Inside Keccak

Guido Bertoni[1] Joan Daemen[1]
Michaël Peeters[2] Gilles Van Assche[1]

[1]STMicroelectronics

[2]NXP Semiconductors

Keccak & SHA-3 Day
Université Libre de Bruxelles
March 27, 2013

Outline

Outline

1. Defining KECCAK
2. Differential and linear trail propagation
3. Alignment
4. Bounding differential and linear trail weights
5. The kernel

The beginning

- SUBTERRANEAN: Daemen (1991)
  - variable-length input and output
  - hashing and stream cipher
  - round function interleaved with input/output
- STEPRIGHTUP: Daemen (1994)
- PANAMA: Daemen and Clapp (1998)
- RADIOGATÚN: Bertoni, Daemen, Peeters and VA (2006)
  - experiments did not inspire confidence in RADIOGATÚN
  - neither did third-party cryptanalysis
    [Bouillaguet, Fouque, SAC 2008] [Fuhr, Peyrin, FSE 2009]
  - NIST SHA-3 deadline approaching ...
  - U-turn: design a sponge with strong permutation f
- KECCAK (2008)

Designing the permutation KECCAK-f

Our mission

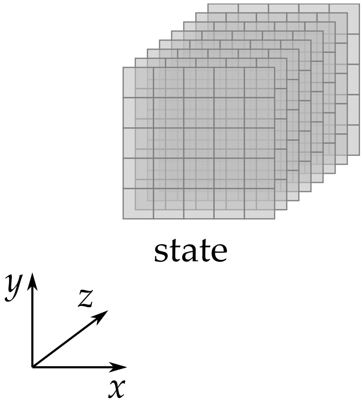
To design a permutation called KECCAK-f that cannot be distinguished from a random permutation.

- Like a block cipher
  - sequence of identical rounds
  - round function that is nonlinear and has good diffusion
- ...but not quite
  - no need for key schedule
  - round constants instead of round keys
  - inverse permutation need not be efficient

KECCAK

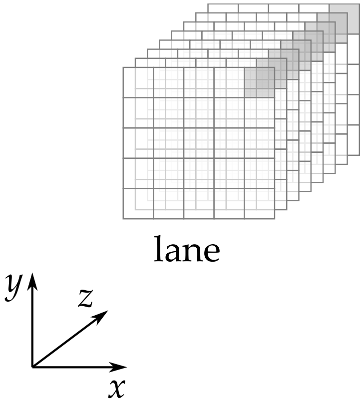
- Instantiation of a *sponge function*
- the permutation KECCAK-f
  - 7 permutations: $b \in \{25, 50, 100, 200, 400, 800, 1600\}$
- Security-speed trade-offs using the same permutation, e.g.,
  - SHA-3 instance: $r = 1088$ and $c = 512$
    - permutation width: 1600
    - security strength 256: post-quantum sufficient
  - Lightweight instance: $r = 40$ and $c = 160$
    - permutation width: 200
    - security strength 80: same as SHA-1

The state: an array of $5 \times 5 \times 2^\ell$ bits



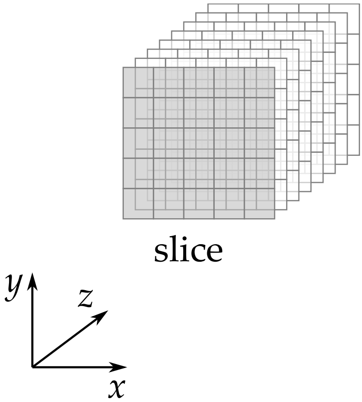
- 5×5 lanes, each containing 2^ℓ bits (1, 2, 4, 8, 16, 32 or 64)
- (5×5)-bit slices, 2^ℓ of them

The state: an array of $5 \times 5 \times 2^{\ell}$ bits



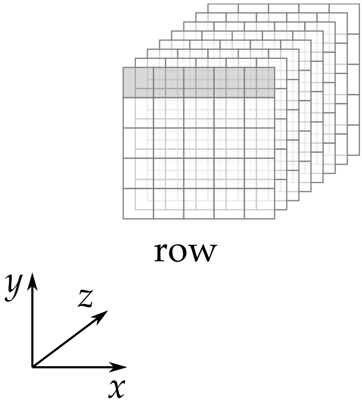
- 5×5 lanes, each containing 2^{ℓ} bits (1, 2, 4, 8, 16, 32 or 64)
- (5×5)-bit slices, 2^{ℓ} of them

The state: an array of $5 \times 5 \times 2^{\ell}$ bits



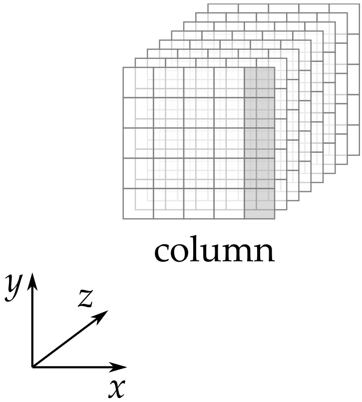
- 5×5 lanes, each containing 2^{ℓ} bits (1, 2, 4, 8, 16, 32 or 64)
- (5×5)-bit slices, 2^{ℓ} of them

The state: an array of $5 \times 5 \times 2^{\ell}$ bits



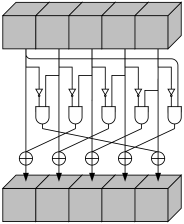
- 5×5 lanes, each containing 2^{ℓ} bits (1, 2, 4, 8, 16, 32 or 64)
- (5×5)-bit slices, 2^{ℓ} of them

The state: an array of $5 \times 5 \times 2^\ell$ bits



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

χ, the nonlinear mapping in KECCAK-f

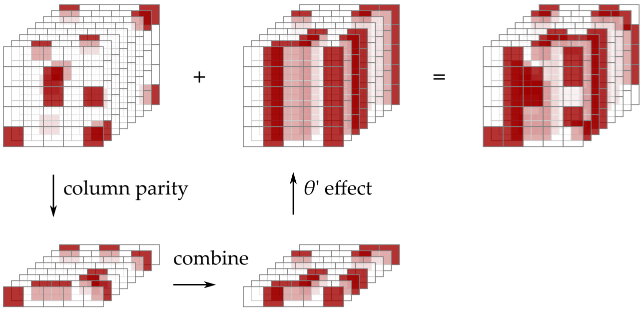


- "Flip bit if neighbors exhibit 01 pattern"
- Operates independently and in parallel on 5-bit rows
- Algebraic degree 2, inverse has degree 3
- LC/DC propagation properties easy to describe and analyze

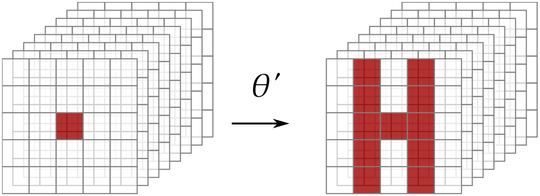
θ', a first attempt at mixing bits

- Compute parity $c_{x,z}$ of each column
- Add to each cell parity of neighboring columns:

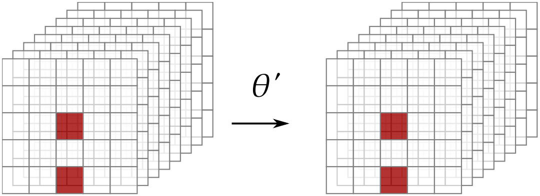
$$b_{x,y,z} = a_{x,y,z} \oplus c_{x-1,z} \oplus c_{x+1,z}$$



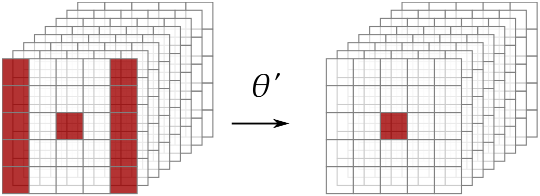
Diffusion of θ'



Diffusion of θ' (kernel)



Diffusion of the inverse of θ'

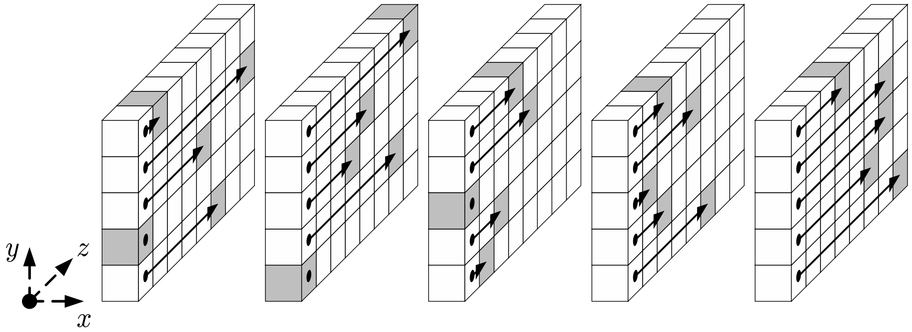


ρ for inter-slice dispersion

- We need diffusion between the slices …
- ρ: cyclic shifts of lanes with offsets

$$i(i+1)/2 \bmod 2^{\ell}$$

- Offsets cycle through all values below 2^{ℓ}

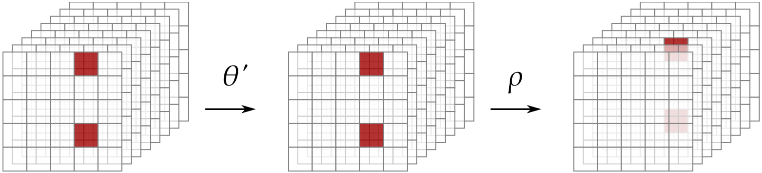


ι to break symmetry

- XOR of round-dependent constant to lane in origin
- Without ι, the round mapping would be symmetric
  - invariant to translation in the z-direction
- Without ι, all rounds would be the same
  - susceptibility to *slide* attacks
  - defective cycle structure
- Without ι, we get simple fixed points (000 and 111)

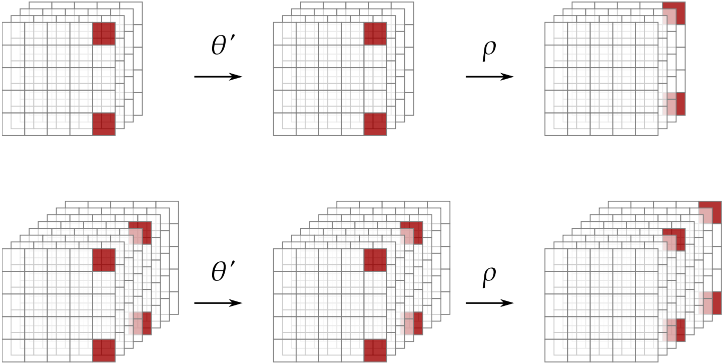
A first attempt at KECCAK-f

- Round function: $\mathrm{R} = \iota \circ \rho \circ \theta' \circ \chi$
- Problem: low-weight periodic trails by chaining:



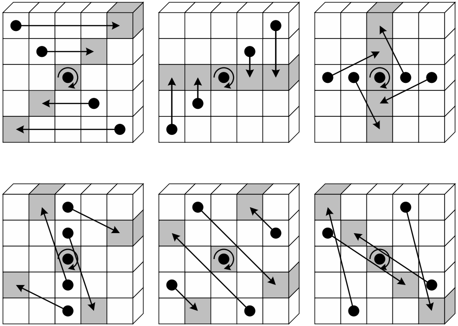
  - χ: may propagate unchanged
  - θ': propagates unchanged, because all column parities are 0
  - ρ: in general moves active bits to different slices ...
  - ...but not always

The Matryoshka property



- Patterns in Q' are z-periodic versions of patterns in Q

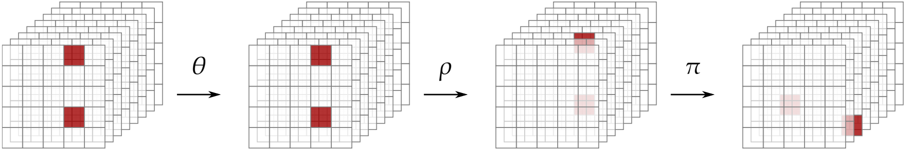
π for disturbing horizontal/vertical alignment



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

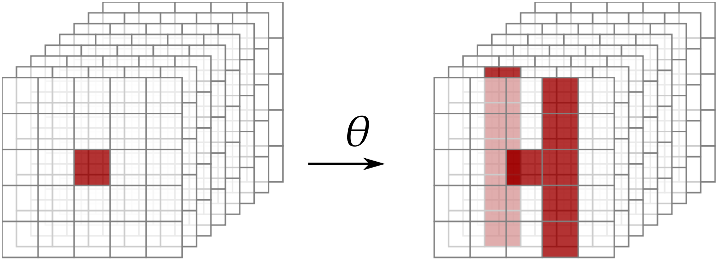
A second attempt at Keccak-f

- Round function: $\mathrm{R} = \iota \circ \pi \circ \rho \circ \theta' \circ \chi$
- Solves problem encountered before:



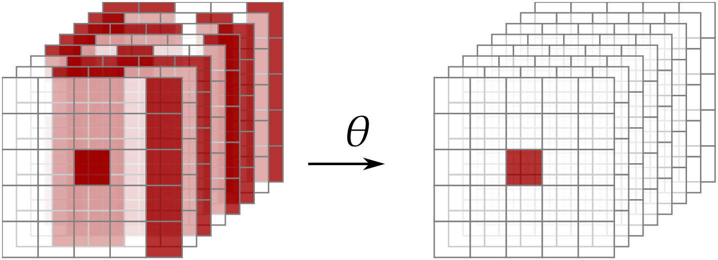
- π moves bits in same column to different columns!

Tweaking θ' to θ



$$b_{x,y,z} = a_{x,y,z} \oplus c_{x-1,z} \oplus c_{x+1,z-1}$$

Inverse of θ



- Diffusion from single-bit output to input very high
- Increases resistance against LC/DC and algebraic attacks

KECCAK-f summary

Round function

$$\text{round} = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$

- Number of rounds: $12 + 2\ell$
  - KECCAK-$f[25]$ has 12 rounds
  - KECCAK-$f[1600]$ has 24 rounds

Design decisions behind KECCAK-f

- Ability to control propagation of differences or linear masks
  - Differential/linear trail analysis
  - Lower bounds for trail weights
  - Alignment and trail clustering
  - $\Rightarrow$ This shaped θ, π and ρ
- Algebraic properties
  - Distribution of $\#$ terms of certain degrees
  - Ability of solving certain problems (CICO) algebraically
  - Zero-sum distinguishers (third party)
  - $\Rightarrow$ This determined the number of rounds
- Analysis of symmetry properties
  $\Rightarrow$ This shaped ι

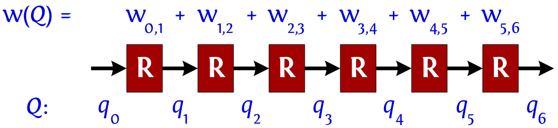
Design decisions behind KECCAK-f

- Ability to control propagation of differences or linear masks
  - **Differential/linear trail analysis**
  - **Lower bounds for trail weights**
  - **Alignment and trail clustering**
  - $\Rightarrow$ This shaped θ, π and ρ
- Algebraic properties
  - Distribution of $\#$ terms of certain degrees
  - Ability of solving certain problems (CICO) algebraically
  - Zero-sum distinguishers (third party)
  - $\Rightarrow$ This determined the number of rounds
- Analysis of symmetry properties

  $\Rightarrow$ This shaped ι

Outline

Differential and linear trails in iterated mappings



- Differential trail: sequence of differences

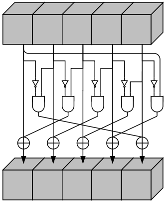
$$\text{weight} = -\log_2(\text{fraction of pairs})$$

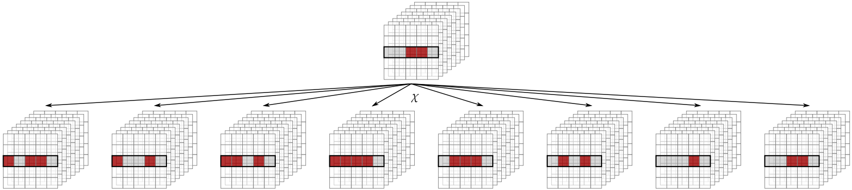
- Linear trail: sequence of linear masks

$$\text{weight} = -2\log_2(\text{correlation contribution})$$

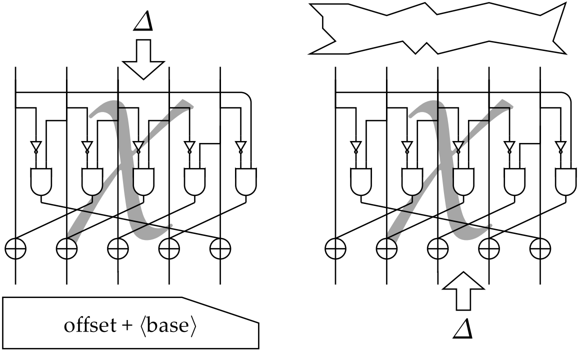
Non-linear mapping χ

- Transforms each **row** independently
- E.g., a difference going through χ
  - Output: affine space



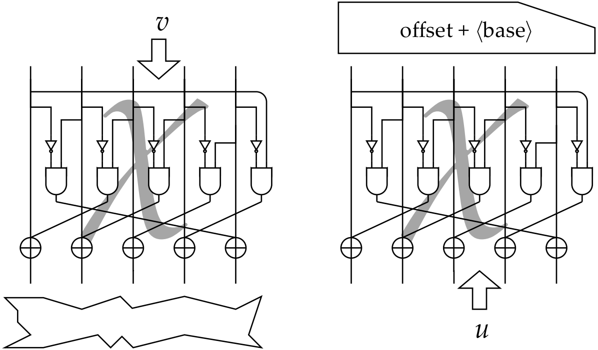


Propagating differences through χ



- The propagation weight...
  - ... is determined by input difference only;
  - ... is the size of the affine base;
  - ... is the number of affine conditions.

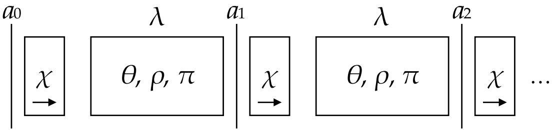
Propagating linear masks through χ



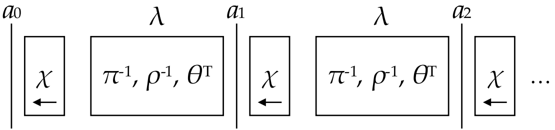
- The propagation weight...
  - ... is determined by output mask only;
  - ... is the size of the affine base.

Differential and linear trails in KECCAKTOOLS

- KECCAKTOOLS
  - *A set of documented C++ classes to help analyze* KECCAK
    Freely available on http://keccak.noekeon.org
  - Implements differential and linear trail propagation
- KeccakFPropagation works in "affine" direction:
  - Differential trails



  - Linear trails: forward propagation means backwards in time



Outline

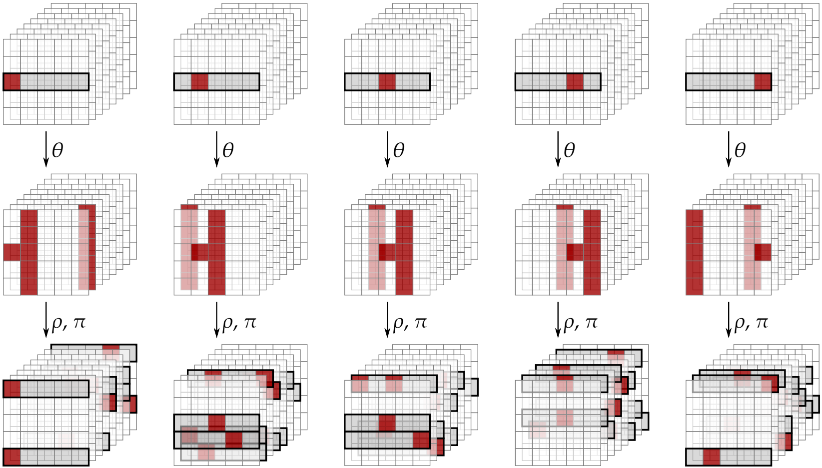
Difference propagation in RIJNDAEL

- Differential trail (fully specified)
  - Deterministic propagation through MixColumns, ShiftRows and AddRoundKey
  - Branching through SubBytes
- Truncated diff. trail specifying active/passive s-boxes
  - Deterministic propagation through SubBytes, ShiftRows and AddRoundKey
  - Branching through MixColumns
    - Sometimes deterministic: 1 byte $\rightarrow$ 4 bytes

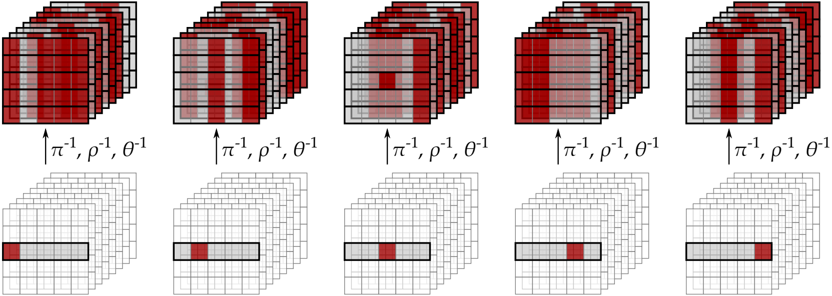
Alignment

- Property of round function
  - relative to partition of state in blocks
- **Strong alignment**
  - Low uncertainty in propagation along block boundaries
  - E.g., RIJNDAEL strongly aligned on byte boundaries
- Weak alignment
  - High uncertainty in propagation along block boundaries
  - E.g., KECCAK weakly aligned on row boundaries…

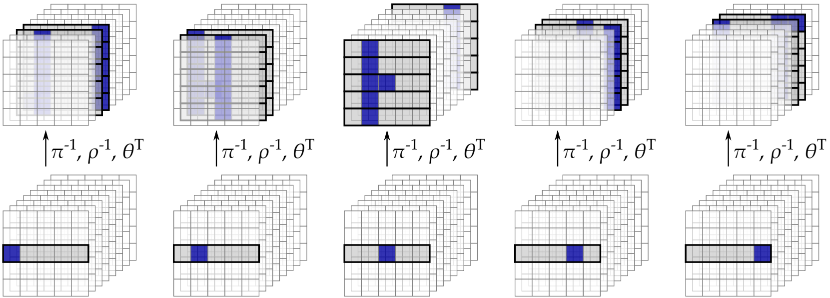
Differential patterns



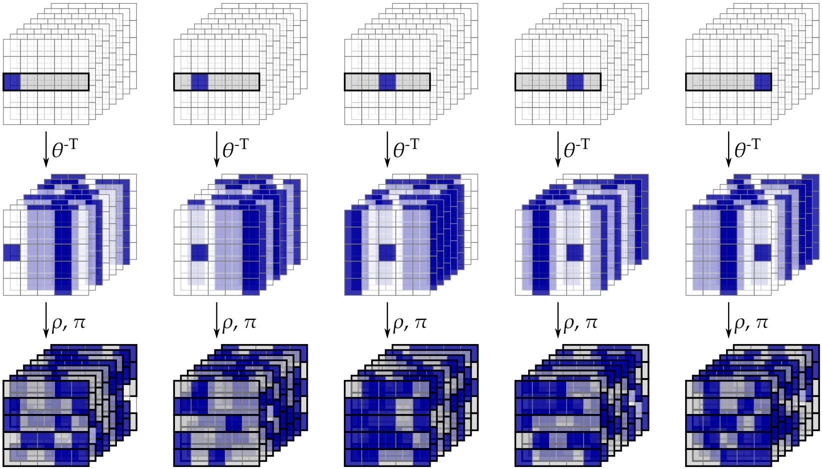
Differential patterns (backwards)



Linear patterns



Linear patterns (backwards)



Benefits of weak alignment

Weak alignment means trails tend to diverge

- Low clustering of trails
  - Differential $b'_0 \rightarrow b'_2$, with $\mathrm{DP}(b'_0, b'_2) = \sum_{b'_1} \mathrm{DP}(b'_0, b'_1, b'_2)$
    - $b'_0 \stackrel{\lambda,\chi}{\rightarrow} b'_1 \stackrel{\lambda,\chi}{\rightarrow} b'_2$
    - $\mathrm{DP} \neq 0 \quad \Rightarrow \quad \mathrm{row}(\lambda(b'_0)) = \mathrm{row}(b'_1) \wedge \mathrm{row}(\lambda(b'_1)) = \mathrm{row}(b'_2)$
    - Weak alignment: not many b'_1 values satisfy this
- Hard to build a truncated differential trail
- Hard to mount a rebound attack
  - See also [Duc et al., Unaligned Rebound Attack: Appl. to KECCAK, FSE 2012]

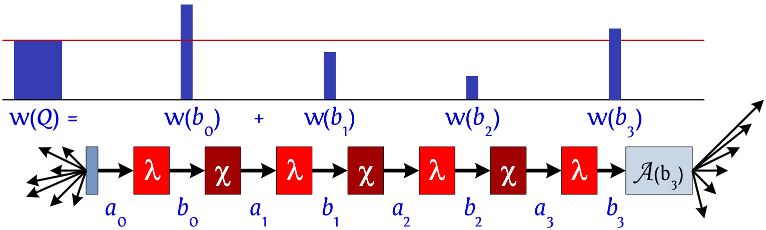
Outline

Why bound trail weights?

- Security of KECCAK relies on **absence** of exploitable trails
  …and not on presumed hardness of finding them
  $\Rightarrow$ Bound differential and linear trails as tightly as possible

How to bound trail weights?

- Bounds vs design strategies
  - ARX: no relevant bounds
  - RIJNDAEL-based: strong and simply provable bounds, **but**
    - Not for truncated differentials and rebound attack
  - Weak alignment: computer-assisted proofs are possible



- Inspired by similar efforts for
  - Noekeon [Nessie, 2000]
  - MD6 [Rivest et al., SHA-3 2008] [Heilman, Ecrypt Hash 2011]

Bounds for small instances of KECCAK

Number of rounds	Differential trails			
	$w = 1$	$w = 2$	$w = 4$	$w = 8$
2	8	8	8	8
3	16	18	19	20
4	23	29	30	46
5	30	42	≤ 54	
6	37	54	≤ 85	
16			≥ 148	
18				≥ 208

Table: Minimum weight of w-symmetric differential trails

Bounds for small instances of KECCAK

Number of rounds	Linear trails			
	$w = 1$	$w = 2$	$w = 4$	$w = 8$
2	8	8	8	8
3	16	16	20	20
4	24	30	38	46
5	30	40	≤ 66	
6	38	52	≤ 94	
16			≥ 152	
18				≥ 208

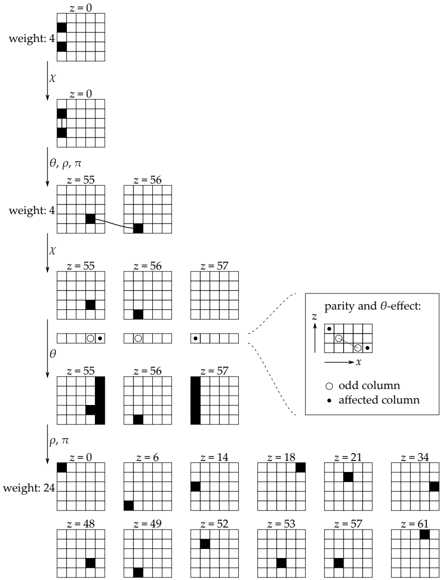
Table: Minimum weight of w-symmetric linear trails

Bounds for differential trails in KECCAK-f[1600]

Rounds	Lower bound		Best known	
1	2		2	
2	8		8	
3	32	[KECCAK team]	32	[Duc et al.]
4			134	[KECCAK team]
5			510	[Naya-Plasencia et al.]
6	74	[KECCAK team]	1360	[KECCAK team]
24	296		???	

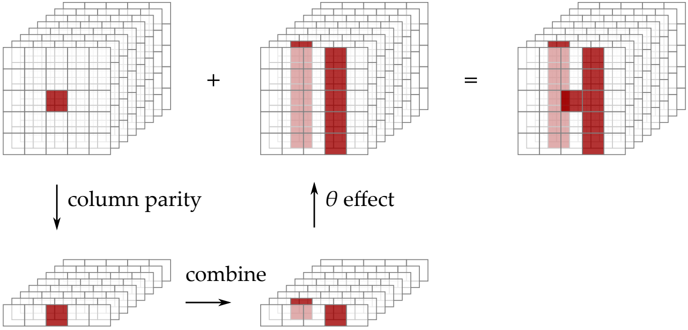
- Pessimistic view
  - Wide gap between bounds and known trails
    Open problem: narrow this gap (and also for linear trails)
  - Bound too loose to prove ideal behavior
- Optimistic view
  - Proven absence of exploitable differential trail
  - Trail weight apparently growing quickly with number of rounds

The best 3-round differential trail in KECCAK-f[1600]



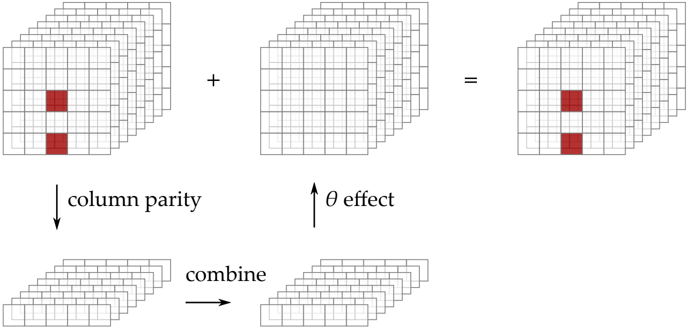
Outline

Reminder: θ, the mixing layer



- Single-bit parity flips already 10 bits
- Other linear mapping ρ and π just move bits around

Reminder: θ, the mixing layer

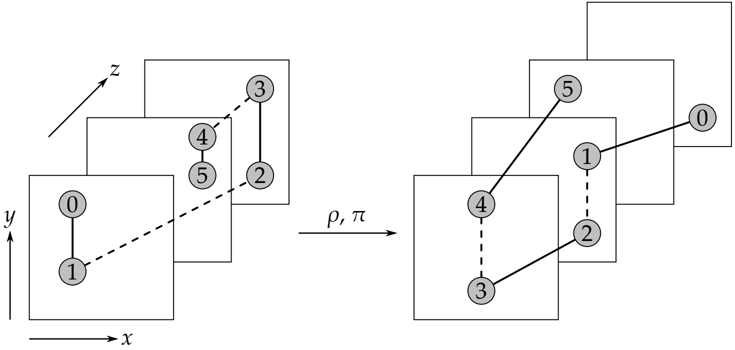


- Effect collapses if parity is zero
- The kernel

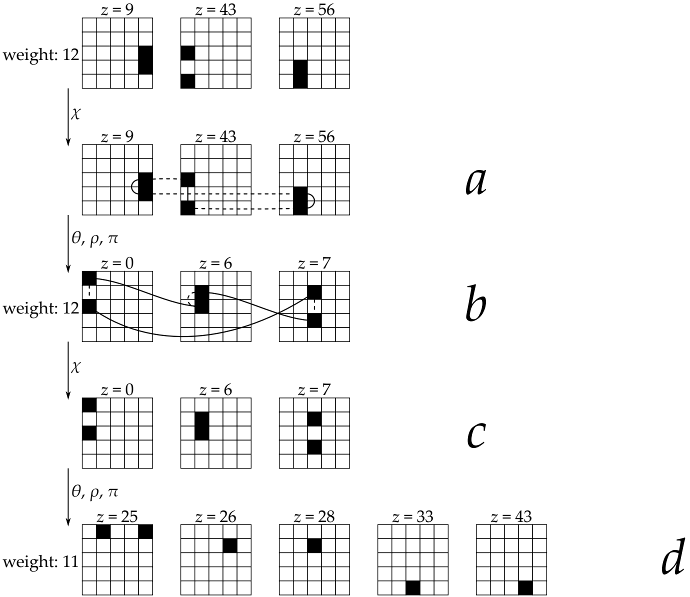
Chains

Sequence of active bits p_i with:

- p_{2i} and p_{2i+1} are in same column in a
- p_{2i+1} and p_{2i} are in same column in b



An in-kernel 3-round trail with a vortex



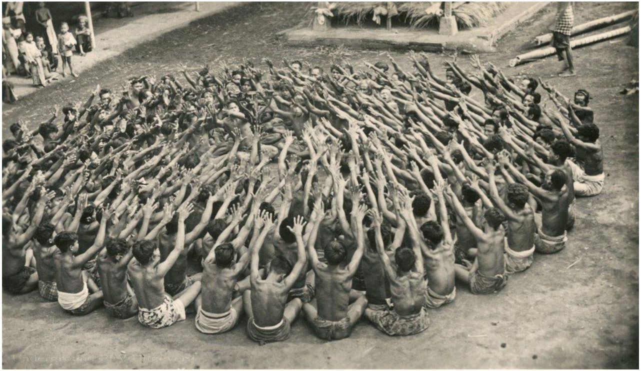
The kernel: an undesired property?

In-kernel vs non-kernel trails

- All trails (both in-kernel and non-kernel):
  - Scanned 3-round trails up to weight 36 (min. found: 32)
  - None extended to 6-round trails with weight below 74
- **In-kernel** trails:
  - Scanned 3-round trails up to weight **54** (min. found: **35**)
  - None extended to 6-round trails with weight below **82**

- Pessimistic view
  - The kernel makes θ act as the identity, clearly an undesired property
- Optimistic view
  - Staying in the kernel constrains the attacker
  - Bounds are easier to prove in the kernel

Questions?



http://sponge.noekeon.org/
http://keccak.noekeon.org/