Security of SHA-3

Jian Guo

CRYPTANALYSIS TASKFORCE

NANYANG TECHNOLOGICAL UNIVERSITY

NTU/SPMS, Singapore

13th June 2019

Acknowledgements

Many thanks go to my collaborators on this topic:

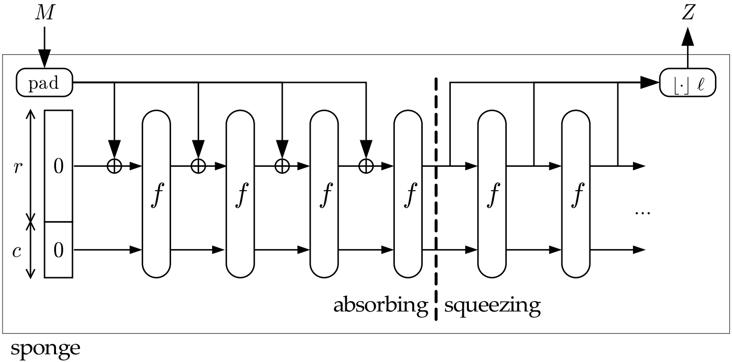
Colin Chaigneau	Alexandre Duc	Thomas Fuhr
Henri Gilbert	Jérémy Jean	Guohong Liao
San Ling	Guozhen Liu	Meicheng Liu
Thomas Peyrin	Kexin Qiao	Jean-René Reinhard
Danping Shi	Ling Song	Lei Wei

Outlines

Outline

SHA-3 (Keccak) Hash Function

The sponge construction [BDPV11]



- b-bit permutation f
- Two parameters: bitrate r, capacity c, and $b = r + c$.
- The message is padded and then split into r-bit blocks.

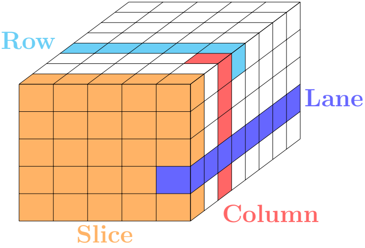
SHA-3 Hash Function

Keccak-f permutation

- 1600 bits: seen as a 5×5 array of 64-bit lanes, $A[x, y], 0 \leq x, y < 5$
- 24 rounds
- each round R consists of five steps:

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

- χ : the only nonlinear operation



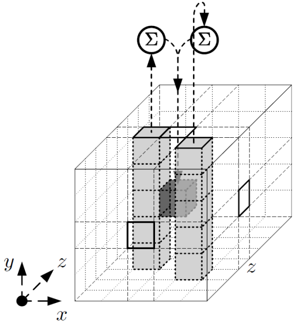
http://www.iacr.org/authors/tikz/

SHA-3 Hash Function

KECCAK permutation: $\iota \circ \chi \circ \pi \circ \rho \circ \theta$

θ step: adding two columns to the current bit

$$
\begin{aligned}
C[x] =& A[x,0] \oplus A[x,1] \oplus A[x,2] \oplus \\
& A[x,3] \oplus A[x,4] \\
D[x] =& C[x-1] \oplus (C[x+1] \lll 1) \\
A[x,y] =& A[x,y] \oplus D[x]
\end{aligned}
$$



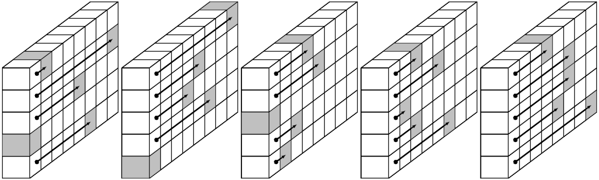
http://keccak.noekeon.org/

- The Column Parity kernel
  - If $C[x] = 0, 0 \leq x < 5$, then the state A is in the CP kernel.

SHA-3 Hash Function

KECCAK permutation: $\iota \circ \chi \circ \pi \circ \rho \circ \theta$

ρ step: lane level rotations, $A[x, y] = A[x, y] \lll r[x, y]$



http://keccak.noekeon.org/

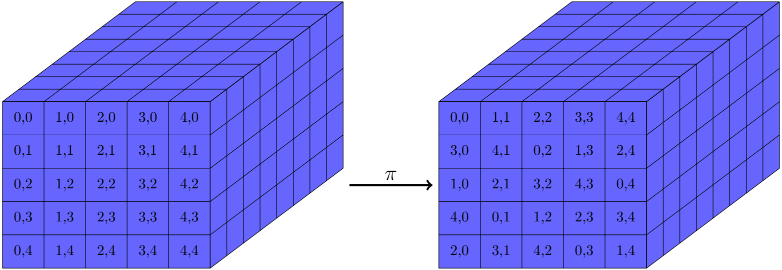
Rotation offsets $r[x, y]$

	$x = 0$	$x = 1$	$x = 2$	$x = 3$	$x = 4$
$y = 0$	0	1	62	28	27
$y = 1$	36	44	6	55	20
$y = 2$	3	10	43	25	39
$y = 3$	41	45	15	21	8
$y = 4$	18	2	61	56	14

SHA-3 Hash Function

KECCAK permutation: $\iota \circ \chi \circ \pi \circ \rho \circ \theta$

π step: permutation on lanes



$$A[y, 2*x+3*y] = A[x, y]$$

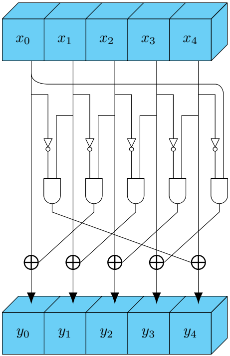
SHA-3 Hash Function

KECCAK permutation: $\iota \circ \chi \circ \pi \circ \rho \circ \theta$

χ step: 5-bit S-boxes, nonlinear operation on rows

$$y_0 = x_0 \oplus (x_1 \oplus 1) \cdot x_2$$
$$y_1 = x_1 \oplus (x_2 \oplus 1) \cdot x_3$$
$$y_2 = x_2 \oplus (x_3 \oplus 1) \cdot x_4$$
$$y_3 = x_3 \oplus (x_4 \oplus 1) \cdot x_0$$
$$y_4 = x_4 \oplus (x_0 \oplus 1) \cdot x_1$$

The algebraic degrees of χ and χ^{-1} are 2 and 3.



SHA-3 Hash Function

KECCAK permutation: $\iota \circ \chi \circ \pi \circ \rho \circ \theta$

ι step: adding a round constant to the state

Adding one round-dependent constant to the first "lane", to destroy the symmetry.

Without ι

- The round function would be symmetric.
- All rounds would be the same.
- Fixed points exist.
- Vulnerable to rotational attacks, slide attacks, ...

SHA-3 Hash Function

Round function of KECCAK-f

Internal state A: a 5×5 array of 64-bit lanes

θ step $C[x] = A[x,0] \oplus A[x,1] \oplus A[x,2] \oplus A[x,3] \oplus A[x,4]$
$D[x] = C[x-1] \oplus (C[x+1] \lll 1)$
$A[x,y] = A[x,y] \oplus D[x]$

ρ step $A[x,y] = A[x,y] \lll r[x,y]$
- The constants $r[x,y]$ are the rotation offsets.

π step $A[y, 2*x+3*y] = A[x,y]$

χ step $A[x,y] = A[x,y] \oplus ((\ A[x+1,y])\& A[x+2,y])$

ι step $A[0,0] = A[0,0] \oplus RC$
- $RC[i]$ are the round constants.

$L \triangleq \pi \circ \rho \circ \theta$

The only non-linear operation is χ step.

Outline

Preimage Attacks — Linear Structures

Core ideas: treat the bits of message block as variables, and convert the preimage finding problem into a system of linear equation; the algebraic degree of the variables is kept to be at most 1 for as many rounds as possible.

- limit the algebraic degrees increased by χ.
- limit the diffusion effect of θ by forcing the variables in CP kernel.

How to keep χ linear

The expression of $b = \chi(a)$ is of algebraic degree 2:
$b_i = a_i + \overline{a_{i+1}} \cdot a_{i+2}$, for $i = 0, 1, \ldots, 4$.

How to keep χ linear

The expression of $b = \chi(a)$ is of algebraic degree 2:
$b_i = a_i + \overline{a_{i+1}} \cdot a_{i+2}$, for $i = 0, 1, \ldots, 4$.

Observation

When there is no neighbouring variables in the input of an Sbox, the application of χ does NOT increase algebraic degrees.

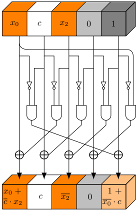
How to keep χ linear

The expression of $b = \chi(a)$ is of algebraic degree 2:

$b_i = a_i + \overline{a_{i+1}} \cdot a_{i+2}$, for $i = 0, 1, \ldots, 4$.

Observation

When there is no neighbouring variables in the input of an Sbox, the application of χ does NOT increase algebraic degrees.



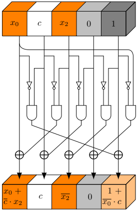
$\checkmark$

How to keep χ linear

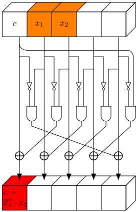
The expression of $b = \chi(a)$ is of algebraic degree 2:
$b_i = a_i + \overline{a_{i+1}} \cdot a_{i+2}$, for $i = 0, 1, \ldots, 4$.

Observation

When there is no neighbouring variables in the input of an Sbox, the application of χ does NOT increase algebraic degrees.



√



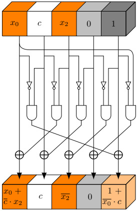
×

How to keep χ linear

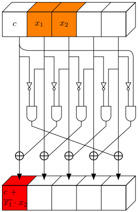
The expression of $b = \chi(a)$ is of algebraic degree 2:
$b_i = a_i + \overline{a_{i+1}} \cdot a_{i+2}$, for $i = 0, 1, \ldots, 4$.

Observation

When there is no neighbouring variables in the input of an Sbox, the application of χ does NOT increase algebraic degrees.



$\surd$



$\times$

Allows at most 2 independent variables, i.e., at least 3 out of 5 bits need to be fixed in each Sbox.

Linear Structure — A Simple Example

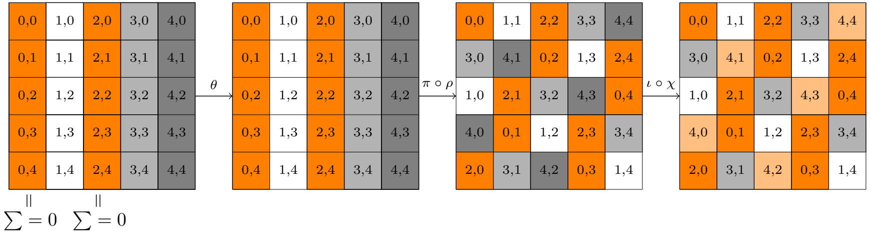


Figure: 1-round linear structure of KECCAK-p*[w] ith the degrees of freedom up to 512, where : variables; : algebraic degree at most 1; : 1; : 0.

Result: one-round linear structure with dimension up to 512.

- All variables do not multiply with each other in the first round.
- The θ effect is limited by forcing $\sum = 0$ (or 1) in two columns.

Preimage Attacks

An Example: 2-Round KECCAK-512

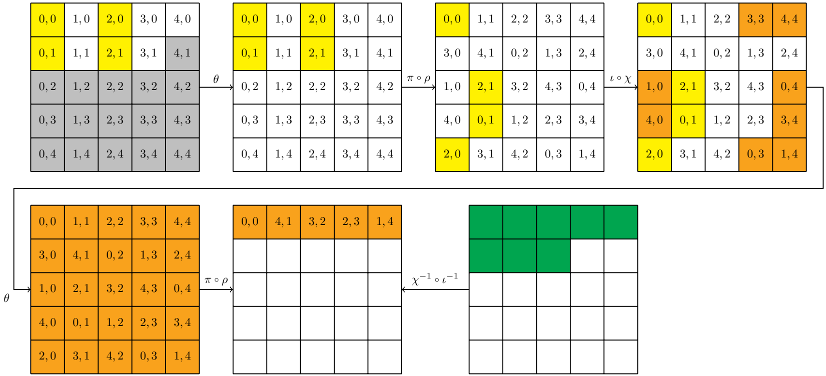


Figure: 2-round KECCAK-512 preimage attack

1-round linear structure of $2 \times 64 = 128$ bits variable.

Preimage Attacks — Inverting One Round

Inverting $\chi : b_i = a_i + \overline{a_{i+1}} \cdot a_{i+2}$

- Linearization: force either a_{i+1} or a_{i+2}, or $a_{i+1} + a_{i+2}$ to be constant, *e.g.*, try both $a_{i+1} = 0$ and $a_{i+1} = 1$.
  (dimension reduces by 1; time complexity reduces when dimension is big enough, otherwise increases by 2^1; space preserves)

- Approximation: $b_i \simeq a_i$, by assuming $\overline{a_{i+1}} \cdot a_{i+2} = 0$, with probability $3/4$.
  (time complexity increases by $4/3$; space reduces to $3/4$)

- Bilinear structure: $b_i = a_i + \overline{b_{i+1}} \cdot a_{i+2}$, when both b_i and b_{i+1} are known.
  (time and space preserve; knowledge of b_i and b_{i+1} is limited by target size and its shape in the $5 \times 5 \times 64$ cube)

Preimage Attacks

Partial linearization

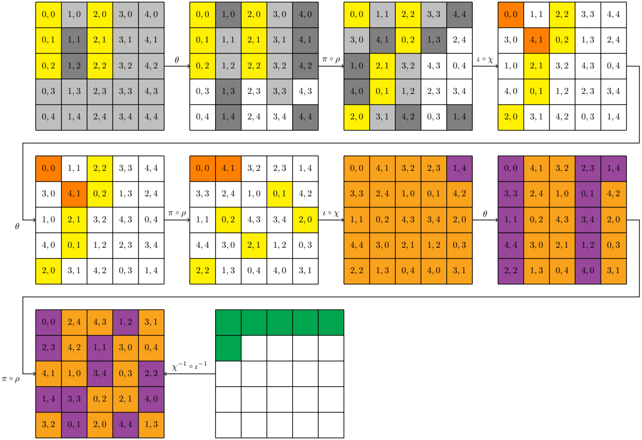


Figure: 3-round KECCAK-384 preimage attack

1 fully linear round + 1 partial linear round + 1 inversion round.

Preimage Attacks — Summary I

Rounds	Target	Complexity	Reference
4	SHA3-384/512	$2^{378}/2^{506}$	[MPS13]
	SHA3-224/256	$2^{213}/2^{251}$	[GLS16]
	SHAKE-128	$2^{106}/2^{106}$	
3	SHA3-384/512	$2^{322}/2^{482}$	
	SHA3-256/SHAKE256	$2^{151}/2^{153}$	[LSLW17]
	SHA3-224	2^{97}	[GLS16]
	SHAKE128	Practical	
2	SHA3-512	2^{384}	
	SHA3-384	2^{89}	[KMS18]
	SHA3-224/256	Practical	[NRM11]
1	SHA3-384/512	Practical	[KRA18]

Preimage Attacks — Summary II

Keccak[$r = 40$, $c = 160$, $n_r = 3$]	?	d8 ed 85 69 2a fb ee 4c 99 ce
Keccak[$r = 240$, $c = 160$, $n_r = 3$]	found by Yao Sun and Ting Li	5c 9d 5e 4b 38 5e 9c 4f 8e 2e
Keccak[$r = 640$, $c = 160$, $n_r = 3$]	found by Jian Guo and Meicheng Liu	00 7b b5 c5 99 80 66 0e 02 93
Keccak[$r = 1440$, $c = 160$, $n_r = 3$]	found by Jian Guo and Meicheng Liu	06 25 a3 46 28 c0 cf e7 6c 75
Keccak[$r = 40$, $c = 160$, $n_r = 4$]	?	74 2c 7e 3c d9 46 1d 0d 03 4e
Keccak[$r = 240$, $c = 160$, $n_r = 4$]	?	0d d2 5e 6d e2 9a 42 ad b3 58
Keccak[$r = 640$, $c = 160$, $n_r = 4$]	?	75 1a 16 e5 e4 95 e1 e2 ff 22
Keccak[$r = 1440$, $c = 160$, $n_r = 4$]	found by Meicheng Liu and Jian Guo	7d aa d8 07 f8 50 6c 9c 02 76

Figure: The status of the Keccak Crunchy Crypto Pre-image Contest, as of 27/03/2019

Ref. https://keccak.team/crunchy_contest.html

Outline

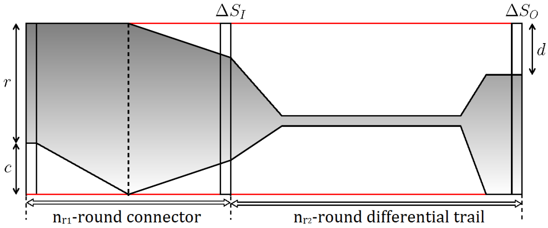
Collision Attack — The State of the Art

Round No.	Target		Complexity	Reference
6	KECCAK	$[r = 1440, c = 160]$	Practical	[SLG17]
5	SHA3-256	$[r = 1088, c = 512]$	Practical	[GLL+19]
5	SHA3-224	$[r = 1152, c = 448]$	Practical	[SLG17]
5	SHAKE128	$[r = 1344, c = 256]$	Practical	[QSLG17]
5	KECCAK	$[r = 640, c = 160]$	Practical	[QSLG17]
4	SHA3-384	$[r = 832, c = 768]$	2^{147}	[DDS13]
4	KECCAK	$[r = 240, c = 160]$	Practical	[KMNS13]
3	SHA3-512	$[r = 576, c = 1024]$	Practical	[DDS13]
3	SHA3-384	$[r = 832, c = 768]$	Practical	[DDS13]
1	KECCAK	$[r = 40, c = 160]$	Practical	[WE17]

Generally, attack becomes more difficult for smaller r and larger c.

Collision Attacks — the Framework

$(n_{r_1} + n_{r_2})$-round collision attacks:



- n_{r_1}-round **connector**: produces message pairs (M_1, M_2) s.t.

$$\mathrm{R}^{n_{r_1}}(\overline{M_1}||0^c) + \mathrm{R}^{n_{r_1}}(\overline{M_2}||0^c) = \Delta S_I, \quad (\mathrm{R}^{n_{r_1}} : n_{r_1} \text{ rounds})$$

$n_{r_1} = 1$ [DDS13] $\longrightarrow$ $n_{r_1} = 2$ [QSLG17] $\longrightarrow$ $n_{r_1} = 3$ [SLG17] .

- n_{r_2}-round **differential**: $\Delta S_I \to \Delta S_O$, with first d bits of ΔS_O being 0, *i.e.*, collision.

Collision Attack — Keccak Sbox Properties

P1: Given compatible I/O differences $(\delta_{in}, \delta_{out})$, the solution set

$$V = \{x \mid S(x) \oplus S(x \oplus \delta_{in}) = \delta_{out}\}$$

forms an affine subspace of size 2, 4, or 8.

Collision Attack — Keccak Sbox Properties

P1: Given compatible I/O differences $(\delta_{in}, \delta_{out})$, the solution set

$$V = \{x \mid S(x) \oplus S(x \oplus \delta_{in}) = \delta_{out}\}$$

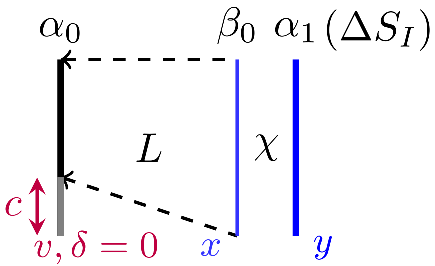
forms an affine subspace of size 2, 4, or 8.

P2: Given the output difference δ_{out}, the compatible input differences

$$\{\delta_{in} \mid \mathsf{DDT}(\delta_{in}, \delta_{out}) > 0\}$$

contains at least 5 2-dimensional affine subspaces.

1-round connector



- **Difference phase**: find a subspace of compatible input difference β_0(*using P2*), under constraint

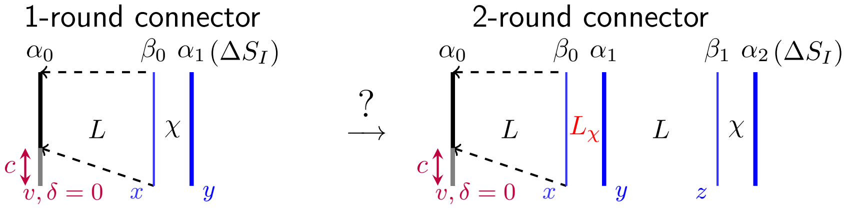
$$\mathsf{last}_c(\alpha_0 = L^{-1}(\beta_0)) = 0$$

- **Value phase**: under fixed β_0 from above, obtain a subspace of input value x that leads to ΔS_I (*using P1*), under constraint

$$\mathsf{last}_c(L^{-1}(x)) = 0$$

2-Round Connectors

Extending the 1-round connector



Idea: Fully linearize the first round,
such that the first 1.5 rounds becomes linear, i.e.,

$$L \circ L_\chi \circ L$$

by linearizing all χ in the first round.

2-Round Connectors

S-box linearization

For an input subspace $V = \{0, 1, 4, 5\}$ which is defined by $\{x_1 = 0, x_3 = 0, x_4 = 0\}$, the S-box is equivalent to the linear transformation

$$y = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot x$$

2-Round Connectors

S-box linearization

For an input subspace $V = \{0, 1, 4, 5\}$ which is defined by $\{x_1 = 0, x_3 = 0, x_4 = 0\}$, the S-box is equivalent to the linear transformation

$$y = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot x$$

Problem: Full linearization allows dimension at most 2 out of 5 affine subspaces. Hence, such linearization can be done at most once.

2-Round Connectors

S-box linearization

For an input subspace $V = \{0, 1, 4, 5\}$ which is defined by $\{x_1 = 0, x_3 = 0, x_4 = 0\}$, the S-box is equivalent to the linear transformation

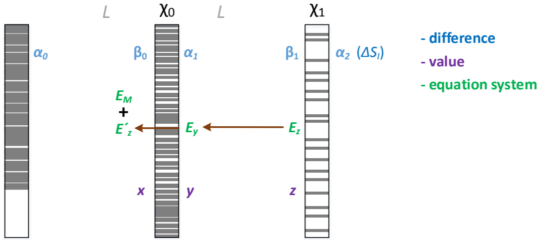
$$y = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot x$$

Problem: Full linearization allows dimension at most 2 out of 5 affine subspaces. Hence, such linearization can be done at most once.

$\longrightarrow$ non-full Sbox linearization

3-Round Connectors

non-full Sbox linearization $\longrightarrow$ partial 3-round connectors



Observation: not all Sboxes are active, and only the input values to the active Sboxes of χ_1 matter, which may come from active/in-active Sboxes of χ_0.

Partial Sbox Linearization I

$$(b_0, b_1, b_2, b_3, b_4) = \mathsf{Sbox}(a_0, a_1, a_2, a_3, a_4)$$

$$\text{fix } a_2 = 0 \text{ !}$$

$$b_0 = a_0 + \overline{a_1} \cdot a_2 = a_0, \text{and}$$

$$b_1 = a_1 + \overline{a_2} \cdot a_3 = a_1 + a_3.$$

Partial Sbox Linearization I

$$(b_0, b_1, b_2, b_3, b_4) = \mathsf{Sbox}(a_0, a_1, a_2, a_3, a_4)$$

fix $a_2 = 0$!

$$b_0 = a_0 + \overline{a_1} \cdot a_2 = a_0, \text{and}$$

$$b_1 = a_1 + \overline{a_2} \cdot a_3 = a_1 + a_3.$$

This costs 1-bit linearization v.s. 3 bits for full linearization.

Partial Sbox Linearization II

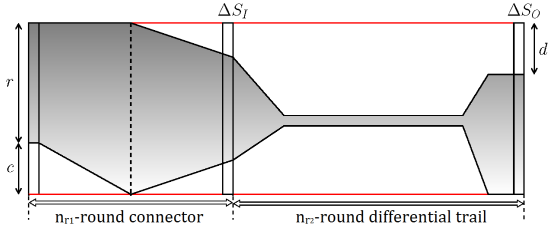
Table: #equations necessary to partially linearize the Sbox

non-active		active	
Mask U	#equations	DDT $\log_2$	#equations
1F(1/32)	3 (3)	1	4
0(1/32)	0 (3)	2	3
T(10/32)	1 (3)	3	2,3
others(20/32)	2 (3)		

Lesser degrees of freedom are consumed for non-full Sbox linearizations, could be used for fulfil Sboxes in the 3rd round.

Collision Attacks — Searching for the Differentials

$(n_{r_1} + n_{r_2})$-round collision attacks:



- high probability, *e.g.*, forcing the differences in 2nd and 3rd rounds of the trail in CP kernel
- $\text{first}_d(\Delta S_O) = 0$
- Consumes as less as possible degrees of freedom, provided by the connectors

GPU Implementation for the bruteforce

$\sim 2^{28}$ (2^{29}) $\text{Keccak-}f$ evaluations per second
on GPU GTX 970 (GTX 1070) v.s.
$\sim 2^{21}$ on CPUs.

Enables computation power up to 2^{50}.
Source code available: `http://catf.crypto.sg`

Collision Attack — Summary I

Round No.	Target		Complexity	Reference
6	Keccak	$[r = 1440, c = 160]$	Practical	[SLG17]
5	SHA3-256	$[r = 1088, c = 512]$	Practical	[GLL+19]
5	SHA3-224	$[r = 1152, c = 448]$	Practical	[SLG17]
5	SHAKE128	$[r = 1344, c = 256]$	Practical	[QSLG17]
5	Keccak	$[r = 640, c = 160]$	Practical	[QSLG17]
4	SHA3-384	$[r = 832, c = 768]$	2^{147}	[DDS13]
4	Keccak	$[r = 240, c = 160]$	Practical	[KMNS13]
3	SHA3-512	$[r = 576, c = 1024]$	Practical	[DDS13]
3	SHA3-384	$[r = 832, c = 768]$	Practical	[DDS13]
1	Keccak	$[r = 40, c = 160]$	Practical	[WE17]

Practical: time complexity $< 2^{54}$.

Collision Attack — Summary II

Keccak[$r = 40$, $c = 160$, $n_r = 5$]	?
Keccak[$r = 240$, $c = 160$, $n_r = 5$]	?
Keccak[$r = 640$, $c = 160$, $n_r = 5$]	found by Kexin Qiao, Ling Song, Meicheng Liu, and Jian Guo
Keccak[$r = 1440$, $c = 160$, $n_r = 5$]	found by Kexin Qiao, Ling Song, Meicheng Liu, and Jian Guo
Keccak[$r = 40$, $c = 160$, $n_r = 6$]	?
Keccak[$r = 240$, $c = 160$, $n_r = 6$]	?
Keccak[$r = 640$, $c = 160$, $n_r = 6$]	?
Keccak[$r = 1440$, $c = 160$, $n_r = 6$]	found by Ling Song, Guohong Liao and Jian Guo

Figure: The status of the Keccak Crunchy Crypto Collision Contest, as of 27/03/2019

Ref. https://keccak.team/crunchy_contest.html

Outline

Distinguishers — Zero-Sum

Zero-Sum Distinguisher

Given function/permutation f, find an input set X, s.t. $\sum_{x \in X} x = 0$ and $\sum_{x \in X} f(x) = 0$, i.e., the sums of input and output set are 0 simultaneously.

Distinguishers — Zero-Sum

Zero-Sum Distinguisher

Given function/permutation f, find an input set X, s.t. $\sum_{x \in X} x = 0$ and $\sum_{x \in X} f(x) = 0$, i.e., the sums of input and output set are 0 simultaneously.

A linear space of dimension $\deg(f) + 1$ fulfils above.

Distinguishers — Zero-Sum

Zero-Sum Distinguisher

Given function/permutation f, find an input set X, s.t. $\sum_{x \in X} x = 0$ and $\sum_{x \in X} f(x) = 0$, i.e., the sums of input and output set are 0 simultaneously.
A linear space of dimension $\deg(f) + 1$ fulfils above.

Application to KECCAK-f

$$\overset{m \text{ rounds}}{\underset{\text{backward}}{\longleftarrow}} \quad \overset{t \text{ rounds}}{\underset{\text{linear structure}}{\longleftrightarrow}} \quad \overset{n \text{ rounds}}{\underset{\text{forward}}{\longrightarrow}}$$

Distinguishers — Zero-Sum

Zero-Sum Distinguisher

Given function/permutation f, find an input set X, s.t. $\sum_{x \in X} x = 0$ and $\sum_{x \in X} f(x) = 0$, i.e., the sums of input and output set are 0 simultaneously.
A linear space of dimension $\deg(f) + 1$ fulfils above.

Application to Keccak-f

m rounds	t rounds	n rounds
$\longleftarrow$	$\longleftrightarrow$	$\longrightarrow$
backward	linear structure	forward

degree of χ: 2; degree of χ^{-1}: 3
degree of n forward rounds: 2^n; degree of m backward rounds: 3^m
Required size of linear structure: $2 \cdot \max(2^n, 3^m)$

2-round Linear Structure of Dimension up to 512

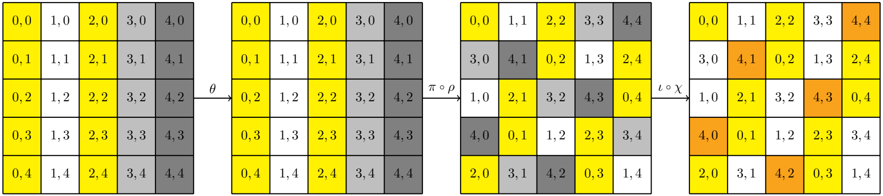


Figure: With one backward round, 2-round Linear Structure of Dimension up to 512

3-round Linear Structure of Dimension up to 194

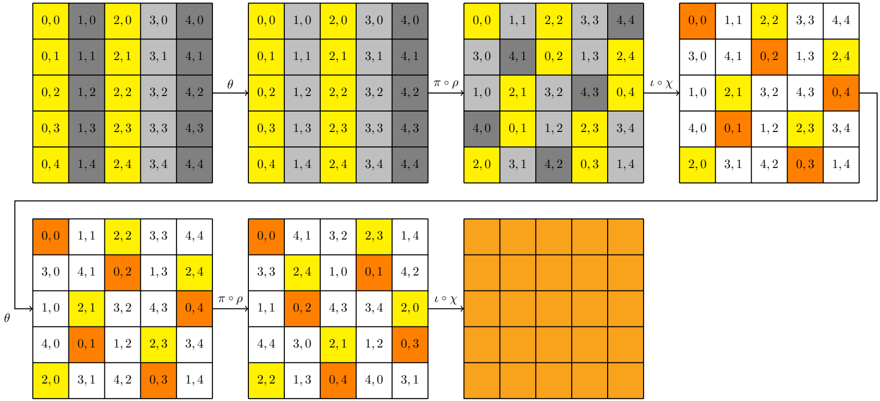


Figure: With one backward round, 3-round Linear Structure of Dimension up to 194

Zero-Sum Distinguisher — Result Summary

#Rounds	back. + l.s. +for.	$3^m, 2^n$	Complexity
7	1+3+3	$3, 8$	2^9
8	2+3+3	$9, 8$	2^{10}
9	2+3+4	$9, 16$	2^{17}
10	3+3+4	$27, 16$	2^{28}
11	3+3+5	$27, 32$	2^{33}
12	3+3+6	$27, 64$	2^{65}
13	4+3+6	$81, 64$	2^{82}
14	4+3+7	$81, 128$	2^{129}
15	5+2+8	$243, 256$	2^{257}

Table: Summary of distinguishers on KECCAK-f permutation

Outline

Key Recovery — The targets I

KMAC, Keyak, Ketje

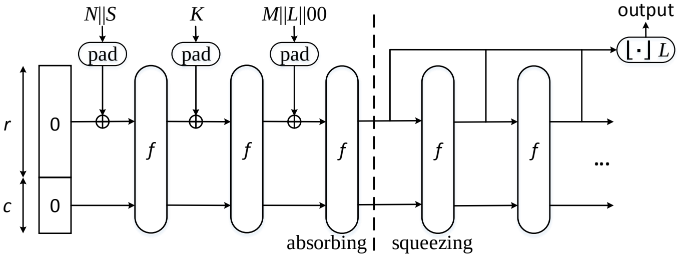


Figure: KMAC processing one message block, K is processed as an independent block before message, with $f = \text{Keccak} - p^*[b = 1600, n_r = 24]$.

Keccak-MAC: $K||M$ is as the message input of Keccak.

Key Recovery — The targets II

KMAC, KEYAK, KETJE

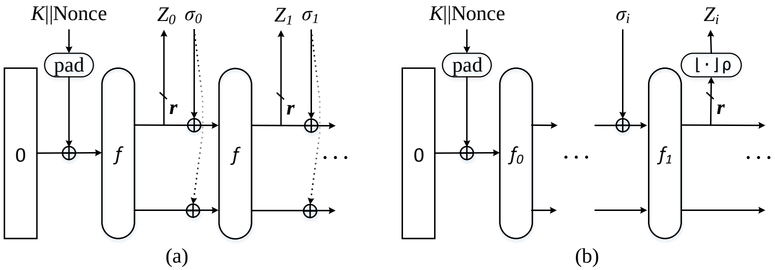


Figure: (a) KEYAK; (b) KETJE.

KEYAK takes KECCAK-p*$[b = 800, 1600]$; KETJE takes KECCAK-p*$[b = 200, 400, 800, 1600]$

Key Recovery — Cube Attacks and Cube-Attack-Like Cryptanalysis I

Given the Boolean polynomial $f(k_0, \ldots, k_{n-1}, v_0, \ldots, v_{m-1})$ and a monomial $t_I = v_{i_1} v_{i_2} \cdots v_{i_r}$, $I = (i_1, \ldots, i_d)$, f can be written as

$$f(k_0, \ldots, k_{n-1}, v_0, \ldots, v_{m-1}) = t_I p_{S_I} + q(k_0, \ldots, k_{n-1}, v_0, \ldots, v_{m-1})$$

where

- q does not contain t_I
- p_{S_I} is the superpoly of I in f
- v's are cube variables, d is the dimension.

The cube sum is

$$\sum_{(v_{i_1}, \ldots, v_{i_r}) \in C_I} f(k_0, \ldots, k_{n-1}, v_0, \ldots, v_{m-1}) = P_{S_I}$$

Key Recovery — Cube Attacks and Cube-Attack-Like Cryptanalysis II

Cube Attack: $P_{S_I} = L(k_0, \ldots, k_{n-1})$ is a linear polynomial.
Conditional Cube Attack: Depending on some (key-dependent) cube variables, P_{S_I} is a linear polynomial.
Cube-Attack-Like: using n_a aux. variables, $P' = L'(k_{i_1}, \ldots, k_{i_{n'}})$, with $n' < n$.

Find cube of size as large as possible, as many round as possible:

CON algebraic degree of m-round $\textsc{Keccak}$-p is 2^m, prepend $1 \sim 3$ rounds and generate a linear space of dimension at least m.

- usually the first round of $\textsc{Keccak}$-p is chosen to be linear
- ultize tools like MILP to find (sub-) optimal choices of conditions, and key variables s.t. [CON] fulfils.

Key Recovery — Summary: MACs

Table: Summary of attacks on KMAC, and KECCAK-MAC

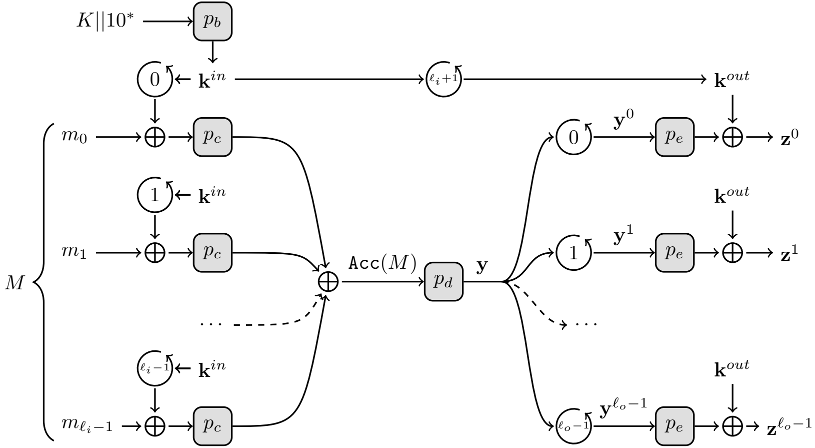
Target	Key Size	Capacity	Rounds	Time (Data)	Reference
KMAC128	128	256	7/24	2^{76}	[SGSL18]
KMAC256	256	512	9/24	2^{147}	
KECCAK-MAC	128	256/512	7/24	2^{72}	[HWX+17]
		768	7/24	2^{75}	[LBDW17]
		1024	6/24	$2^{58.3}$	
		1024	6/24	2^{40}	[SGSL18]
		1024	7/24	2^{111}	[SG18]

Key Recovery — Summary: AEs

Table: Summary of Attacks on KEYAK and KETJE

Target	Key Size	Rounds	Time (Data)	Memory	nonce-respected	Reference
Lake KEYAK	128	6/12	2^{37}	-	Yes	[DMP+15]
	128	8/12	2^{74}	-	No	[HWX+17]
	128	8/12	$2^{71.01}$	-	Yes	[SGSL18]
	256	9/14	$2^{137.05}$	-	Yes	
River KEYAK	128	8/12	2^{77}	-	Yes	[SGSL18]
KETJE Major	128	7/13	2^{83}	-	Yes	[LBDW17]
	128	7/13	$2^{71.24}$	-	Yes	[SGSL18]
KETJE Minor	128	7/13	2^{81}	-	Yes	[LBDW17]
	128	7/13	$2^{73.03}$	-	Yes	[SGSL18]
KETJE SR v1	128	7/13	2^{115}	2^{50}	Yes	[DLWQ17]
	128	7/13	2^{91}	-	Yes	[SGSL18]
FKD[1600]	128	9/-	2^{90}	-	No	[SGSL18]
KETJE Jr v1	96	5/13	$2^{36.86}$	2^{18}	Yes	[SG18]
KETJE Jr v2	96	5/13	$2^{34.91}$	2^{15}	Yes	
KETJE Sr v2	128	7/13	2^{99}	2^{33}	Yes	

Kravatte



p_b, p_c, p_d, p_e being 4 or 6 round Keccak-p

MITM and Linear Recurrence Attacks due to low algebraic degree and linear rolling functions.

Outline

Conclusion Remarks

In Summary:

- 5 and 4 rounds of SHA-3 can be attacked, w.r.t. collision and preimage resistance, out of 24 rounds (huge security margin).
- key-recovery attack works up to 9 rounds, intensive cryptanalysis is necessary when weak permutation is used.

More information is available via:

http://catf.crypto.sg/keccak

Thank You !

References I

Itai Dinur, Orr Dunkelman, and Adi Shamir.
Collision Attacks on Up to 5 Rounds of SHA-3 Using Generalized Internal Differentials.
In Shiho Moriai, editor, *FSE 2013*, volume 8424 of *LNCS*, pages 219–240. Springer, 2013.

Xiaoyang Dong, Zheng Li, Xiaoyun Wang, and Ling Qin.
Cube-like Attack on Round-Reduced Initialization of Ketje Sr.
IACR Trans. Symmetric Cryptol., 2017(1):259–280, 2017.

Itai Dinur, Pawel Morawiecki, Josef Pieprzyk, Marian Srebrny, and Michal Straus.
Cube Attacks and Cube-Attack-Like Cryptanalysis on the Round-Reduced Keccak Sponge Function.
In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 733–761. Springer, 2015.

Jian Guo, Guohong Liao, Guozhen Liu, Meicheng Liu, Kexin Qiao, and Ling Song.
Practical Collision Attacks against Round-Reduced SHA-3.
Journal of Cryptology, 2019.

Jian Guo, Meicheng Liu, and Ling Song.
Linear Structures: Applications to Cryptanalysis of Round-Reduced Keccak.
In *ASIACRYPT 2018 (1)*, pages 249–274, 2016.

References II

Senyang Huang, Xiaoyun Wang, Guangwu Xu, Meiqin Wang, and Jingyuan Zhao.
Conditional Cube Attack on Reduced-Round Keccak Sponge Function.
In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 259–288, 2017.

Stefan Kölbl, Florian Mendel, Tomislav Nad, and Martin Schläffer.
Differential Cryptanalysis of Keccak Variants.
In *IMACC 2013*, pages 141–157, 2013.

Rajendra Kumar, Nikhil Mittal, and Shashank Singh.
Cryptanalysis of 2 round Keccak-384.
In Debrup Chakraborty and Tetsu Iwata, editors, *Indocrypt*, volume 11356, pages 120–133. Springer, 2018.

Rajendra Kumar, Mahesh Sreekumar Rajasree, and Hoda AlKhzaimi.
Cryptanalysis of 1-round KECCAK.
In *AFRICACRYPT 2018*, pages 124–137, 2018.

Zheng Li, Wenquan Bi, Xiaoyang Dong, and Xiaoyun Wang.
Improved Conditional Cube Attacks on Keccak Keyed Modes with MILP Method.
In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 99–127. Springer, 2017.

References III

Ting Li, Yao Sun, Maodong Liao, and Dingkang Wang.
Preimage Attacks on the Round-reduced Keccak with Cross-linear Structures.
IACR Transactions on Symmetric Cryptology, 2017(4):39–57, Dec. 2017.

Pawel Morawiecki, Josef Pieprzyk, and Marian Srebrny.
Rotational Cryptanalysis of Round-Reduced Keccak.
In *FSE 2013*, pages 241–262, 2013.

María Naya-Plasencia, Andrea Röck, and Willi Meier.
Practical Analysis of Reduced-Round Keccak.
In Daniel J. Bernstein and Sanjit Chatterjee, editors, *INDOCRYPT 2011*, volume 7107 of *LNCS*, pages 236–254. Springer, 2011.

Kexin Qiao, Ling Song, Meicheng Liu, and Jian Guo.
New Collision Attacks on Round-Reduced Keccak.
In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017 (III)*, volume 10212 of *LNCS*, pages 216–243, 2017.

Ling Song and Jian Guo.
Cube-Attack-Like Cryptanalysis of Round-Reduced Keccak Using MILP.
IACR Trans. Symmetric Cryptol., 2018(3):182–214, 2018.

References IV

Ling Song, Jian Guo, Danping Shi, and San Ling.
New MILP Modeling: Improved Conditional Cube Attacks on Keccak-Based Constructions.
In Thomas Peyrin and Steven D. Galbraith, editors, *ASIACRYPT 2018*, volume 11273 of *Lecture Notes in Computer Science*, pages 65–95. Springer, 2018.

Ling Song, Guohong Liao, and Jian Guo.
Non-full Sbox Linearization: Applications to Collision Attacks on Round-Reduced Keccak.
In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017 (II)*, volume 10402 of *LNCS*, pages 428–451. Springer, 2017.