Collisions on Keccak

An Other Cryptanalysis on Keccak !

Yann Rotella

Paris-Saclay University, LMV

7 May 2021

Laboratoire de mathématiques de Versailles - CNRS UMR 8100

OUTLINE

KECCAK :

- A familiy of Hash functions designed by Guido Bertoni, Joan Daemen, Michaël Peeters and Gilles Van Assche
- Winner in 2012 of NIST competition.
- Some instances standardized : SHA-3

Outline

KECCAK :

- A familiy of Hash functions designed by Guido Bertoni, Joan Daemen, Michaël Peeters and Gilles Van Assche
- Winner in 2012 of NIST competition.
- Some instances standardized : SHA-3

This work :

- Rachelle Heim, Camille Noûs and Yann Rotella
- Crunchy Contest
  `https://keccak.team/crunchy_contest.html`

OUTLINE

KECCAK :

- A familiy of Hash functions designed by Guido Bertoni, Joan Daemen, Michaël Peeters and Gilles Van Assche
- Winner in 2012 of NIST competition.
- Some instances standardized : SHA-3

This work :

- Rachelle Heim, Camille Noûs and Yann Rotella
- Crunchy Contest
  `https://keccak.team/crunchy_contest.html`
- *Algebraic Collision Attacks on Keccak*, ToSC 2021 (Issue 1).

OUTLINE

KECCAK :

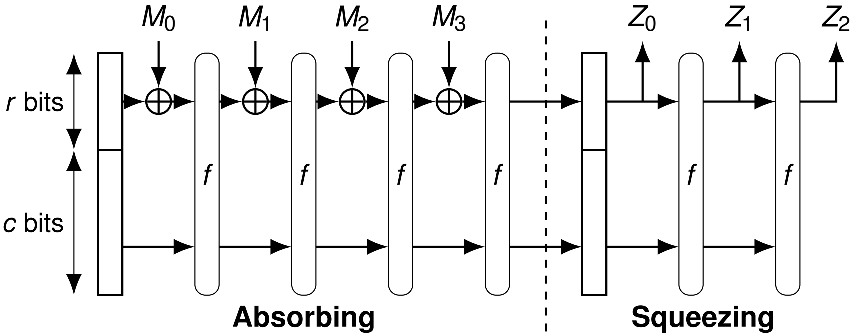
- A familiy of Hash functions designed by Guido Bertoni, Joan Daemen, Michaël Peeters and Gilles Van Assche
- Winner in 2012 of NIST competition.
- Some instances standardized : SHA-3

This work :

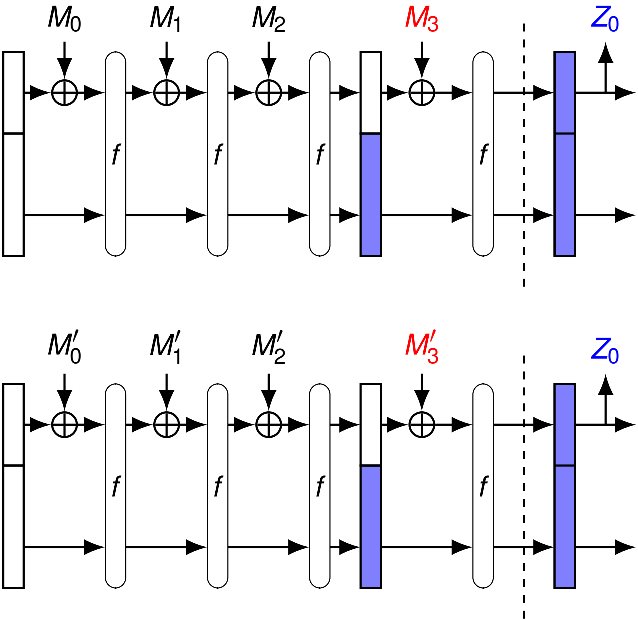
- Rachelle Heim, Camille Noûs and Yann Rotella
- Crunchy Contest
  `https://keccak.team/crunchy_contest.html`
- *Algebraic Collision Attacks on Keccak*, ToSC 2021 (Issue 1).
- FSE 2022 (Hopefully in Greece)

The Sponge Construction

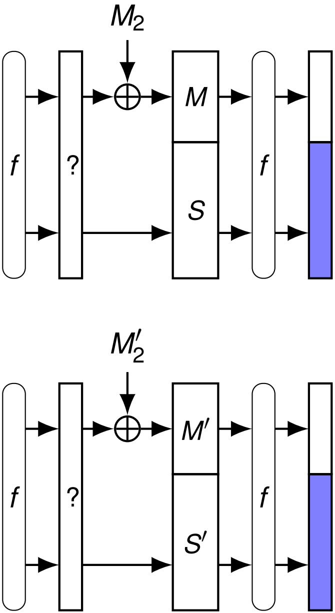
$f : \mathbb{F}_2^b \to \mathbb{F}_2^b$
$b = r + c$
r is the rate and c is the capacity.



Inner Collisions



OUR TARGET



Our Target

Equivalent to solve

$$\begin{cases} f_0(m_0,\ldots,m_{r-1},s_0,\ldots,s_{c-1}) = f_0(m'_0,\ldots,m'_{r-1},s'_0,\ldots,s'_{c-1}) \\ f_1(m_0,\ldots,m_{r-1},s_0,\ldots,s_{c-1}) = f_1(m'_0,\ldots,m'_{r-1},s'_0,\ldots,s'_{c-1}) \\ \cdots \\ f_{c-1}(m_0,\ldots,m_{r-1},s_0,\ldots,s_{c-1}) = f_{c-1}(m'_0,\ldots,m'_{r-1},s'_0,\ldots,s'_{c-1}) \end{cases} \tag{$\mathcal{S}$}$$

Where s_i and s'_i are "random".

$f =$ KECCAK-$p[b, n_r]$ act on a state of size $b = 25 \times \omega$ where $\omega \in \{8, 16, 32, 64\}$

Row

Column

Line

Slice

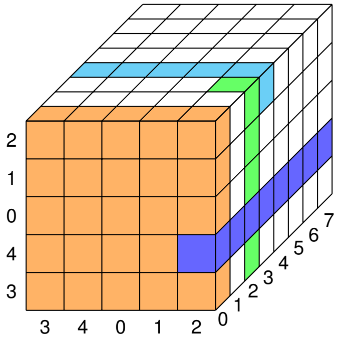
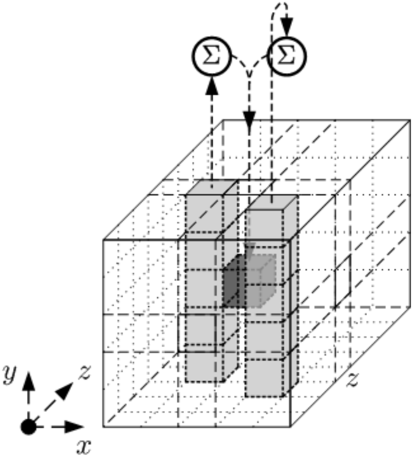


FIGURE – KECCAK state where $\omega = 8$

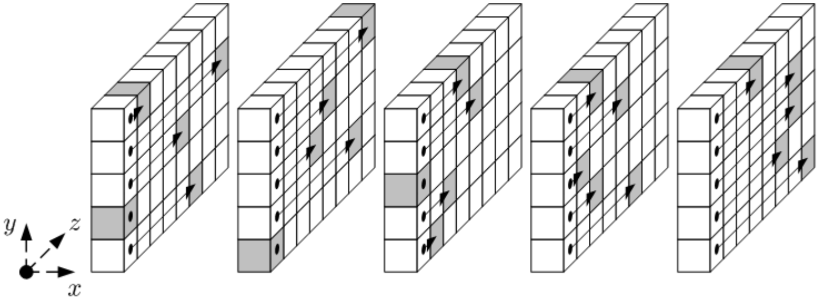
A KECCAK ROUND

$$R = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$

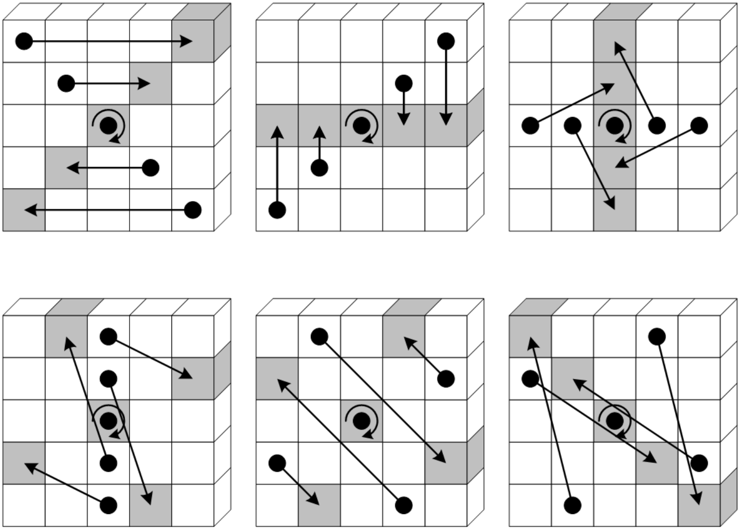
Theta



RHO

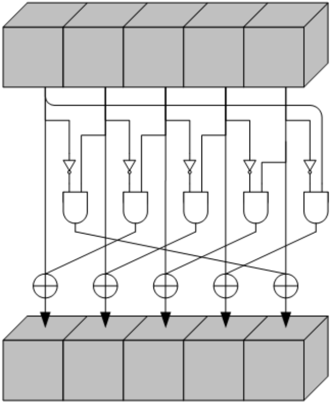


Pi

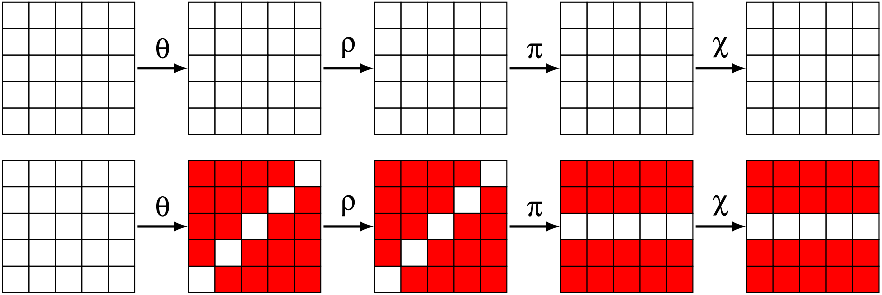


Chi

$$b_i = a_i + (a_{i+1} + 1)a_{i+2}$$



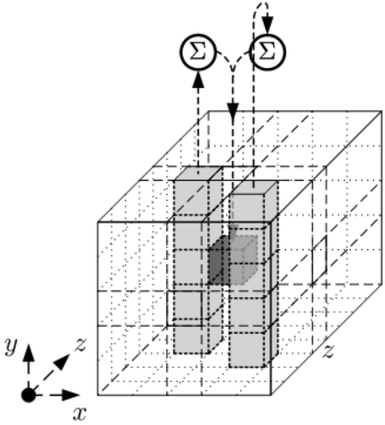
Almost one round for free



THETA PROPERTY

$$b_1 = \theta(a_1) = a_1 + c$$
$$b_2 = \theta(a_2) = a_2 + c$$

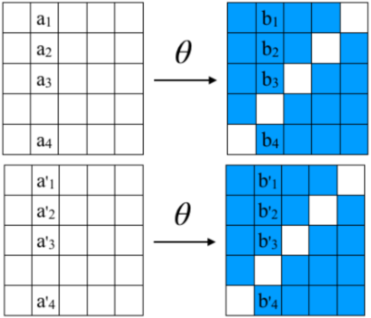
$$b_1 + b_2 = a_1 + a_2$$



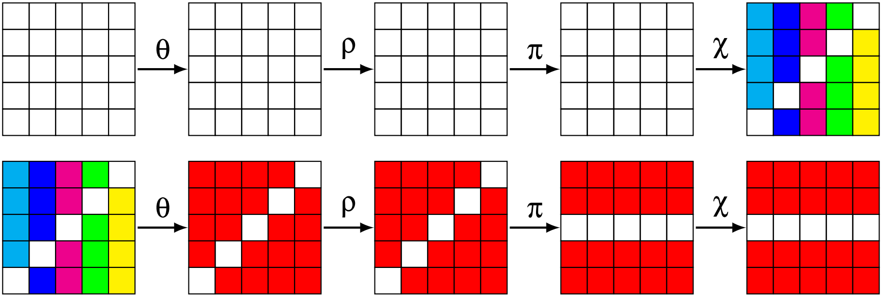
EQUIVALENT SYSTEM

$$\begin{cases} b_1 = b'_1 \\ b_2 = b'_2 \\ b_3 = b'_3 \\ b_4 = b'_4 \end{cases} \Leftrightarrow \begin{cases} b_1 = b'_1 \\ b_1 + b_2 = b'_1 + b'_2 \\ b_2 + b_3 = b'_2 + b'_3 \\ b_3 + b_4 = b'_3 + b'_4 \end{cases}$$

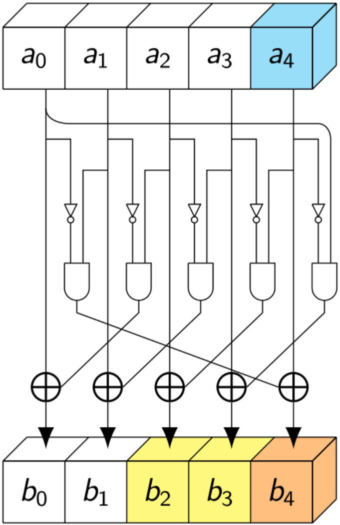
$$\Leftrightarrow \begin{cases} b_1 = b'_1 \\ a_1 + a_2 = a'_1 + a'_2 \\ a_2 + a_3 = a'_2 + a'_3 \\ a_3 + a_4 = a'_3 + a'_4 \,. \end{cases}$$



ONE ROUND FOR $2^{5\omega}$



Chi Linearisation

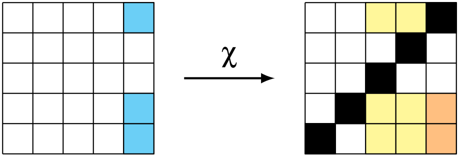


$$\boxed{b_2} = a_2 + (a_3 + 1) \times \boxed{a_4}$$

$$\boxed{b_3} = a_3 + (\boxed{a_4} + 1) \times a_0$$

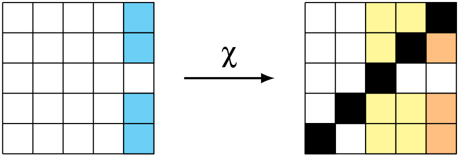
$$\boxed{b_4} = \boxed{a_4} + (a_0 + 1) \times a_1$$

The attack



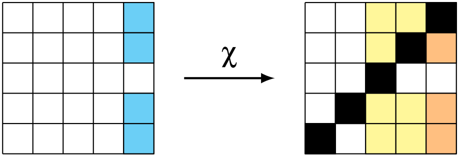
- Fix $3n$ bits in blue before θ ($3n$ equations)
- Add equations related to bits in yellow ($4n$ equations)
- $2n$ bits are constant with probability $\frac{5}{8}$

Time-Memory Trade-Offs



- Fix more bits in blue in each slices

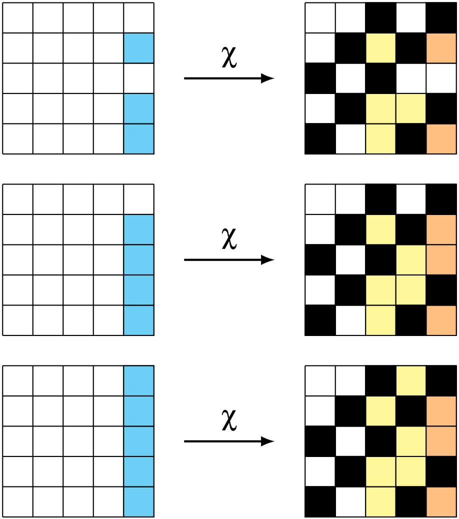
Time-Memory Trade-Offs



- Fix more bits in blue in each slices

But We don't care

ATTACKING ALL INSTANCES



COMPLEXITY

KECCAK[40, 160]	KECCAK[72, 128]	KECCAK[144, 256]
2^{73}	$2^{52.5}$	$2^{101.5}$

COMPLEXITY

KECCAK[40, 160]	KECCAK[72, 128]	KECCAK[144, 256]
2^{73}	$2^{52.5}$	$2^{101.5}$

Verified on KECCAK[30, 70].

Conclusion

- Linearisation helps collision finding
- Needs dedicated cryptanalysis for small instances

Conclusion

- Linearisation helps collision finding
- Needs dedicated cryptanalysis for small instances

Thanks !

Conclusion

- Linearisation helps collision finding
- Needs dedicated cryptanalysis for small instances

Thanks !

And thanks Jeremy Jean, Keccak Team and Rachelle Heim for all figures