Collision attacks on small KECCAK

Rachelle Heim Boissier, Yann Rotella

Paris-Saclay University - Versailles University

24 March 2022

KECCAK hash functions

- KECCAK is a **hash function** designed by Guido Bertoni, Joan Daemen, Michaël Peeters and Gilles Van Assche
- In 2012, four KECCAK instances are standardised as SHA-3
- Permutation-based mode of operation : the **sponge construction**
- Underlying permutation : KECCAK-$f[b]$, b state length in bits
  - Standardised instances : $b = 1600$
  - Instances of interest here : "Small" KECCAK $b = 200$ or $b = 400$

Motivation for the analysis of small Keccak

- **Crunchy contest :** cryptanalysis challenges on round-reduced Keccak instances

  "Remarkably, the smaller versions are harder to break"

- Small Keccak hash functions used in a proposal for RFID [KY10]

Motivation for the analysis of small KECCAK

Function	Rounds	Complexity (Time)
SHA3-224	2	Practical [NRM11] [HMRS12]
	4	Practical [DDS12]
	5	Practical [GLLQS19]
SHA3-256	2	Practical [NRM11]
	4	Practical [DDS12]
	5	2^{115} [DDS13]
		Practical [GLLQS19]
SHA3-384	3	Pactical [DDS13]
	4	2^{147} [DDS13] 2^{60} [HABYDM22]
SHA3-512	3	Practical [DDS13]
Keccak[40,160,1]	1	Practical [WE17]

Summary of our results

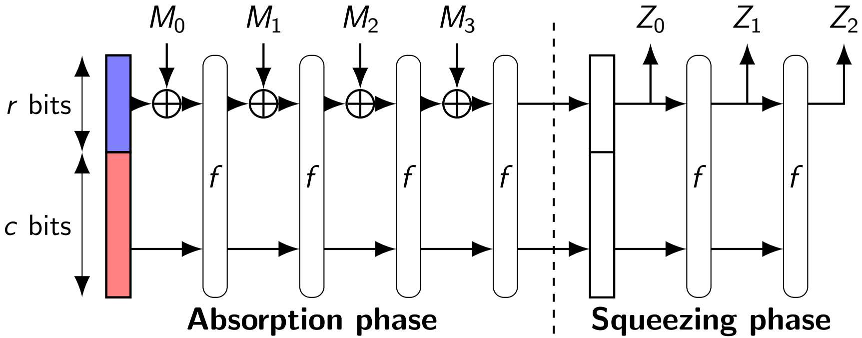
Parameters ($n_r = 2$)	$b = 200$, $c = 160$	$b = 200$, $c = 128$	$b = 400$, $c = 256$
Generic security	2^{80}	2^{64}	2^{128}
Time complexity of our attack	2^{73}	2^{53}	2^{102}

Implementation & verification:

- Attack implemented and verified in C on toy versions ($b = 100$)
- Practical complexities match the theory

Plan

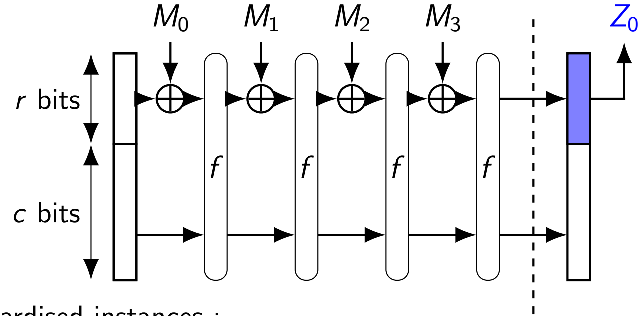
The sponge construction



- Permutation f is applied to a state of length $b = r + c$, where c is the **capacity** and r is the **bitrate**.
- **outer state**
- **inner state**
- d is the ouput length

Generic collision attacks on the sponge mode

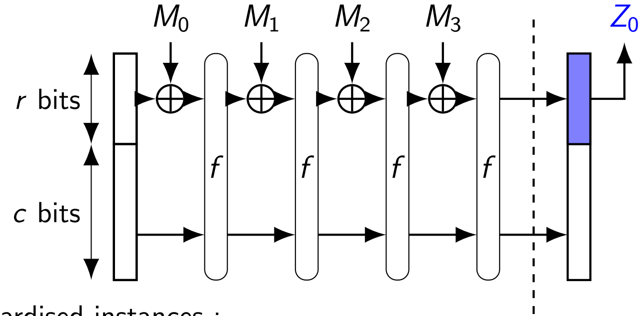
Output collisions



- Standardised instances :
  - $d < r$
  - $c = 2d$
- Small instances :
  - $d > r$: output collision requires several outer state collisions
  - Instead : inner (state) collisions
  - Since $d = c$, same generic security as output collisions

Generic collision attacks on the sponge mode

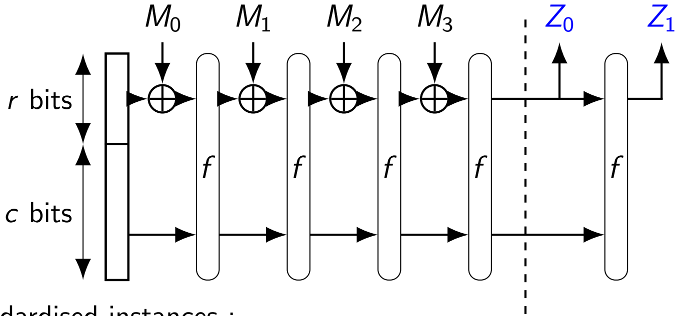
Output collisions



- Standardised instances :
  - $d < r$
  - $c = 2d$
- Small instances :
  - $d > r$: output collision requires several outer state collisions
  - Instead : inner (state) collisions
  - Since $d = c$, same generic security as output collisions

Generic collision attack on the sponge mode

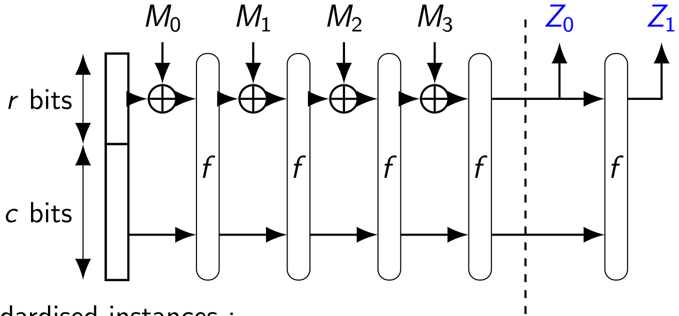
Output collisions



- Standardised instances :
  - $d < r$
  - $c = 2d$
- Small instances :
  - $d > r$: output collision requires several outer state collisions
  - Instead : inner (state) collisions
  - Since $d = c$, same generic security as output collisions

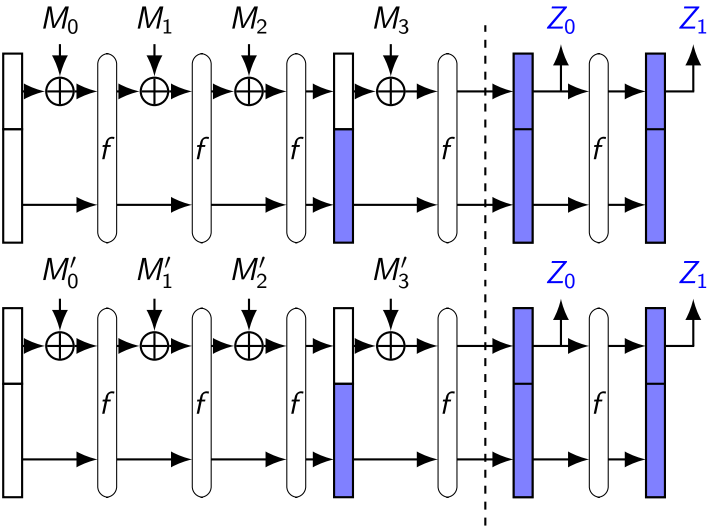
Generic collision attack on the sponge mode

Output collisions



- Standardised instances :
  - $d < r$
  - $c = 2d$
- Small instances :
  - $d > r$: output collision requires several outer state collisions
  - Instead : inner (state) collisions
  - Since $d = c$, same generic security as output collisions

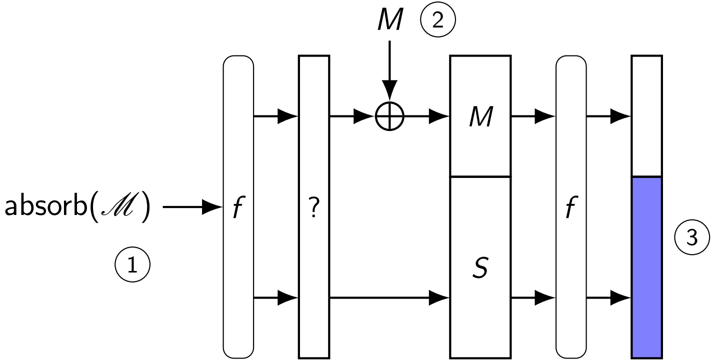
Inner collision attack on small sponges



- Despite $r < d$, the collision propagates to every output

General description of the attack

1. Generate and absorb a random long message to obtain a random inner state S
2. Given S, exploit the properties of f to find a message block M such that the inner state of $f(M||S)$ belongs to a proper subset of $\mathbb{F}_2^c$
3. Find collisions using the birthday paradox



Plan

The KECCAK-f permutation

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

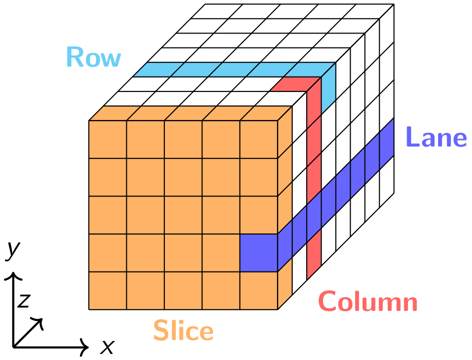
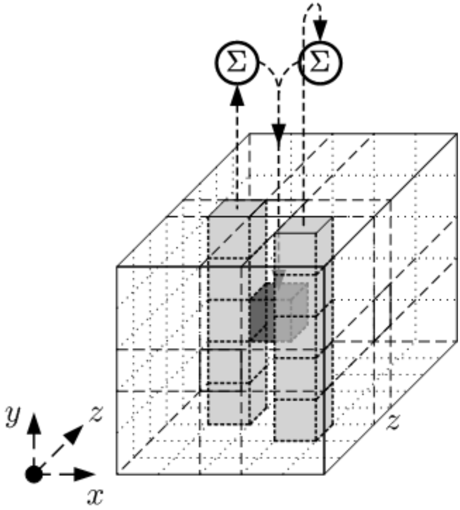


Figure: KECCAK state for $\omega = 8$

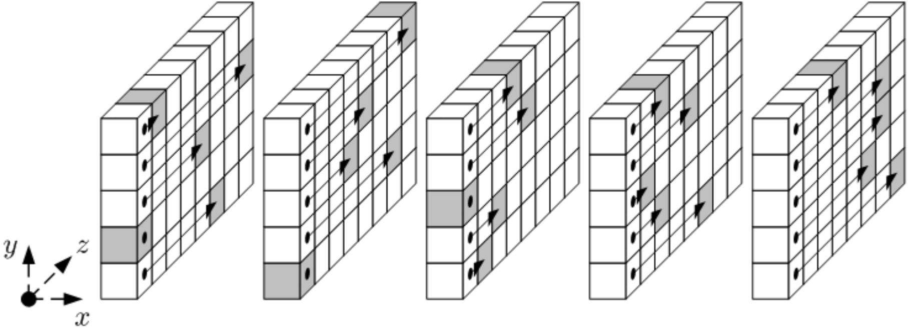
- A round of KECCAK-f[b]: $R = \iota \circ \chi \circ \pi \circ \rho \circ \theta$
- We study a round-reduced version with $f = R^2$

Permutation θ



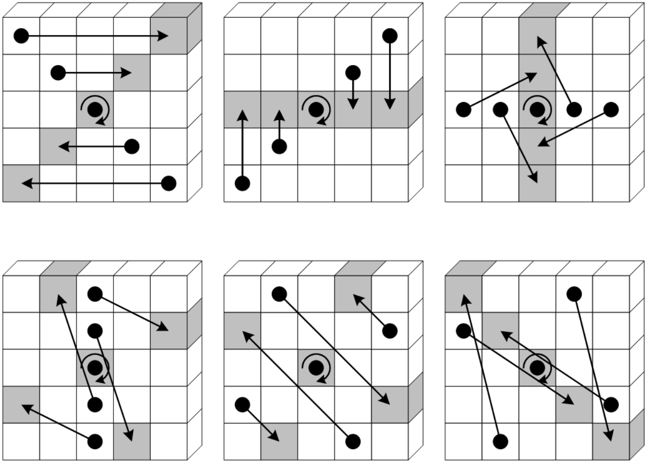
Source : https://keccak.team/figures.html

Permutation ρ



Source : https://keccak.team/figures.html

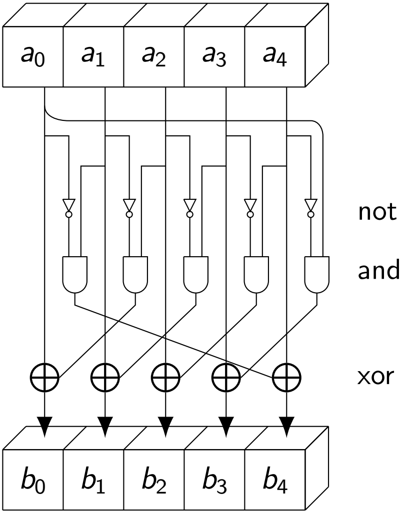
Permutation π



Source : https://keccak.team/figures.html

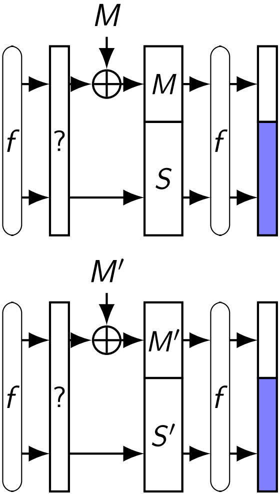
Permutation χ

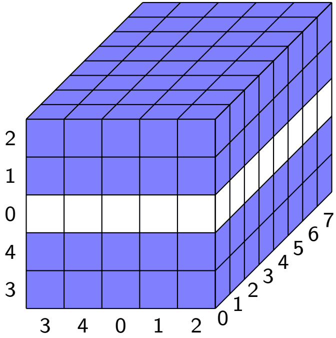
$$b_0 = a_0 + (a_1 + 1) \times a_2$$
$$b_1 = a_1 + (a_2 + 1) \times a_3$$
$$b_2 = a_2 + (a_3 + 1) \times a_4$$
$$b_3 = a_3 + (a_4 + 1) \times a_0$$
$$b_4 = a_4 + (a_0 + 1) \times a_1$$



Plan

Back to inner collisions





KECCAK state with $r = 40$, $c = 160$. In blue, the inner state

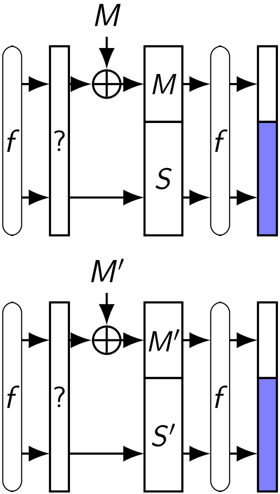
Back to inner collisions

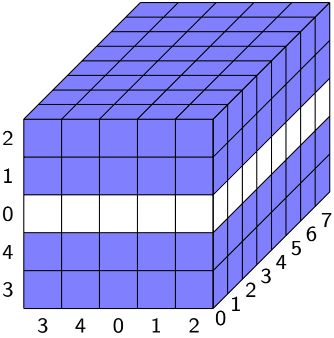
We wish to find a solution to a system of c equations of the form:

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

For KECCAK-f reduced to two rounds, the equations have degree 4.

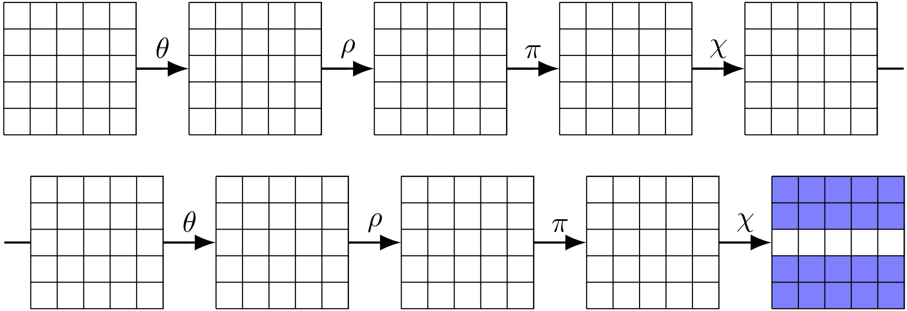
Back to inner collisions





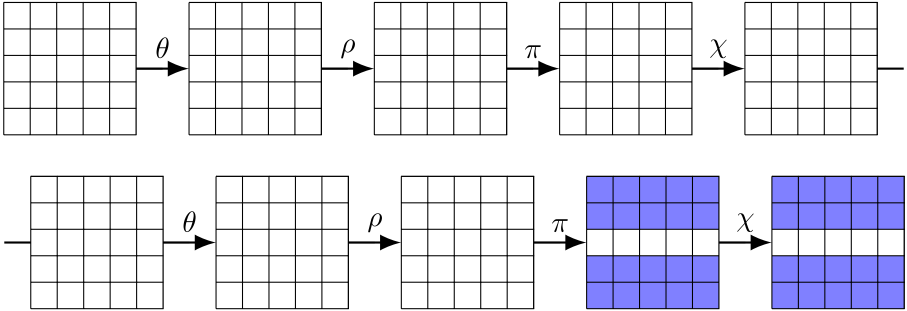
Keccak state with $r = 40$, $c = 160$. In blue, the inner state

Two rounds of KECCAK-f



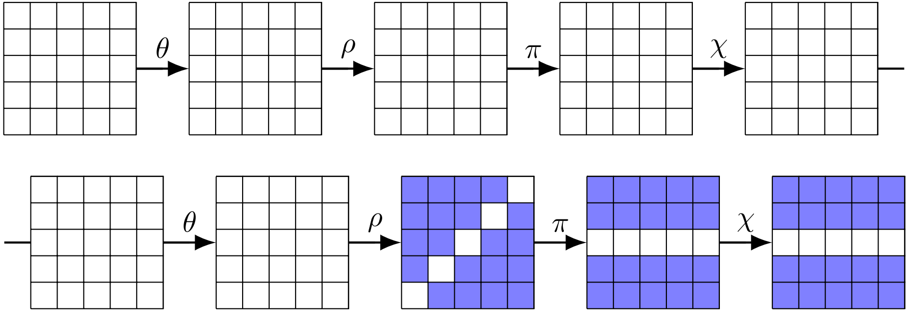
2 rounds of KECCAK-f

Two rounds of KECCAK-f



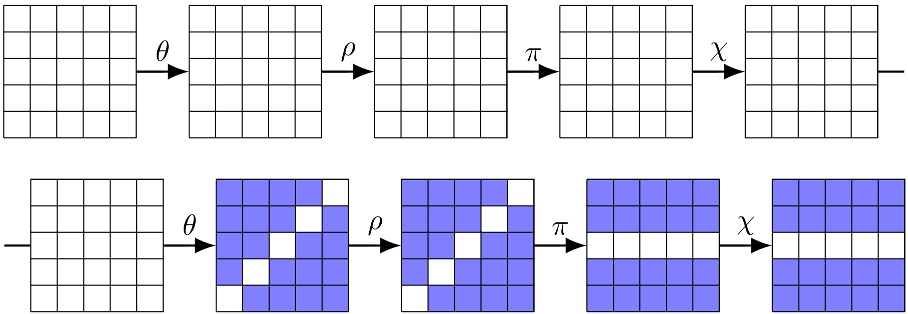
2 rounds of KECCAK-f

Two rounds of KECCAK-f



2 rounds of KECCAK-f

Two rounds of KECCAK-*f*



2 rounds of KECCAK-*f*

θ property

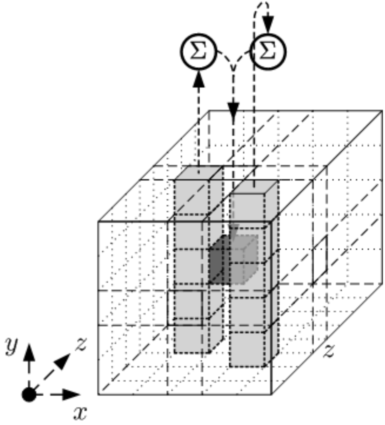
Let $a = (a_0, ..., a_4)$ be a column at the input of θ, let $b = (b_0, ..., b_4)$ be the same column at the output of θ.

For any $0 \leq i, j < 5$,

$$b_i = a_i + c$$
$$b_j = a_j + c$$

and thus

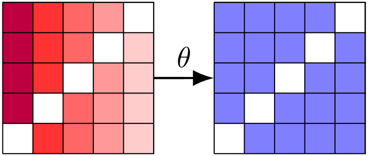
$$b_i + b_j = a_i + a_j$$



Source : https://keccak.team/figures.html

Exploiting θ's property

$$\begin{cases} b_1 = b_1' \\ b_2 = b_2' \\ b_3 = b_3' \\ b_4 = b_4' \end{cases} \Leftrightarrow \begin{cases} b_1 = b_1' \\ b_1 + b_2 = b_1' + b_2' \\ b_2 + b_3 = b_2' + b_3' \\ b_3 + b_4 = b_3' + b_4' \end{cases}$$



From θ's property, we deduce

$$\begin{cases} b_1 + b_2 = b_1' + b_2' \\ b_2 + b_3 = b_2' + b_3' \\ b_3 + b_4 = b_3' + b_4' \end{cases} \Leftrightarrow \begin{cases} a_1 + a_2 = a_1' + a_2' \\ a_2 + a_3 = a_2' + a_3' \\ a_3 + a_4 = a_3' + a_4' \, . \end{cases}$$

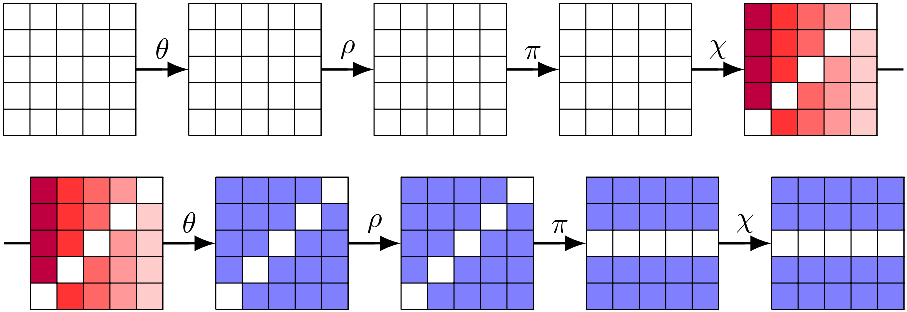
which is equivalent to

$$a_1 + a_1' = a_2 + a_2' = a_3 + a_3' = a_4 + a_4'$$

Property

Having a constant difference on k bits of a column is equivalent to satisfying $k-1$ equations of $(\mathscr{S})$.

Two rounds of KECCAK-f



2 rounds of KECCAK-f

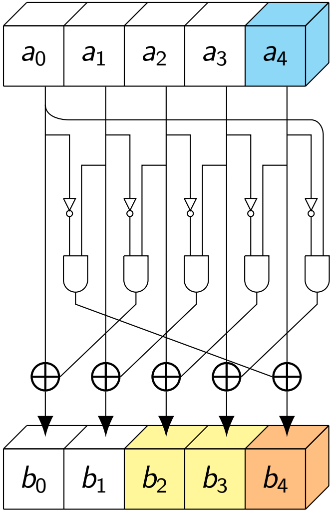
If one generates a set of states that are all constant on columns, then the difference between any two of these states is also constant on columns

Properties

If one sets $a_4 = 0$

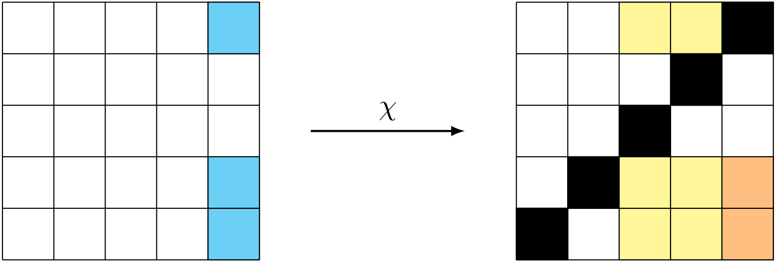
1. b_2 and b_3 can be expressed linearly
2. $b_4 = 0$ with probability $\frac{3}{4}$

$$b_2 = a_2 + (a_3 + 1) \times a_4$$
$$b_3 = a_3 + (a_4 + 1) \times a_0$$
$$b_4 = a_4 + (a_0 + 1) \times a_1$$



Plan

Example of allocation strategy on a slice



Setting the blue bits to 0 $\rightarrow$ 3 linear equations

Ensuring constancy on the yellow bits $\rightarrow$ 4 linear equations

} We satisty 4 equations of $(\mathscr{S})$

With probability $\frac{5}{8}$, we satisfy an extra equation of $(\mathscr{S})$

Example of state allocation strategy

On 5 slices, we set 3 bits to 0
On 1 slice, we set 2 bits to 0

We add 39 equations to a linear system

For any pair of state in the output set :

- 21 equations of $(\mathscr{S})$ are satisfied automatically
- 6 equations of $(\mathscr{S})$ are satisfied with probability $\left(\frac{17}{32}\right)^6$

The probability of inner collision is : $p = 2^{21+6-c}\left(\frac{17}{32}\right)^6$

Conclusion

The time complexity of our attack is $2g\sqrt{p^{-1}} \approx 2^{70}g$

where g will be specified (roughly the "cost of finding a solution to the linear system ")

Computing g

(1) The value of g **does not depend** on the rank of the linear system.

Let e be the size of $\mathscr{L}$.

- Probability of finding a solution : $2^{rank(\mathscr{L})-e}$
- Number of free variables : $r - rank(\mathscr{L})$
- Number of solutions obtained : $2^{r-rank(\mathscr{L})}$

$\rightarrow$ On average, each Gaussian elimination provides 2^{r-e} solutions. Thus,

$$\boxed{g = \frac{e^3}{n_o} 2^{e-r}}$$

where n_o is the number of logical operations in KECCAK-f

Computing g

(1) The value of g **does not depend** on the rank of the linear system.
$\rightarrow$ On average, each Gaussian elimination provides 2^{r-e} solutions. Thus,

$$\boxed{g = \frac{e^3}{n_o} 2^{e-r}}$$

(2) We can **precompute** the Gaussian elimination
$\rightarrow$ Cost of computing solutions: multiplication matrix-vector in $e \times c$ operations.

Computing g

(1) The value of g **does not depend** on the rank of the linear system.
$\rightarrow$ On average, each Gaussian elimination provides 2^{r-e} solutions.

(2) We can **precompute** the Gaussian elimination
$\rightarrow$ Cost of computing solutions: multiplication matrix-vector in $e \times c$ operations.

$$\boxed{g = \frac{ec}{n_o} 2^{e-r}}$$

Computing g

(1) The value of g **does not depend** on the rank of the linear system.
$\rightarrow$ On average, each Gaussian elimination provides 2^{r-e} solutions.

(2) We can **precompute** the Gaussian elimination
$\rightarrow$ Cost of computing solutions: multiplication matrix-vector in $e \times c$ operations.

$$\boxed{g = \frac{ec}{n_o} 2^{e-r}}$$

Application to our attack example:

$$g = \frac{39 \times 161}{410} 2^{-1} \approx 2^3$$

The time complexity is thus of $2^{70+3} = 2^{73}$

Conclusion

- Indeed, the smaller versions are hard to break
- Need for a dedicated analysis of small KECCAK instances

Thanks to Léo Perrin and Jérémy Jean

Thank you for your attention, questions?

Conclusion

- Indeed, the smaller versions are hard to break
- Need for a dedicated analysis of small KECCAK instances

Thanks to Léo Perrin and Jérémy Jean

Thank you for your attention, questions?