On Some Variants of Cube-Attack-Like Cryptanalysis on SHA-3 Designs

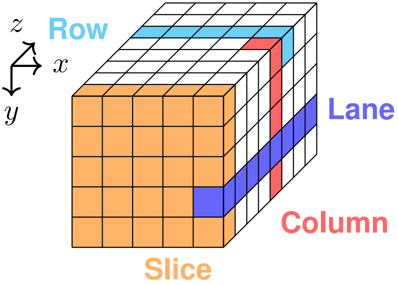
Mohammad Vaziri

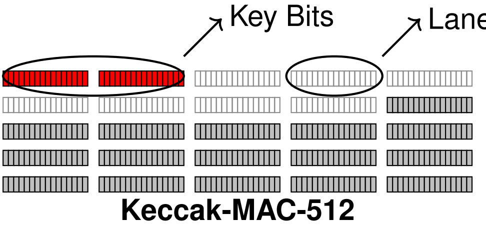
Permutation-based Crypto 2025

4 May 2025

Table of Contents

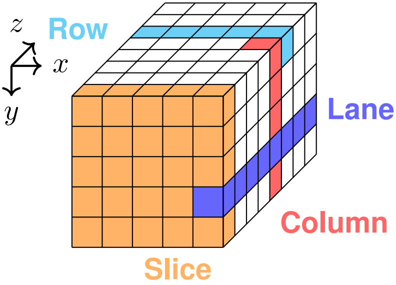
Keccak Sponge Function (Description)

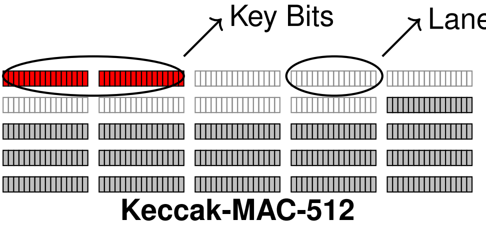




$A_r[i][j][k] \rightarrow$ the bit indexed by (i, j, k) of state A, round $r + 1$.

Keccak Sponge Function (Description)

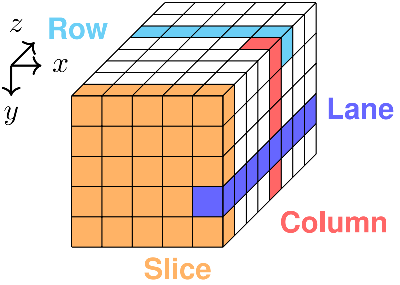


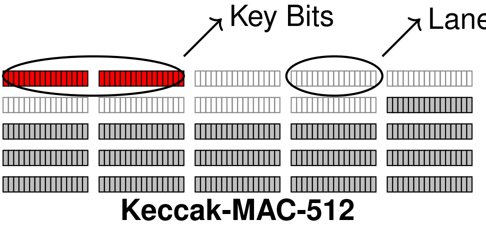


$A_r[i][j][k] \rightarrow$ the bit indexed by (i, j, k) of state A, round $r+1$.

$$\theta : A[x][y] = A[x][y] \bigoplus \sum_{j=0}^{4} (A[x-1][j] \bigoplus (A[x+1][j] \lll 1)).$$

Keccak Sponge Function (Description)



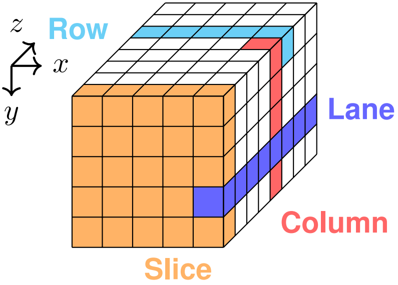


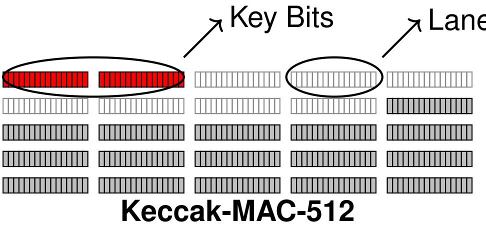
$A_r[i][j][k] \rightarrow$ the bit indexed by (i, j, k) of state A, round $r + 1$.

$$\theta : A[x][y] = A[x][y] \bigoplus \sum_{j=0}^{4} (A[x-1][j] \bigoplus (A[x+1][j] \lll 1)).$$

$$\rho : A[x][y] = A[x][y] \lll r[x, y].$$

Keccak Sponge Function (Description)





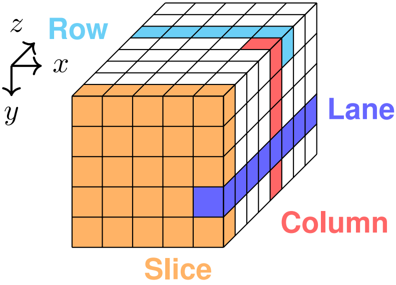
$A_r[i][j][k] \rightarrow$ the bit indexed by (i, j, k) of state A, round $r + 1$.

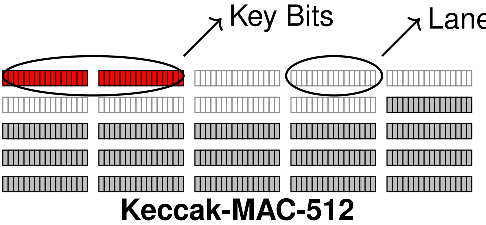
$$\theta : A[x][y] = A[x][y] \bigoplus \sum_{j=0}^{4} (A[x-1][j] \bigoplus (A[x+1][j] \lll 1)).$$

$$\rho : A[x][y] = A[x][y] \lll r[x, y].$$

$$\Pi : A[y][2x + 3y] = A[x][y].$$

Keccak Sponge Function (Description)





$A_r[i][j][k] \rightarrow$ the bit indexed by (i, j, k) of state A, round $r+1$.

$$\theta : A[x][y] = A[x][y] \bigoplus \sum_{j=0}^{4} (A[x-1][j] \bigoplus (A[x+1][j] \lll 1)).$$

$$\rho : A[x][y] = A[x][y] \lll r[x, y].$$

$$\Pi : A[y][2x+3y] = A[x][y].$$

$$\chi : A[x][y] = A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y]).$$

Cube Attack(Simple Example and Terminology)

- Assume that f is:

$$f(v_1, v_2, k_1, k_2) = v_1 + v_1 v_2 k_1 + v_1 v_2 k_1 k_2 + v_2 k_1 + k_1 k_2.$$

Cube Attack(Simple Example and Terminology)

- Assume that f is:

$$f(v_1, v_2, k_1, k_2) = v_1 + v_1v_2k_1 + v_1v_2k_1k_2 + v_2k_1 + k_1k_2.$$

- For recovering the $\{k_1, k_2\}$, we choose v_1v_2 as cube, so the f can be represented as:

$$f(v_1, v_2, k_1, k_2) = v_1v_2(k_1 + k_1k_2) + v_1 + v_2k_1 + k_1k_2.$$

Cube Attack(Simple Example and Terminology)

- Assume that f is:
$$f(v_1, v_2, k_1, k_2) = v_1 + v_1v_2k_1 + v_1v_2k_1k_2 + v_2k_1 + k_1k_2.$$

- For recovering the $\{k_1, k_2\}$, we choose v_1v_2 as cube, so the f can be represented as:
$$f(v_1, v_2, k_1, k_2) = v_1v_2(k_1 + k_1k_2) + v_1 + v_2k_1 + k_1k_2.$$

- By doing summation on all of the possible values of the cube variables v_1 and v_2 we have:
$$f(0, 0, k_1, k_2) + f(0, 1, k_1, k_2) + f(1, 0, k_1, k_2) + f(1, 1, k_1, k_2) = k_1 + k_1k_2.$$

Cube Attack(Simple Example and Terminology)

- Assume that f is:

$$f(v_1, v_2, k_1, k_2) = v_1 + v_1 v_2 k_1 + v_1 v_2 k_1 k_2 + v_2 k_1 + k_1 k_2.$$

- For recovering the $\{k_1, k_2\}$, we choose $v_1 v_2$ as cube, so the f can be represented as:

$$f(v_1, v_2, k_1, k_2) = v_1 v_2 (k_1 + k_1 k_2) + v_1 + v_2 k_1 + k_1 k_2.$$

- By doing summation on all of the possible values of the cube variables v_1 and v_2 we have:

$$f(0, 0, k_1, k_2) + f(0, 1, k_1, k_2) + f(1, 0, k_1, k_2) + f(1, 1, k_1, k_2) = k_1 + k_1 k_2.$$

Cube Attack Terminology

$f : X^n \rightarrow \{0, 1\} \quad \longrightarrow \quad f(x) = t.P_t(x) + Q(x), \quad t = x_0 ... x_{k-1}$

Then sum of f over all values of t is:

$$\sum_{x' = (x_0, ..., x_{k-1}) \in C_t} f(x', x) = P_t(\underbrace{1, ..., 1}_{k}, x_k, ..., x_{n-1})$$

C_t contains all binary vectors of the length k and $P_t(x)$ is called **superpoly** of t.

Cube-Attack-Like Cryptanalysis on Keccak

- Is divided into two Offline and Online Phases.
- The cube variables do not multiply together in the first round.
- By linearizing the first round, attacking r-round, require 2^{r-1} cube variables.
- Cube variables instead multiply with certain key bits (related-key-bits).
- In offline phase, for every possible related-key-bit value, the cube sum over 2^{r-1} cube variables is computed and stored in a list L.
- In online phase, the cube sum over 2^{r-1} cube variables is computed, while the secret key is set.
- For each match found in L, the corresponding candidate values for the related-key-bits are retrieved.

Cube-Attack-Like Cryptanalysis on Keccak

- Attack complexity depends on the number of cube variables S and related-key-bits T.

Trade-off Between Offline and Online Phases

-A trade-off between offline and online phases lowers attack complexity.
-In the Offline phase, controlling diffusion for half the related-key bits prevents multiplications with cube variables.
-This is achieved by setting auxiliary variables to match values in their same column.
-In the Online phase, this trade-off requires computing the cube sum for each possible value of the auxiliary-variables.

- Since S is fixed, minimizing T reduces attack complexity.

- To minimize T, the works [des, 2019] and [tosc, 2018] independently propose techniques based on Mixed-Integer-Linear-Programming (MILP).

Cube-Attack-Like Cryptanalysis on Keccak

Attack Process:

- Retrieve the position of cube variables and related-key-bits in the initial state by the MILP model proposed by [des, 2019].

Offline Phase

-Choose half of the related-keys as guessing-keys and the rest as auxiliary-variables.
-For each of the $2^{T/2}$ possible values of guessing-keys calculate the cube sum and store the amounts the list L.
-The time complexity is $2^{S+T/2}$ and the memory complexity is $2^{T/2}$.

Attack Process:

- Retrieve the position of cube variables and related-key-bits in the initial state by the MILP model proposed by [des, 2019].

Offline Phase

-Choose half of the related-keys as guessing-keys and the rest as auxiliary-variables.
-For each of the $2^{T/2}$ possible values of guessing-keys calculate the cube sum and store the amounts the list L.
-The time complexity is $2^{S+T/2}$ and the memory complexity is $2^{T/2}$.

Online Phase

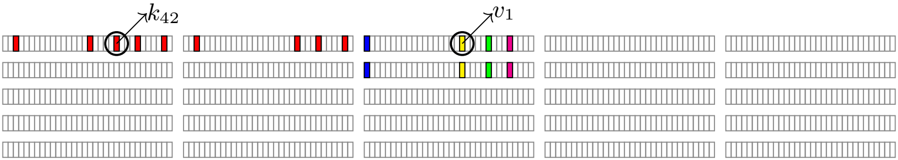
-For each of the $2^{T/2}$ possible values of auxiliary-variables Calculate the cube sum and search for a match in L.
-The time complexity is $2^{S+T/2}$.

- The total time, memory and data complexity for recovering T related-key-bits is $2^{S+T/2} + 2^{S+T/2}$, $2^{T/2}$ and 2^{S}.

Cube-Attack-Like Cryptanalysis on Keccak

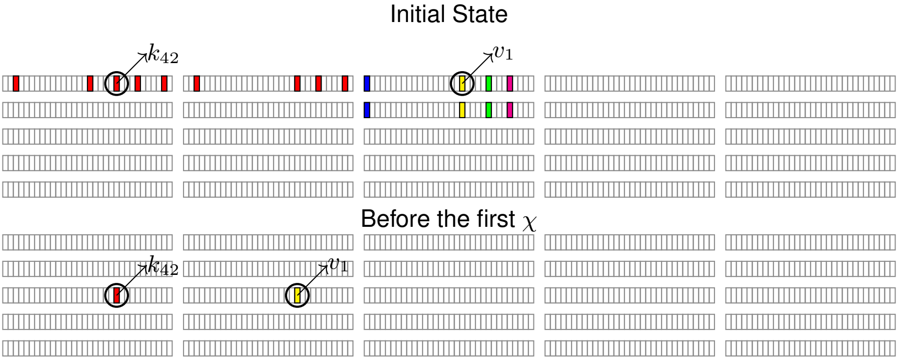
- Let's assume that we want to attack 3 rounds of Keccak-Mac-512.

Initial State



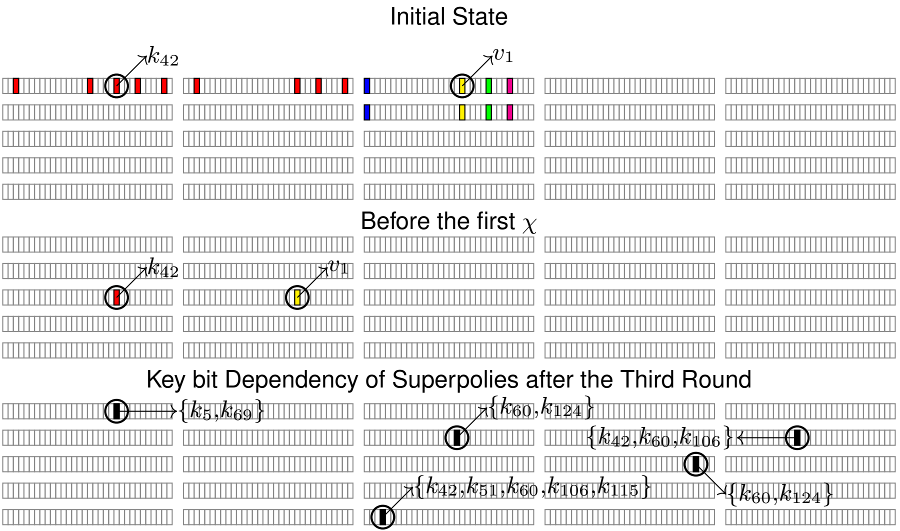
Cube-Attack-Like Cryptanalysis on Keccak

- Let's assume that we want to attack 3 rounds of Keccak-Mac-512.



Cube-Attack-Like Cryptanalysis on Keccak

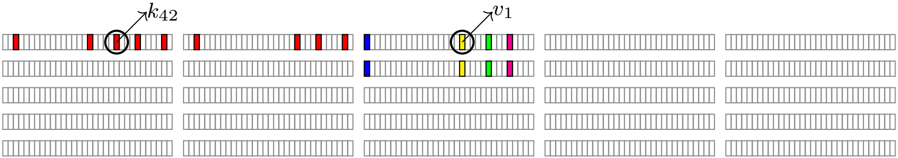
- Let's assume that we want to attack 3 rounds of Keccak-Mac-512.



Improvement with Exploiting State Differences

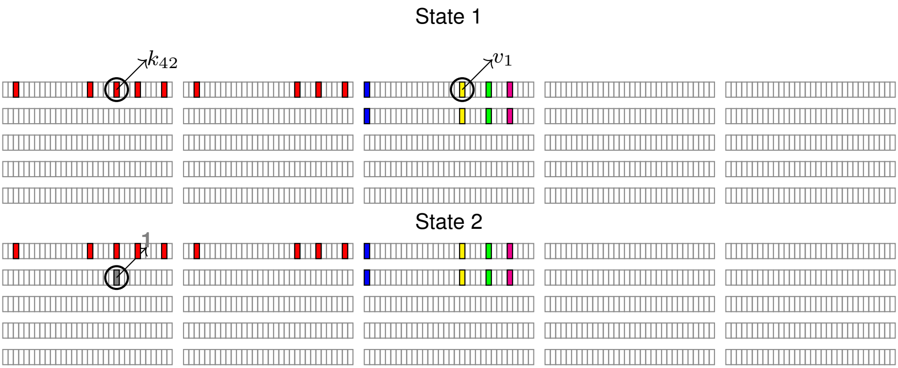
- To attack 3 rounds of Keccak-MAC-512, consider the following two states.

State 1



Improvement with Exploiting State Differences

- To attack 3 rounds of Keccak-MAC-512, consider the following two states.



Improvement with Exploiting State Differences

- The differences of State 1 and State 2 after some operations.

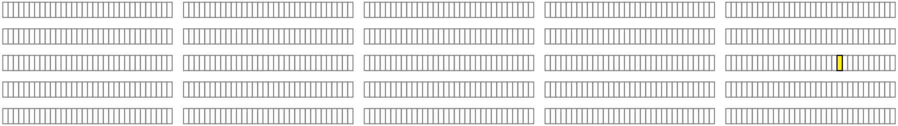
Initial difference between State 1 and State 2

Improvement with Exploiting State Differences

- The differences of State 1 and State 2 after some operations.

Initial difference between State 1 and State 2

Difference between State 1 and State 2 after the first χ operation

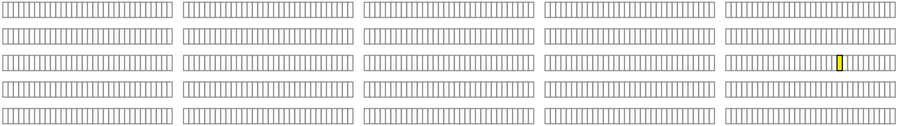


Improvement with Exploiting State Differences

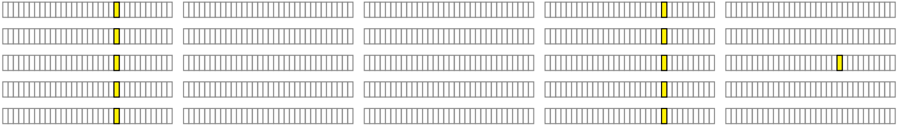
- The differences of State 1 and State 2 after some operations.

Initial difference between State 1 and State 2

Difference between State 1 and State 2 after the first χ operation



Difference between State 1 and State 2 after the second θ operation



Improvement with Exploiting State Differences

Difference between State 1 and State 2 after the third χ operation

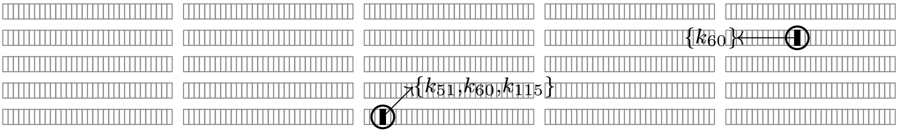


Table: Comparison of superpolies between two techniques

Bit Position	First Technique	Second Technique
$A_2[4][1][26]$	$\{k_{42}, k_{60}, k_{106}\}$	k_{60}
$A_2[2][4][7]$	$\{k_{42}, k_{51}, k_{60}, k_{106}, k_{115}\}$	$\{k_{51}, k_{60}, k_{115}\}$
$A_2[0][0][43]$	$\{k_5, k_{69}\}$	Canceled
$A_2[2][1][35]$	$\{k_{60}, k_{124}\}$	Canceled
$A_2[3][2][57]$	$\{k_{60}, k_{124}\}$	Canceled

Improvement with Exploiting State Differences

- **Axis-Cube-Variable:** A cube variable that:
  -Persists in the state difference pattern following the first χ operation.
- **Axis-Related-Key:** A related-key-bit that:
  -Only multiplies with axis cube variables after first χ-layer,
  -Negated after first θ in one of the states for creating state differences.

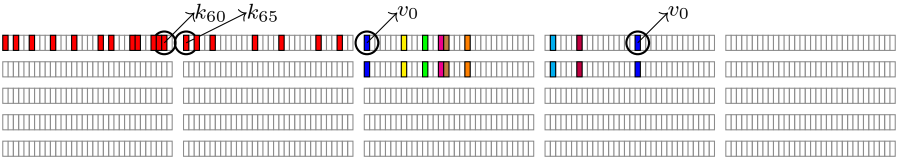
- Our technique's key strategy is Constructing two minimally distinct states.
- After independent cube summations, differential analysis reveals extensive cancellations in superpoly terms.
- the presence of axis cube variables introduces differences between the states.

Better control of axis cube variable propagation directly enhances state similarity.

Improvement with Exploiting State Differences

- To attack 4 rounds of Keccak-MAC-512, consider the following two states.

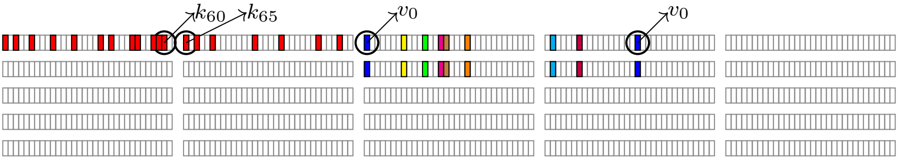
State 1



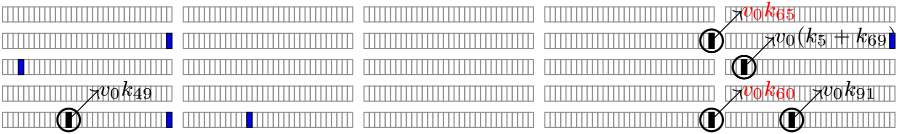
Improvement with Exploiting State Differences

- To attack 4 rounds of Keccak-MAC-512, consider the following two states.

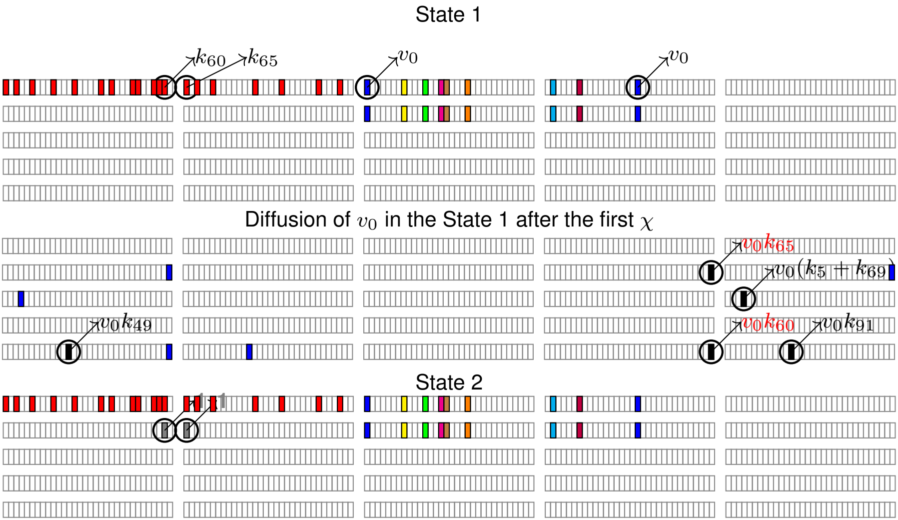
State 1



Diffusion of v_0 in the State 1 after the first χ



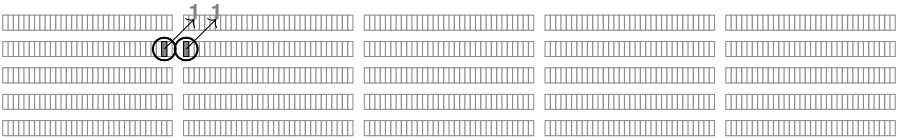
- To attack 4 rounds of Keccak-MAC-512, consider the following two states.



Improvement with Exploiting State Differences

- The differences of State 1 and State 2 after some operations.

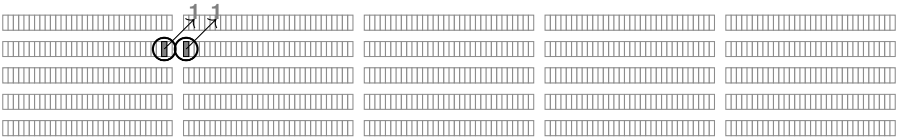
Initial difference between State 1 and State 2



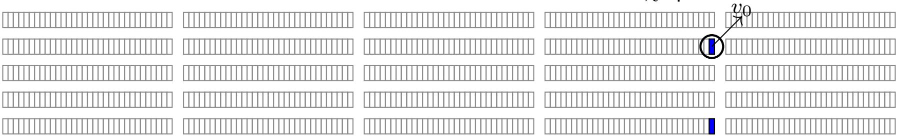
Improvement with Exploiting State Differences

- The differences of State 1 and State 2 after some operations.

Initial difference between State 1 and State 2



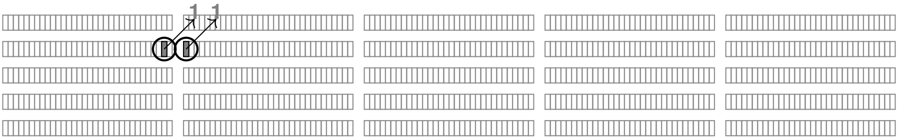
Difference between State 1 and State 2 after the first χ operation



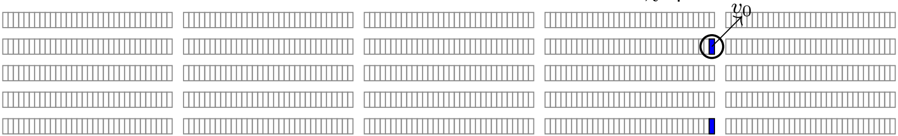
Improvement with Exploiting State Differences

- The differences of State 1 and State 2 after some operations.

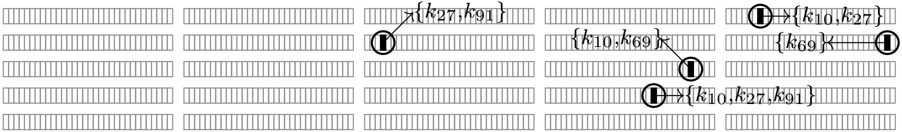
Initial difference between State 1 and State 2



Difference between State 1 and State 2 after the first χ operation



Difference between State 1 and State 2 after the fourth χ operation



Compute Limitations

- In the technique, we have computationally verified the improvements.
- We developed a dedicated tool capable of recovering all terms in all superpolies and supports multi-threading.
- we observed that tracking all terms is unnecessary.
- Recovering $t \cdot P_t(x)$ requires only terms containing all cube variables.
- χ for the second round onward can be replaced as follows:

$$b_i = a_i + (a_{i+1} + 1) \cdot a_{i+2} \quad \Rightarrow \quad b_i = a_{i+1} \cdot a_{i+2}$$

Compute Limitations

- In the technique, we have computationally verified the improvements.
- We developed a dedicated tool capable of recovering all terms in all superpolies and supports multi-threading.
- we observed that tracking all terms is unnecessary.
- Recovering $t \cdot P_t(x)$ requires only terms containing all cube variables.
- χ for the second round onward can be replaced as follows:

$$b_i = a_i + (a_{i+1} + 1) \cdot a_{i+2} \quad \Rightarrow \quad b_i = a_{i+1} \cdot a_{i+2}$$

Observation

We observed that in each of the cases, the superpolies do not depend on the axis-related-keys.

Application to 5-round Keccak-MAC-512

Automated Technique [DES, 2019]:

- MILP model: **30 related-key bits**
  - 15 guessing keys (Offline)
  - 15 auxiliary vars (Online)
- Complexities:

Time:	Offline: $2^{15} \times 2^{16} = 2^{31}$
	Online: $2^{15} \times 2^{16} = 2^{31}$
	Full key: $4 \times 2^{32} + 2^{8} \approx 2^{34}$
Memory:	$4 \times 2^{15} = 2^{17}$
Data:	2^{16}

Application to 5-round Keccak-MAC-512

Automated Technique [DES, 2019]:

- MILP model: **30 related-key bits**
  - 15 guessing keys (Offline)
  - 15 auxiliary vars (Online)
- Complexities:
  - Time: Offline: $2^{15} \times 2^{16} = 2^{31}$
  - Online: $2^{15} \times 2^{16} = 2^{31}$
  - Full key: $4 \times 2^{32} + 2^{8} \approx 2^{34}$
  - Memory: $4 \times 2^{15} = 2^{17}$
  - Data: 2^{16}

Our Technique:

- Bit position $A_4[4][3][15]$:
  - 4 guessing keys (Offline)
  - 8 auxiliary vars (Online)
- Complexities:
  - Time: Offline: $4 \times 2^{4} \times 2^{17} = 2^{23}$
  - Online: $8 \times 2^{8} \times 2^{17} = 2^{28}$
  - Full key: $10 \times (2^{23} + 2^{28}) + 2^{8} \approx 2^{31.3}$
  - Memory: $10 \times 4 \times 2^{4} \approx 2^{9.3}$
  - Data: $2^{17} \times 12 \approx 2^{21}$

Thank you for your attention!
Questions?