New MILP Modeling: Improved Conditional Cube Attacks on Keccak-Based Constructions

Ling Song, Jian Guo, Danping Shi, San Ling

NANYANG TECHNOLOGICAL UNIVERSITY

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

4 Dec 2018 @ Brisbane, Australia

Outline

Outline

KECCAK

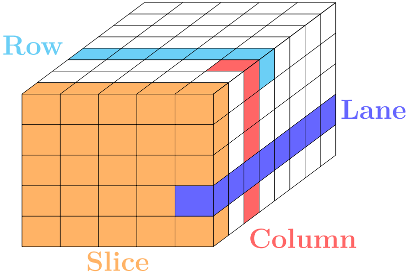
- Permutation-based hash function
  - Designed by Guido Bertoni, Joan Daemen, Michaël Peeters and Gilles Van Assche
  - Selected as `SHA-3` standard
  - Underlying permutation: KECCAK-$p[1600, 24]$
- KECCAK under keyed modes: `KMAC`, KECCAK-MAC
- Its relatives
  - Authenticated encryption: KEYAK, KETJE
  - Pseudorandom function: KRAVATTE
  - Permutation: XOODOO

KECCAK-$p[b, n_r]$ Permutation

- b bits: seen as a 5×5 array of $\frac{b}{25}$-bit lanes, $A[x, y]$
- n_r rounds
- each round R consists of five steps:

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

- χ : S-box on each row
- π, ρ: change the position of state bits

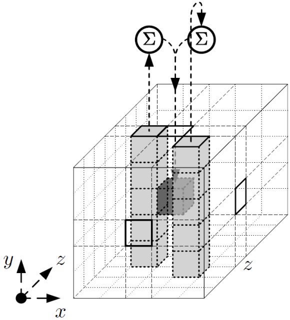


http://www.iacr.org/authors/tikz/

KECCAK-p Round Function: θ

θ step: adding two columns to the current bit

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

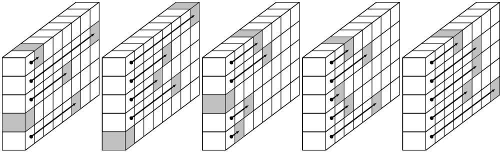


http://keccak.noekeon.org/

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

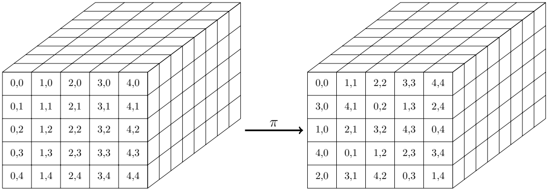
Keccak-p Round Function: ρ, π

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



http://keccak.noekeon.org/

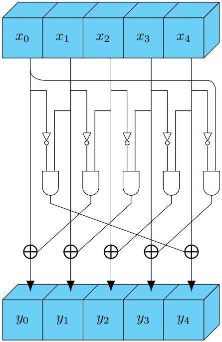
π step: permutation on lanes, $A[y, 2*x+3*y] = A[x, y]$



KECCAK-p Round Function: χ

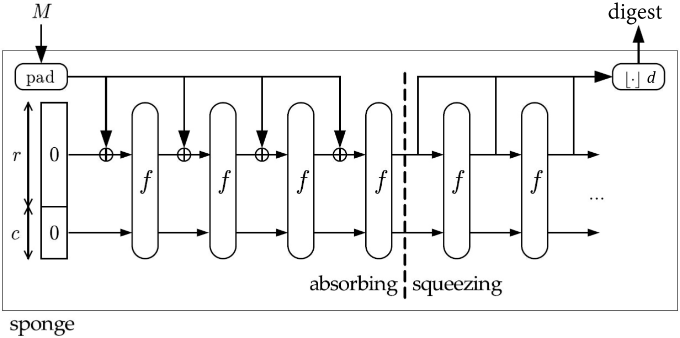
χ step: 5-bit S-boxes, nonlinear operation on rows

$$
\begin{aligned}
y_0 &= x_0 + (x_1 + 1) \cdot x_2, \\
y_1 &= x_1 + (x_2 + 1) \cdot x_3, \\
y_2 &= x_2 + (x_3 + 1) \cdot x_4, \\
y_3 &= x_3 + (x_4 + 1) \cdot x_0, \\
y_4 &= x_4 + (x_0 + 1) \cdot x_1.
\end{aligned}
$$



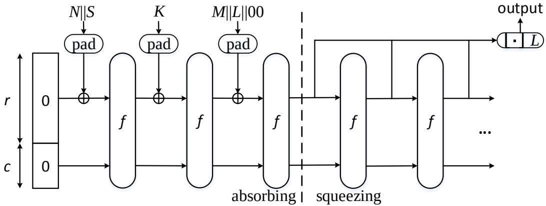
- Nonlinear term: product of two adjacent bits in a row.
- The algebraic degree of n rounds is 2^n.

KECCAK: KECCAK-$p[1600, 24] + $ Sponge

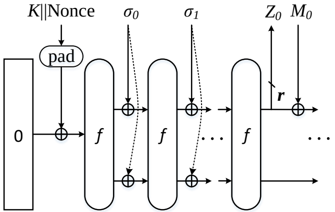


- Sponge construction [BDPV11]
  - b-bit permutation f
  - Two parameters: bitrate r, capacity c, and $b = r + c$.
- KECCAK-MAC
  - Take $K||M$ as input

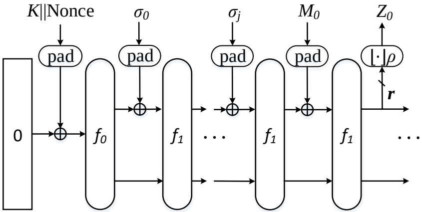
Keyed KECCAK Constructions



KMAC



KEYAK



KETJE

Key Recovery Attacks

Intuition: $deg(\chi) = 2$. Consider algebraic cryptanalsis, in paticular, cube attacks.

Key Recovery Attacks

Intuition: $deg(\chi) = 2$. Consider algebraic cryptanalsis, in paticular, cube attacks.

Contributions

- Mixed Integer Linear Programming models for searching two types of cube attacks
- Best key recovery attacks on round-reduced KMAC, KEYAK and larger versions of KETJE so far
- Solve the open problem of "Full State Keyed Duplex (Sponge)"

Key Recovery Attacks

Intuition: $deg(\chi) = 2$. Consider algebraic cryptanalsis, in paticular, cube attacks.

Contributions

- Mixed Integer Linear Programming models for searching two types of cube attacks
- Best key recovery attacks on round-reduced KMAC, KEYAK and larger versions of KETJE so far
- Solve the open problem of "Full State Keyed Duplex (Sponge)"

"Whether these attacks can still be extended to more rounds by exploiting full-state absorbing remains an open question".

— *the* KEYAK *designers*

Outline

Cube Attacks [DS09]

Higher Order Differential Cryptanalysis [Lai94]

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

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

  - q contains terms that are not divisible by t_I
  - p_{S_I} is called the superpoly of I in f
  - $v_{i_1}, \ldots, v_{i_d}$ are called cube variables. d is the dimension.
- The the cube sum is exactly

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

- Cube attacks: p_{S_I} is a linear polynomial in key bits.
- Cube testers: distinguish p_{S_I} from a random function.
- If $deg(f) < d$, $p_{S_I} = 0$

Conditional Cube Testers of Keccak [HWX+17]

Renamed conCube

conCube

- Linearize the first round.
- There exist p cube variables that are not multiplied with any cube variable even in the second round under certain *conditions*.

We classify two types of conditional cubes:

Type I conCube

- $p = 1$.
- Given such a cube with $d = 2^{n-1}$, $p_{S_I} = 0$ for n-round $\textsc{Keccak}$ if the conditions are met.

Type II conCube

- $p = d$.
- Given such a cube with $d = 2^{n-2} + 1$, $p_{S_I} = 0$ for n-round $\textsc{Keccak}$ if the conditions are met.

ConCube on Keccak

If the conditions involve the key, the conditional cube can be used to recover the key.

ConCube on KECCAK

If the conditions involve the key, the conditional cube can be used to recover the key.

How to find good cubes?

ConCube on KECCAK

If the conditions involve the key, the conditional cube can be used to recover the key.

How to find good cubes?

Task of the MILP Model

1. Find Type I (II) cubes with dimension 2^{n-1} $(2^{n-2}+1)$ where n is as large as possible; (attack more rounds).
2. The number of conditions is minimized. (low complexity).

Outline

Mixed Integer Linear Programming

- An MILP problem is of the form

$$
\begin{aligned}
\texttt{min} \quad & c^T x \\
Ax & \geq b \\
x_i & \geq 0 \\
x_i & \in \mathbb{Z}
\end{aligned}
$$

- Solvers
  - Gurobi, CPLEX, SCIP, ...
- Application to cryptanalysis since Mouha et al.'s pioneering work [MWGP11]

MILP Model of Searching Cubes

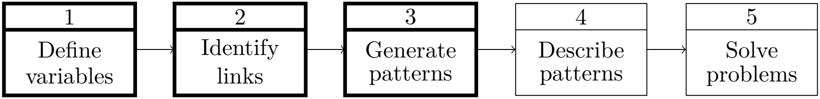
- Similar to modeling differential cryptanalysis
- Model the propagation of activeness through each step

$$\chi \circ \pi \circ \rho \circ \theta \circ \chi \circ \pi \circ \rho \circ \theta$$

- Modeling ρ, π is trivial.

MILP-based Cryptanalysis

1. Define variables which are mostly binary for the crypto problem.
2. Identify links between the variables
3. Generate all valid patterns for the variables
4. Describe valid patterns with inequalities
5. Solve the MILP problem



MILP-based Cryptanalysis

1. Define variables which are mostly binary for the crypto problem.
2. Identify links between the variables
3. Generate all valid patterns for the variables
4. Describe valid patterns with inequalities
5. Solve the MILP problem

1		2		3		4		5
Define variables	→	Identify links	→	Generate patterns	→	Describe patterns	→	Solve problems

Example: Modeling the first χ

1. Define Variables

Let $a[x][y][z]$ be the state:

$$a \xrightarrow{\pi\circ\rho\circ\theta} \mathbf{b} \xrightarrow{\chi} \mathbf{c} \xrightarrow{\pi\circ\rho\circ\theta} \mathbf{d} \xrightarrow{\chi} e$$

$A[x][y][z] = 1$ if $a[x][y][z]$ is active, *i.e.*, containing cube variables:

$$A \xrightarrow{\pi\circ\rho\circ\theta} \mathbf{B} \xrightarrow{\chi} \mathbf{C} \xrightarrow{\pi\circ\rho\circ\theta} \mathbf{D} \xrightarrow{\chi} E$$

$V[x][y][z] = 1$ indicates that bit $b[x][y][z]$ is constrained to the value of $H[x][y][z]$.

2. Identify Links

Propagation of variables through χ

Observation

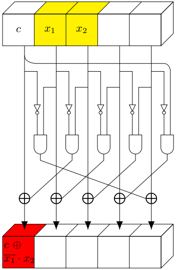
1. Linearize χ by avoiding adjacent variables in the input.
2. Bit 1 (0) on the left (right) of the variable helps to restrict the diffusion of variables through χ, while an unknown constant diffuses the variable in an uncertain way.

2. Identify Links

Propagation of variables through χ

Observation

1. Linearize χ by avoiding adjacent variables in the input.
2. Bit 1 (0) on the left (right) of the variable helps to restrict the diffusion of variables through χ, while an unknown constant diffuses the variable in an uncertain way.

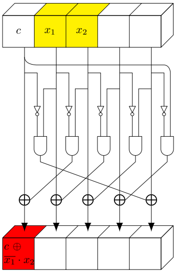


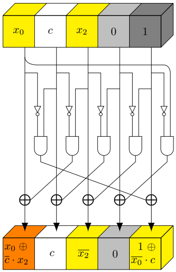
2. Identify Links

Propagation of variables through χ

Observation

1. Linearize χ by avoiding adjacent variables in the input.
2. Bit 1 (0) on the left (right) of the variable helps to restrict the diffusion of variables through χ, while an unknown constant diffuses the variable in an uncertain way.





3. Generate Valid Patterns

$$c[x] = b[x] + (b[x+1] + 1) \cdot b[x+2]^1$$

$b[x]$	$b[x+1]$	$b[x+2]$	$c[x]$

[1] Omit coordinates $[y][z]$.

3. Generate Valid Patterns

$$c[x] = b[x] + (b[x+1] + 1) \cdot b[x+2]^1$$

$b[x]$	$b[x+1]$	$b[x+2]$	$c[x]$
cst	cst	cst	cst

[1] Omit coordinates $[y][z]$.

3. Generate Valid Patterns

$$c[x] = b[x] + (b[x+1] + 1) \cdot b[x+2]^1$$

$b[x]$	$b[x+1]$	$b[x+2]$	$c[x]$
cst	cst	cst	cst
var	cst	*	var

[1] Omit coordinates $[y][z]$.

3. Generate Valid Patterns

$$c[x] = b[x] + (b[x+1] + 1) \cdot b[x+2]^{1}$$

$b[x]$	$b[x+1]$	$b[x+2]$	$c[x]$
cst	cst	cst	cst
var	cst	*	var
cst	cst	var	var (deg ≤ 1)

[1] Omit coordinates $[y][z]$.

3. Generate Valid Patterns

$$c[x] = b[x] + (b[x+1] + 1) \cdot b[x+2]^1$$

$b[x]$	$b[x+1]$	$b[x+2]$	$c[x]$
cst	cst	cst	cst
var	cst	*	var
cst	cst	var	var (deg $\leq$ 1)
cst	1	var	cst

[1] Omit coordinates $[y][z]$.

3. Generate Valid Patterns

$$c[x] = b[x] + (b[x+1] + 1) \cdot b[x+2]^1$$

$b[x]$	$b[x+1]$	$b[x+2]$	$c[x]$
cst	cst	cst	cst
var	cst	*	var
cst	cst	var	var (deg ≤ 1)
cst	1	var	cst
$\vdots$	$\vdots$	$\vdots$	$\vdots$

[1] Omit coordinates $[y][z]$.

3. Generate Valid Patterns

$$B[x] = \begin{cases} 0, & b[x] \text{ is a constant;} \\ 1, & b[x] \text{ is a var.} \end{cases} \qquad V[x] = \begin{cases} 0, & \text{no condidtion on } b[x]; \\ 1, & b[x] \text{ is restricted to } 0/1. \end{cases}$$

3. Generate Valid Patterns

$$B[x] = \begin{cases} 0, & b[x] \text{ is a constant;} \\ 1, & b[x] \text{ is a var.} \end{cases} \qquad V[x] = \begin{cases} 0, & \text{no condidtion on } b[x]; \\ 1, & b[x] \text{ is restricted to } 0/1. \end{cases}$$

Table: Diffusion of variables through χ

$B[x]$	$B[x+1]$	$B[x+2]$	$V[x+1]$	$V[x+2]$	$H[x+1]$	$H[x+2]$	$C[x]$
0	0	0	*	*	*	*	0
1	0	0	*	*	*	*	1
0	0	1	0	0	*	*	1
0	0	1	1	0	1	*	0
0	0	1	1	0	0	*	1
0	1	0	0	0	*	*	1
0	1	0	0	1	*	0	0
0	1	0	0	1	*	1	1
1	0	1	0	0	*	*	1
1	0	1	1	0	*	*	1

Modeling the First χ

4. Describe valid patterns with inequality

By generating the convex hull of the set of patterns [SHW+14], we get

$$\begin{aligned}
-B[x] - B[x+1] &\geq -1\\
-B[x] + C[x] &\geq 0\\
-B[x+2] - V[x+2] &\geq -1\\
-B[x+1] - V[x+1] &\geq -1\\
-B[x] - B[x+1] - H[x+2] + C[x] &\geq -1\\
B[x] - V[x+1] - H[x+1] - C[x] &\geq -2\\
B[x] - V[x+2] + H[x+2] - C[x] &\geq -1\\
B[x] + B[x+1] + B[x+2] - C[x] &\geq 0\\
-B[x+1] - B[x+2] + V[x+1] + V[x+2] + C[x] &\geq 0\\
-B[x+1] - B[x+2] + V[x+2] + H[x+1] + C[x] &\geq 0
\end{aligned}$$

Modeling Other Steps

- Modeling the activeness of column sums in the first/second round
- Modeling χ in the second round

$\Rightarrow$ See the paper.

Modeling Other Steps

- Modeling the activeness of column sums in the first/second round
- Modeling χ in the second round

$\Rightarrow$ See the paper.

Property

The model contains no unnecessary conditions, hence could be able to find optimal conditional cubes.

Outline

Results of Key Recovery Attacks

- First analytical results on KMAC
- Improve the attack against Lake Keyak (128) from 6 to 8 rounds in the NR setting, and attack 9 rounds if the key size is 256 bits.
- Solve the FKD open problem

Target	$\|K\|$	c	Rounds	Time	Reference	Type
KMAC128	128	256	7/24	2^{76}	this	conCube
KMAC256	256	512	9/24	2^{147}	this	

Target	$\|K\|$	NR	Rounds	Time	Reference	Type
Lake Keyak	128	Yes	6/12	2^{37}	[DMP+15]	cube
	128	No	8/12	2^{74}	[HWX+17]	conCube
	128	Yes	8/12	$2^{71.01}$	this	conCube
	256	Yes	9/14	$2^{137.05}$	this	
River Keyak	128	Yes	8/12	2^{77}	this	
FKD[1600]	128	No	9/-	2^{90}	this	

NR: nonce-respected

Improved attacks on Ketje and Keccak-MAC

Target	$\lvert K \rvert$	Rounds	T	M	Reference	Type
Ketje Major	128	7/13	2^{83}	-	[LBD+17]	conCube
	128	7/13	$2^{71.24}$	-	this	
Ketje Minor	128	7/13	2^{81}	-	[LBD+17]	
	128	7/13	$2^{73.03}$	-	this	
Ketje Sr V1	128	7/13	2^{115}	2^{50}	[DMP+15]	auxCube†
	128	7/13	2^{91}	-	this	conCube
Keccak-MAC	128	256/512	7/24	2^{72}	[HWX+17]	conCube
		768	7/24	2^{75}	[LBD+17]	
		1024	6/24	$2^{58.3}$		
		1024	6/24	2^{40}	this	

† auxCube: cube-attack-like cryptanalysis

Conclusion

1. MILP models for searching two types of cubes for KECCAK.
2. First attacks on `KMAC`, and improved attacks on KEYAK and KETJE.
3. Solve the FKD open problem.
4. The security of Keccak-based constructions is far from being threatened.

Conclusion

1. MILP models for searching two types of cubes for KECCAK.
2. First attacks on `KMAC`, and improved attacks on KEYAK and KETJE.
3. Solve the FKD open problem.
4. The security of Keccak-based constructions is far from being threatened.

Thank you for your attention!