determine design goals
2. Pick m, n, and cell width
3. Pick ‘good’ ‘efficient’ non-linear S-box

Building a permutation with a CPM

1. Determine design goals
2. Pick m, n, and cell width
3. Pick 'good' 'efficient' non-linear S-box
4. Consider (truncated) trails in the kernel (independent of Z)

Building a permutation with a CPM

1. Determine design goals
2. Pick m, n, and cell width
3. Pick ‘good’ ‘efficient’ non-linear S-box
4. Consider (truncated) trails in the kernel (independent of Z)
5. Determine ‘good’ transposition

Building a permutation with a CPM

1. Determine design goals
2. Pick m, n, and cell width
3. Pick ‘good’ ‘efficient’ non-linear S-box
4. Consider (truncated) trails in the kernel (independent of Z)
5. Determine ‘good’ transposition
6. Consider (truncated) trails outside the kernel

Building a permutation with a CPM

1. Determine design goals
2. Pick m, n, and cell width
3. Pick 'good' 'efficient' non-linear S-box
4. Consider (truncated) trails in the kernel (independent of Z)
5. Determine 'good' transposition
6. Consider (truncated) trails outside the kernel
7. Determine 'good' Z

Building a permutation with a CPM

1. Determine design goals
2. Pick m, n, and cell width
3. Pick ‘good’ ‘efficient’ non-linear S-box
4. Consider (truncated) trails in the kernel (independent of Z)
5. Determine ‘good’ transposition
6. Consider (truncated) trails outside the kernel
7. Determine ‘good’ Z
8. Pick ‘good’ round constants to beat all kinds of invariant attacks [BCLR17]

Building a permutation with a CPM

1. Determine design goals
2. Pick m, n, and cell width
3. Pick 'good' 'efficient' non-linear S-box
4. Consider (truncated) trails in the kernel (independent of Z)
5. Determine 'good' transposition
6. Consider (truncated) trails outside the kernel
7. Determine 'good' Z
8. Pick 'good' round constants to beat all kinds of invariant attacks [BCLR17]
9. Do more analysis

Building a permutation with a CPM

1. Determine design goals
2. Pick m, n, and cell width
3. Pick 'good' 'efficient' non-linear S-box
4. Consider (truncated) trails in the kernel (independent of Z)
5. Determine 'good' transposition
6. Consider (truncated) trails outside the kernel
7. Determine 'good' Z
8. Pick 'good' round constants to beat all kinds of invariant attacks [BCLR17]
9. Do more analysis
10. Determine the number of rounds

Building a permutation with a CPM

1. Determine design goals
2. Pick m, n, and cell width
3. Pick 'good' 'efficient' non-linear S-box
4. Consider (truncated) trails in the kernel (independent of Z)
5. Determine 'good' transposition
6. Consider (truncated) trails outside the kernel
7. Determine 'good' Z
8. Pick 'good' round constants to beat all kinds of invariant attacks [BCLR17]
9. Do more analysis
10. Determine the number of rounds
11. Implement it

Building a permutation with a CPM

1. Determine design goals
2. Pick m, n, and cell width
3. Pick 'good' 'efficient' non-linear S-box
4. Consider (truncated) trails in the kernel (independent of Z)
5. Determine 'good' transposition
6. Consider (truncated) trails outside the kernel
7. Determine 'good' Z
8. Pick 'good' round constants to beat all kinds of invariant attacks [BCLR17]
9. Do more analysis
10. Determine the number of rounds
11. Implement it
12. Give it a name

(Truncated) trail search

- r-round trail with weight W has differential with weight $L \leq \left\lfloor \frac{W}{r} \right\rfloor$

(Truncated) trail search

- r-round trail with weight W has differential with weight $L \leq \left\lfloor \frac{W}{r} \right\rfloor$
- Observation in [MDA17]: less 2-round trail cores with weight $\leq 2L$ than differentials $\leq L$

(Truncated) trail search

- r-round trail with weight W has differential with weight $L \leq \left\lfloor \frac{W}{r} \right\rfloor$
- Observation in [MDA17]: less 2-round trail cores with weight $\leq 2L$ than differentials $\leq L$
- Generate 2-round trail cores, extend $r - 2$

(Truncated) trail search

- r-round trail with weight W has differential with weight $L \leq \left\lfloor \frac{W}{r} \right\rfloor$
- Observation in [MDA17]: less 2-round trail cores with weight $\leq 2L$ than differentials $\leq L$
- Generate 2-round trail cores, extend $r - 2$
- Model generation as tree traversal, following [MDA17]

(Truncated) trail search

- r-round trail with weight W has differential with weight $L \leq \left\lfloor \frac{W}{r} \right\rfloor$
- Observation in [MDA17]: less 2-round trail cores with weight $\leq 2L$ than differentials $\leq L$
- Generate 2-round trail cores, extend $r - 2$
- Model generation as tree traversal, following [MDA17]
- Use rotational symmetry and monotonically increasing weight for pruning

(Truncated) trail search

- r-round trail with weight W has differential with weight $L \leq \left\lfloor \frac{W}{r} \right\rfloor$
- Observation in [MDA17]: less 2-round trail cores with weight $\leq 2L$ than differentials $\leq L$
- Generate 2-round trail cores, extend $r - 2$
- Model generation as tree traversal, following [MDA17]
- Use rotational symmetry and monotonically increasing weight for pruning
- CPM causes heavy search space branching

(Truncated) trail search

- r-round trail with weight W has differential with weight $L \leq \left\lfloor \frac{W}{r} \right\rfloor$
- Observation in [MDA17]: less 2-round trail cores with weight $\leq 2L$ than differentials $\leq L$
- Generate 2-round trail cores, extend $r - 2$
- Model generation as tree traversal, following [MDA17]
- Use rotational symmetry and monotonically increasing weight for pruning
- CPM causes heavy search space branching
- Dedicated software for CPM-based ciphers/permutations

Mixifer

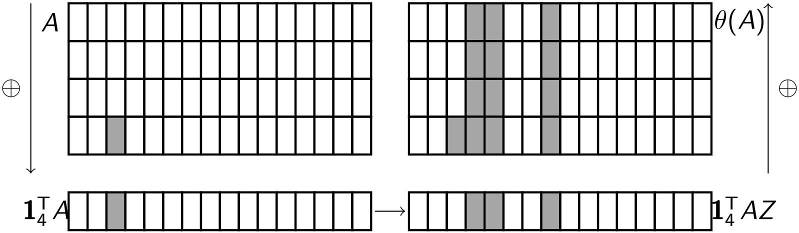
- 16 rounds $(\iota \circ \rho \circ \pi \circ \theta \circ \gamma)$, $4 \times 16 \times 4 = 256$ bits permutation

Mixifer

- 16 rounds ($\iota \circ \rho \circ \pi \circ \theta \circ \gamma$), $4 \times 16 \times 4 = 256$ bits permutation
- γ: rotational symmetric, $b_0 = a_1 + a_2 + a_0 a_2 + a_1 a_2 + a_1 a_2 a_3$

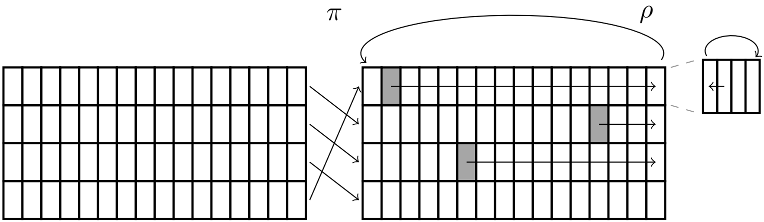
Mixifer

- 16 rounds $(\iota \circ \rho \circ \pi \circ \theta \circ \gamma)$, $4 \times 16 \times 4 = 256$ bits permutation
- γ: rotational symmetric, $b_0 = a_1 + a_2 + a_0 a_2 + a_1 a_2 + a_1 a_2 a_3$
- θ: Z is circulant, first row $[0, 1, 1, 0, 0, 1, 0, 0, 0, \ldots, 0]$



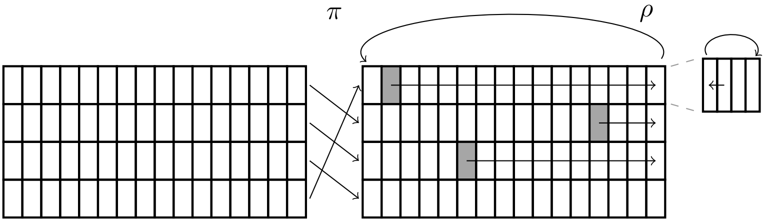
Mixifer

- 16 rounds $(\iota \circ \rho \circ \pi \circ \theta \circ \gamma)$, $4 \times 16 \times 4 = 256$ bits permutation
- γ: rotational symmetric, $b_0 = a_1 + a_2 + a_0 a_2 + a_1 a_2 + a_1 a_2 a_3$
- θ: Z is circulant, first row $[0, 1, 1, 0, 0, 1, 0, 0, 0, \ldots, 0]$
- π: rotate rows down



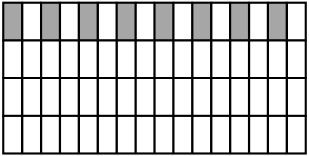
Mixifer

- 16 rounds $(\iota \circ \rho \circ \pi \circ \theta \circ \gamma)$, $4 \times 16 \times 4 = 256$ bits permutation
- γ: rotational symmetric, $b_0 = a_1 + a_2 + a_0 a_2 + a_1 a_2 + a_1 a_2 a_3$
- θ: Z is circulant, first row $[0, 1, 1, 0, 0, 1, 0, 0, 0, \ldots, 0]$
- π: rotate rows down
- ρ: rotate rows cell-wise to the right by $\{14, 3, 10, 0\}$



Mixifer

- 16 rounds $(\iota \circ \rho \circ \pi \circ \theta \circ \gamma)$, $4 \times 16 \times 4 = 256$ bits permutation
- γ: rotational symmetric, $b_0 = a_1 + a_2 + a_0 a_2 + a_1 a_2 + a_1 a_2 a_3$
- θ: Z is circulant, first row $[0, 1, 1, 0, 0, 1, 0, 0, 0, \ldots, 0]$
- π: rotate rows down
- ρ: rotate rows cell-wise to the right by $\{14, 3, 10, 0\}$
- ι: add $\texttt{0xF3485763} \gg i$ in round i to every other cell of top row



Mixifer analysis

- Strict avalanche criterion after 3 rounds, full diffusion after 5

Mixifer analysis

- Strict avalanche criterion after 3 rounds, full diffusion after 5
- After 4 rounds:

Mixifer analysis

- Strict avalanche criterion after 3 rounds, full diffusion after 5
- After 4 rounds:
  - In kernel: ≥ 52 active cells

Mixifer analysis

- Strict avalanche criterion after 3 rounds, full diffusion after 5
- After 4 rounds:
  - In kernel: ≥ 52 active cells
  - Outside kernel: ≥ 46 active cells (differential), DP 2^{-92}

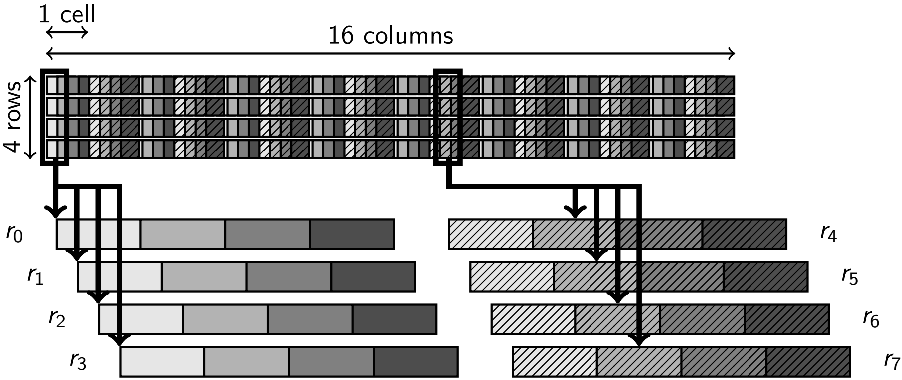
Mixifer analysis

- Strict avalanche criterion after 3 rounds, full diffusion after 5
- After 4 rounds:
  - In kernel: ≥ 52 active cells
  - Outside kernel: ≥ 46 active cells (differential), DP 2^{-92}
  - Outside kernel: ≥ 40 active cells (linear), LP 2^{-80}

Mixifer analysis

- Strict avalanche criterion after 3 rounds, full diffusion after 5
- After 4 rounds:
  - In kernel: ≥ 52 active cells
  - Outside kernel: ≥ 46 active cells (differential), DP 2^{-92}
  - Outside kernel: ≥ 40 active cells (linear), LP 2^{-80}
- Preliminary study makes us believe trail clustering, impossible differentials, invariant attacks are not a concern

Mixifer implementation



Mixifer comparison (ARM Cortex-M4)

Cipher	Width (bits)	r	Speed (cpb) Full	Speed (cpb) $/r$	Bound trails r	Bound trails W	Bound trails $/r$
AES bitsliced	128	10	50.52	5.05	4	150	37.5
AES tables			39.97	4.00			
Gimli	384	24	21.81	0.91	8	52	6.5
Keccak-f[400]	400	20	106	5.3	6	92	15.3
Keccak-f[800]	800	22	48.02	2.18	6	92	15.3
Salsa20/20	512	20	13.88	0.69	3	18	6
Mixifer	256	16	36.69	2.33	4	92	23

Thanks. . .

. . . for your attention

Questions?

References I

Christof Beierle, Anne Canteaut, Gregor Leander, and Yann Rotella.
Proving resistance against invariant attacks: How to choose the round constants.
In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 647–678, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

Silvia Mella, Joan Daemen, and Gilles Van Assche.
New techniques for trail bounds and application to differential trails in Keccak.
IACR Transactions on Symmetric Cryptology, 2017(1):329–357, 2017.