Cube Attacks

Presented by
Donghoon Chang
NIST

August 28 2020

Cube Attacks

- Basic Cube Attacks were introduced in EUROCRYPT 2009.
- There two types of Enhanced Cube Attacks.
  - **Divide-and-Conquer Cube Attack (EUROCRYPT 2015)**
  - **Conditional Cube Attack (EUROCRYPT 2017)**
- These two enhanced attacks are the best known cube attacks till today which are key (or state)-recovery attacks on MAC or AEAD, so it is worthwhile to understand it.
- The papers in 2017-2020 follow these two enhanced attack techniques.

Basic Notations and Results

Cubes #1

- Given a monomial $x_1x_2x_3$,

  $C_{x_1x_2x_3}$:={(0,0,0),(0,0,1),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0),(1,1,1)},

  $C_{x_1x_2x_3}$ is defined as the set of all binary vectors of the length 3,

  $C_{x_1x_2x_3}$ is called *the cube* with *3 cube variables,* x_1,x_2,x_3.

*We follow the notations defined in EUROCRYPT 2015 paper, “Cube Attacks and Cube-attack-like Cryptanalysis on the Round-reduced Keccak Sponge Function”.

Cubes #2

- Given a monomial x_4x_5,

  $C_{x_4x_5}$:={(0,0),(0,1),(1,0),(1,1)},

  $C_{x_4x_5}$ is defined as the set of all binary vectors of the length 2,

  $C_{x_4x_5}$ is called *the cube* with *2 cube variables,* x_4,x_5.

Cubes #3

- Given a monomial x_1,

  C_{x_1} :={0,1},

  C_{x_1} is defined as the set of all binary vectors of the length 1,

  C_{x_1} is called *the cube* with *1 cube variable,* x_1.

Cubes #4

- Given a monomial t with k variables,

  C_t :={(0,…,0,0),(0,…,0,1),(0,…,1,0),(0,…,1,1),….,(1,…,1,1)},

  C_t is the set of all binary vectors of the length k.

  C_t is called *the cube* with k *cube variables*.

Division #1

- Given a polynomial $f(x_1,x_2,x_3,x_4)=x_1+x_1x_2x_3+x_1x_2x_4+x_3$,
- we want to divide f by x_1.
- We know that there are unique Q (quotient) and R (remainder) such that $f(x_1,x_2,x_3,x_4)= x_1Q+R$.
- $f(x_1,x_2,x_3,x_4) = x_1 (1+x_2x_3+x_2x_4) + x_3$
- So, $Q = 1+x_2x_3+x_2x_4$ and $R = x_3$.
- Note that there is no x_1 in $Q = 1+x_2x_3+x_2x_4$ and x_1 does not divide each term of $R = x_3$.
- Therefore, $Q = Q(x_2,x_3,x_4)=1+x_2x_3+x_2x_4$, $R = R(x_2,x_3,x_4) = x_3$

Division #2

- Given a polynomial $f(x_1,x_2,x_3,x_4)=x_1+x_1x_2x_3+x_1x_2x_4+x_3$,
- we want to divide f by x_1x_2.
- We know that there are unique Q (quotient) and R (remainder) such that $f(x_1,x_2,x_3,x_4)= x_1x_2\ Q+R$.
- $f(x_1,x_2,x_3,x_4) = x_1x_2\ (x_3+x_4) + x_1+x_3$
- So, $Q = x_3+x_4$ and $R = x_1+x_3$.
- Note that there is neither x_1 or x_2 in $Q = x_3+x_4$ and x_1x_2 does not divide each term of $R = x_1+x_3$.
- Therefore, $Q = Q\ (x_3,x_4)=x_3+x_4$, $R = R(x_1,x_2,x_3,x_4) = x_1+x_3$

Cube and Quotient

- Given a polynomial $f(x_1,x_2,x_3,x_4)$,
- we want to divide f by x_1x_2.
- Then, we can describe $f(x_1,x_2,x_3,x_4) = x_1x_2\,Q(x_3,x_4) + R(x_1,x_2,x_3,x_4)$, where Q is the quotient and R is the remainder.
- We define the cube $C_{x_1x_2} := \{(0,0),(0,1),(1,0),(1,1)\}$.
- Then, Xuejia Lai* proved

$$\sum_{(x_1,x_2)\in C_{x_1x_2}} f(x_1,x_2,x_3,x_4) = Q(x_3,x_4)$$

* "Higher Order Derivatives and Differential Cryptanalysis", Communications and Cryptography 1994.

Cube and Quotient (Example #1)

- Given a polynomial $f(x_1,x_2,x_3,x_4)=x_1+x_1x_2x_3+x_1x_2x_4+x_3$,
- we want to divide f by x_1x_2.
- We know $f(x_1,x_2,x_3,x_4) = x_1x_2\,(x_3+x_4) + x_1+x_3$.
- We define the cube $C_{x_1x_2}$:={(0,0),(0,1),(1,0),(1,1)}. Then,

$$\sum_{(x_1,x_2)\in C_{x_1x_2}} f(x_1,x_2,x_3,x_4) = x_3+x_4$$

Cube and Quotient (Example #1)

- Given a polynomial $f(x_1,x_2,x_3,x_4)=x_1+x_1x_2x_3+x_1x_2x_4+x_3$,
- we want to divide f by x_1x_2.
- We know $f(x_1,x_2,x_3,x_4) = x_1x_2\,(x_3+x_4) + x_1+x_3$.
- We define the cube $C_{x_1x_2}$:={(0,0),(0,1),(1,0),(1,1)}. Then,

$$\sum_{(x_1,x_2)\in C_{x_1x_2}} f(x_1,x_2,x_3,x_4) = x_3+x_4$$

called the superpoly of $C_{x_1x_2}$

Cube and Quotient (Example #2)

- Given a polynomial $f(x_1,x_2,x_3,x_4)=x_1+x_1x_2x_3+x_1x_2x_4+x_3$,
- we want to divide f by x_1.
- We know $f(x_1,x_2,x_3,x_4) = x_1 (x_2x_3 +x_2x_4+1) + x_3$.
- We define the cube $C_{x_1} :=\{0,1\}$. Then,

$$\sum_{x_1 \in C_{x_1}} f(x_1,x_2,x_3,x_4) = x_2x_3 +x_2x_4+1$$

Cube and Quotient (Example #2)

- Given a polynomial $f(x_1,x_2,x_3,x_4)=x_1+x_1x_2x_3+x_1x_2x_4+x_3$,
- we want to divide f by x_1.
- We know $f(x_1,x_2,x_3,x_4) = x_1 (x_2x_3 +x_2x_4+1) + x_3$.
- We define the cube C_{x_1} :={0,1}. Then,

$$\sum_{x_1 \in C_{x_1}} f(x_1,x_2,x_3,x_4) = x_2x_3 +x_2x_4+1$$

called the superpoly of C_{x_1}

Cube and Quotient (Example #3)

- Given a polynomial $f(x_1,x_2,x_3,x_4)=x_1+x_1x_2x_3+x_1x_2x_4+x_3$,
- we want to divide f by $x_1x_2x_3$.
- We know $f(x_1,x_2,x_3,x_4) = x_1x_2x_3\,(1) + x_1+x_1x_2x_4+x_3$.
- We define the cube $C_{x_1x_2x_3}$:={(0,0,0),(0,0,1),…,(1,1,1)}. Then,

$$\sum_{(x_1,x_2,x_3)\in C_{x_1x_2x_3}} f(x_1,x_2,x_3,x_4) = 1$$

Cube and Quotient (Example #4)

- Given a polynomial $f(x_1,x_2,x_3,x_4)=x_1+x_1x_2x_3+x_1x_2x_4+x_3$,
- we want to divide f by $x_2x_3x_4$.
- We know $f(x_1,x_2,x_3,x_4) = x_2x_3x_4\,(0) + x_1+x_1x_2x_3+x_1x_2x_4+x_3$.
- We define the cube $C_{x_2x_3x_4}$:={(0,0,0),(0,0,1),…,(1,1,1)}. Then,

$$\sum_{(x_2,x_3,x_4)\in C_{x_2x_3x_4}} f(x_1,x_2,x_3,x_4) = 0$$

Cube and Quotient (Example #5)

- $f(x_1,x_2,x_3,x_4) = x_1x_2\,(x_3+x_4) + x_1+x_3$.
- Cube $C_{x_1x_2}$:={(0,0),(0,1),(1,0),(1,1)}.
- Then,

$$\sum_{(x_1,x_2)\in C_{x_1x_2}} f(x_1,x_2,x_3,x_4) = x_3+x_4$$

Cube and Quotient (Example #5)

- $f(x_1,x_2,x_3,x_4) = x_1x_2\,(x_3+x_4) + x_1+x_3.$
- Cube $C_{x_1x_2}$:={(0,0),(0,1),(1,0),(1,1)}.
- Then,

$$\sum_{(x_1,x_2)\in C_{x_1x_2}} f(x_1,x_2,x_3,x_4) = x_3+x_4$$

The coefficient of x_3 is 1.

Cube and Quotient (Example #5)

- $f(x_1,x_2,x_3,x_4) = x_1x_2\,(x_3+x_4) + x_1+x_3$.
- Cube $C_{x_1x_2}$:={(0,0),(0,1),(1,0),(1,1)}.
- Then,

The coefficient of x_3 is 1.

$$\sum_{(x_1,x_2)\in C_{x_1x_2}} f(x_1,x_2,x_3,x_4) = x_3+x_4$$

$$\sum_{(x_1,x_2)\in C_{x_1x_2}} f(x_1,x_2,1,0) = \text{The coefficient of } x_3 = 1$$

Cube and Quotient (Example #5)

- $f(x_1,x_2,x_3,x_4) = x_1x_2\,(x_3+x_4) + x_1+x_3.$
- Cube $C_{x_1x_2}$:={(0,0),(0,1),(1,0),(1,1)}.
- Then,

$$\sum_{(x_1,x_2)\in C_{x_1x_2}} f(x_1,x_2,x_3,x_4) = x_3+x_4$$

The coefficient of x_4 is 1.

Cube and Quotient (Example #5)

- $f(x_1,x_2,x_3,x_4) = x_1x_2\,(x_3+x_4) + x_1+x_3.$
- Cube $C_{x_1x_2}$:={(0,0),(0,1),(1,0),(1,1)}.
- Then,

The coefficient of x_4 is 1.

$$\sum_{(x_1,x_2)\in C_{x_1x_2}} f(x_1,x_2,x_3,x_4) = x_3+x_4$$

$$\sum_{(x_1,x_2)\in C_{x_1x_2}} f(x_1,x_2,0,1) = \text{The coefficient of } x_4 = 1$$

Cube and Quotient (Example #6)

- $f(x_1,x_2,x_3,x_4) = x_1 (x_2x_3 + x_2x_4 + 1) + x_3$.
- Cube $C_{x_1} := \{0,1\}$.
- Then,

$$\sum_{x_1 \in C_{x_1}} f(x_1,x_2,x_3,x_4) = x_2x_3 + x_2x_4 + 1$$

Cube and Quotient (Example #6)

- $f(x_1,x_2,x_3,x_4) = x_1 (x_2x_3 + x_2x_4 + 1) + x_3$.
- Cube C_{x_1} :={0,1}.
- Then,

$$\sum_{x_1 \in C_{x_1}} f(x_1,x_2,x_3,x_4) = x_2x_3 + x_2x_4 + 1$$

the constant is 1.

Cube and Quotient (Example #6)

- $f(x_1,x_2,x_3,x_4) = x_1 (x_2x_3 + x_2x_4 + 1) + x_3$.
- Cube C_{x_1} :={0,1}.
- Then,

the constant is 1.

$$\sum_{x_1 \in C_{x_1}} f(x_1,x_2,x_3,x_4) = x_2x_3 + x_2x_4 + 1$$

$$\sum_{x_1 \in C_{x_1}} f(x_1, 0, 0, 0) = \text{The constant} = 1$$

Cube and Quotient (Example #6)

- $f(x_1,x_2,x_3,x_4) = x_1 (x_2x_3 + x_2x_4 + 1) + x_3$.
- Cube C_{x_1} :={0,1}.
- Then,

$$\sum_{x_1 \in C_{x_1}} f(x_1,x_2,x_3,x_4) = x_2x_3 + x_2x_4 + 1$$

The coefficient of x_2x_3 is 1.

Cube and Quotient (Example #6)

- $f(x_1,x_2,x_3,x_4) = x_1 (x_2x_3 + x_2x_4 + 1) + x_3$.
- Cube C_{x_1} :={0,1}.
- Then,

The coefficient of x_2x_3 is 1.

$$\sum_{x_1 \in C_{x_1}} f(x_1,x_2,x_3,x_4) = x_2x_3 + x_2x_4 + 1$$

$$\sum_{x_1 \in C_{x_1}} f(x_1, 1, 1, 0) = \text{The coefficient of } x_2x_3 + \text{The constant} = 1 + 1 = 0$$

Cube Attacks and Cube-attack-like Cryptanalysis on the Round-reduced Keccak Sponge Function

By Dinur, Morawiecki, Pieprzyk, Srebrny, Straus (**EUROCRYPT 2015**)

EUROCRYPT 2015 paper introduced

- Three kinds of Cube Attack Approaches
  - **The First: Divide-and-Conquer Attack with Partial Input Key**
  - The Second: Balanced Attack with Partial Input Key
  - The Third: the first&second Attacks with Partial Secret Internal Information

First Attack Idea

Superpolys with Some (not all) Variables #1

- $f(x_1,x_2,x_3,x_4) = x_1 (x_3+x_4) + x_2x_4$.
- Cube C_{x_1}:={0,1}.
- Then,

$$\sum_{x_1 \in C_{x_1}} f(x_1,x_2,x_3,x_4) = x_3+x_4$$

$$\sum_{x_1 \in C_{x_1}} f(x_1, 0,x_3,x_4) = x_3+x_4$$

$$\sum_{x_1 \in C_{x_1}} f(x_1, 1,x_3,x_4) = x_3+x_4$$

First Attack Idea

Superpolys with Some (not all) Variables #1

- $f(x_1,x_2,x_3,x_4) = x_1 (x_3+x_4) + x_2x_4$.
- Cube C_{x_1}:={0,1}.
- Then,

$$\sum_{x_1 \in C_{x_1}} f(x_1,x_2,x_3,x_4) = x_3+x_4$$

First Attack Idea

Superpolys with Some (not all) Variables #1

- $f(x_1,x_2,x_3,x_4) = x_1 (x_3+x_4) + x_2x_4$.
- Cube C_{x_1}:={0,1}.
- Then,

$$\sum_{x_1 \in C_{x_1}} f(x_1,x_2,x_3,x_4) = x_3+x_4$$

$$\sum_{x_1 \in C_{x_1}} f(x_1, 0,x_3,x_4) = x_3+x_4$$

$$\sum_{x_1 \in C_{x_1}} f(x_1, 1,x_3,x_4) = x_3+x_4$$

The Superpoly of the cube C_{x_1} remains same regardless of the value of x_2.

First Attack Idea

Superpolys with Some (not all) Variables #2

- $f(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4) = M_1M_2Q(K_1,K_2,M_3,M_4)+ R(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4)$.
- Assume that
  - K_1,K_2,K_3,K_4 are 4-bit secret keys and
  - By the oracle $f(K_1,K_2,K_3,K_4,\cdot,\cdot,\cdot,\cdot)$-queries, the attacker can make a query (M_1,M_2,M_3,M_4) and then get the value of $f(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4)$.

First Attack Idea

Superpolys with Some (not all) Variables #2

- $f(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4) = M_1M_2Q(K_1,K_2,M_3,M_4)+ R(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4)$.
- The Superpoly $Q(K_1,K_2,M_3,M_4)$ does not depend on either K_3 or K_4.
- For each $(K_1,K_2) \in \{(0,0),(0,1),(1,0),(1,1)\}$, the attacker computes the followings (offline):

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(0,0,0,0,M_1,M_2,0,0) = Q(0,0,0,0)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(0,0,0,0,M_1,M_2,0,1) = Q(0,0,0,1)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(0,0,0,0,M_1,M_2,1,0) = Q(0,0,1,0)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(0,0,0,0,M_1,M_2,1,1) = Q(0,0,1,1)$$

First Attack Idea

Superpolys with Some (not all) Variables #2

- $f(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4) = M_1M_2Q(K_1,K_2,M_3,M_4)+ R(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4)$.
- The Superpoly $Q(K_1,K_2,M_3,M_4)$ does not depend on either K_3 or K_4.
- For each $(K_1,K_2) \in \{(0,0),(0,1),(1,0),(1,1)\}$, the attacker computes the followings (offline):

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(0,1,0,0,M_1,M_2,0,0) = Q(0,1,0,0)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(0,1,0,0,M_1,M_2,0,1) = Q(0,1,0,1)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(0,1,0,0,M_1,M_2,1,0) = Q(0,1,1,0)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(0,1,0,0,M_1,M_2,1,1) = Q(0,1,1,1)$$

First Attack Idea

Superpolys with Some (not all) Variables #2

- $f(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4) = M_1M_2Q(K_1,K_2,M_3,M_4)+ R(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4)$.
- The Superpoly $Q(K_1,K_2,M_3,M_4)$ does not depend on either K_3 or K_4.
- For each $(K_1,K_2) \in \{(0,0),(0,1),(1,0),(1,1)\}$, the attacker computes the followings (offline):

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(1,0,0,0,M_1,M_2,0,0) = Q(1,0,0,0)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(1,0,0,0,M_1,M_2,0,1) = Q(1,0,0,1)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(1,0,0,0,M_1,M_2,1,0) = Q(1,0,1,0)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(1,0,0,0,M_1,M_2,1,1) = Q(1,0,1,1)$$

First Attack Idea

Superpolys with Some (not all) Variables #2

- $f(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4) = M_1M_2Q(K_1,K_2,M_3,M_4)+ R(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4)$.
- The Superpoly $Q(K_1,K_2,M_3,M_4)$ does not depend on either K_3 or K_4.
- For each $(K_1,K_2) \in \{(0,0),(0,1),(1,0),(1,1)\}$, the attacker computes the followings (offline):

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(1,1,0,0,M_1,M_2,0,0) = Q(1,1,0,0)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(1,1,0,0,M_1,M_2,0,1) = Q(1,1,0,1)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(1,1,0,0,M_1,M_2,1,0) = Q(1,1,1,0)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(1,1,0,0,M_1,M_2,1,1) = Q(1,1,1,1)$$

First Attack Idea

Superpolys with Some (not all) Variables #2

- $f(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4) = M_1M_2Q(K_1,K_2,M_3,M_4)+ R(K_1,K_2,K_3,K_4,M_1,M_2,M_3,M_4)$.
- The Superpoly $Q(K_1,K_2,M_3,M_4)$ does not depend on either K_3 or K_4.
- For each $(K_1,K_2) \in \{(0,0),(0,1),(1,0),(1,1)\}$, the attacker computes the followings (offline):
- Store the following off-line cube-sum values.

Q(0,0,0,0)	Q(0,1,0,0)	Q(1,0,0,0)	Q(1,1,0,0)
Q(0,0,0,1)	Q(0,1,0,1)	Q(1,0,0,1)	Q(1,1,0,1)
Q(0,0,1,0)	Q(0,1,1,0)	Q(1,0,1,0)	Q(1,1,1,0)
Q(0,0,1,1)	Q(0,1,1,1)	Q(1,0,1,1)	Q(1,1,1,1)

First Attack Idea

Superpolys with Some (not all) Variables #2

- Now, we move to the online phase. The attacker does not know the 4-bit actual secret key K_1,K_2,K_3,K_4.
- The Attacker makes oracle $f(K_1,K_2,K_3,K_4,\cdot,\cdot,\cdot,\cdot)$-queries and compute the on-line cube-sums:

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(K_1,K_2,K_3,K_4,M_1,M_2,0,0) = Q(K_1,K_2,0,0)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(K_1,K_2,K_3,K_4,M_1,M_2,0,1) = Q(K_1,K_2,0,1)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(K_1,K_2,K_3,K_4,M_1,M_2,1,0) = Q(K_1,K_2,1,0)$$

$$\sum_{(M_1,M_2)\in C_{M_1M_2}} f(K_1,K_2,K_3,K_4,M_1,M_2,1,1) = Q(K_1,K_2,1,1)$$

First Attack Idea

Superpolys with Some (not all) Variables #2

There exists a matching between off-line cube-sum and on-line cube-sum.

?

$Q(K_1,K_2,0,0)$
$Q(K_1,K_2,0,1)$
$Q(K_1,K_2,1,0)$
$Q(K_1,K_2,1,1)$

Q(0,0,0,0)
Q(0,0,0,1)
Q(0,0,1,0)
Q(0,0,1,1)

Q(0,1,0,0)
Q(0,1,0,1)
Q(0,1,1,0)
Q(0,1,1,1)

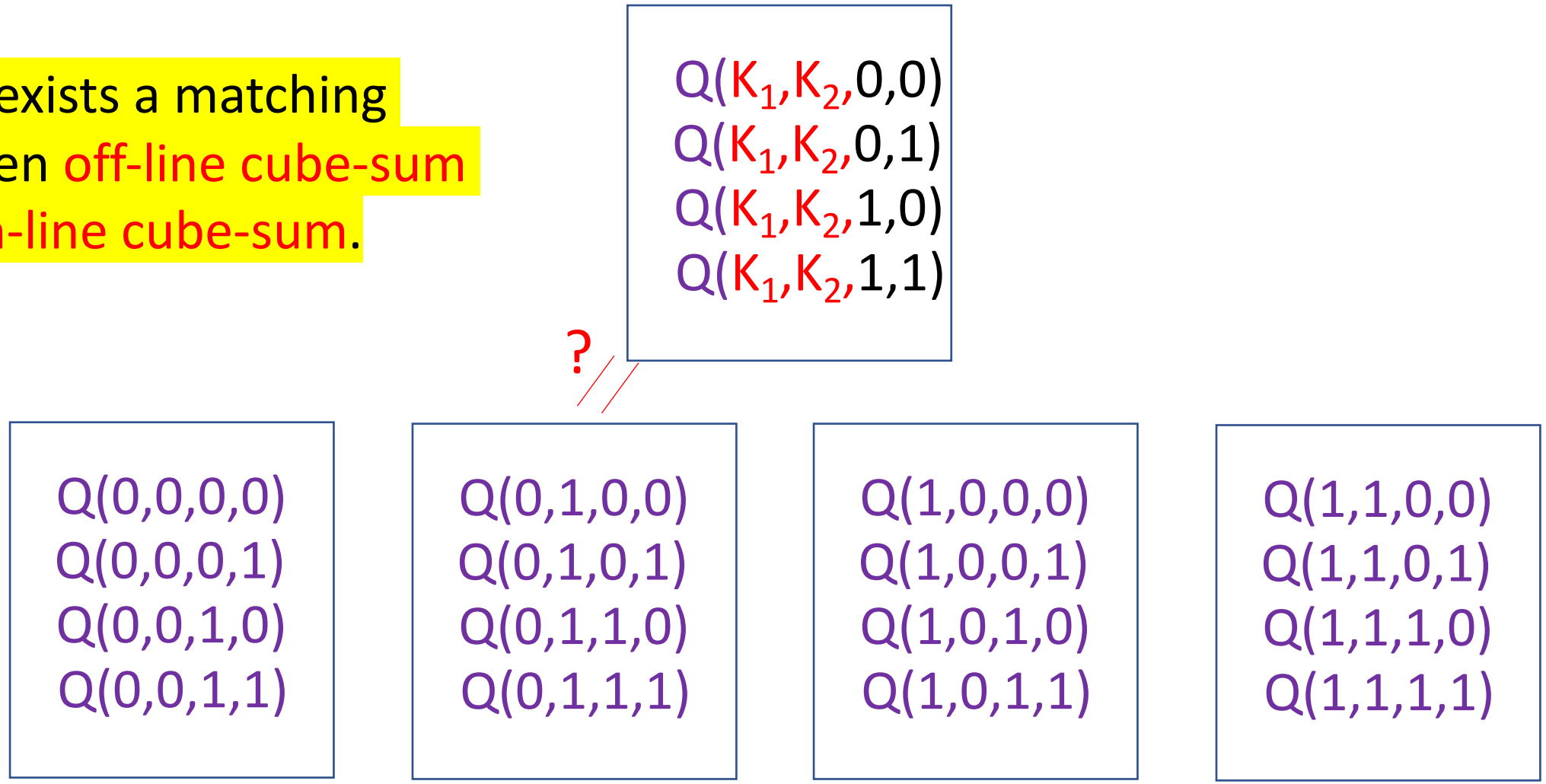
Q(1,0,0,0)
Q(1,0,0,1)
Q(1,0,1,0)
Q(1,0,1,1)

Q(1,1,0,0)
Q(1,1,0,1)
Q(1,1,1,0)
Q(1,1,1,1)

First Attack Idea

Superpolys with Some (not all) Variables #2

There exists a matching between off-line cube-sum and on-line cube-sum.



First Attack Idea

Superpolys with Some (not all) Variables #2

There exists a matching between off-line cube-sum and on-line cube-sum.

$Q(K_1,K_2,0,0)$
$Q(K_1,K_2,0,1)$
$Q(K_1,K_2,1,0)$
$Q(K_1,K_2,1,1)$

?

Q(0,0,0,0)
Q(0,0,0,1)
Q(0,0,1,0)
Q(0,0,1,1)

Q(0,1,0,0)
Q(0,1,0,1)
Q(0,1,1,0)
Q(0,1,1,1)

Q(1,0,0,0)
Q(1,0,0,1)
Q(1,0,1,0)
Q(1,0,1,1)

Q(1,1,0,0)
Q(1,1,0,1)
Q(1,1,1,0)
Q(1,1,1,1)

First Attack Idea

Superpolys with Some (not all) Variables #2

There exists a matching between off-line cube-sum and on-line cube-sum.

$Q(K_1,K_2,0,0)$
$Q(K_1,K_2,0,1)$
$Q(K_1,K_2,1,0)$
$Q(K_1,K_2,1,1)$

?

Q(0,0,0,0)	Q(0,1,0,0)	Q(1,0,0,0)	Q(1,1,0,0)
Q(0,0,0,1)	Q(0,1,0,1)	Q(1,0,0,1)	Q(1,1,0,1)
Q(0,0,1,0)	Q(0,1,1,0)	Q(1,0,1,0)	Q(1,1,1,0)
Q(0,0,1,1)	Q(0,1,1,1)	Q(1,0,1,1)	Q(1,1,1,1)

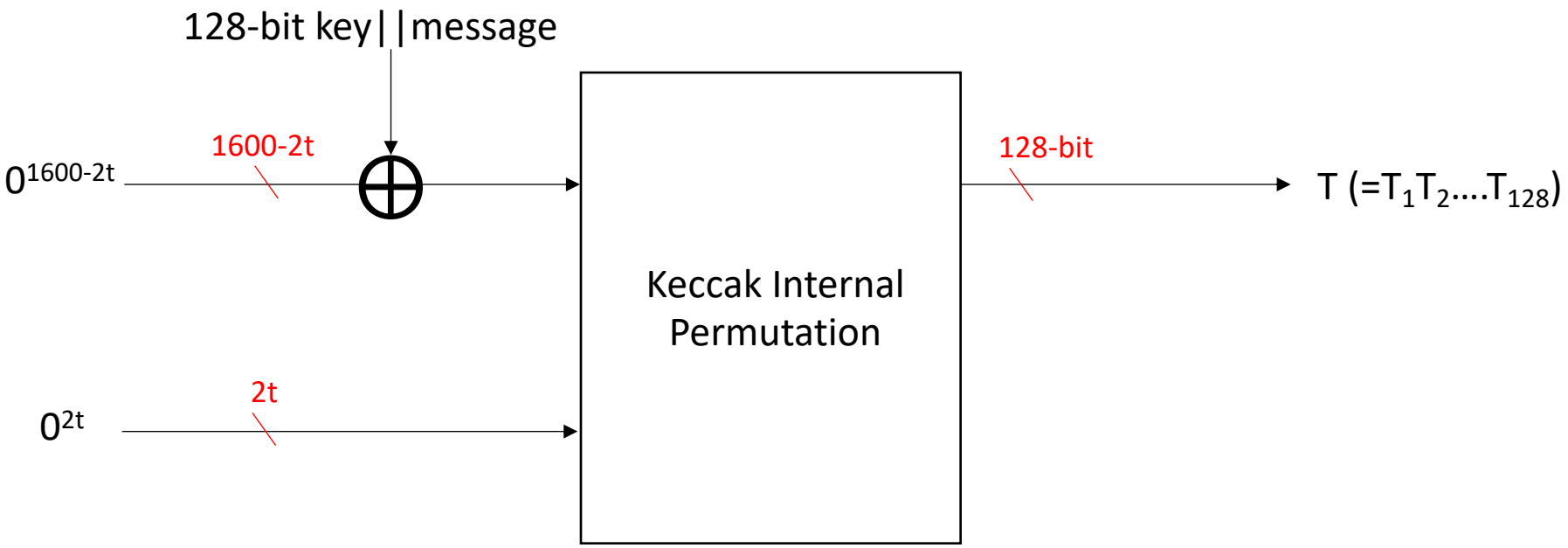
First Attack Idea

Superpolys with Some (not all) Variables #2

- Once a matching happens, we can find K_1,K_2.
- Then, we exhaustively search the remaining two bits, K_3,K_4. (Divide-and-Conquer Attack)

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



Construction of Keccak-MAC-t

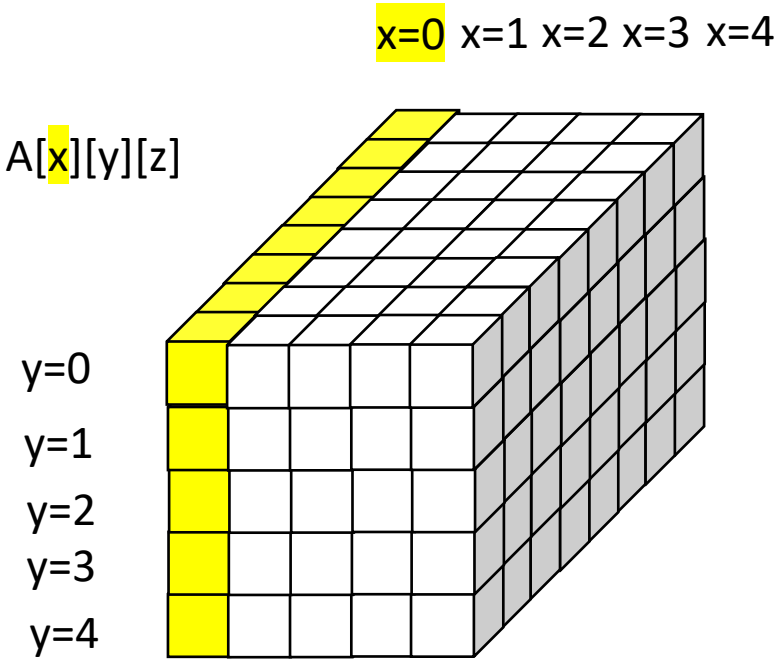
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

- The secret key K is 128-bit.
- Let $K=K_0||K_1$, where $|K_0|=|K_1|=64$.
- We need to search a cube with some cube variables whose superpolys depends only K_0.
- How can we find such cube?

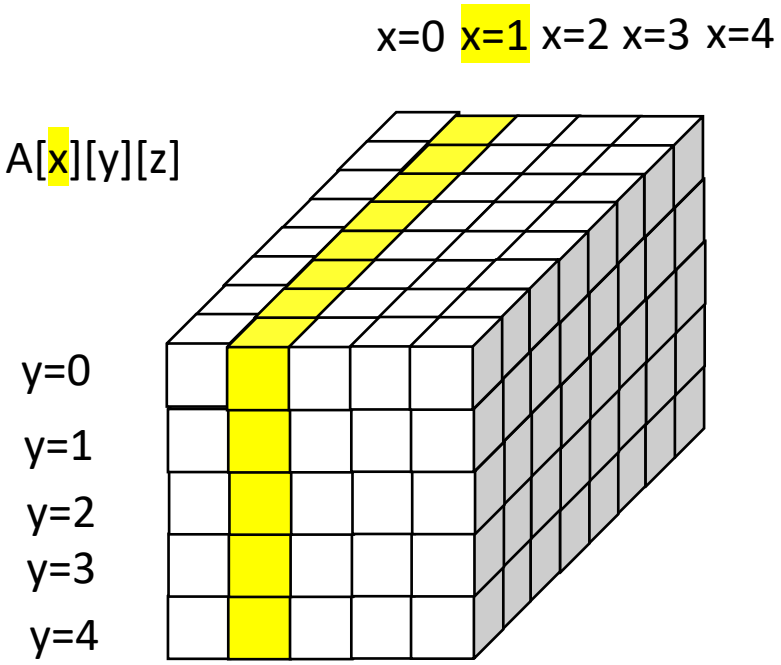
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



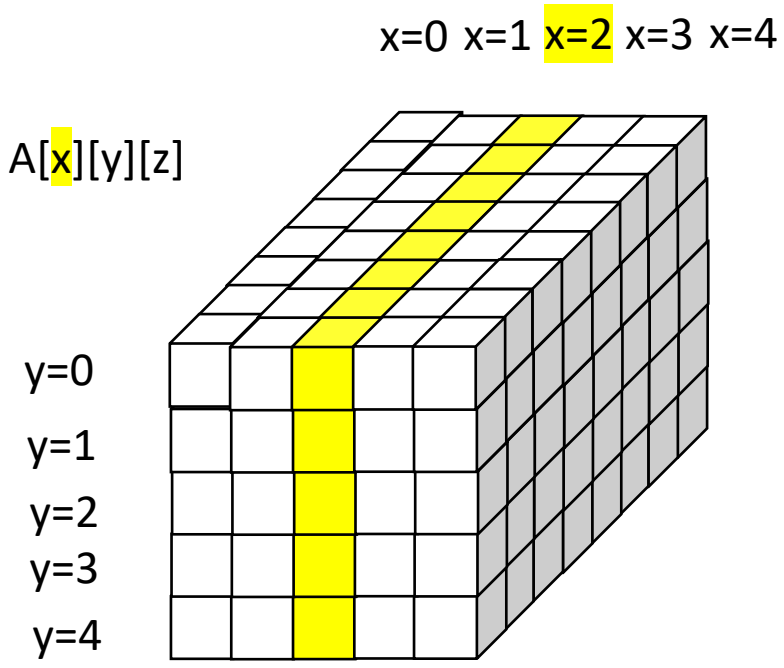
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



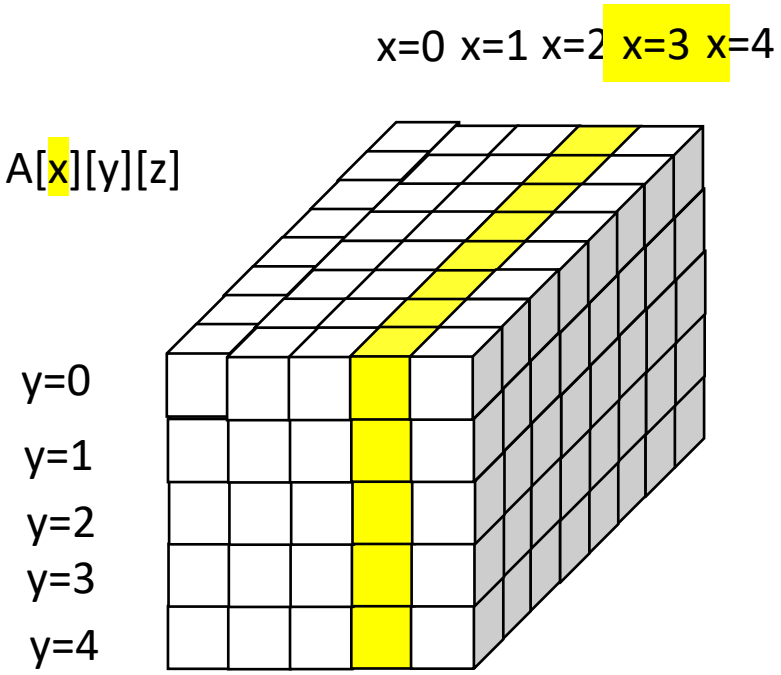
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



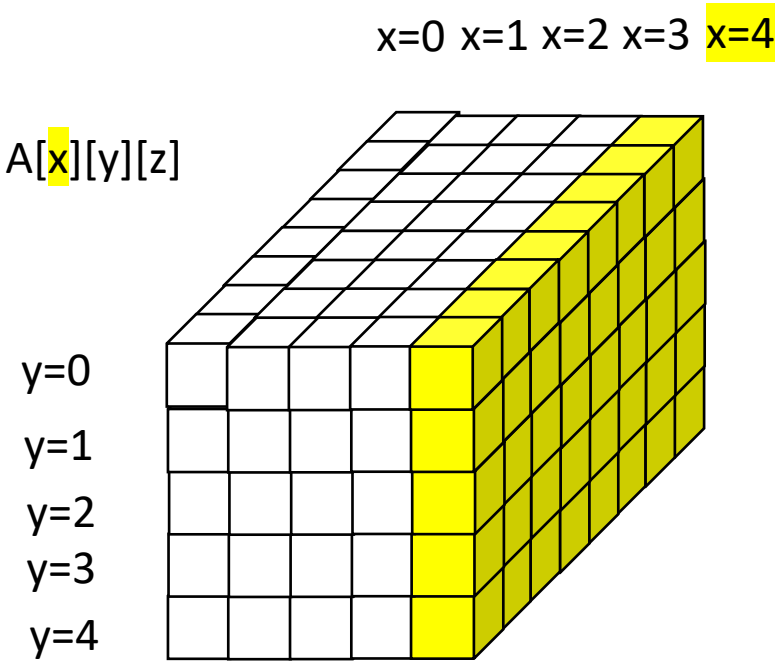
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



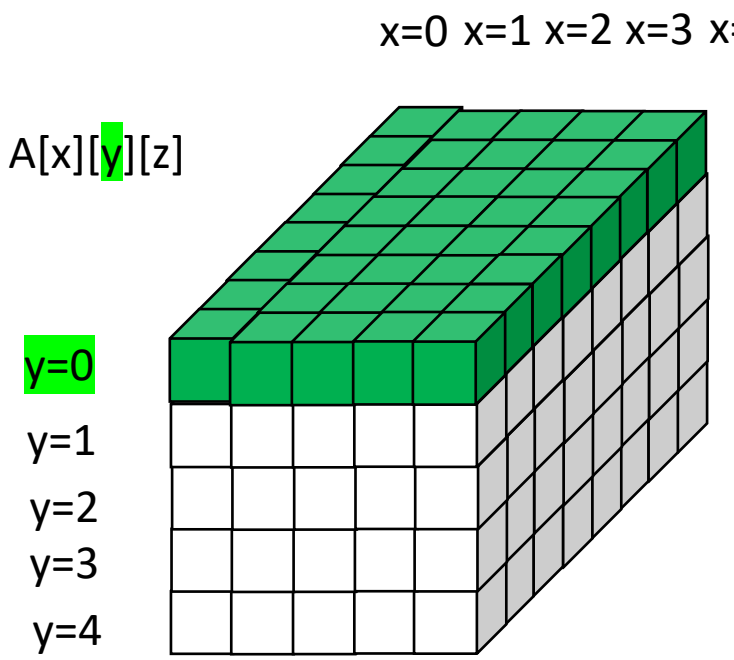
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



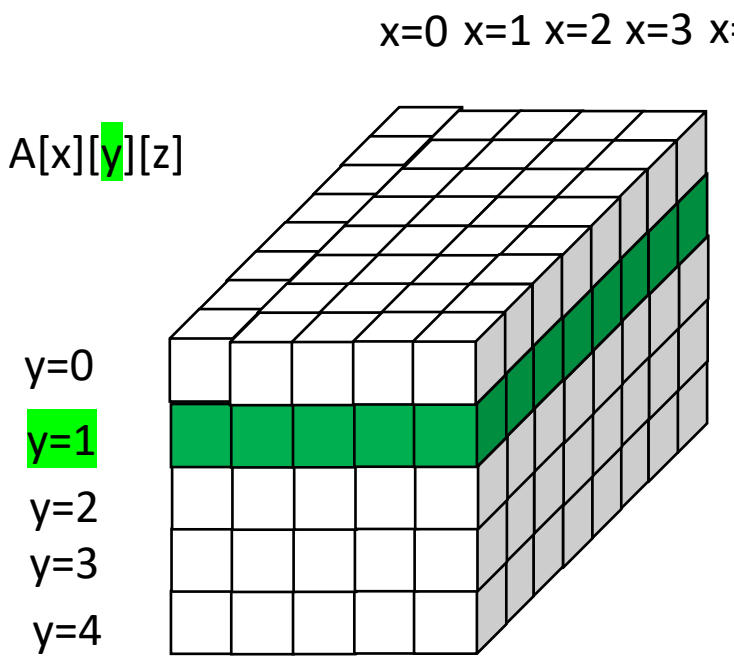
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



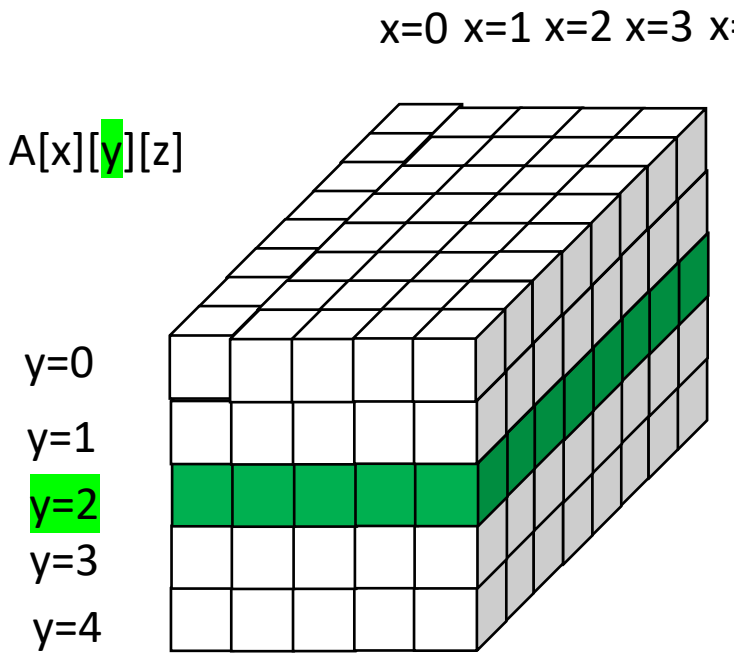
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



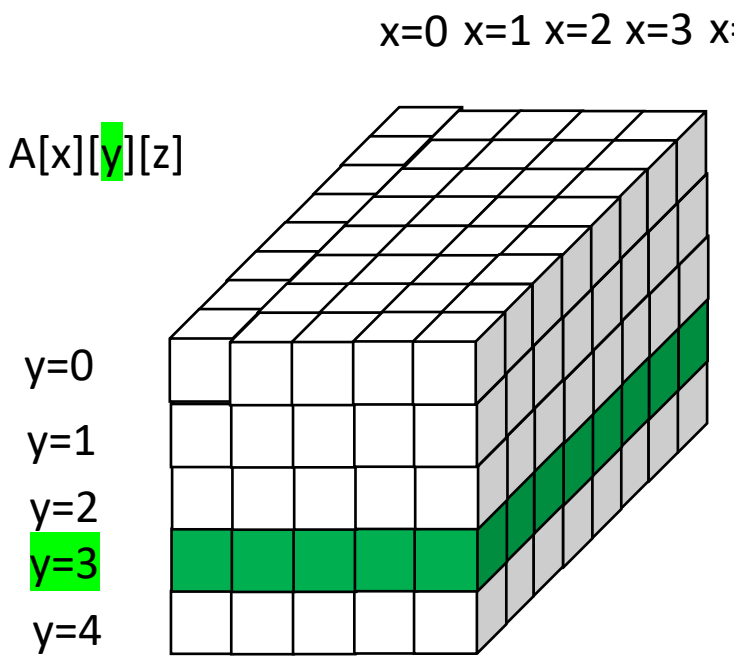
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



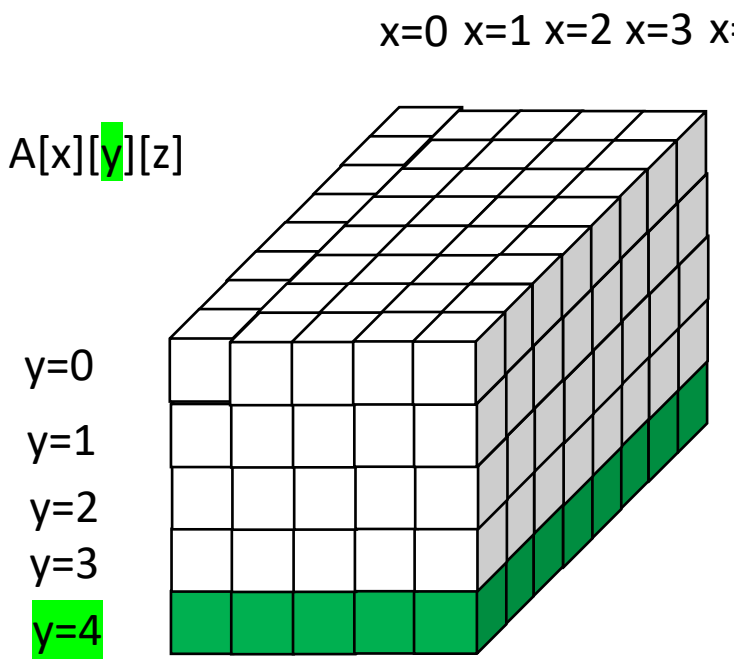
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



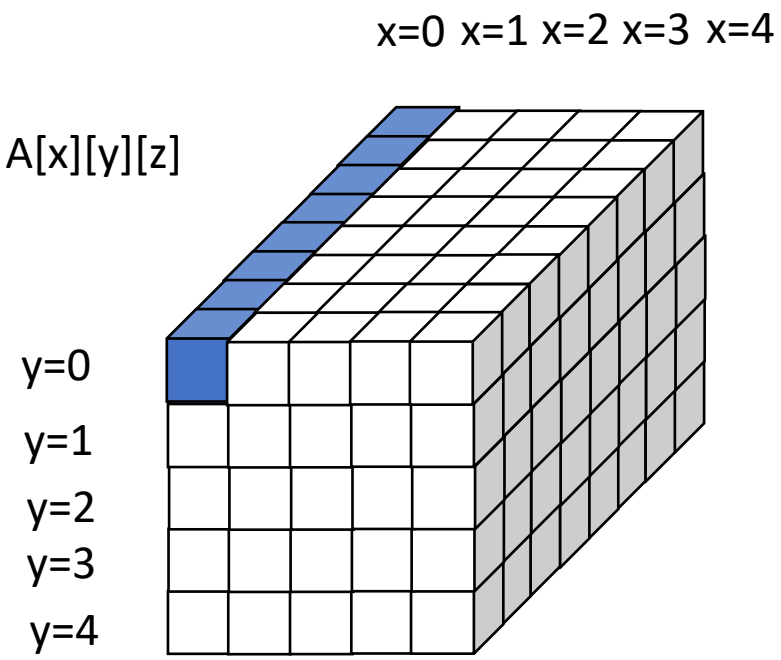
First Attack Idea

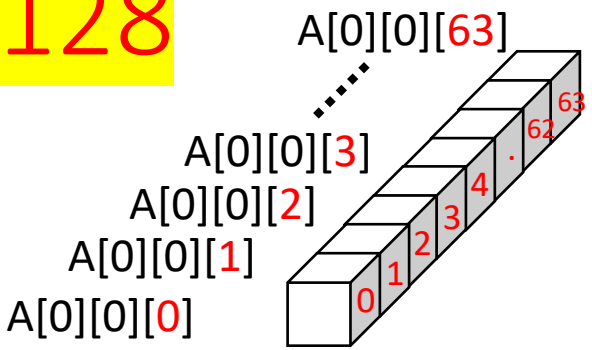
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



First Attack Idea

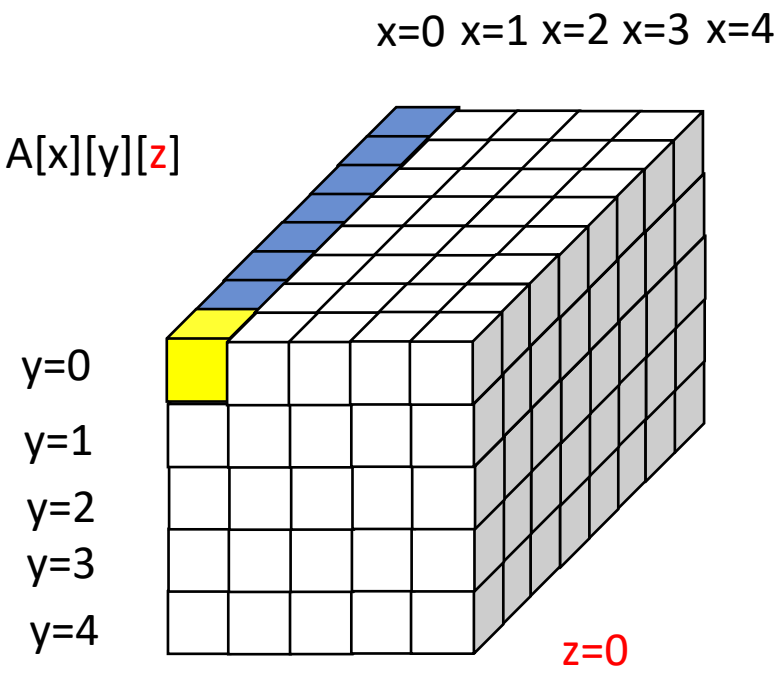
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

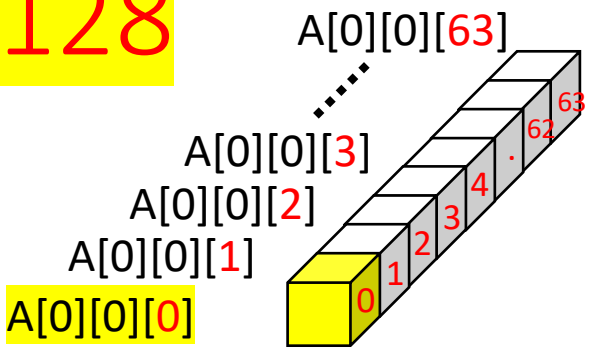




First Attack Idea

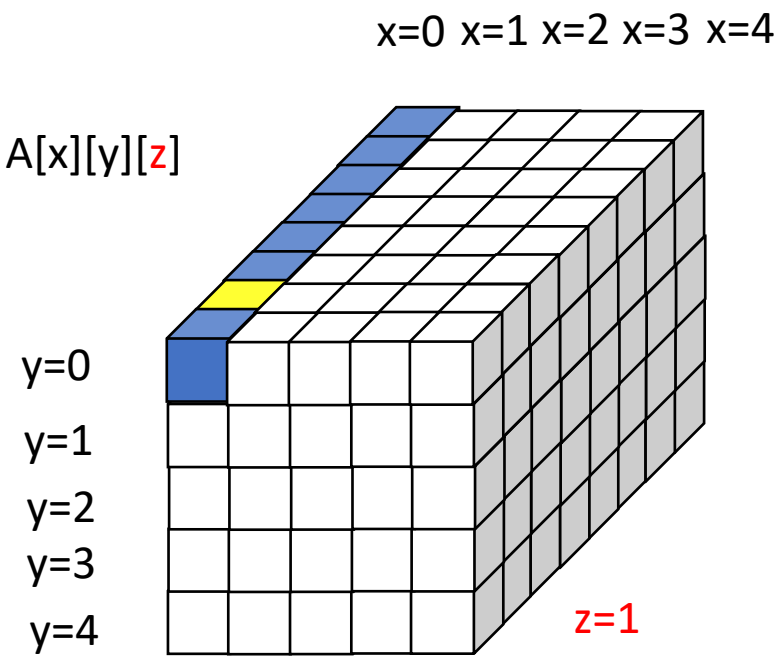
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

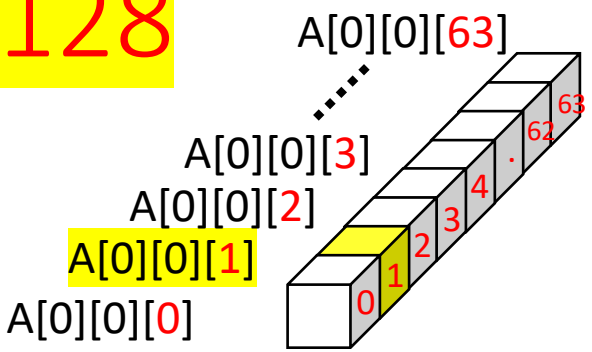




First Attack Idea

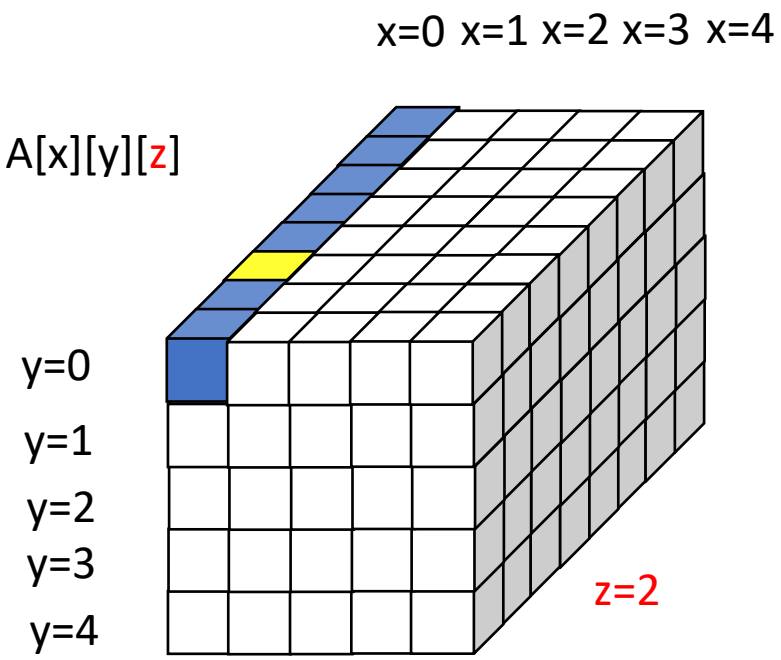
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

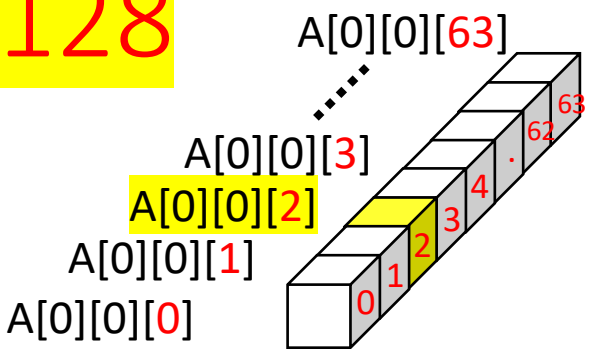




First Attack Idea

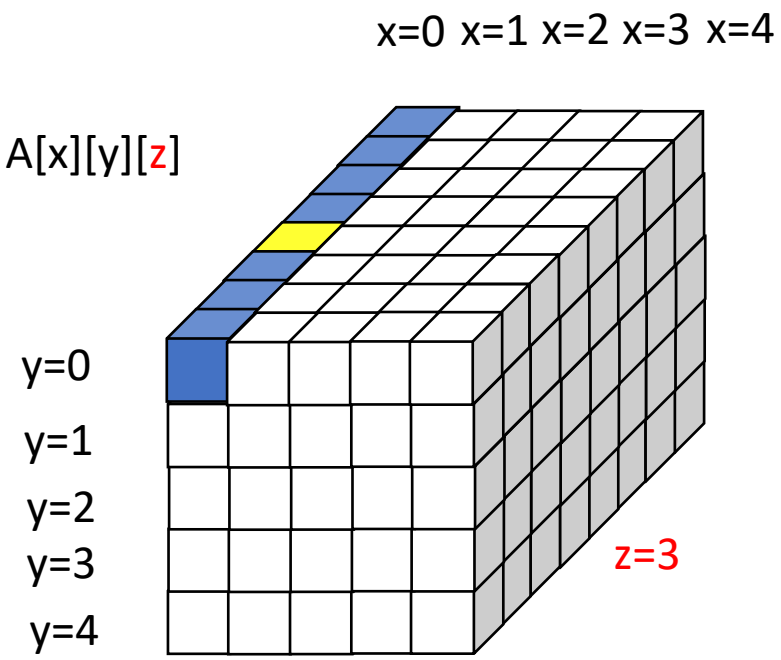
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

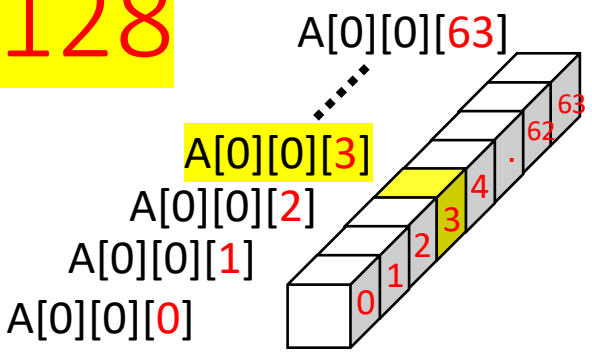




First Attack Idea

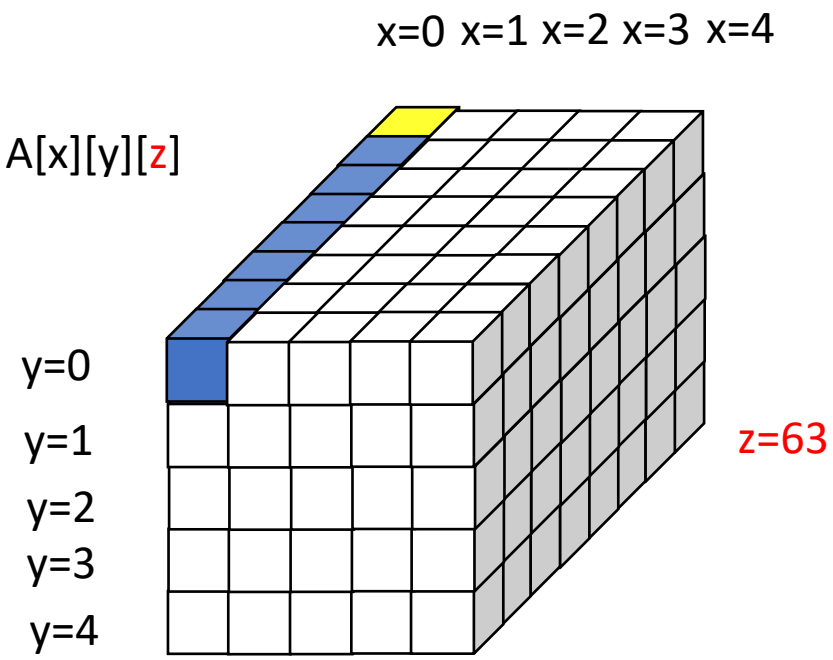
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

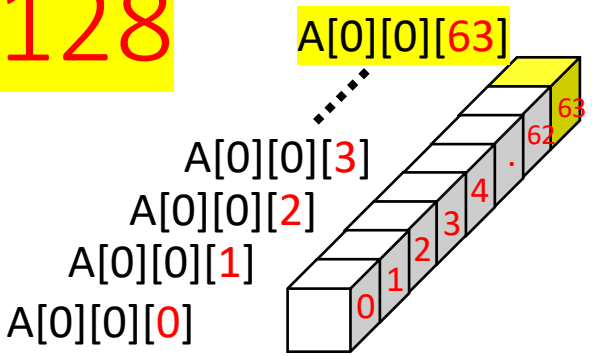




First Attack Idea

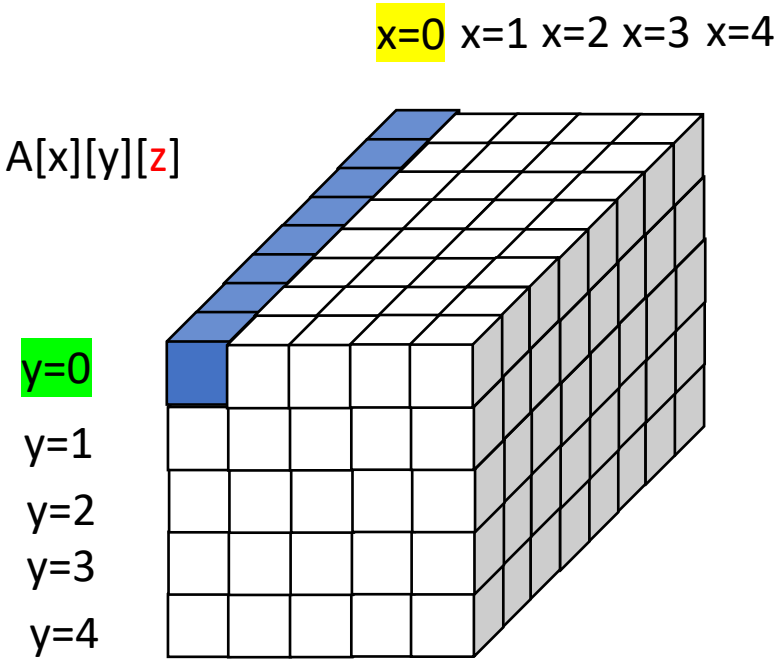
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

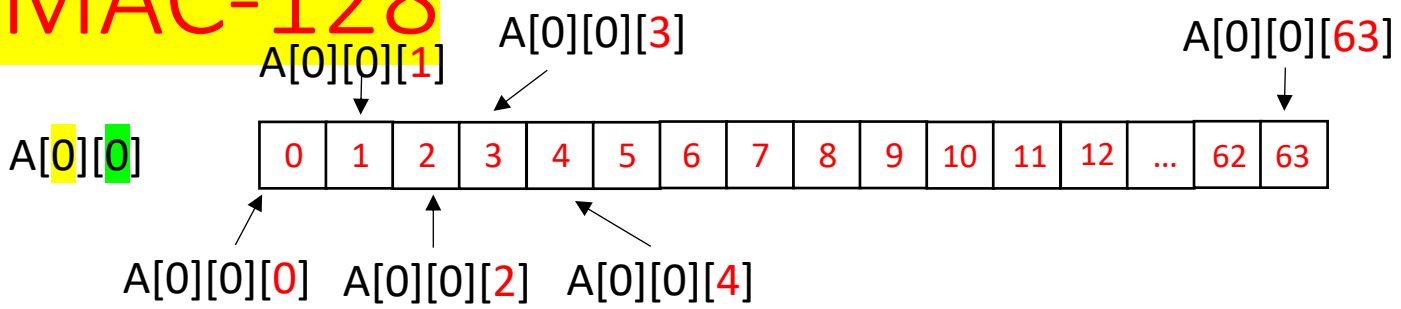




First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



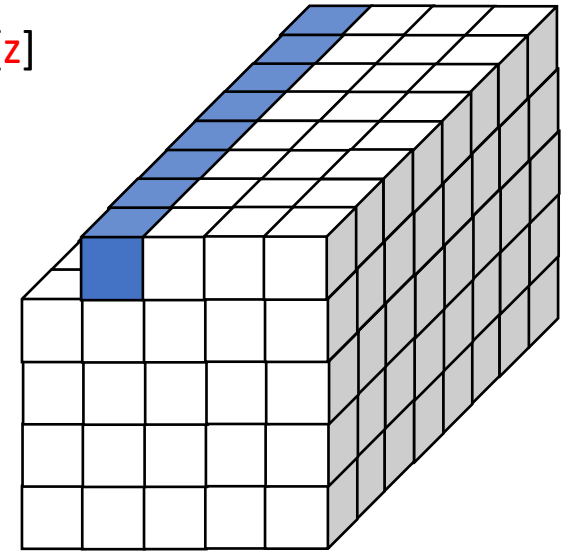


First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

A[x][y][z]



y=0

y=1

y=2

y=3

y=4

A[0][0]

A[1][0]

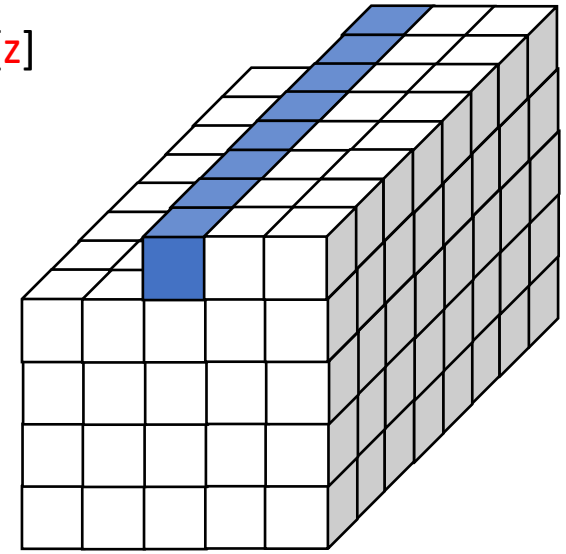
0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

A[x][y][z]



y=0

y=1

y=2

y=3

y=4

A[0][0]

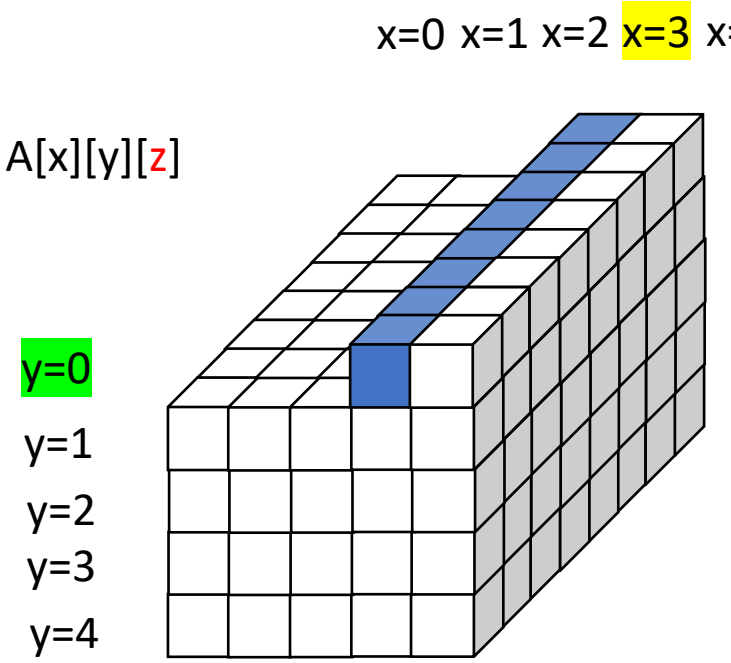
A[1][0]

A[2][0]

0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63

First Attack Idea

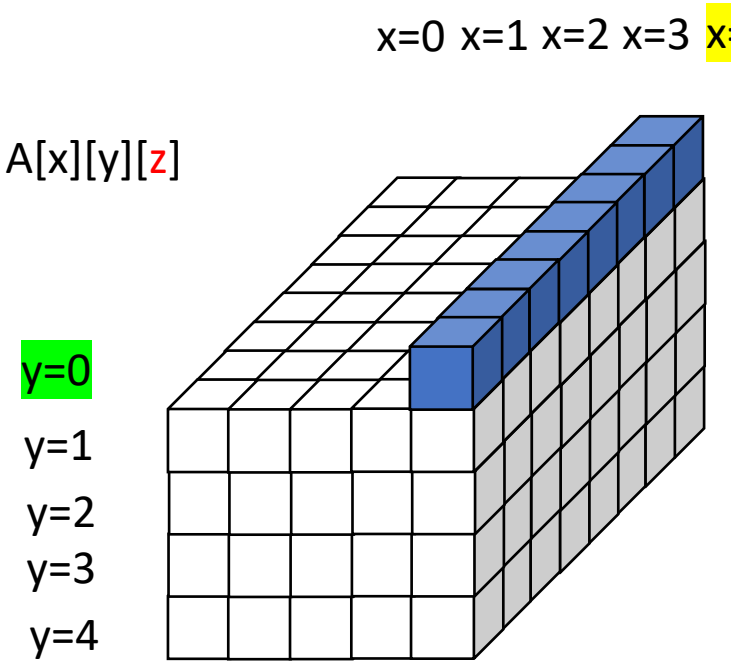
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



A[0][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63

First Attack Idea

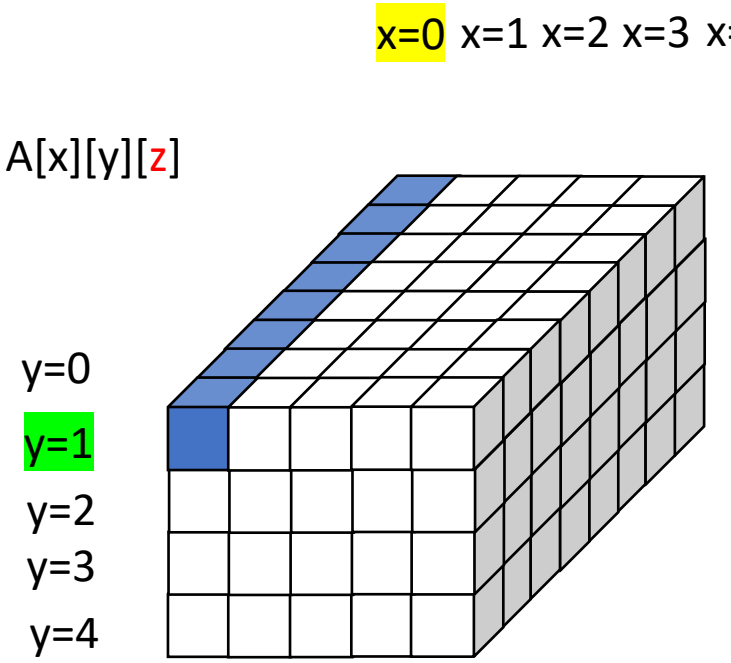
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



A[0][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63

First Attack Idea

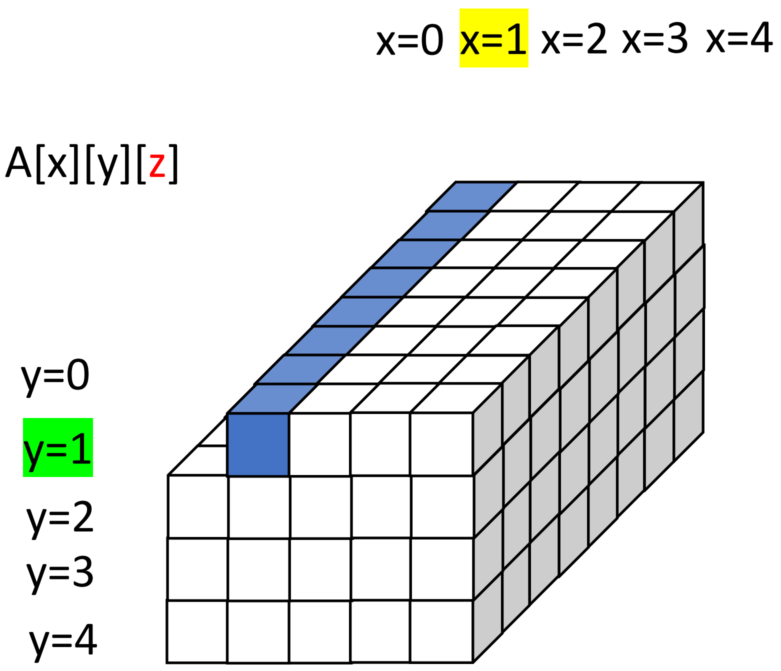
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



A[0][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[0][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63

First Attack Idea

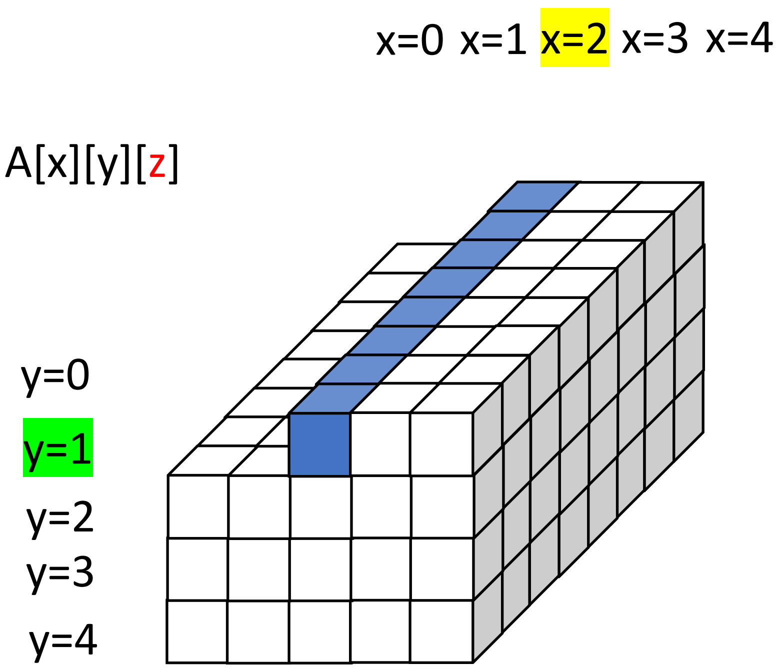
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



A[0][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[0][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63

First Attack Idea

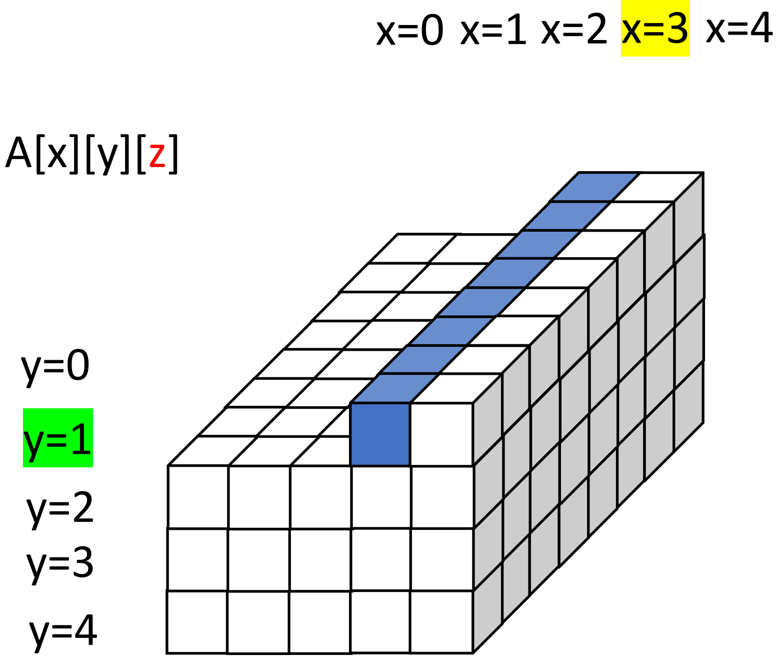
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



A[0][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[1][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[2][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[3][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[4][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[0][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[1][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[2][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



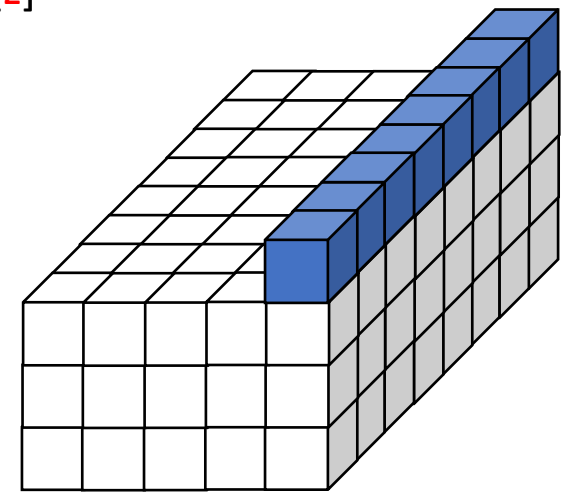
A[0][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[1][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[2][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[3][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[4][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[0][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[1][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[2][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[3][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

A[x][y][z]



A[0][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[0][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63

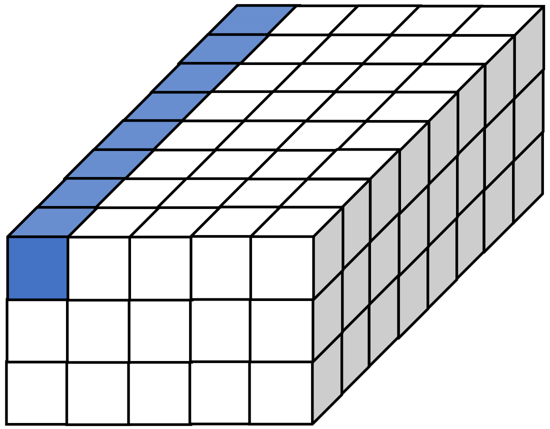
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

A[x][y][z]

y=0
y=1
y=2
y=3
y=4



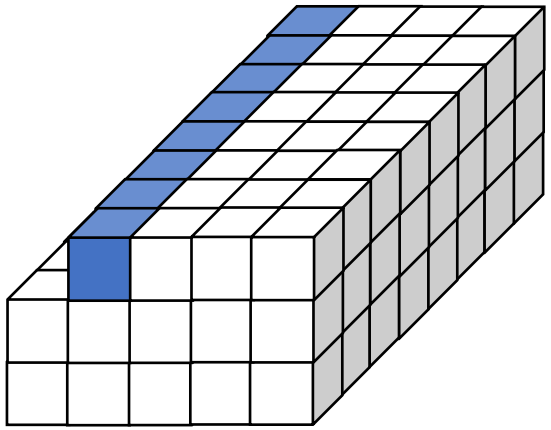
A[0][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[0][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[0][2]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

A[x][y][z]



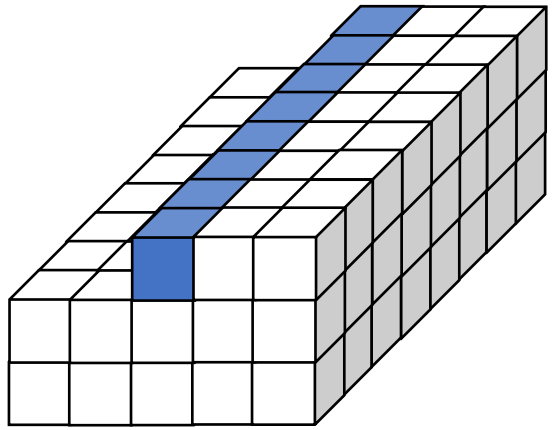
A[0][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[0][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[0][2]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][2]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

A[x][y][z]



A[0][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[0][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[0][2]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][2]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][2]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

A[x][y][z]

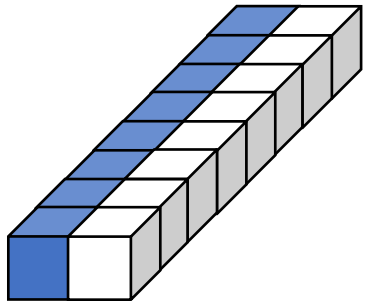
y=0

y=1

y=2

y=3

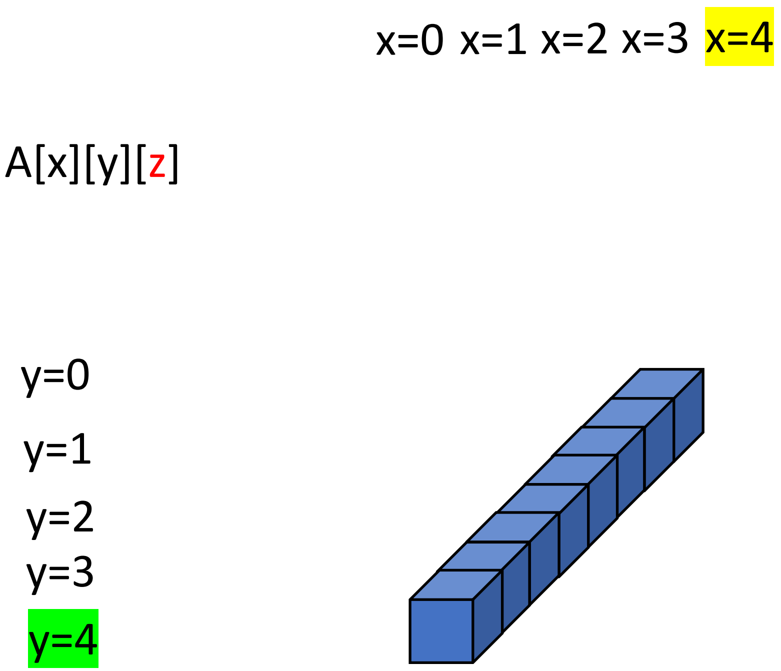
y=4



A[0][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[0][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[3][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[4][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[0][2]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[1][2]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
A[2][2]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63
⋮																
A[3][4]	0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63

First Attack Idea

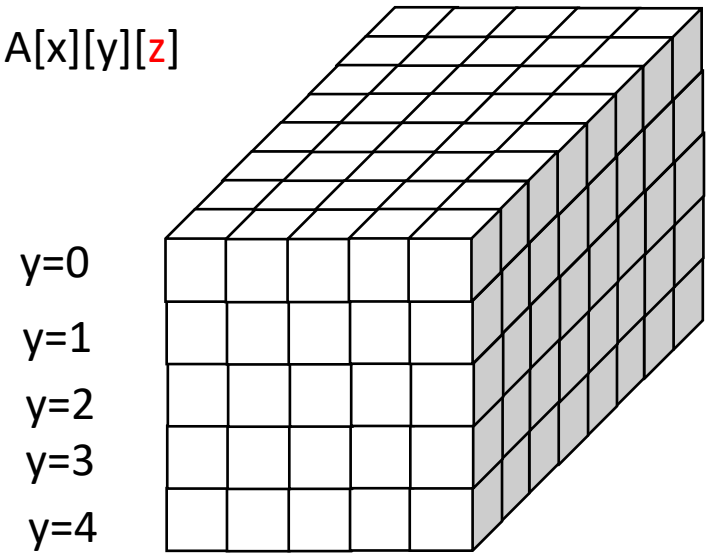
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



A[0][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[1][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[2][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[3][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[4][0]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[0][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[1][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[2][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[3][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[4][1]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[0][2]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[1][2]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[2][2]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
⋮																
A[3][4]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63
A[4][4]	0	1	2	3	4	5	6	7	8	9	10	11	12	...	62	63

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



A (1600-bit)

Round *i*

0.5 round

$\boldsymbol{\theta}$: $A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\big(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\big)$.

$\boldsymbol{\rho}$: $A[x][y] \leftarrow A[x][y] \ggg r[x][y]$.

$\boldsymbol{\pi}$: $A[y][2x+3y] \leftarrow A[x][y]$.

0.5 round

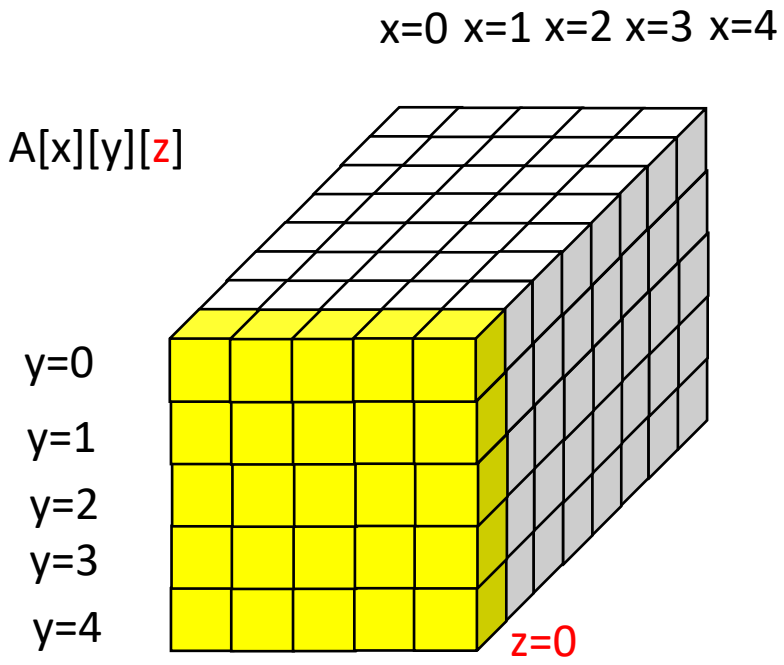
$\boldsymbol{\chi}$: $A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$

$\boldsymbol{\iota}$: $A[0][0] \leftarrow A[0][0] \oplus RC[i]$.

A (1600-bit)

First Attack Idea

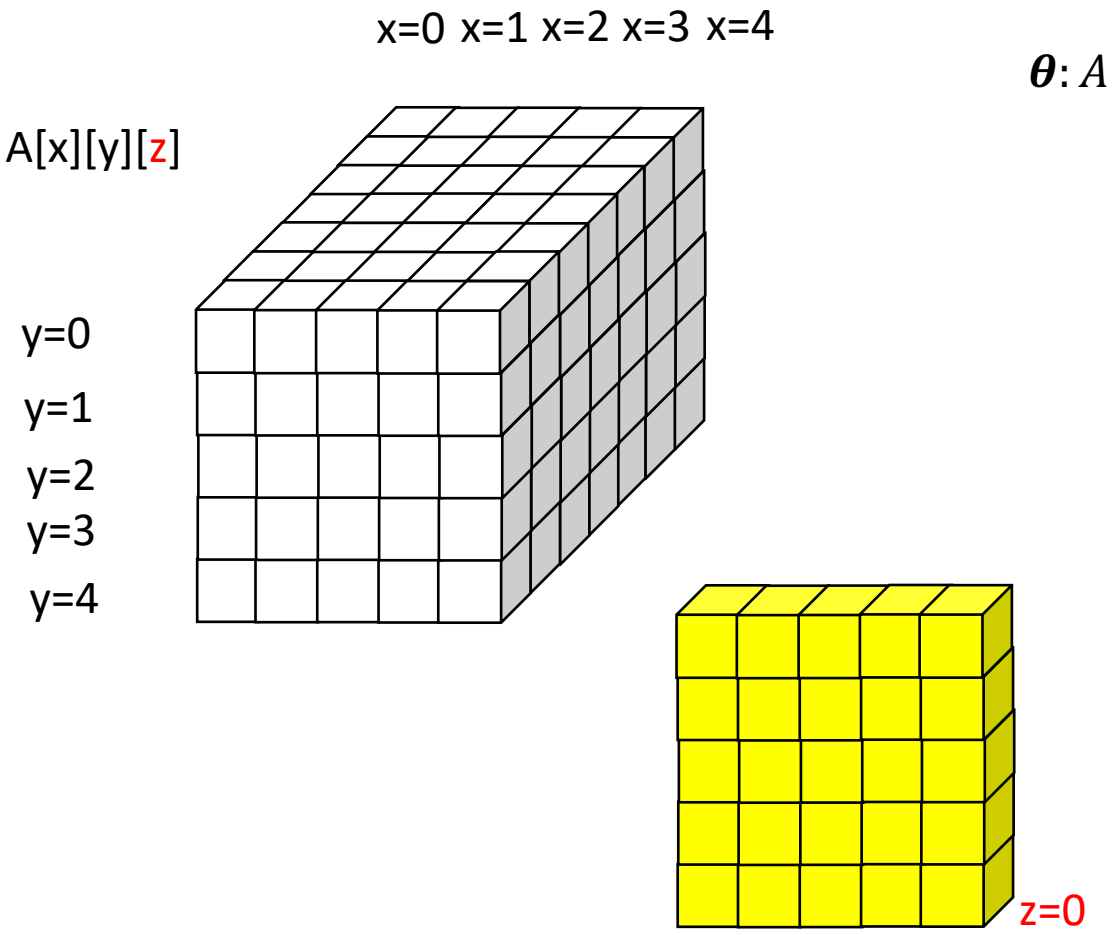
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



$$\boldsymbol{\theta}\colon A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\left(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\right).$$

First Attack Idea

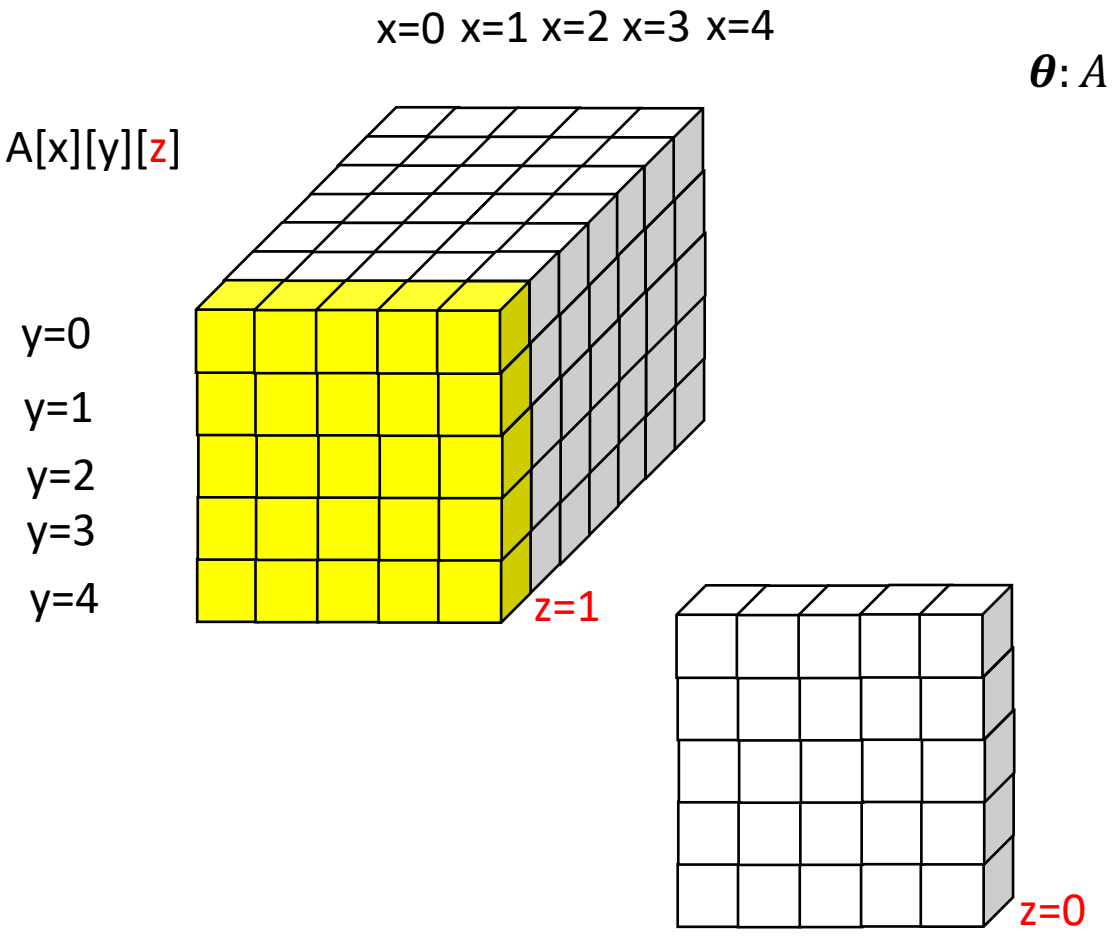
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



$$\boldsymbol{\theta}\colon A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\left(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\right).$$

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



$$\boldsymbol{\theta}\colon A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\left(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\right).$$

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

A[x][y][z]

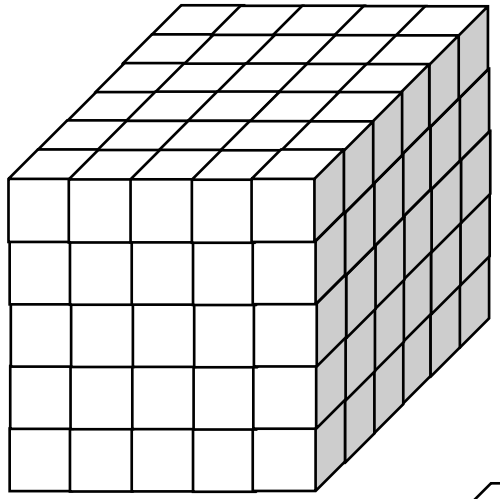
y=0

y=1

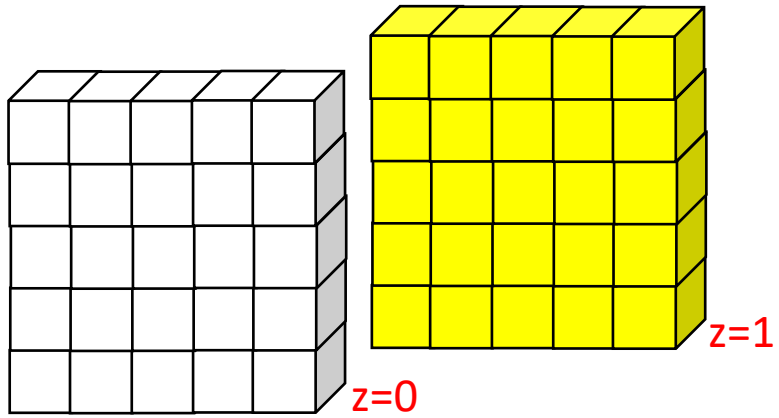
y=2

y=3

y=4



$\boldsymbol{\theta}$: $A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\left(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\right)$.

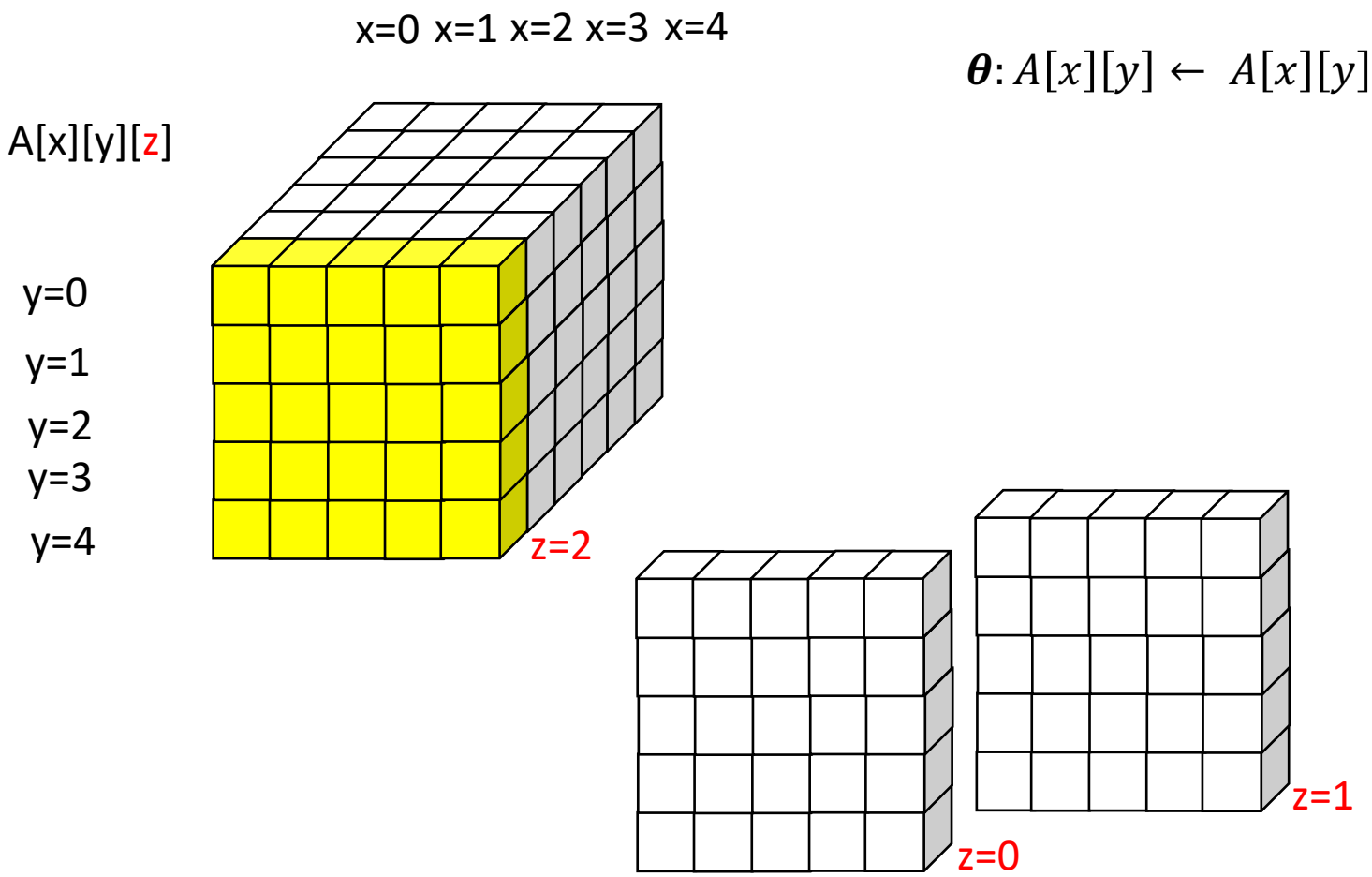


z=1

z=0

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



$$\boldsymbol{\theta}\colon A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\left(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\right).$$

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

A[x][y][z]

y=0

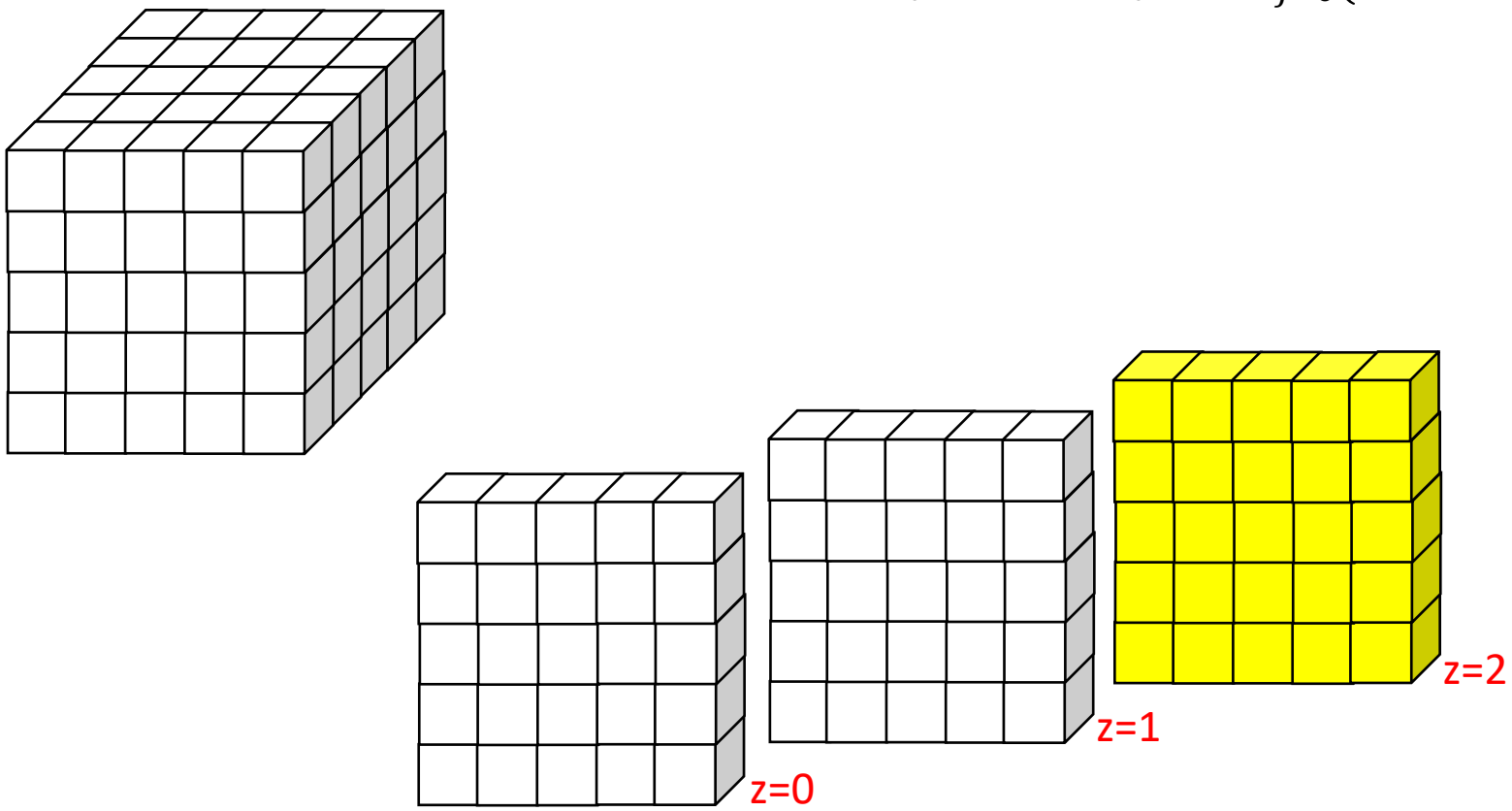
y=1

y=2

y=3

y=4

$\boldsymbol{\theta}$: $A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\left(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\right)$.



First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

A[x][y][z]

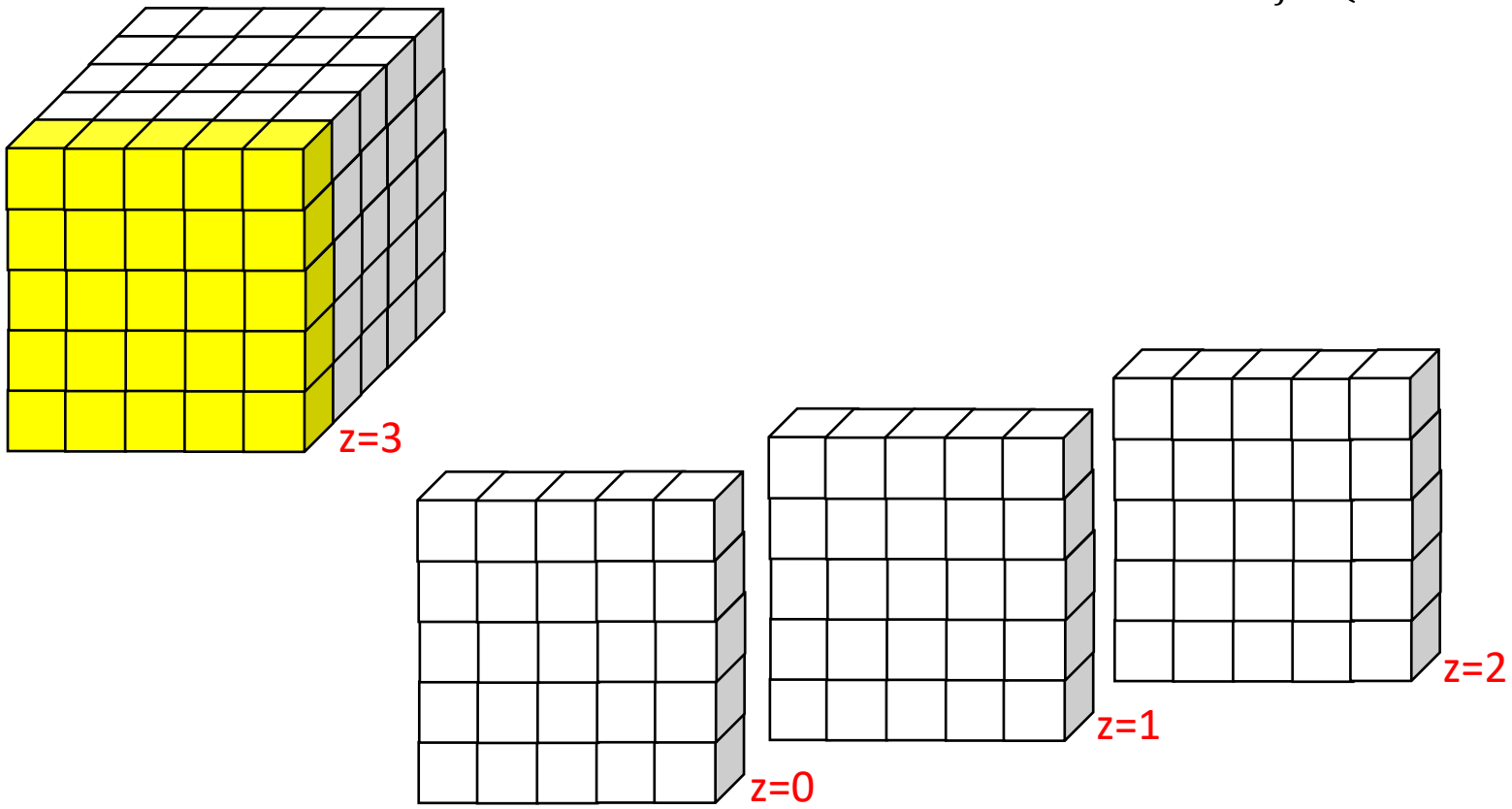
y=0

y=1

y=2

y=3

y=4



$\boldsymbol{\theta}$: $A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\big(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\big)$.

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

$\boldsymbol{\theta}$: $A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\left(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\right)$.

A[x][y][z]

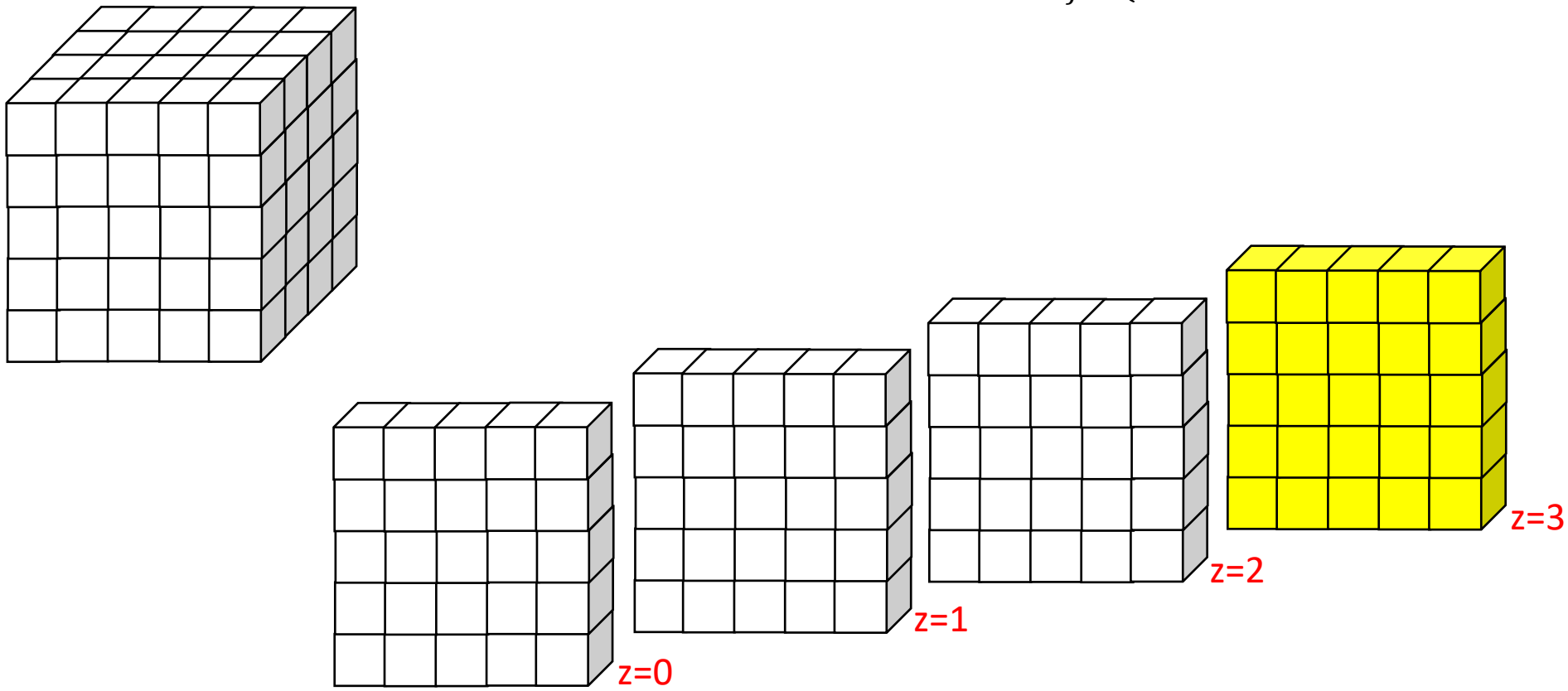
y=0

y=1

y=2

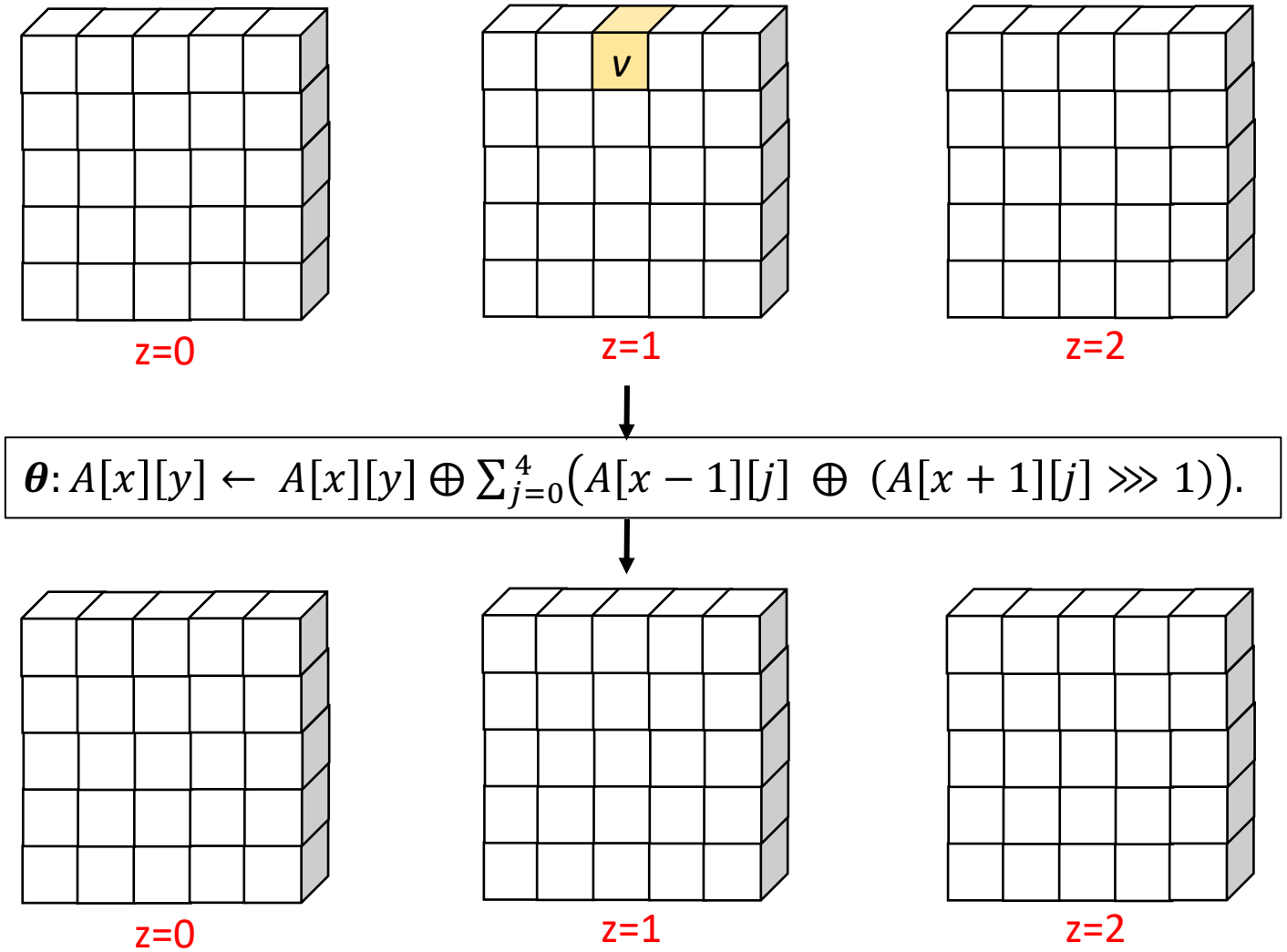
y=3

y=4



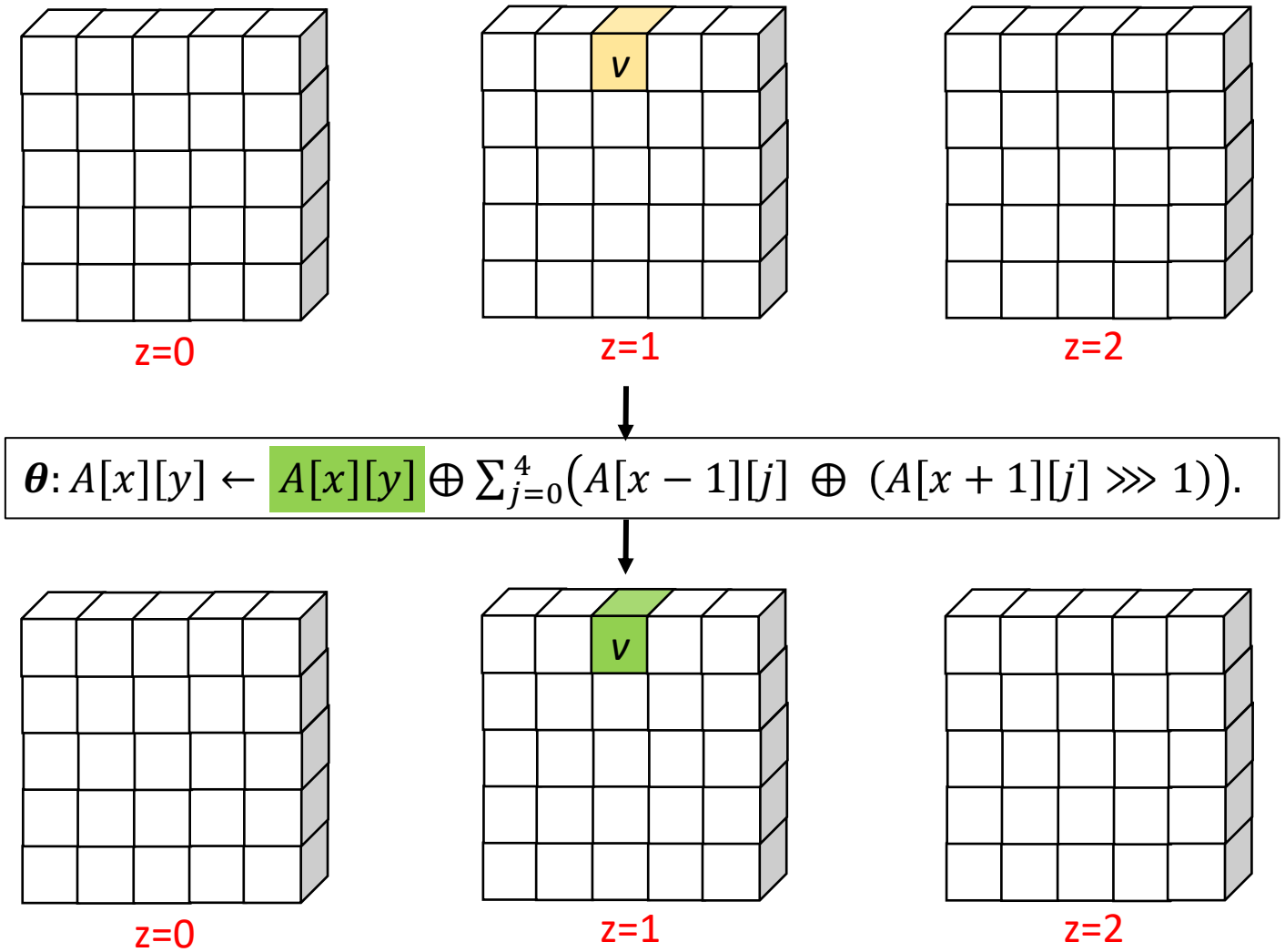
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



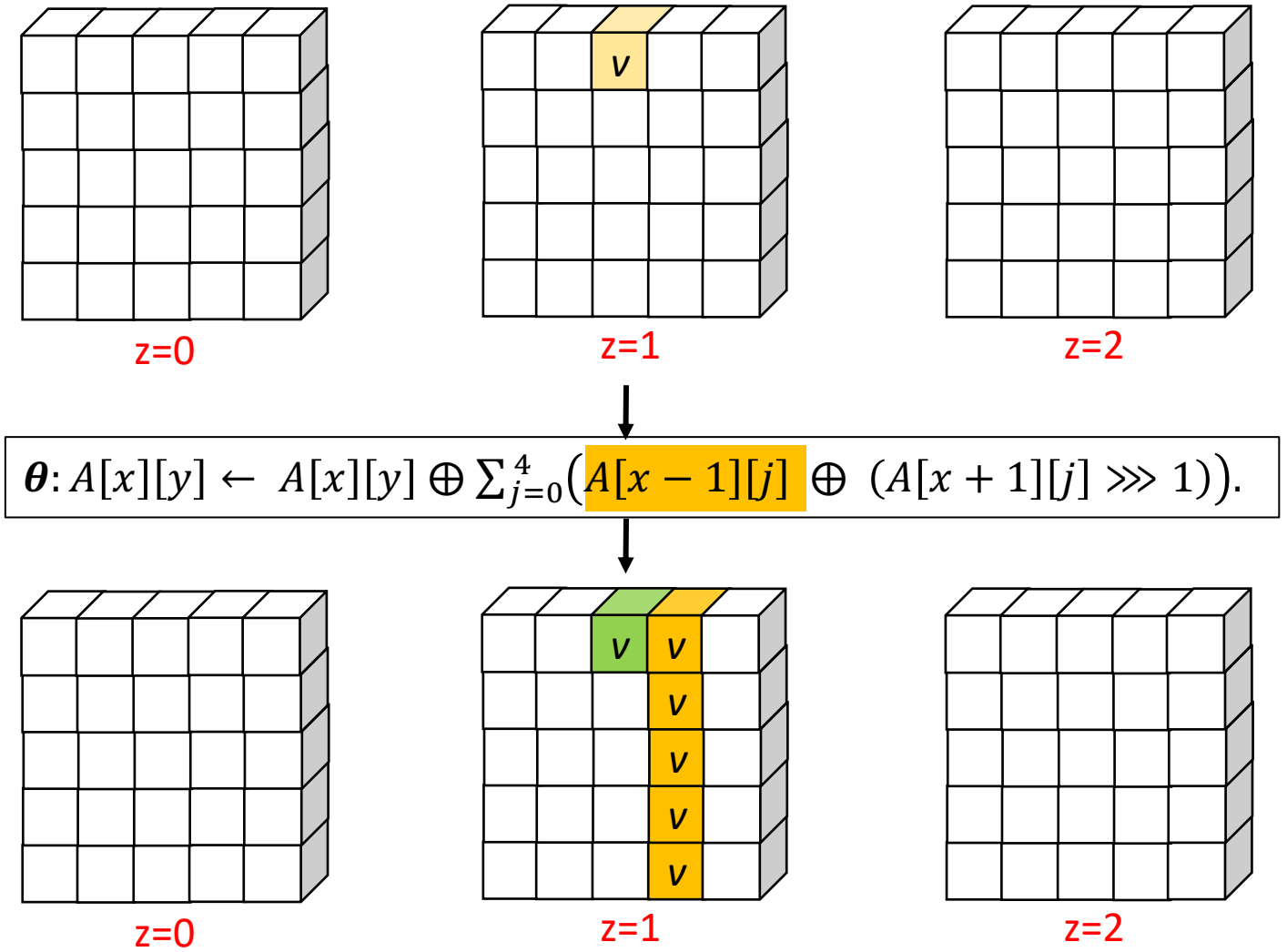
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



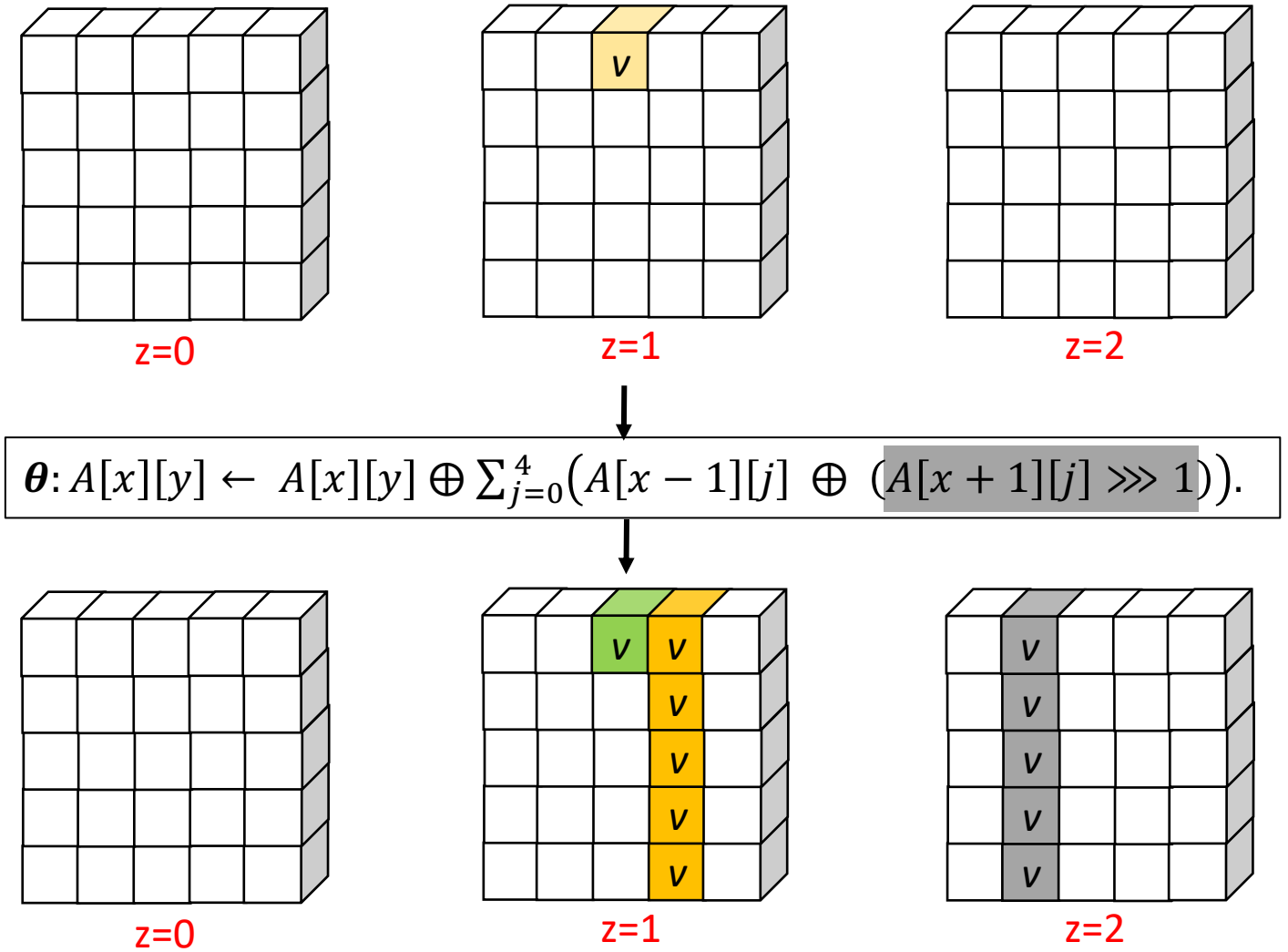
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



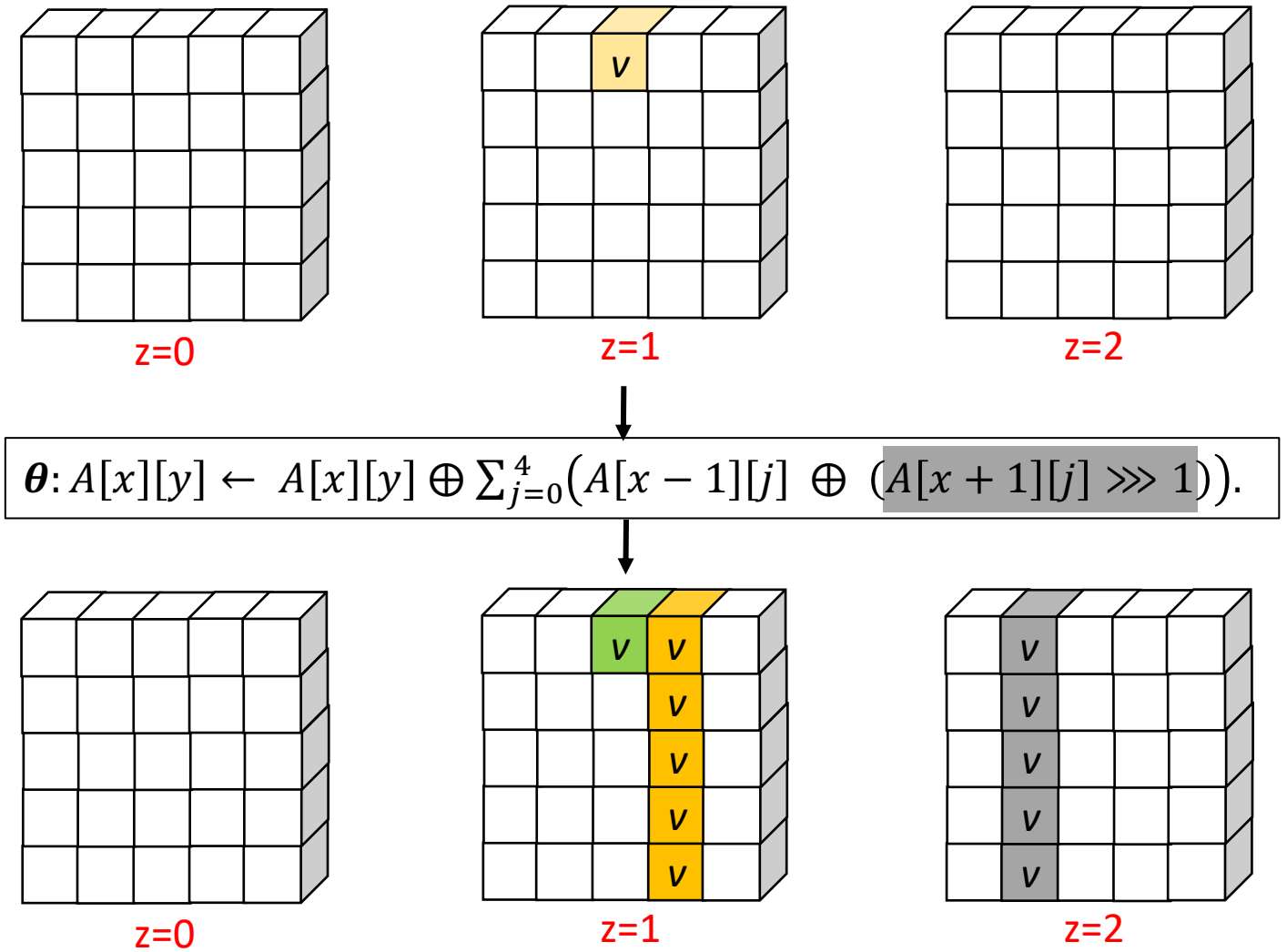
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



First Attack Idea

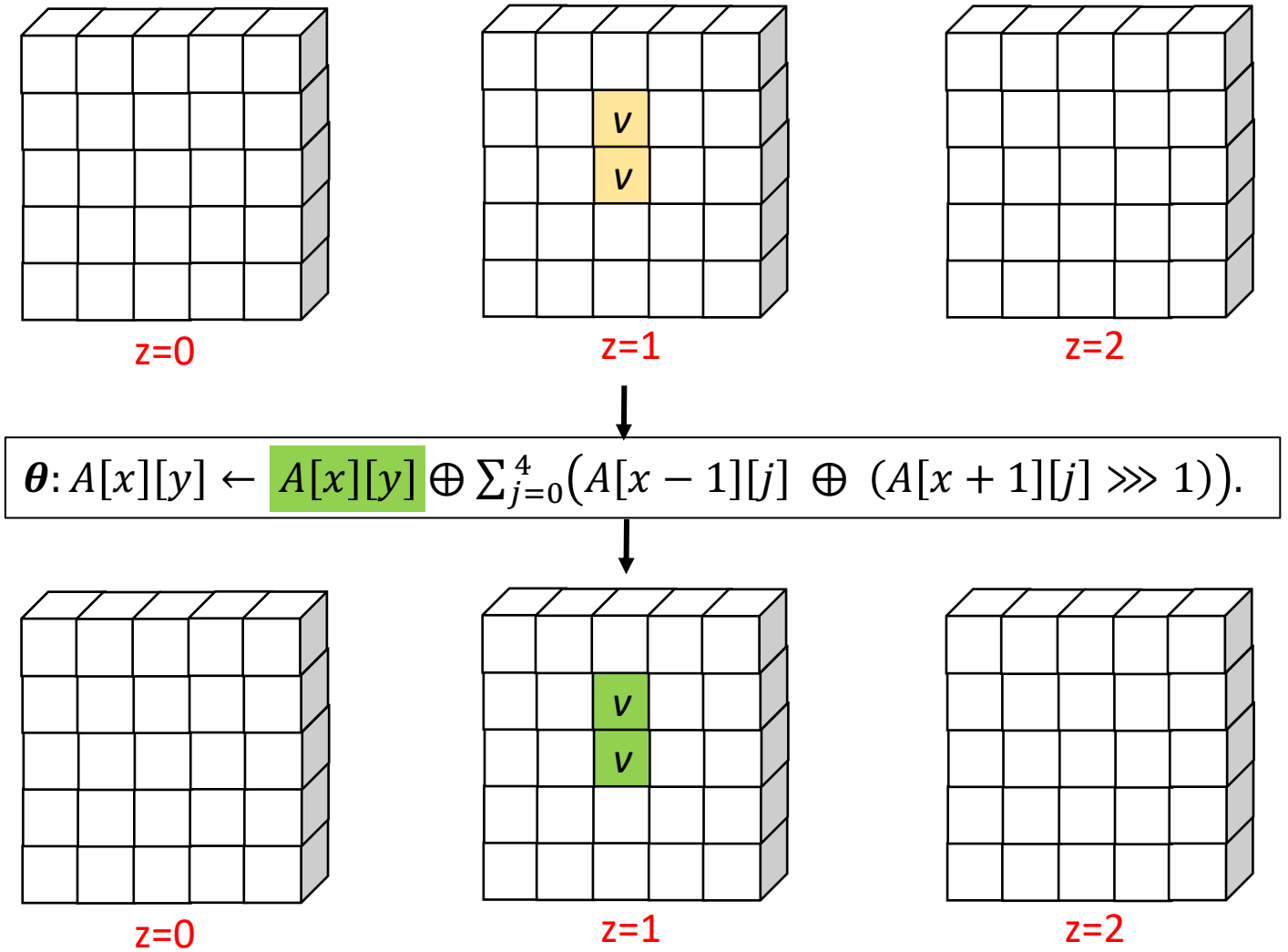
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



Through $\boldsymbol{\theta}$, a variable v influences 11-bit positions.

First Attack Idea

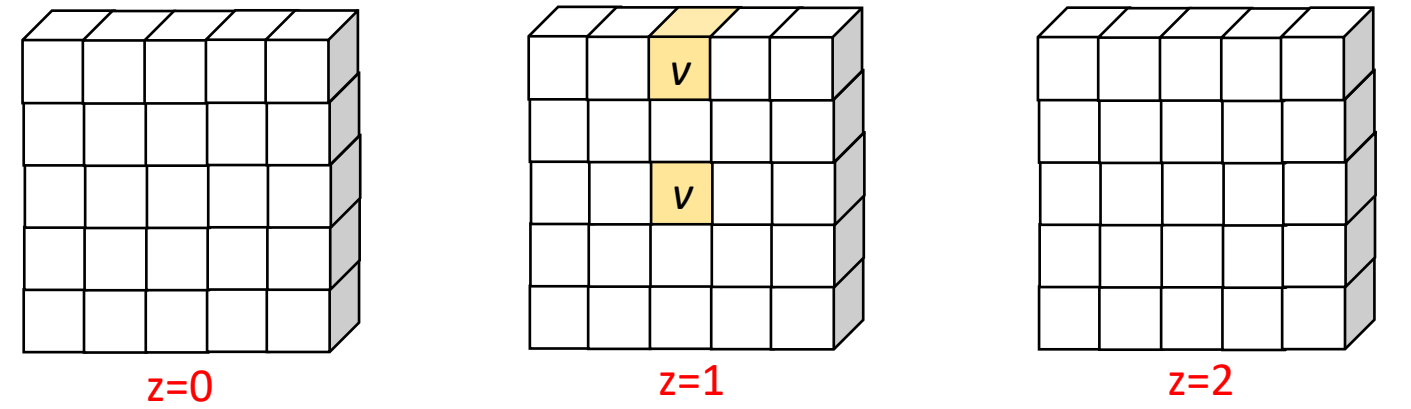
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



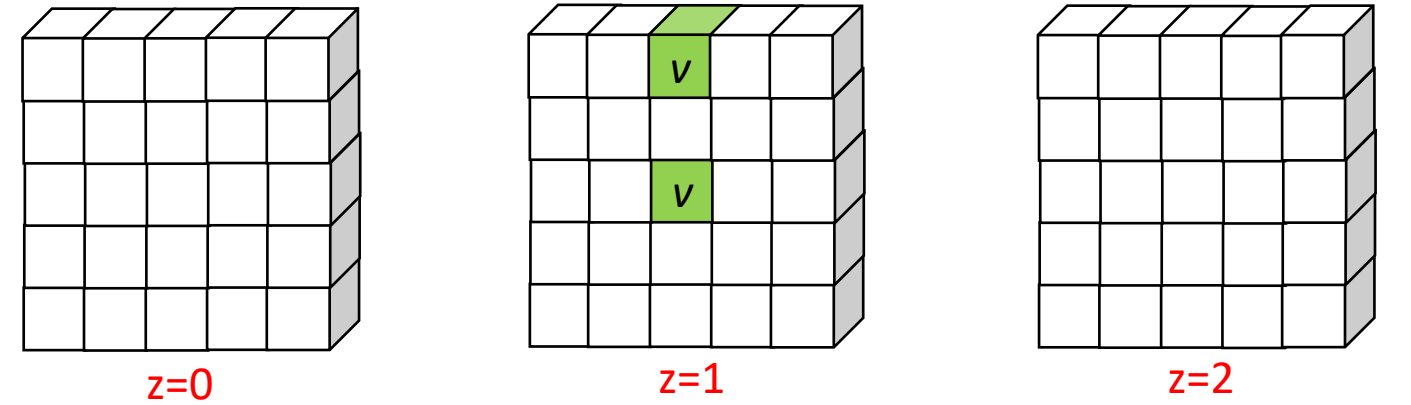
Through $\boldsymbol{\theta}$, a variable v influences Only 2-bit positions.

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

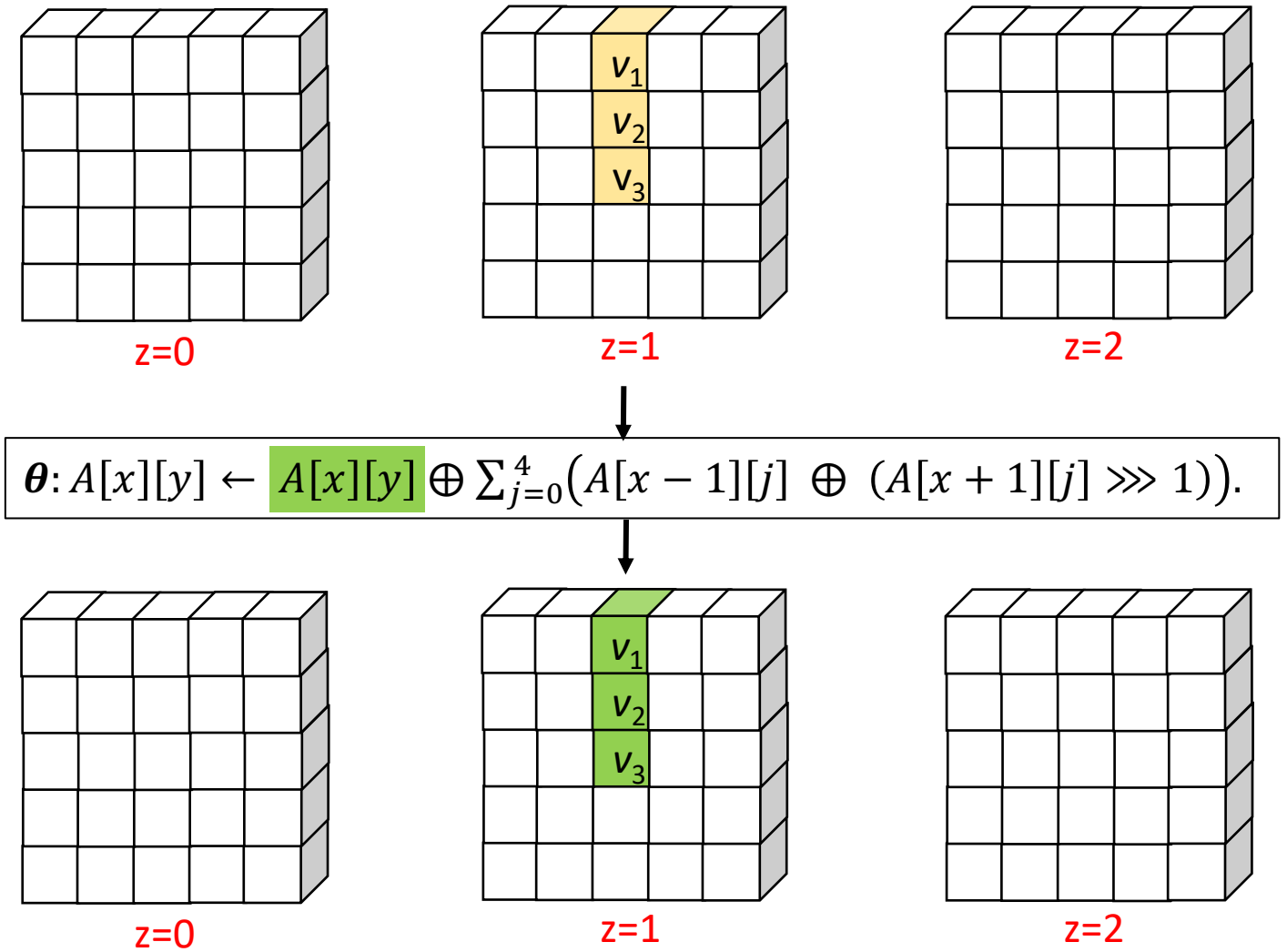


$$\boldsymbol{\theta}: A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\left(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\right).$$



First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



Through $\boldsymbol{\theta}$, three variables v_1, v_2, v_3 (s.t. $v_1+v_2=v_3$) influences Only 3-bit positions.

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

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

$$\boldsymbol{\rho}: A[x][y] \leftarrow A[x][y] \ggg r[x][y].$$

A[0][2]

0	1	2	3	4	5	6	7	8	9	10	11	12	…	62	63

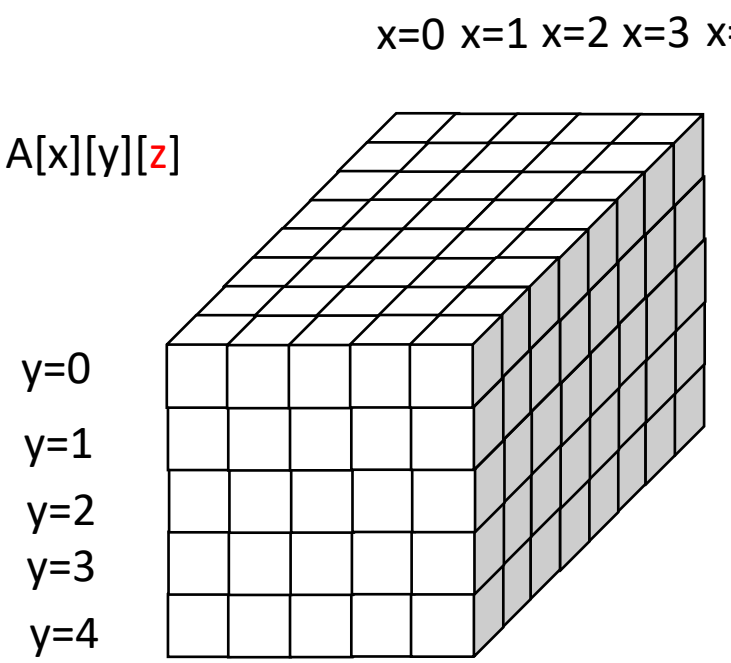
Right circular rotation by r[0][2]=3

A[0][2]

61	62	63	0	1	2	3	4	5	6	7	8	9	…	59	60

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



A (1600-bit)

Round *i*

$\boldsymbol{\theta}$: $A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\big(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\big)$.

$\boldsymbol{\rho}$: $A[x][y] \leftarrow A[x][y] \ggg r[x][y]$.

$\boldsymbol{\pi}$: $A[y][2x+3y] \leftarrow A[x][y]$.

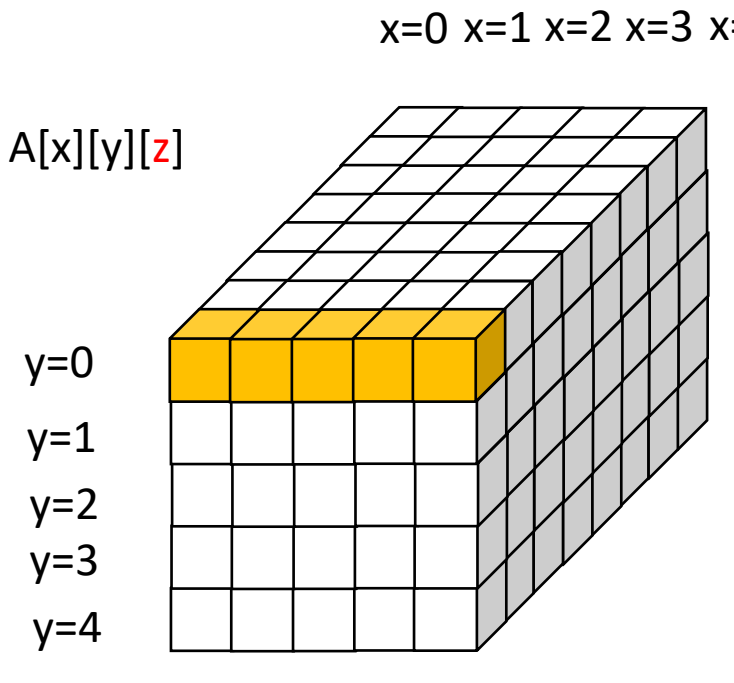
$\boldsymbol{\chi}$: $A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$

$\boldsymbol{\iota}$: $A[0][0] \leftarrow A[0][0] \oplus RC[i]$.

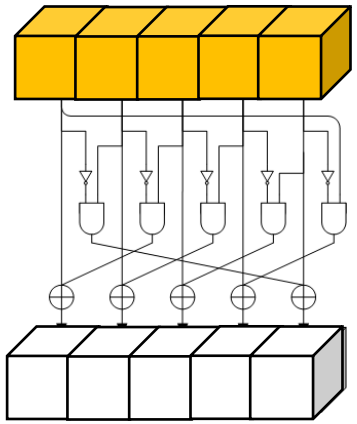
A (1600-bit)

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

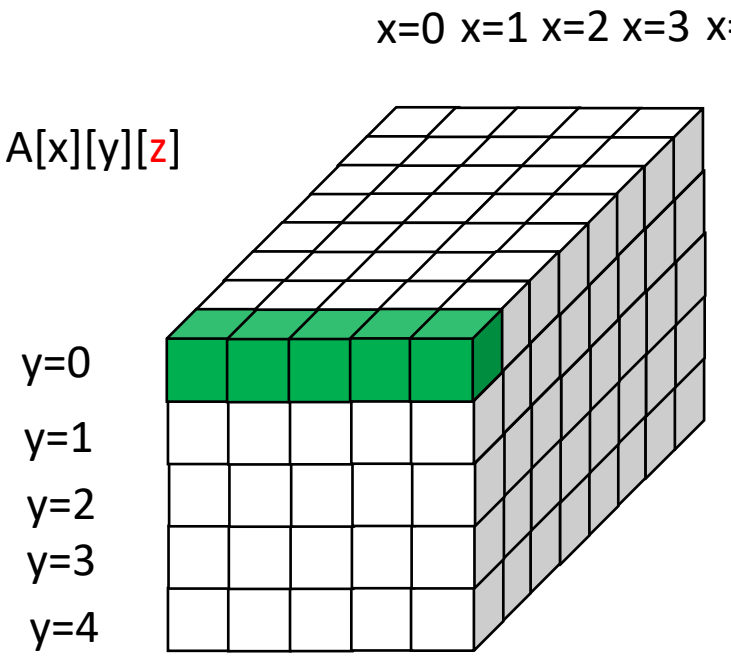


$$\chi: A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$$

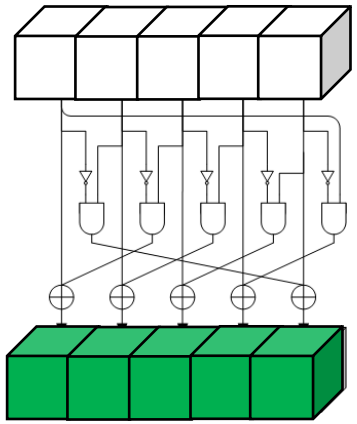


First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

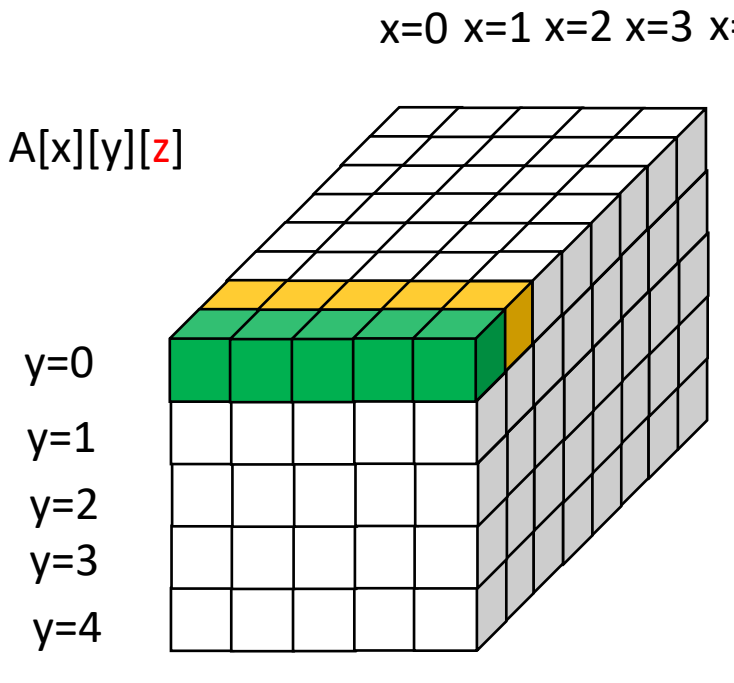


$$\chi: A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$$

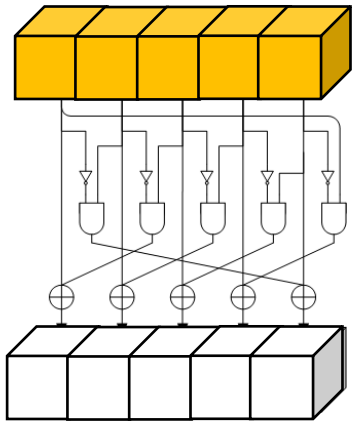


First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



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



First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

A[x][y][z]

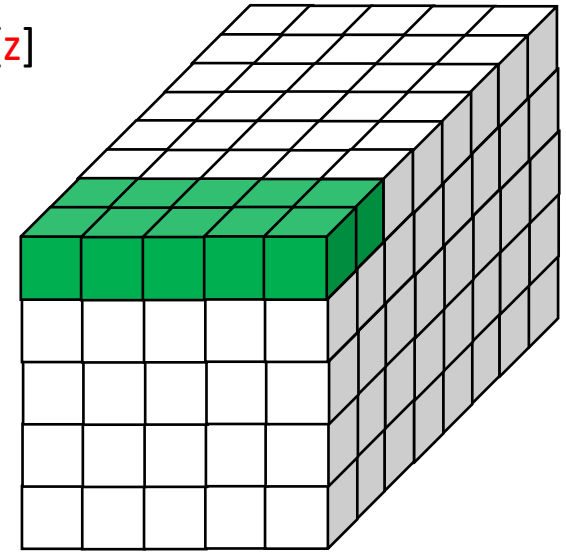
y=0

y=1

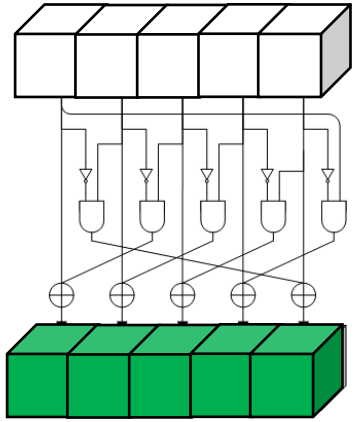
y=2

y=3

y=4

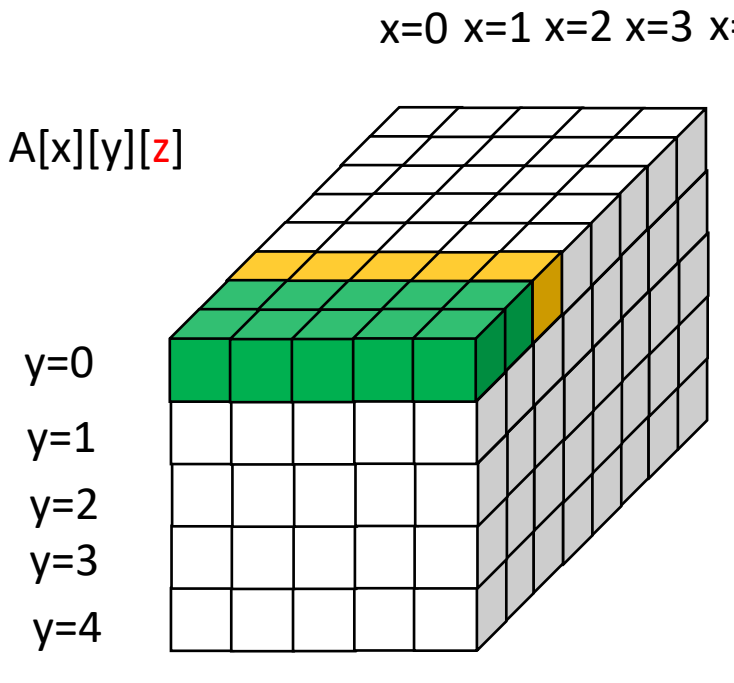


$$\chi: A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$$

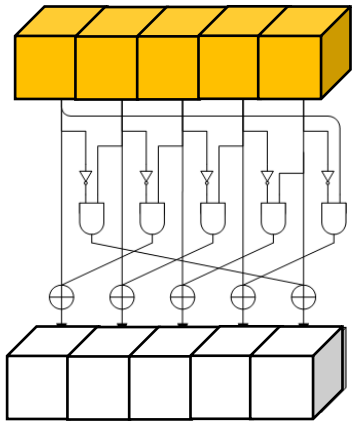


First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



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



First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

x=0 x=1 x=2 x=3 x=4

A[x][y][z]

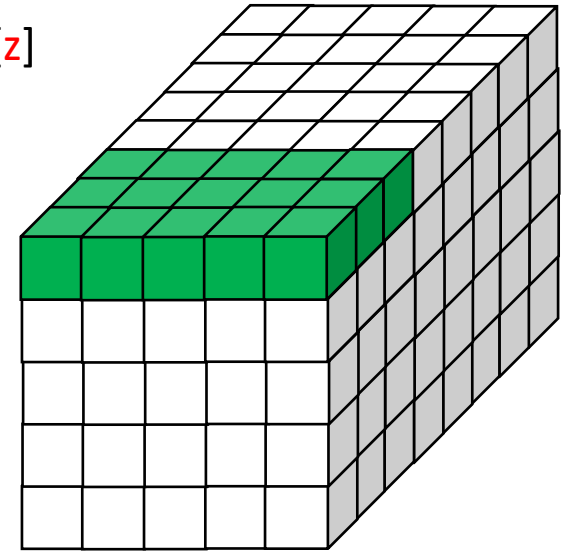
y=0

y=1

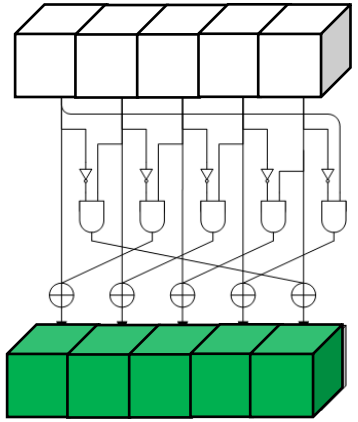
y=2

y=3

y=4

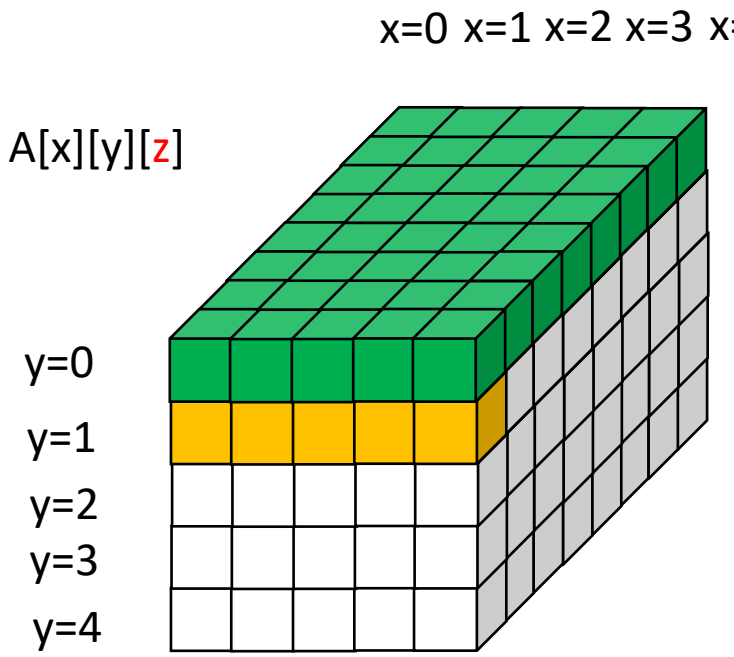


$$\chi: A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$$

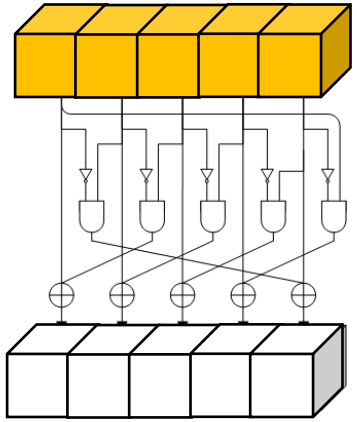


First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

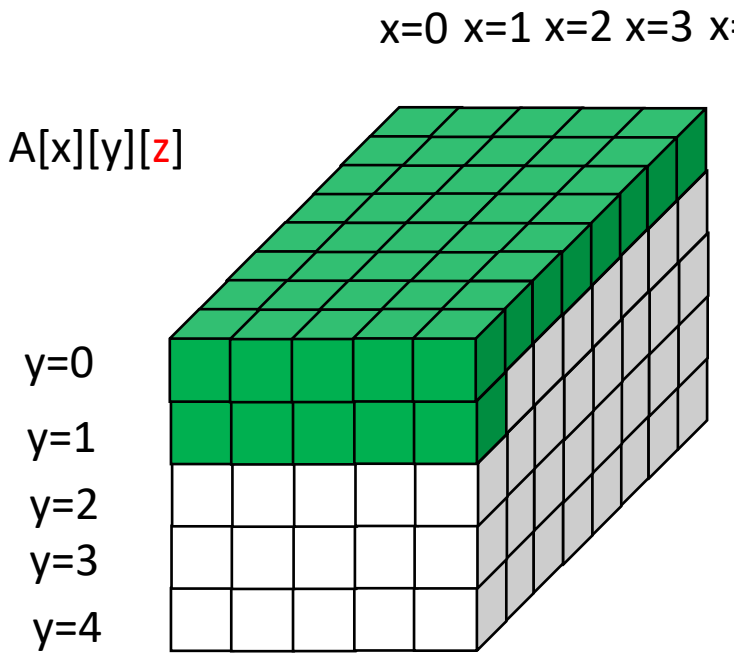


$$\chi: A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$$

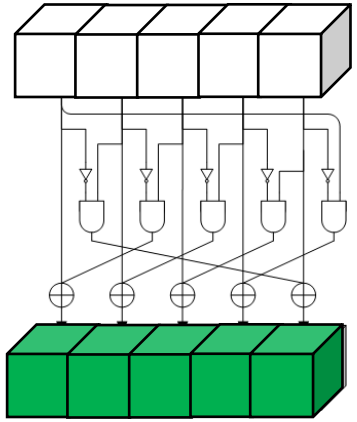


First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

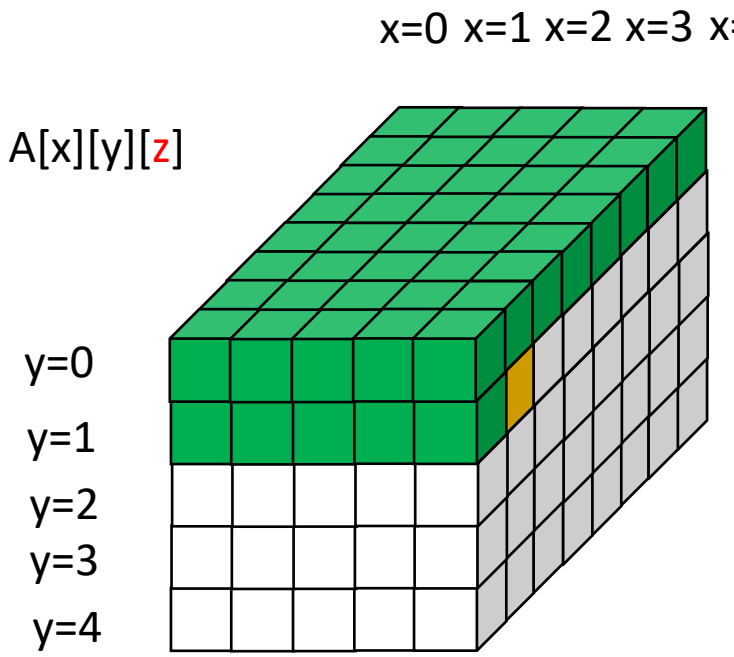


$$\chi: A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$$

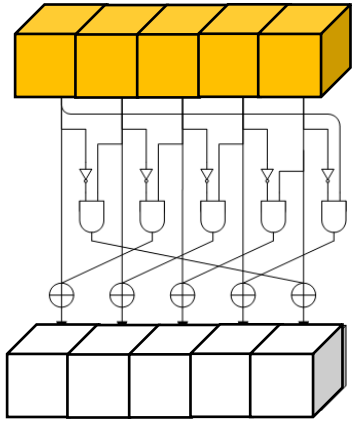


First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

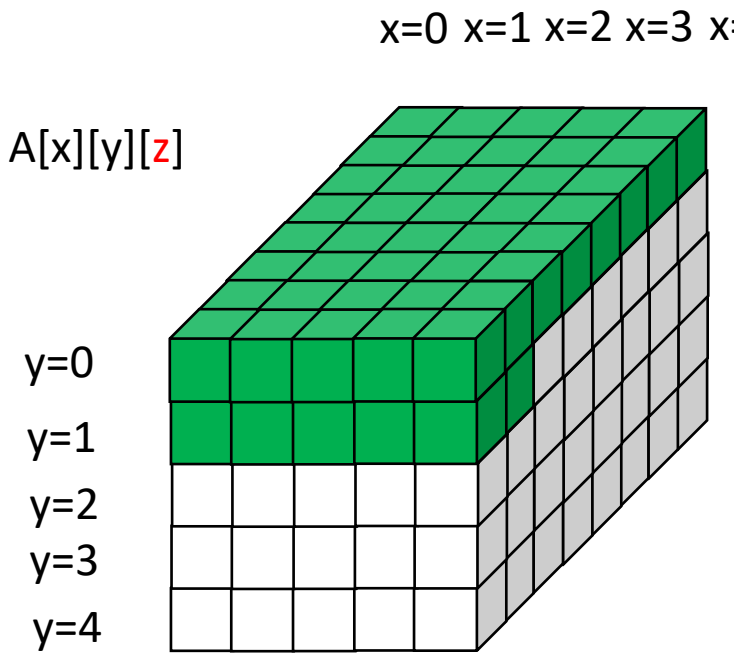


$$\chi: A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$$

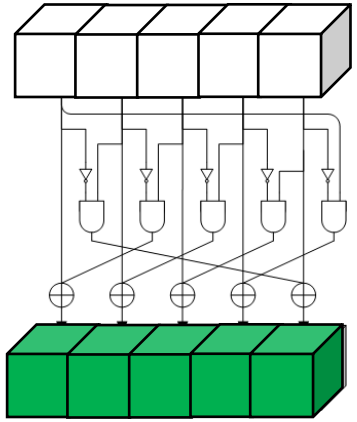


First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

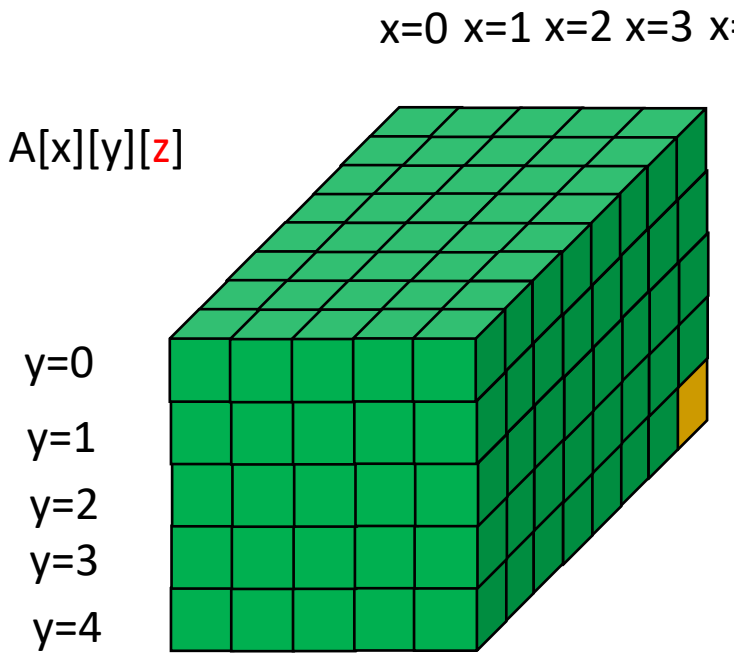


$$\chi: A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$$

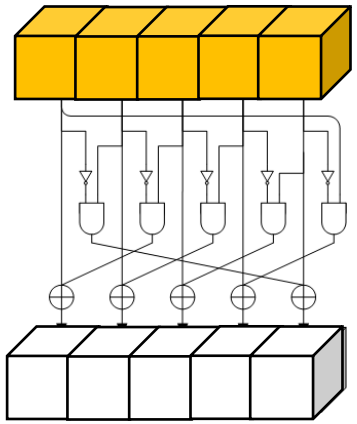


First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

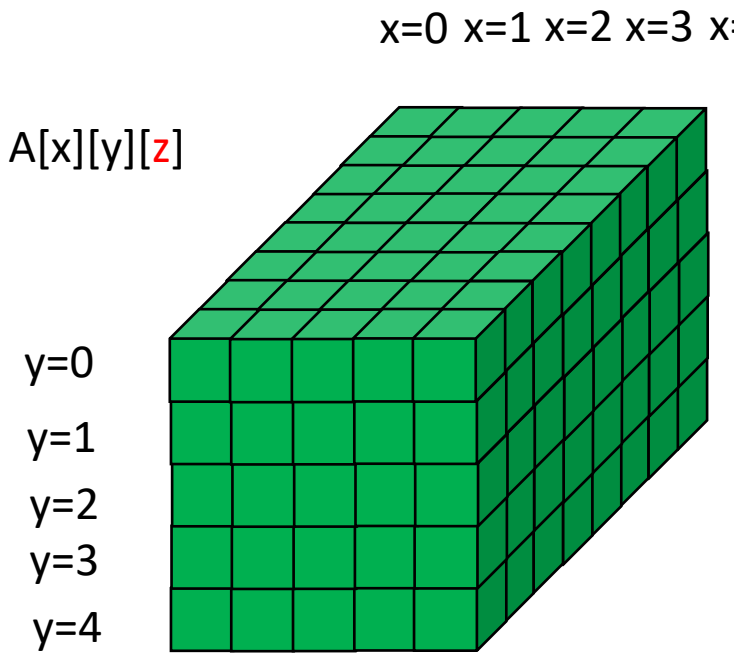


$$\chi: A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$$

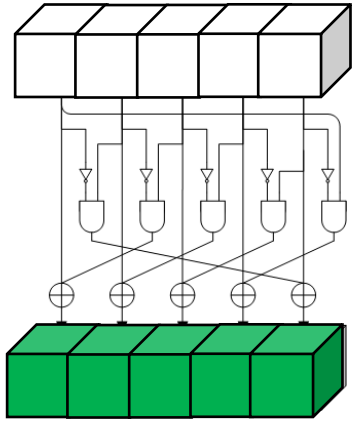


First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

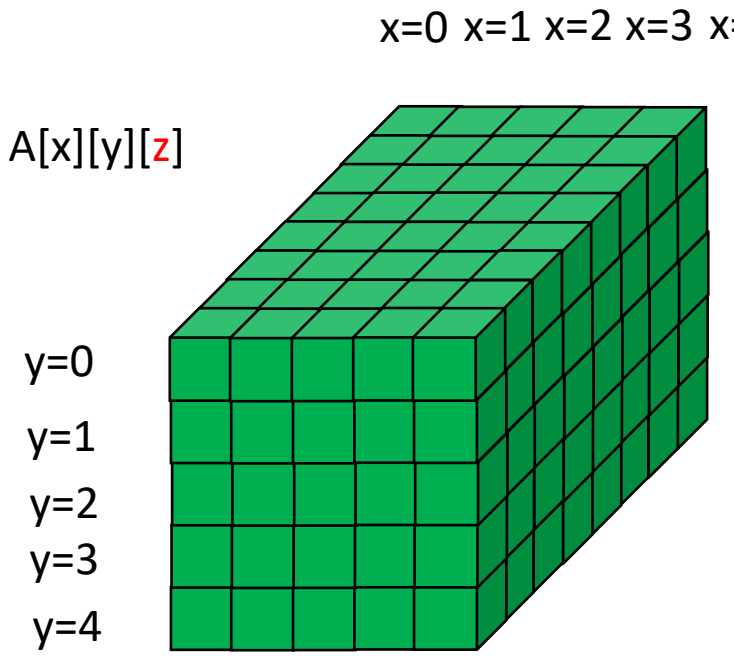


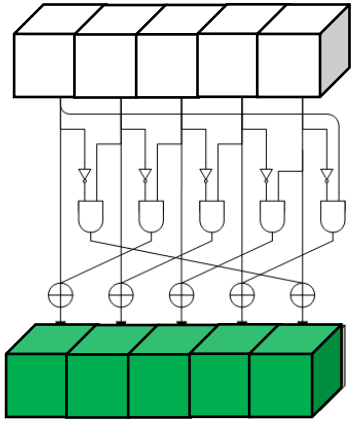
$$\chi: A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$$



First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128





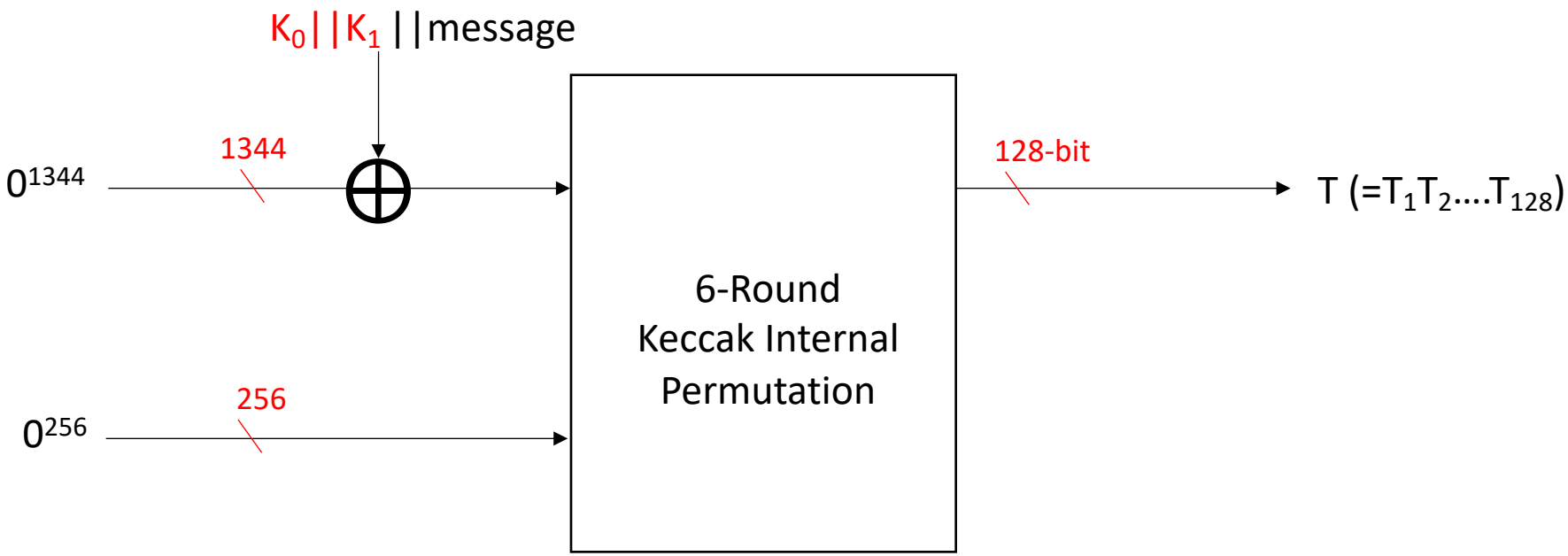
$$\chi: A[x][y] \leftarrow A[x][y] \oplus ((\neg A[x+1][y]) \wedge A[x+2][y])$$

Only neighbors are multiplied by each other. (Degree is 2.)

First Attack Idea

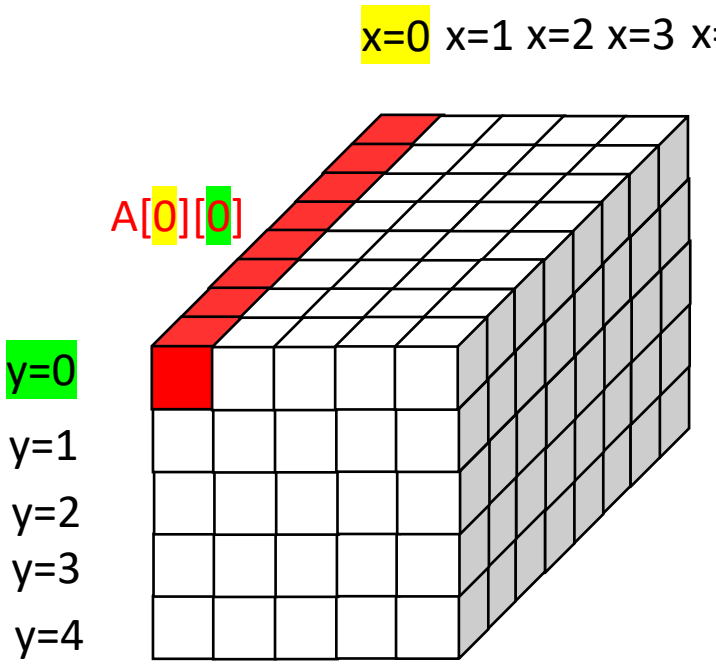
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

- The secret key K is 128-bit.
- Let $K=K_0||K_1$, where $|K_0|=|K_1|=64$.



First Attack Idea

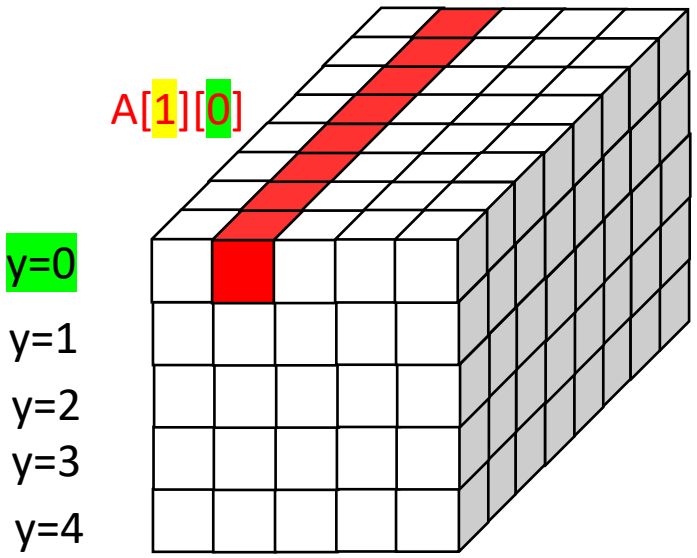
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



K_0=A[0][0] and K_1=A[1][0].

First Attack Idea

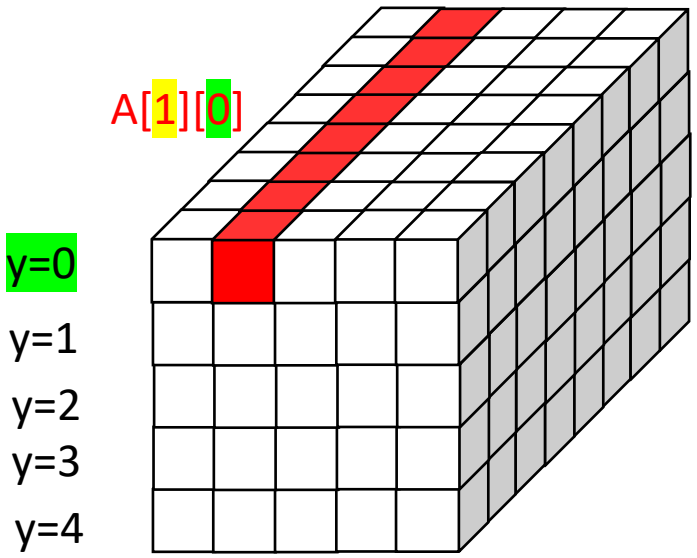
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



K_0=A[0][0] and K_1=A[1][0].

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

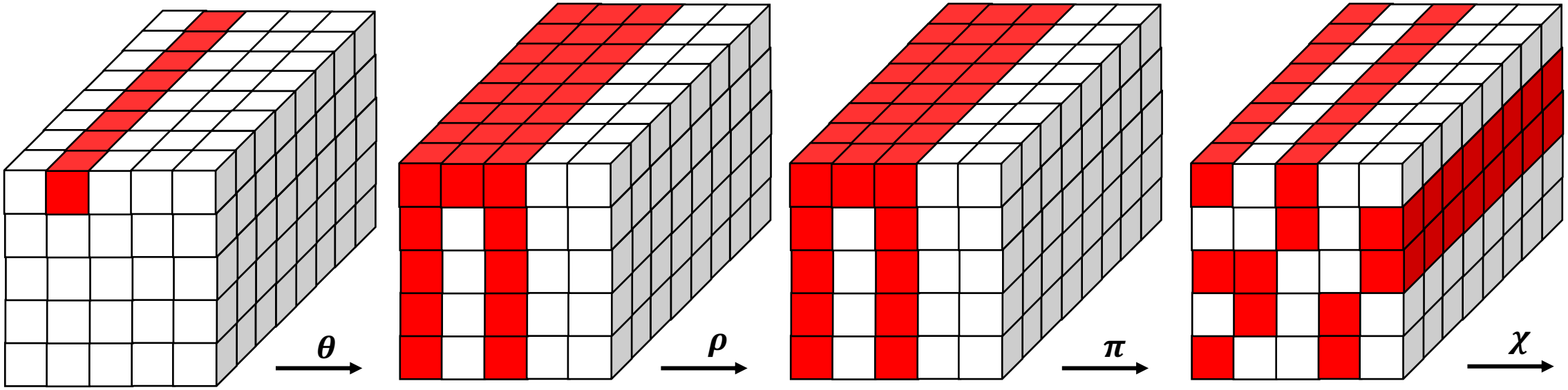


K_0=A[0][0] and K_1=A[1][0].

First Attack Idea

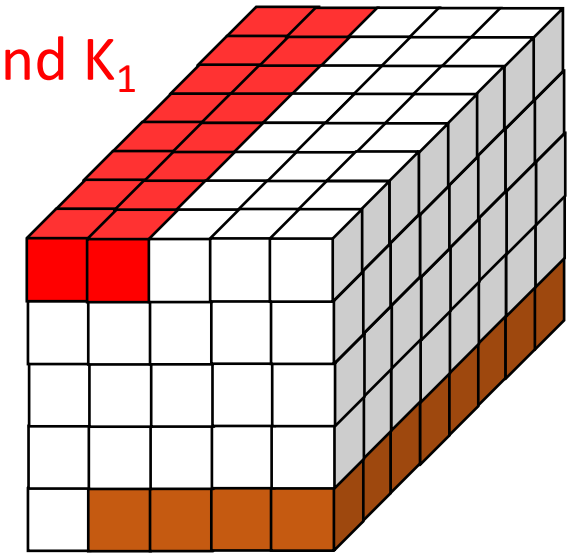
Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

- Let us see the propagation of K_1=A[1][0].



First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



K_0 and K_1

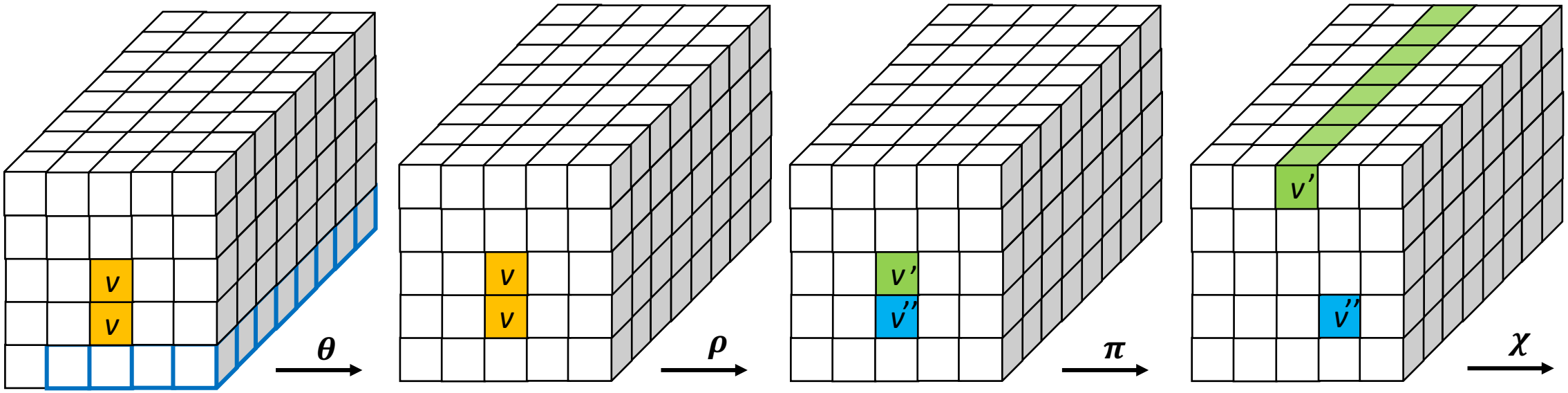
The white portion is controlled by the attacker.

Capacity part is fixed as 0^{256}

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

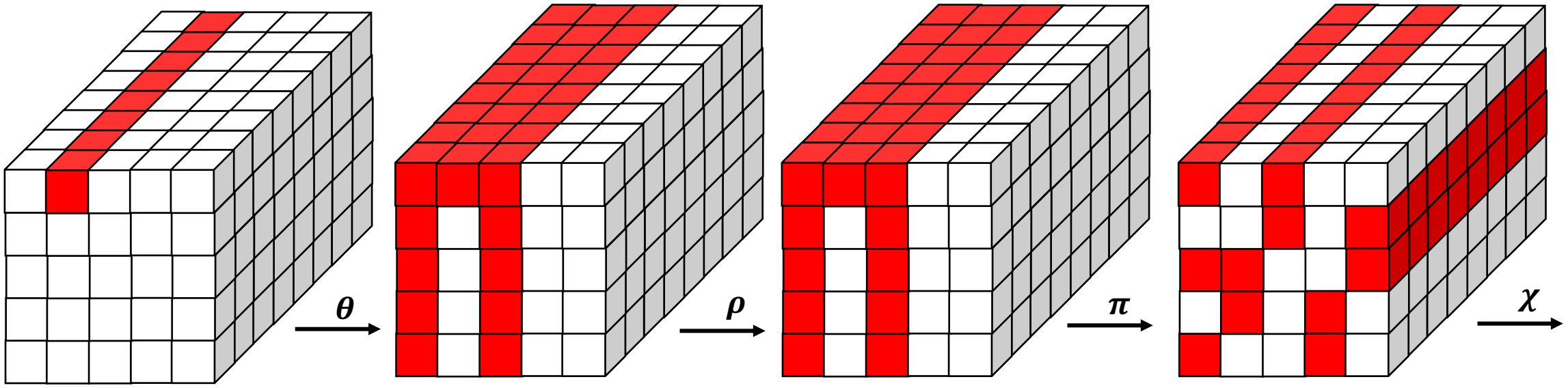
- v=(v_1,v_2,…..,v_{32},0,0,0,….,0) is a 64-bit tuple with 32 cube variables.
- Let A[2][2]=A[2][3]=v. Then, see the propagation of v.



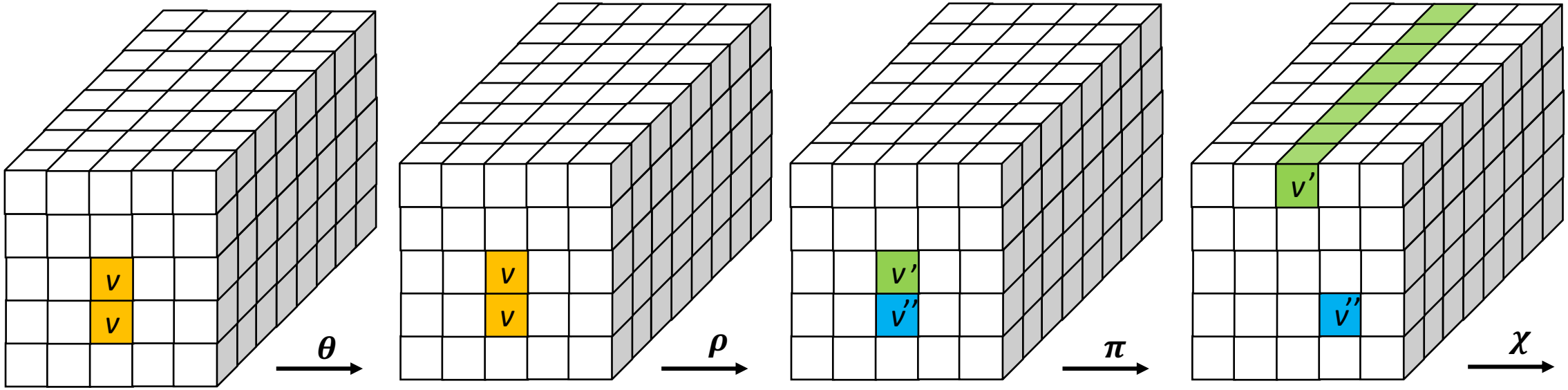
v'= v ⋙ r[2][2],
v''= v ⋙ r[2][3]

First Attack Idea

Propagation of K_1

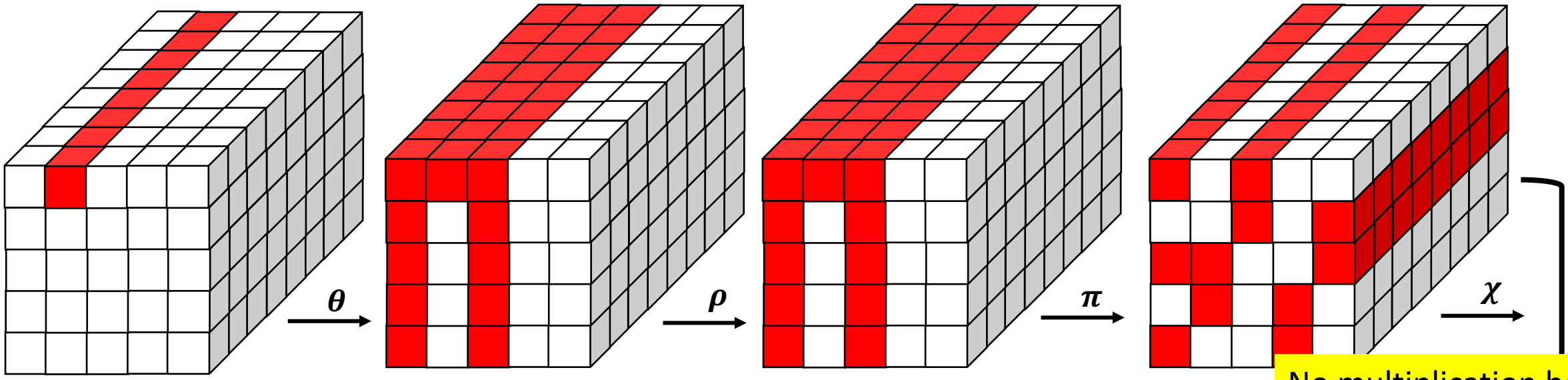


Propagation of v



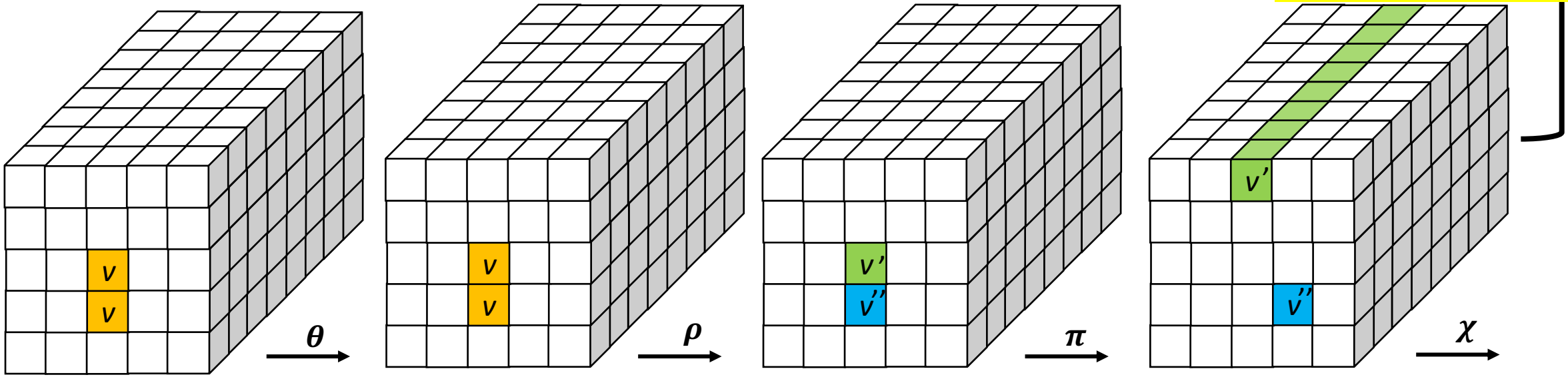
First Attack Idea

Propagation of K_1

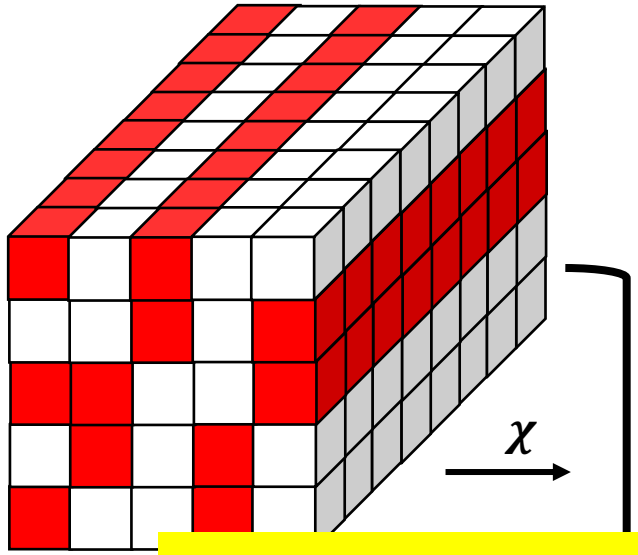


No multiplication between K_1 and 32 variables.

Propagation of v



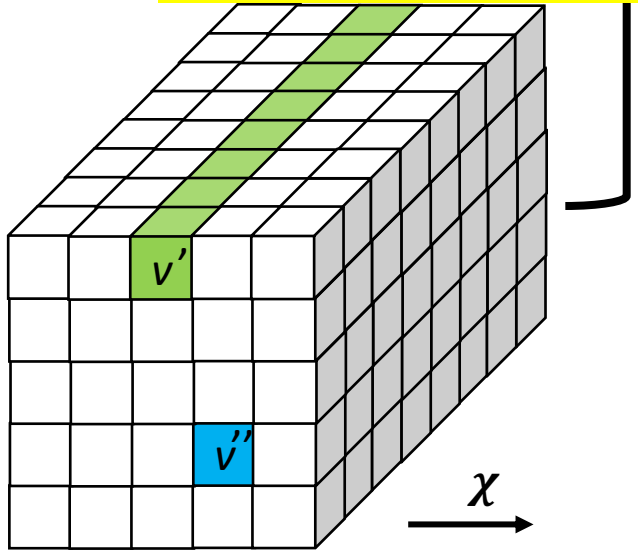
First Attack Idea



No multiplication between K_1 and 32 variables.

And, no multiplication among 32 variables.

5 Rounds

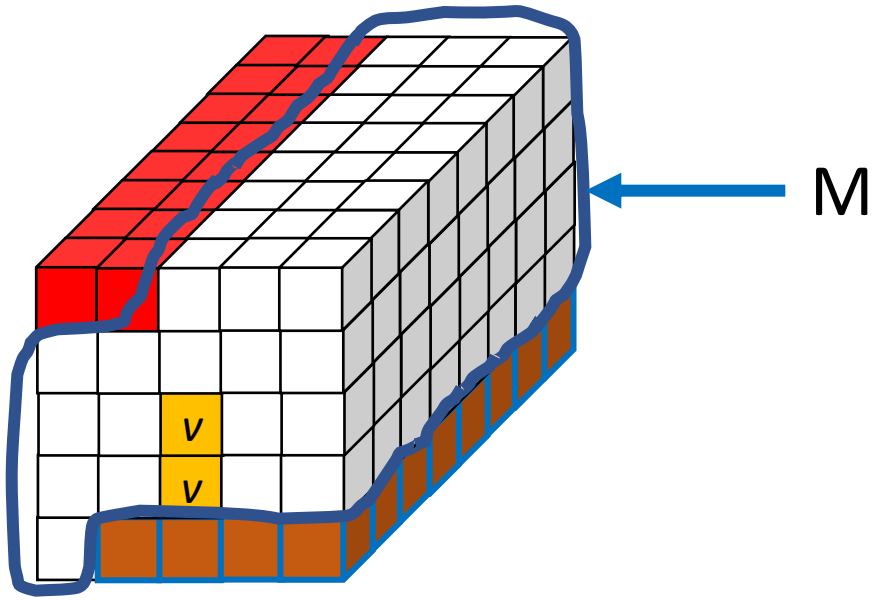


Therefore, the superpoly of the cube $C_{v_1 v_2 \ldots v_{32}}$ for each output-bit T_i ($1 \leq i \leq 128$) does not depend on K_1. In other words, the superpolys depends only on K_0.

First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128

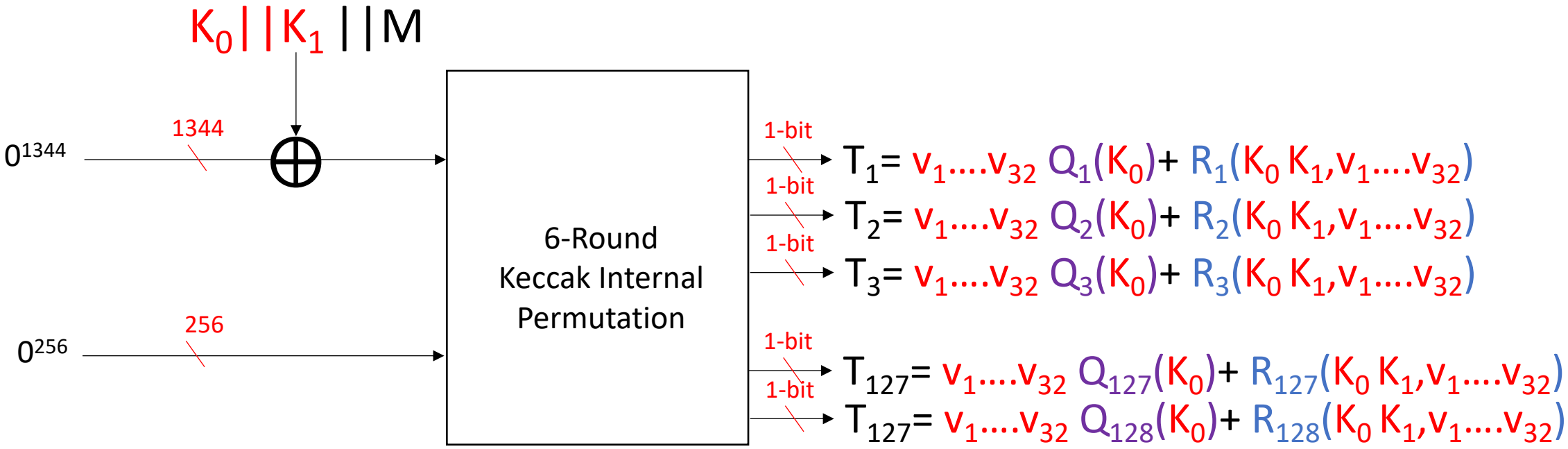
- $v=(v_1,v_2,\ldots..,v_{32},0,0,0,\ldots.,0)$ is a 64-bit tuple with 32 cube variables.
- Let A[2][2]=A[2][3]=v and arbitrarily choose the values of remaining other white part. Then, we say that the message is M.



M

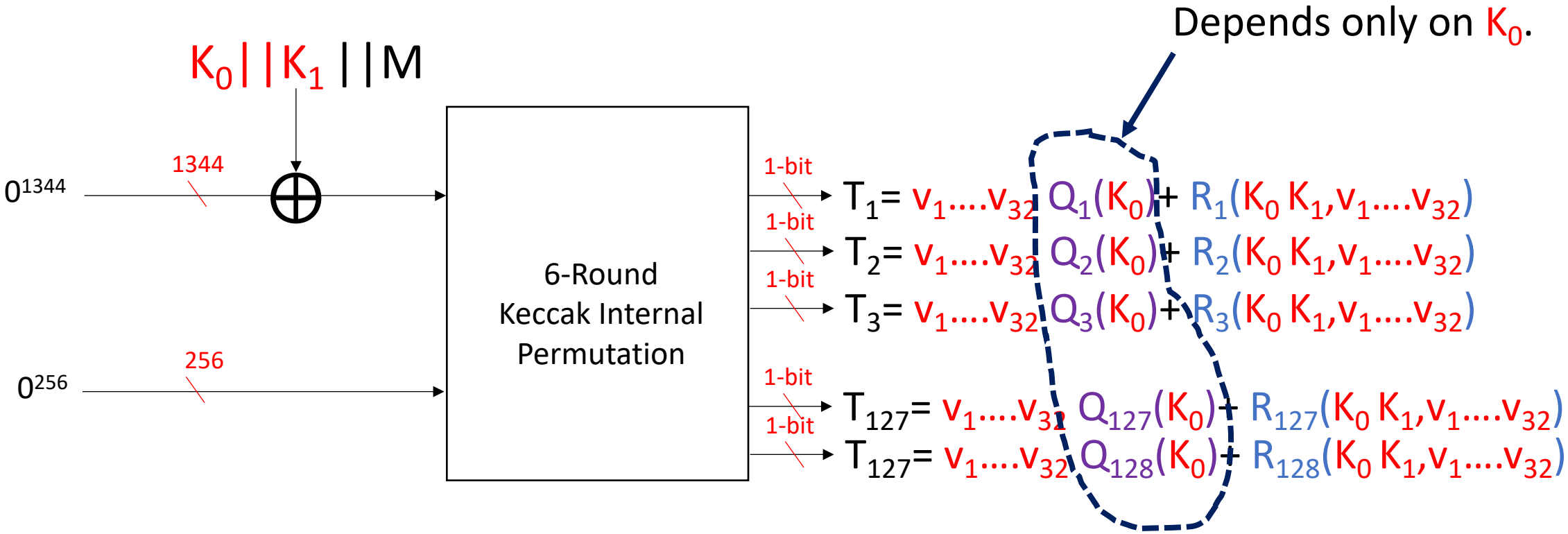
First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



First Attack Idea

Finding Superpolys with Partial Key Variables for 6-round Keccak-MAC-128



First Attack Idea

Off-line Phase using Superpolys with K_0 Variables

- We fixed K_1 as 0^{64}, because the superpolys do not depend on K_1.
- For each $j \in \{0,1\}^{64}$, the attacker computes the followings:

$$\sum_{(v_1,\ldots,v_{32})\in C_{v_1\ldots v_{32}}} \mathrm{Keccak-MAC}(j\,||0^{64},M) = Q_1(j)||Q_2(j)||\ldots||Q_{128}(j)$$

- Above cube-sum requires 2^{32} $\mathrm{Keccak-MAC}$ -computations and 128-bit memory.
- We have to perform the off-line cube-sum 2^{64} times.
- In total, for the offline phase, 2^{96} $\mathrm{Keccak-MAC}$ -computations and 2^{71} bit memory are required.

First Attack Idea

On-line Phase using Superpolys with Actual Secret key $K_0||K_1$

- The Attacker makes oracle $\mathrm{Keccak-MAC}(K_0||K_1,\cdot)$-queries and compute the following online cube-sum:

$$\sum_{(v_1,\ldots,v_{32})\in C_{v_1\ldots v_{32}}} \mathrm{Keccak-MAC}(K_0||K_1,M) = Q_1(K_0)||Q_2(K_0)||\ldots||Q_{128}(K_0)$$

- Above cube-sum requires 2^{32} Keccak-MAC$(K_0||K_1,\cdot)$-queries and 128-bit memory.
- Then, find j^* from the memory table constructed in Off-line Phase such that

$$Q_1(K_0)||Q_2(K_0)||\ldots||Q_{128}(K_0) = Q_1(j^*)||Q_2(j^*)||\ldots||Q_{128}(j^*)$$

- We conclude that $K_0=j^*$ with high probability. (Note that with high probability, there is a matching between off-line cube-sum and on-line cube-sum.)

First Attack Idea

Searching for K_1

- After finding the 64-bit value of K_0, we exhaustively search the remaining 64-bit K_1 by 2^{64} Keccak-MAC-computations. (Divided-and-Conquer Attack)

Conditional Cube Attack on Reduced-Round Keccak Sponge Function

By Senyang Huang, Xiaoyun Wang, Guangwu Xu, Meiqin Wang, Jingyuan Zhao
(**EUROCRYPT 2017**)

Conditional Cube Variables, Ordinary Cube Variables

- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.

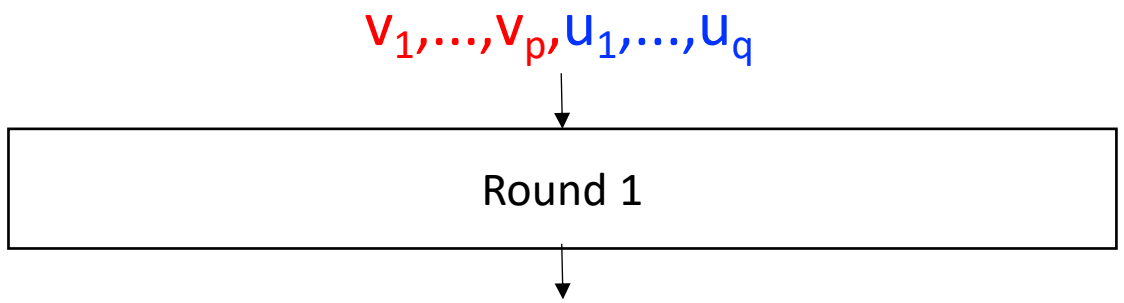
Conditional Cube Variables, Ordinary Cube Variables

- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.

Let us see the propagation of these variables to explain the meanings of conditional and ordinary cube variables.

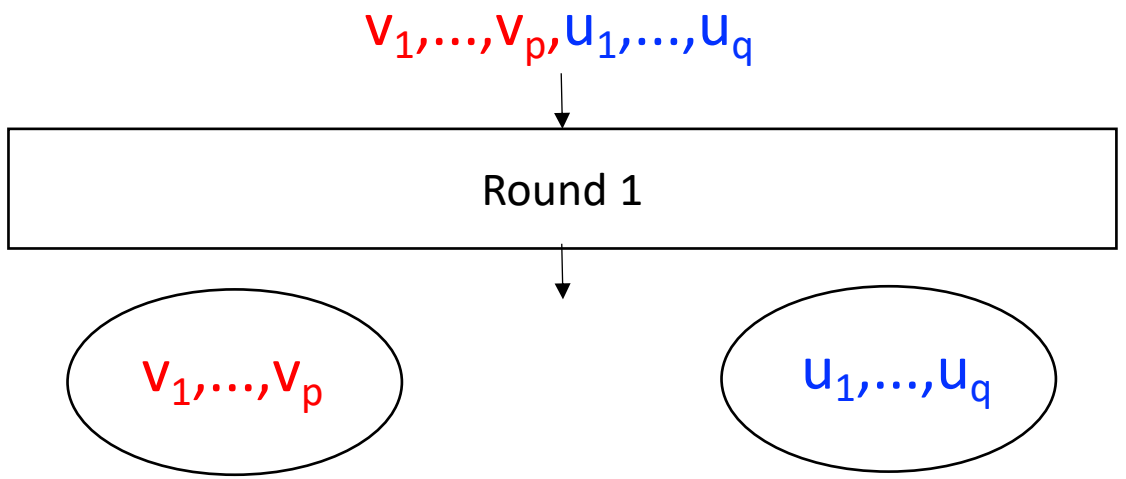
Conditional Cube Variables, Ordinary Cube Variables

- There are $p+q$ cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



Conditional Cube Variables, Ordinary Cube Variables

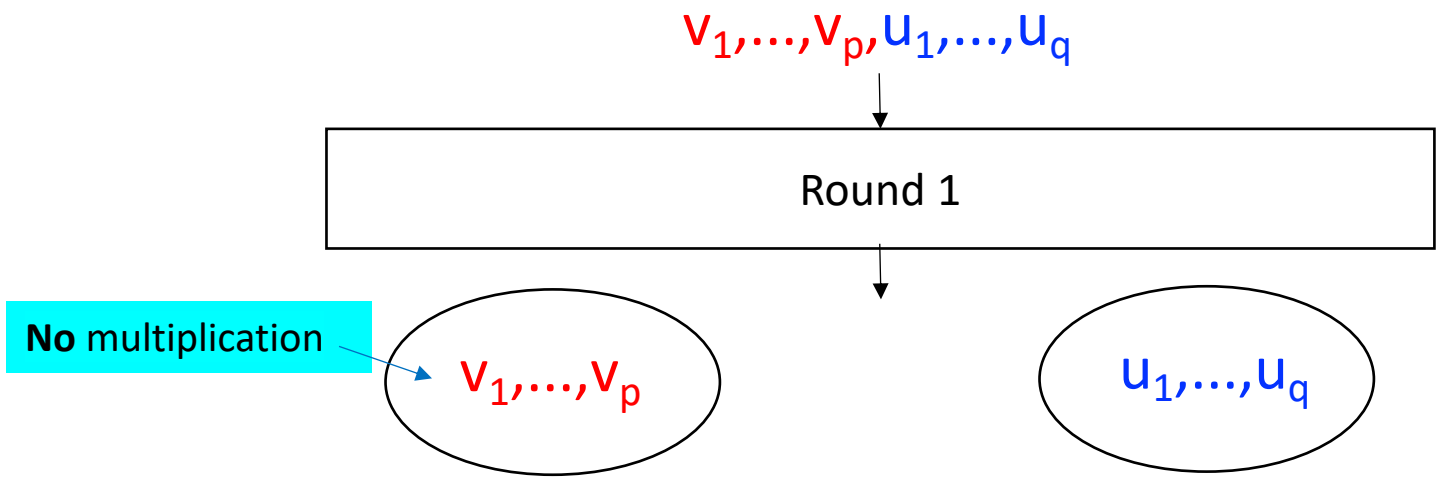
- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



Let us consider two groups.

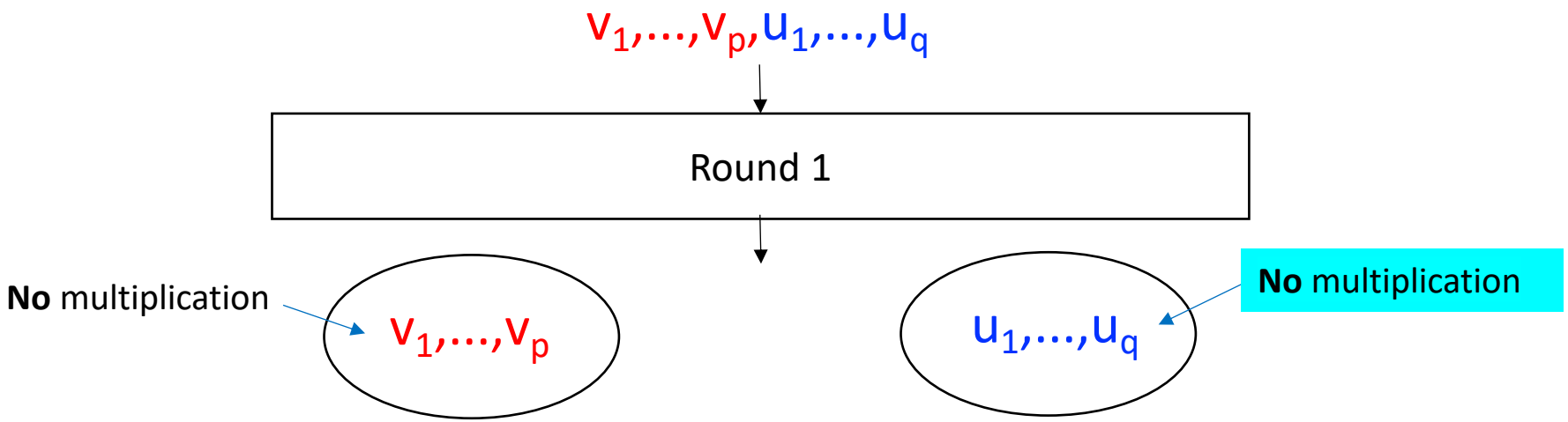
Conditional Cube Variables, Ordinary Cube Variables

- There are $p+q$ cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



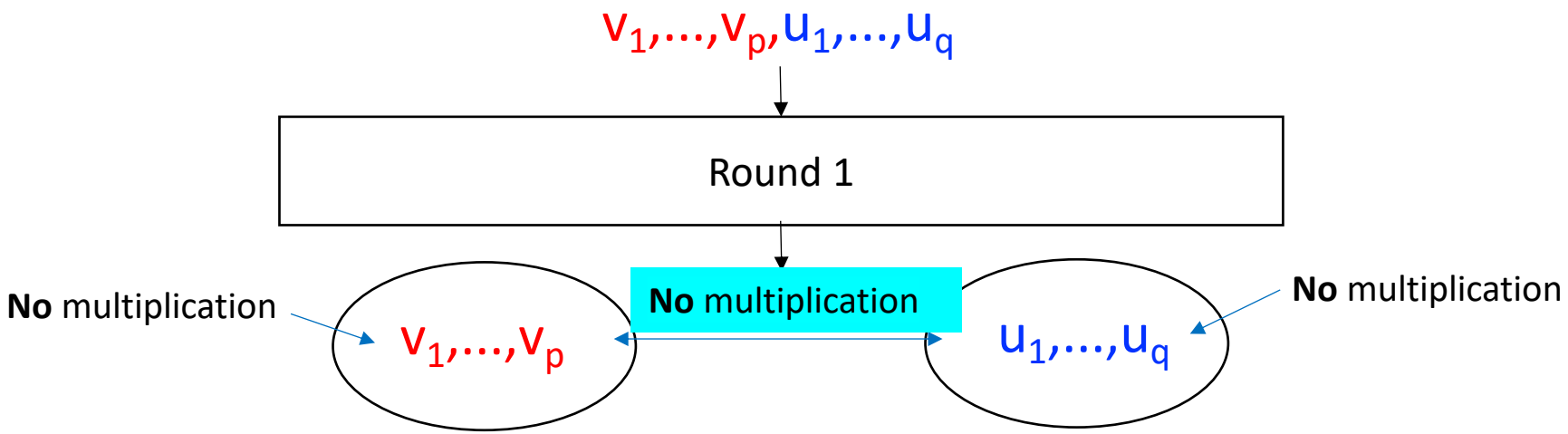
Conditional Cube Variables, Ordinary Cube Variables

- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



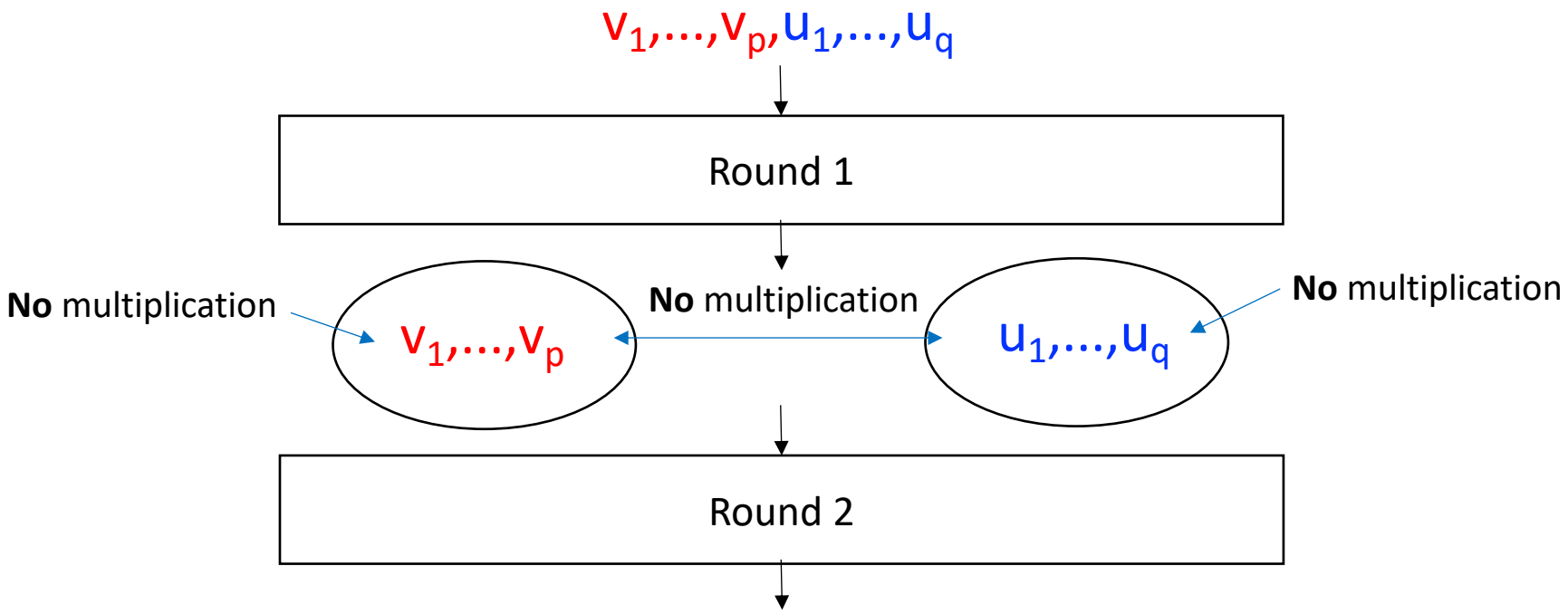
Conditional Cube Variables, Ordinary Cube Variables

- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



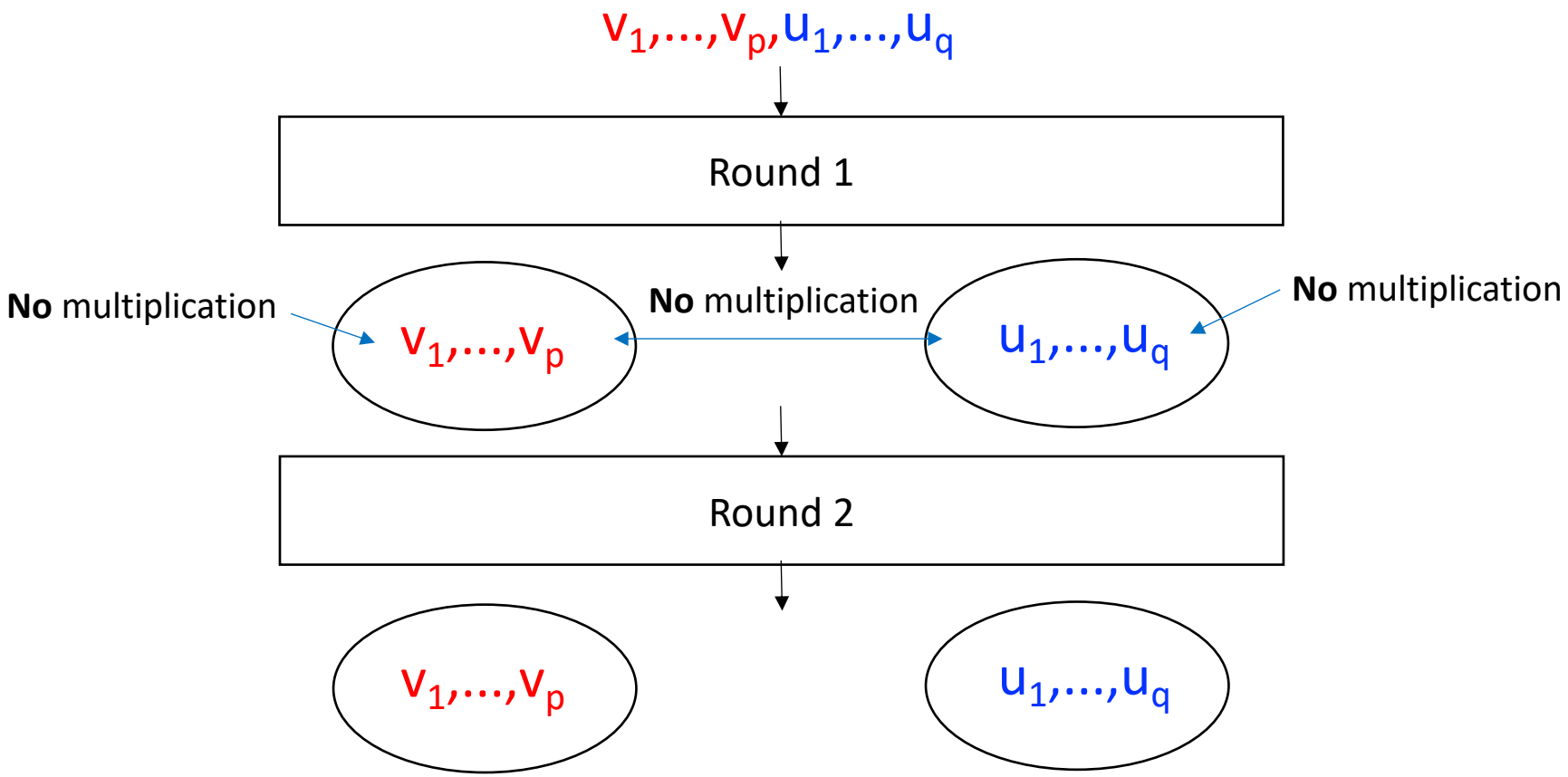
Conditional Cube Variables, Ordinary Cube Variables

- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



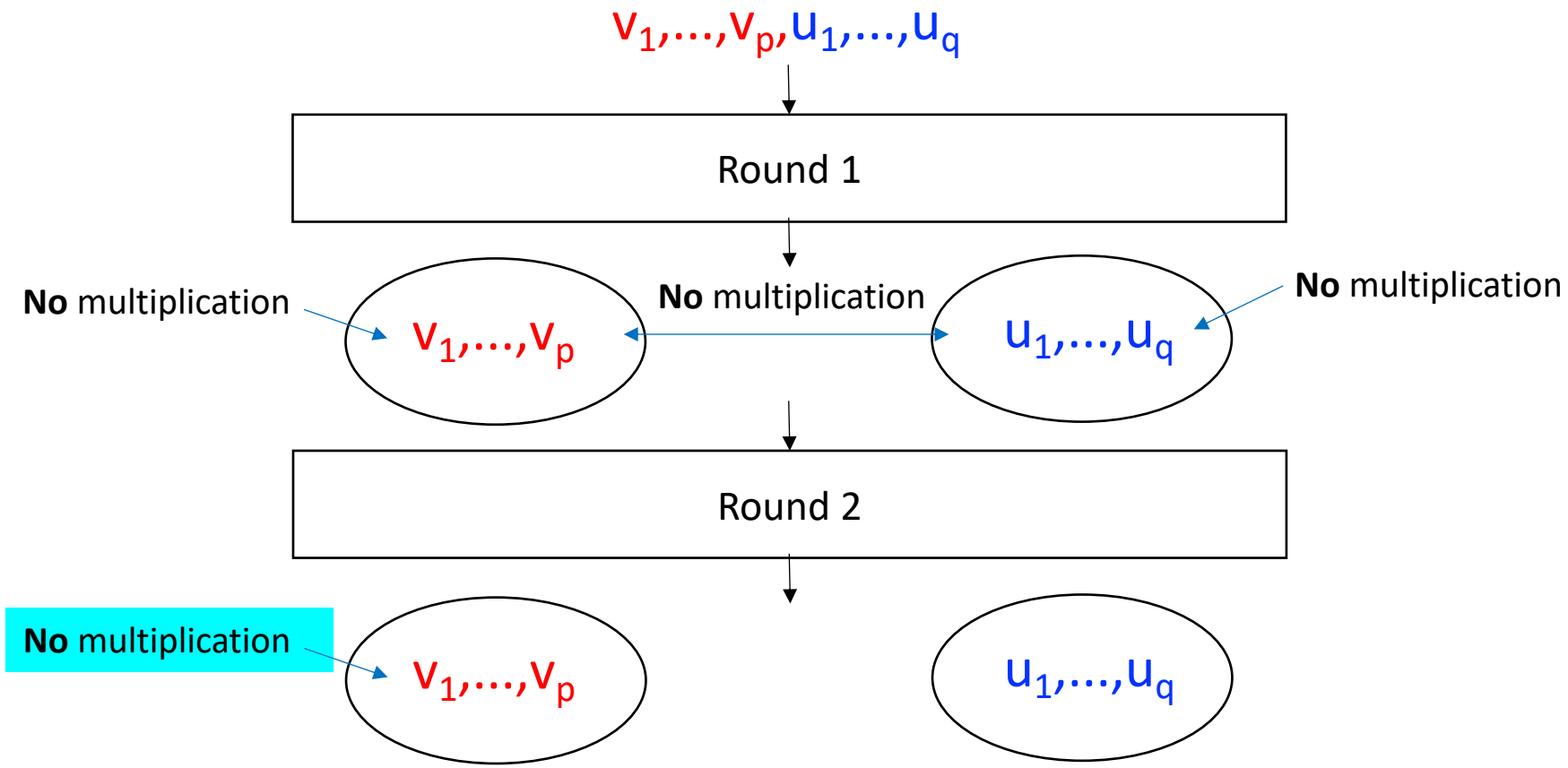
Conditional Cube Variables, Ordinary Cube Variables

- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



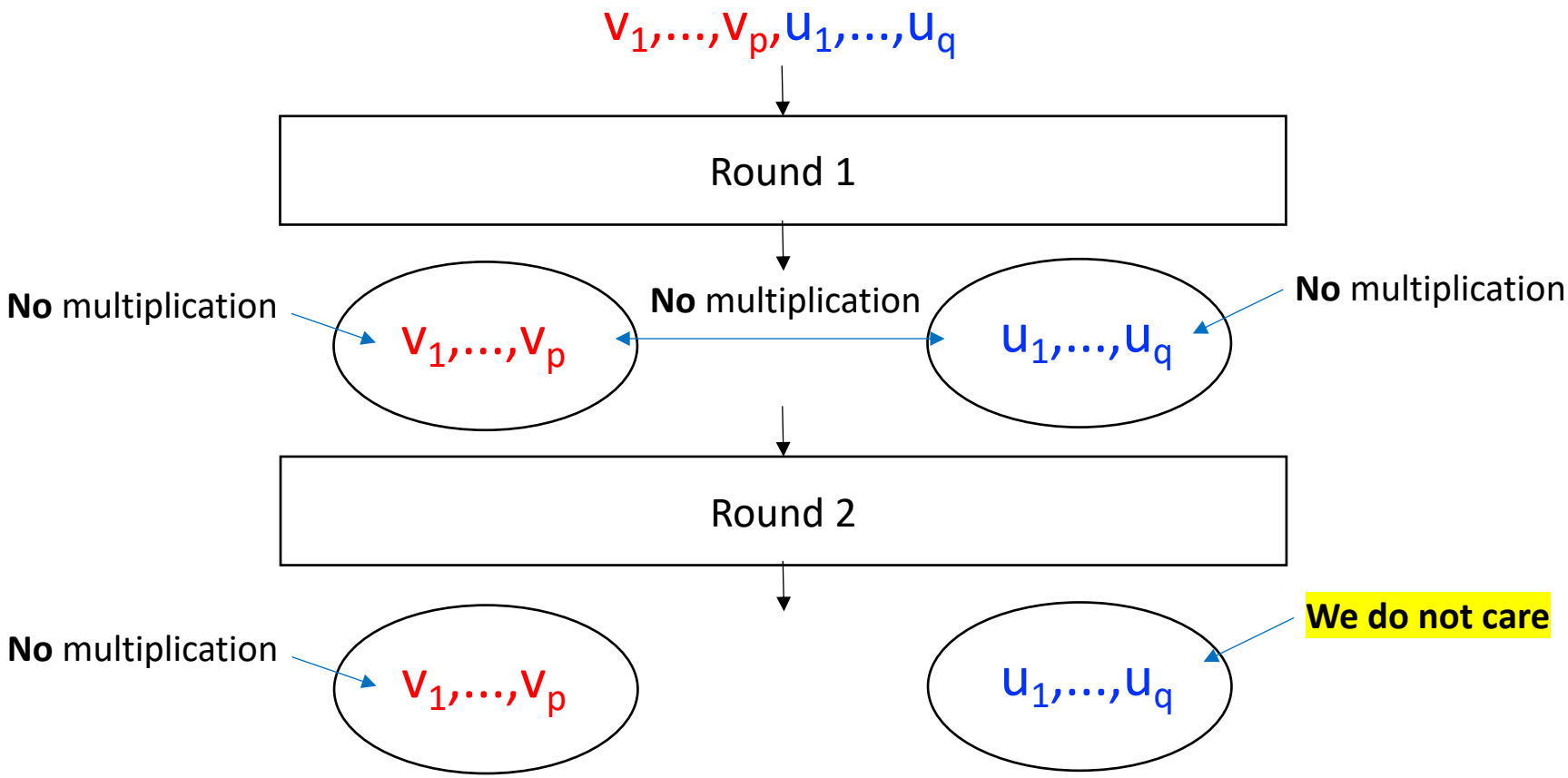
Conditional Cube Variables, Ordinary Cube Variables

- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



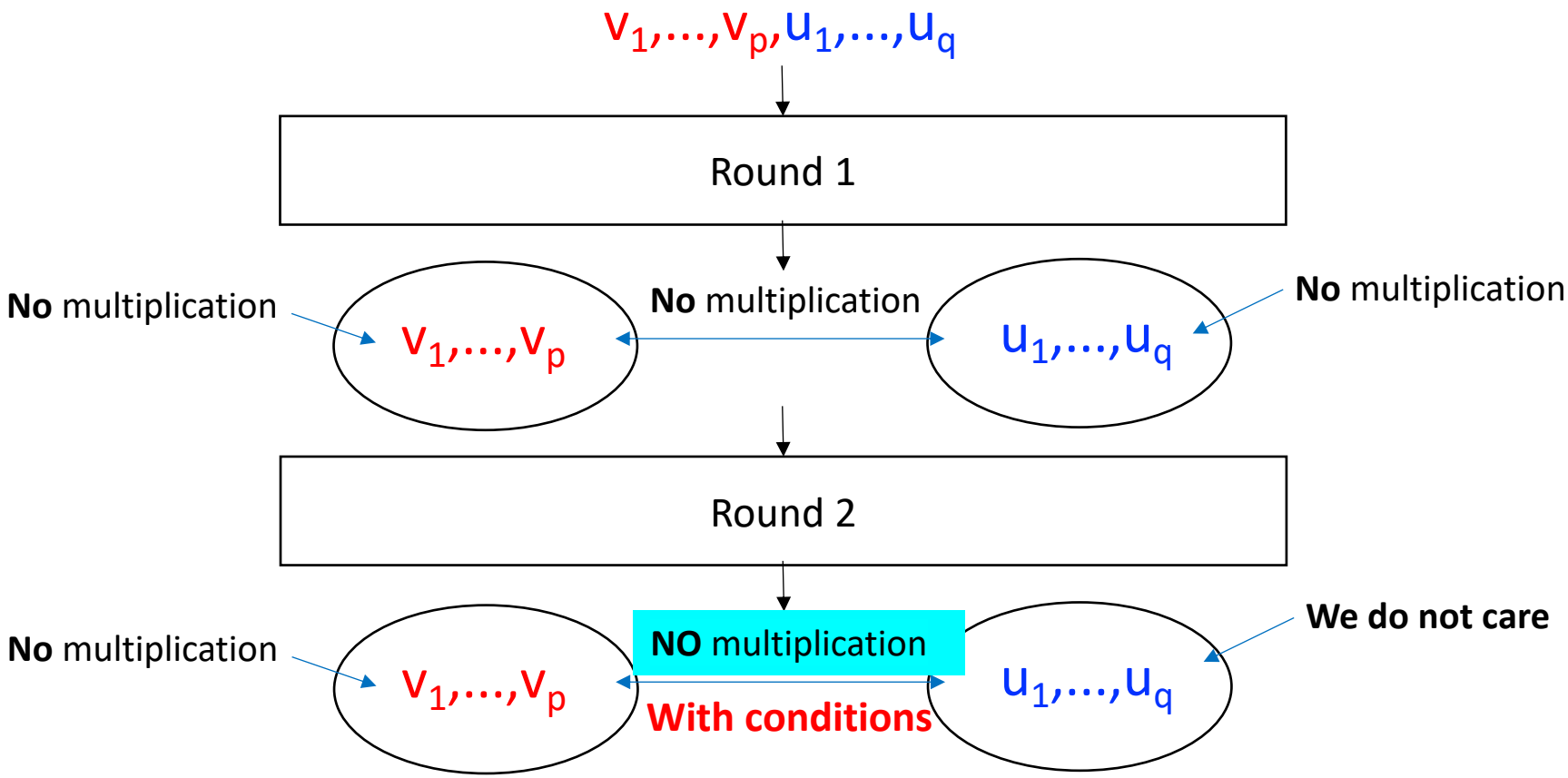
Conditional Cube Variables, Ordinary Cube Variables

- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



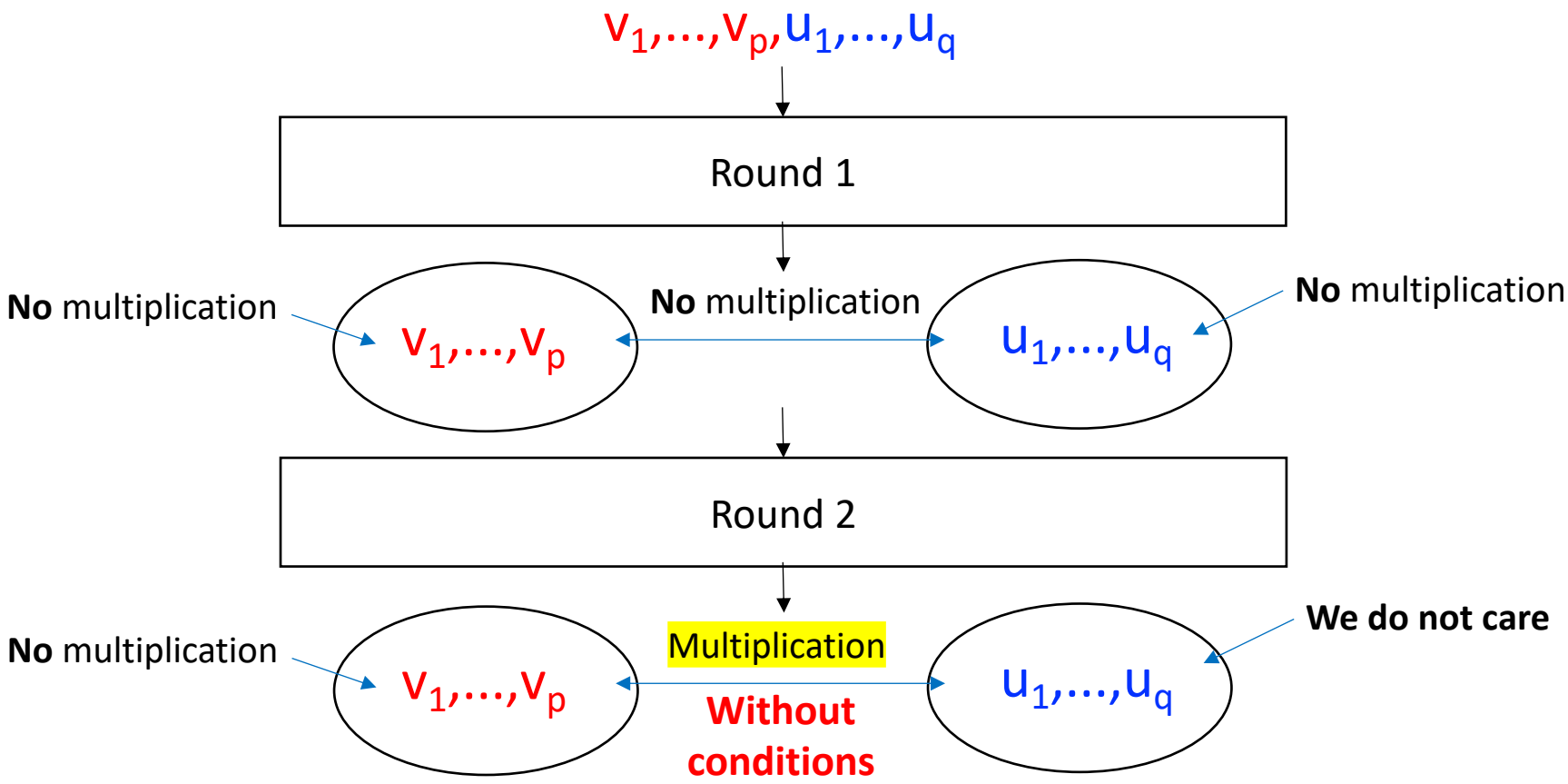
Conditional Cube Variables, Ordinary Cube Variables

- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



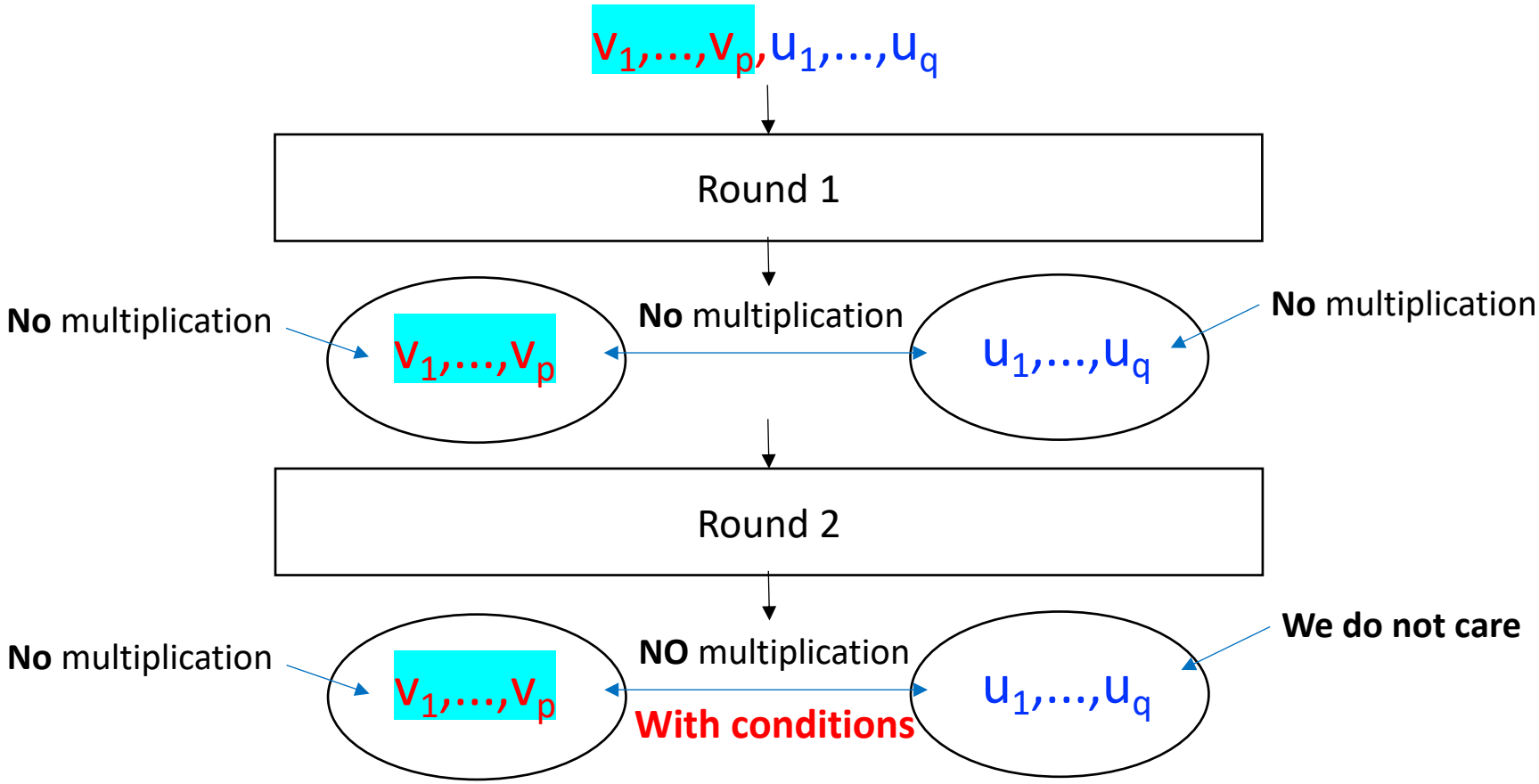
Conditional Cube Variables, Ordinary Cube Variables

- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



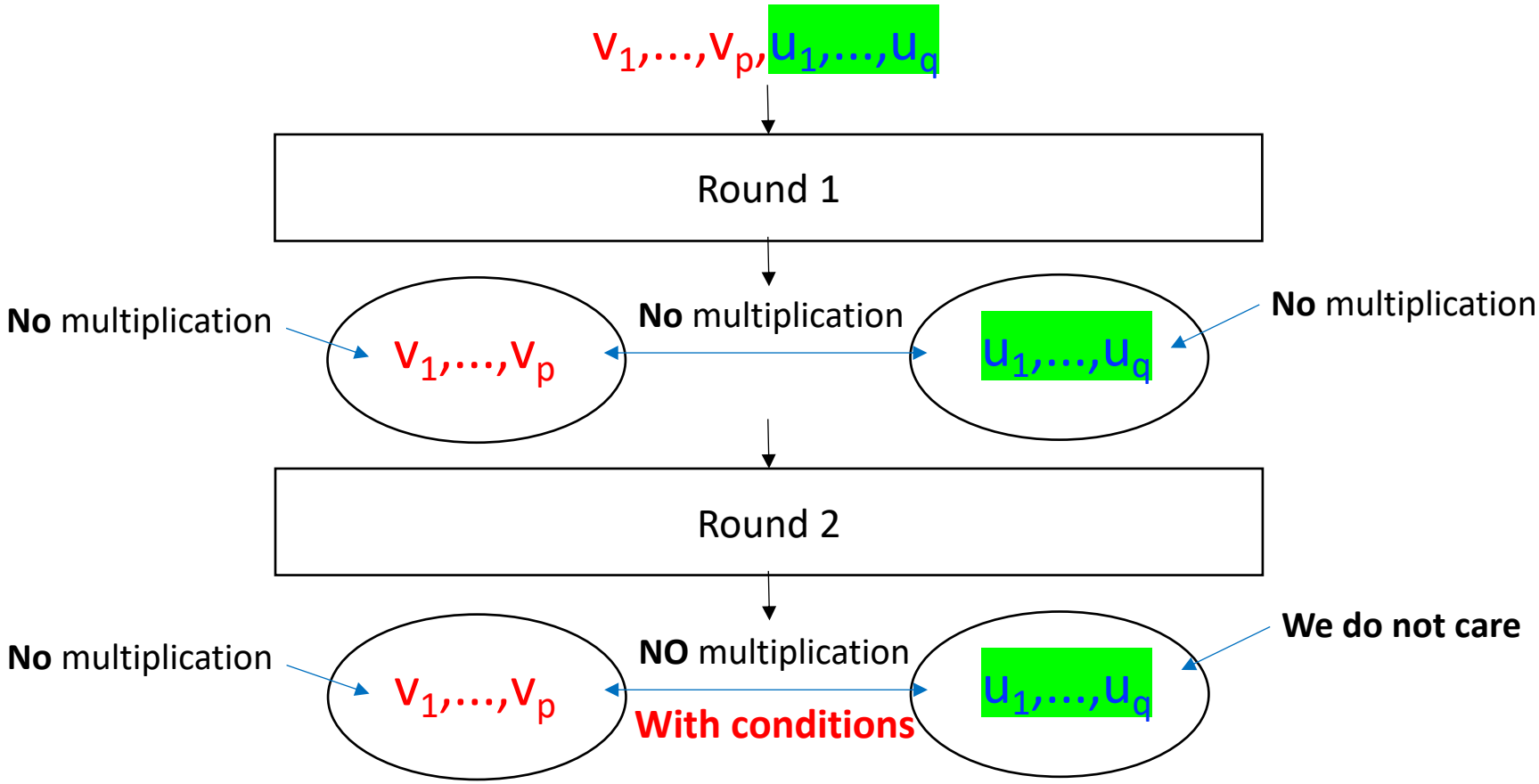
Conditional Cube Variables, Ordinary Cube Variables

- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



Conditional Cube Variables, Ordinary Cube Variables

- There are p+q cube variables, $v_1,\ldots,v_p,u_1,\ldots,u_q$.



A question is

- Given p conditional cube variables and the number (**n+2**) of rounds,

  - What is the minimum number (q) of ordinary cube variables to guarantee that after **n+2** rounds there is no (p+q)-degree term, $v_1 v_2 \cdots v_p u_1 u_2 \cdots u_q$?

Example #1: p=1, n=1

- Given 1 conditional cube variables and the number (**1+2**) of rounds,

  - What is the minimum number (q) of ordinary cube variables to guarantee that after **1+2** rounds there is no (1+q)-degree term, $v_1 u_1 u_2 \cdots u_q$?

Example #1: p=1, n=1

- Assume that there are 1 conditional cube variable v_1 and q ordinary cube variables, $u_1,u_2,\ldots,u_q$, and the conditions are true.

Example #1: p=1, n=1

- Assume that there are 1 conditional cube variable v_1 and q ordinary cube variables, $u_1, u_2, \ldots, u_q$, and the conditions are true.
- After two rounds, these are all possible terms of degree 1 or 2. (in case of Keccak permutation)
  - Degree 1: v_1, u_1, $u_2, \ldots, u_q$
  - Degree 2: u_1u_2, u_1u_3, $u_2u_3, \ldots, u_{q-1}u_q$

Example #1: p=1, n=1

- Assume that there are 1 conditional cube variable v_1 and q ordinary cube variables, $u_1, u_2, \ldots, u_q$, and the conditions are true.
- After two rounds, these are all possible terms of degree 1 or 2. (in case of Keccak permutation)
  - Degree 1: v_1, u_1, $u_2, \ldots, u_q$
  - Degree 2: u_1u_2, u_1u_3, $u_2u_3, \ldots, u_{q-1}u_q$

- After 1 (n=1) more round,

Example #1: p=1, n=1

- Assume that there are 1 conditional cube variable v_1 and q ordinary cube variables, $u_1, u_2, \ldots, u_q$, and the conditions are true.
- After two rounds, these are all possible terms of degree 1 or 2. (in case of Keccak permutation)
  - Degree 1: v_1, u_1, $u_2, \ldots, u_q$
  - Degree 2: u_1u_2, u_1u_3, $u_2u_3, \ldots, u_{q-1}u_q$
- After 1 (n=1) more round,

By choosing 2 terms including 1 term v_1, the possible maximum degree is 3.

Example #1: p=1, n=1

- Assume that there are 1 conditional cube variable v_1 and q ordinary cube variables, $u_1, u_2, \ldots, u_q$, and the conditions are true.
- After two rounds, these are all possible terms of degree 1 or 2. (in case of Keccak permutation)
  - Degree 1: v_1, u_1, $u_2, \ldots, u_q$
  - Degree 2: u_1u_2, u_1u_3, $u_2u_3, \ldots, u_{q-1}u_q$
- After 1 (n=1) more round,
  - $\nexists$ $v_1u_1u_2u_3$

By choosing 2 terms including 1 term v_1, the possible maximum degree is 3.

Example #1: p=1, n=1

- Assume that there are 1 conditional cube variable v_1 and q ordinary cube variables, $u_1, u_2, \ldots, u_q$, and the conditions are true.
- After two rounds, these are all possible terms of degree 1 or 2. (in case of Keccak permutation)
  - Degree 1: v_1, u_1, $u_2, \ldots, u_q$
  - Degree 2: u_1u_2, u_1u_3, $u_2u_3, \ldots, u_{q-1}u_q$

- After 1 (n=1) more round,
  - $\nexists$ $v_1u_1u_2u_3$

By choosing 2 terms including 1 term v_1, the possible maximum degree is 3.
Therefore, the min of q is 3 such that $\nexists$ $v_1u_1u_2 \ldots u_q$

Example #2: p=2, n=2

- Given 2 conditional cube variables and the number (**2+2**) of rounds,

  - What is the minimum number (q) of ordinary cube variables to guarantee that after **2+2** rounds there is no (2+q)-degree term, $v_1v_2u_1u_2\cdots u_q$?

Example #2: p=2, n=2

- Assume that there are 2 conditional cube variables v_1,v_2 and q ordinary cube variables, u_1,u_2,…,u_q, and the conditions are true.

Example #2: p=2, n=2

- Assume that there are 2 conditional cube variables v_1, v_2 and q ordinary cube variables, $u_1, u_2, \ldots, u_q$, and the conditions are true.
- After two rounds, these are all possible terms of degree 1 or 2. (in case of Keccak permutation)
  - Degree 1: $v_1, v_2, u_1, u_2, \ldots, u_q$
  - Degree 2: $u_1u_2, u_1u_3, u_2u_3, \ldots, u_{q-1}u_q$

Example #2: p=2, n=2

- Assume that there are 2 conditional cube variables v_1, v_2 and q ordinary cube variables, $u_1, u_2, \ldots, u_q$, and the conditions are true.
- After two rounds, these are all possible terms of degree 1 or 2. (in case of Keccak permutation)
  - Degree 1: $v_1, v_2, u_1, u_2, \ldots, u_q$
  - Degree 2: $u_1u_2, u_1u_3, u_2u_3, \ldots, u_{q-1}u_q$

- After 2 (n=2) more rounds,

Example #2: p=2, n=2

- Assume that there are 2 conditional cube variables v_1, v_2 and q ordinary cube variables, $u_1, u_2, \ldots, u_q$, and the conditions are true.
- After two rounds, these are all possible terms of degree 1 or 2. (in case of Keccak permutation)
  - Degree 1: $v_1, v_2, u_1, u_2, \ldots, u_q$
  - Degree 2: $u_1u_2, u_1u_3, u_2u_3, \ldots, u_{q-1}u_q$
- After 2 (n=2) more rounds,

By choosing 4 terms including 2 terms v_1, v_2, the possible maximum degree is 6.

Example #2: p=2, n=2

- Assume that there are 2 conditional cube variables v_1, v_2 and q ordinary cube variables, $u_1, u_2, \ldots, u_q$, and the conditions are true.
- After two rounds, these are all possible terms of degree 1 or 2. (in case of Keccak permutation)
  - Degree 1: $v_1, v_2, u_1, u_2, \ldots, u_q$
  - Degree 2: $u_1u_2, u_1u_3, u_2u_3, \ldots, u_{q-1}u_q$

- After 2 (n=2) more rounds,
  - $\nexists\ v_1v_2u_1u_2u_3u_4u_5$

By choosing 4 terms including 2 terms v_1, v_2, the possible maximum degree is 6.

Example #2: p=2, n=2

- Assume that there are 2 conditional cube variables v_1, v_2 and q ordinary cube variables, $u_1, u_2, \ldots, u_q$, and the conditions are true.
- After two rounds, these are all possible terms of degree 1 or 2. (in case of Keccak permutation)
  - Degree 1: $v_1, v_2, u_1, u_2, \ldots, u_q$
  - Degree 2: $u_1u_2, u_1u_3, u_2u_3, \ldots, u_{q-1}u_q$

- After 2 (n=2) more rounds,
  - $\nexists\ v_1v_2u_1u_2u_3u_4u_5$

By choosing 4 terms including 2 terms v_1, v_2, the possible maximum degree is 6.
Therefore, the min of q is 5 such that $\nexists\ v_1v_2u_1u_2\ldots u_q$

Theorem 2 (Generalization)

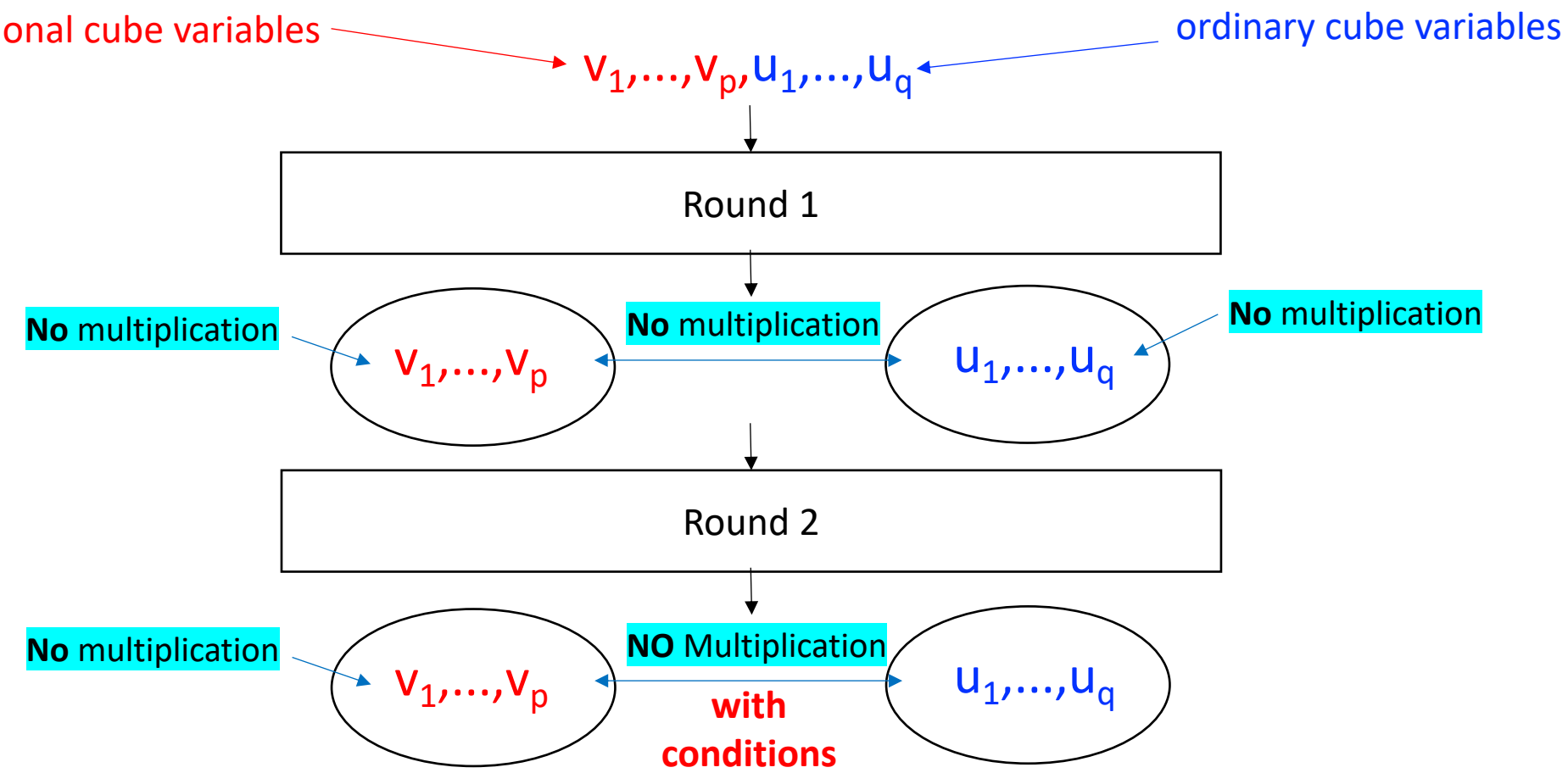
- Assume that there are p conditional cube variables $v_1,\ldots,v_p$ and q ordinary cube variables $u_1,\ldots,u_q$, where

  (1) $p,q \geq 1$ and $q=2^{n+1}-2p+1$

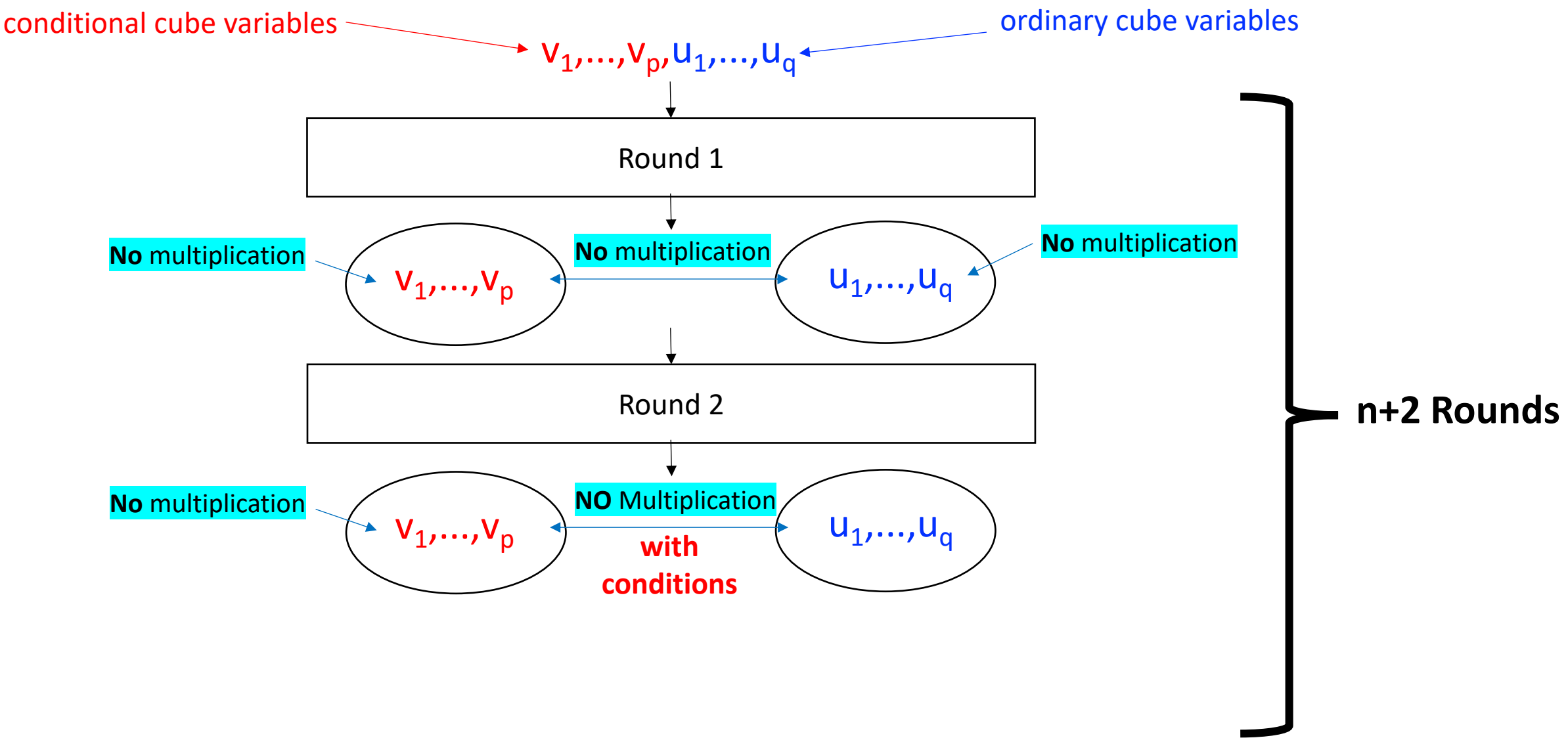
  or (2) $q=0$ and $p=2^n+1$

- If the conditions are true, then the term $v_1v_2\ldots v_pu_1u_2\ldots u_q$ will not appear in the output polynomials of **(n+2)**-round Keccak sponge function.

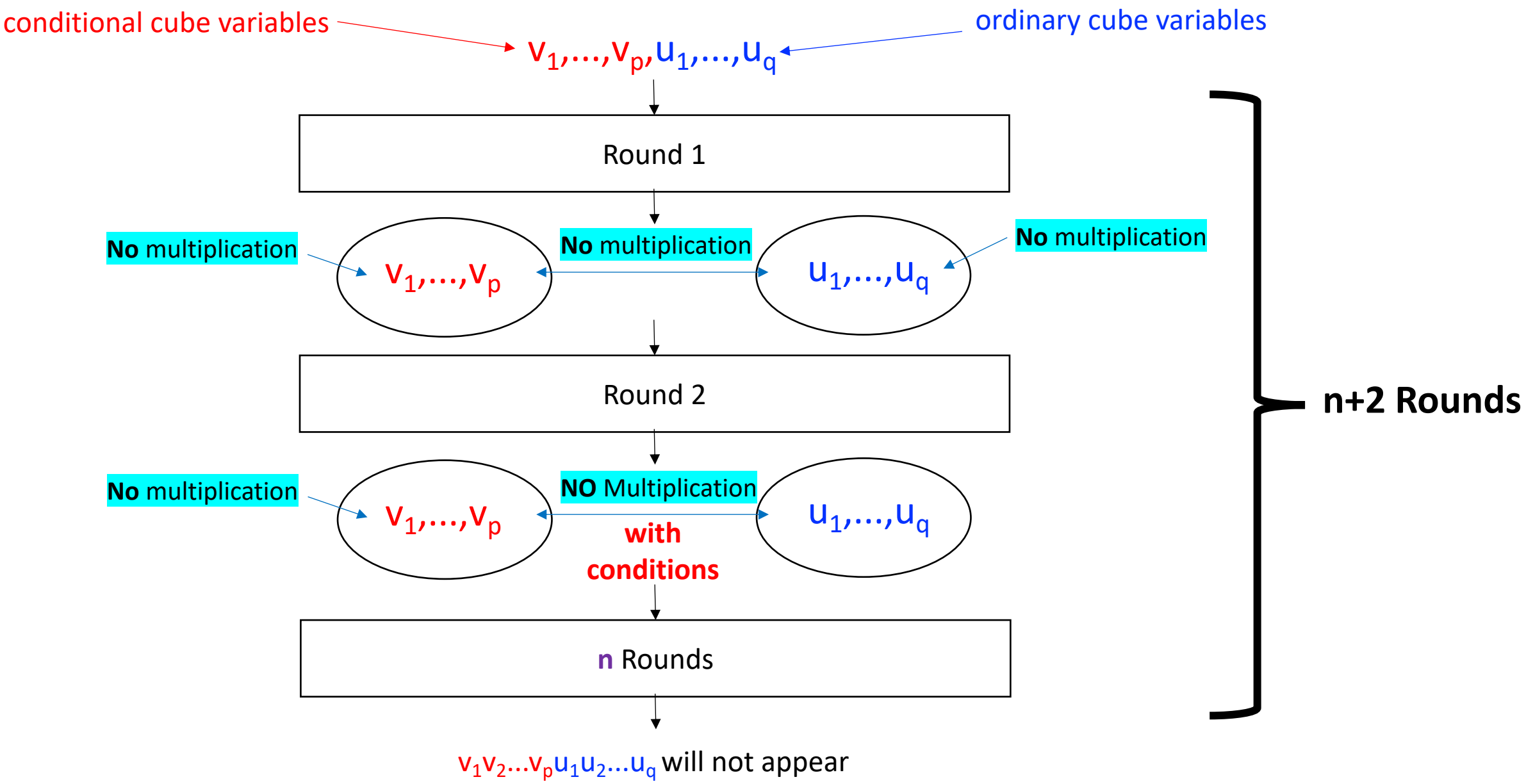
Theorem 2. ($p,q \geq 1$ and $q=2^{n+1}-2p+1$) or ($q=0$ and $p=2^n+1$)



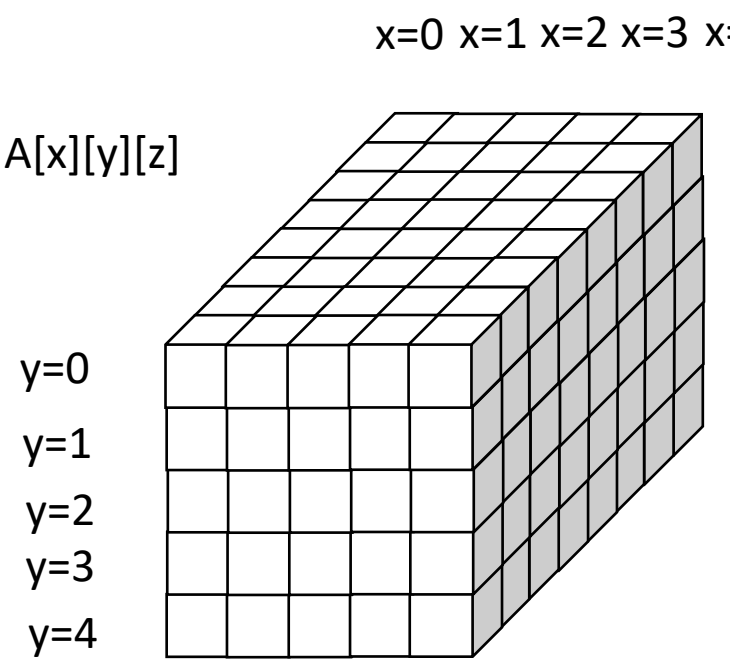
Theorem 2. ($p,q \geq 1$ and $q=2^{n+1}-2p+1$) or ($q=0$ and $p=2^n+1$)



Theorem 2. ($p,q \geq 1$ and $q=2^{n+1}-2p+1$) or ($q=0$ and $p=2^n+1$)



Keccak Round Permutation



A (1600-bit)

Round *i*

0.5 round

$\boldsymbol{\theta}$: $B[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\big(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\big)$.

$\boldsymbol{\rho}$: $C[x][y] \leftarrow B[x][y] \ggg r[x][y]$.

$\boldsymbol{\pi}$: $D[y][2x+3y] \leftarrow C[x][y]$.

0.5 round

$\boldsymbol{\chi}$: $E[x][y] \leftarrow D[x][y] \oplus ((\neg D[x+1][y]) \wedge D[x+2][y])$

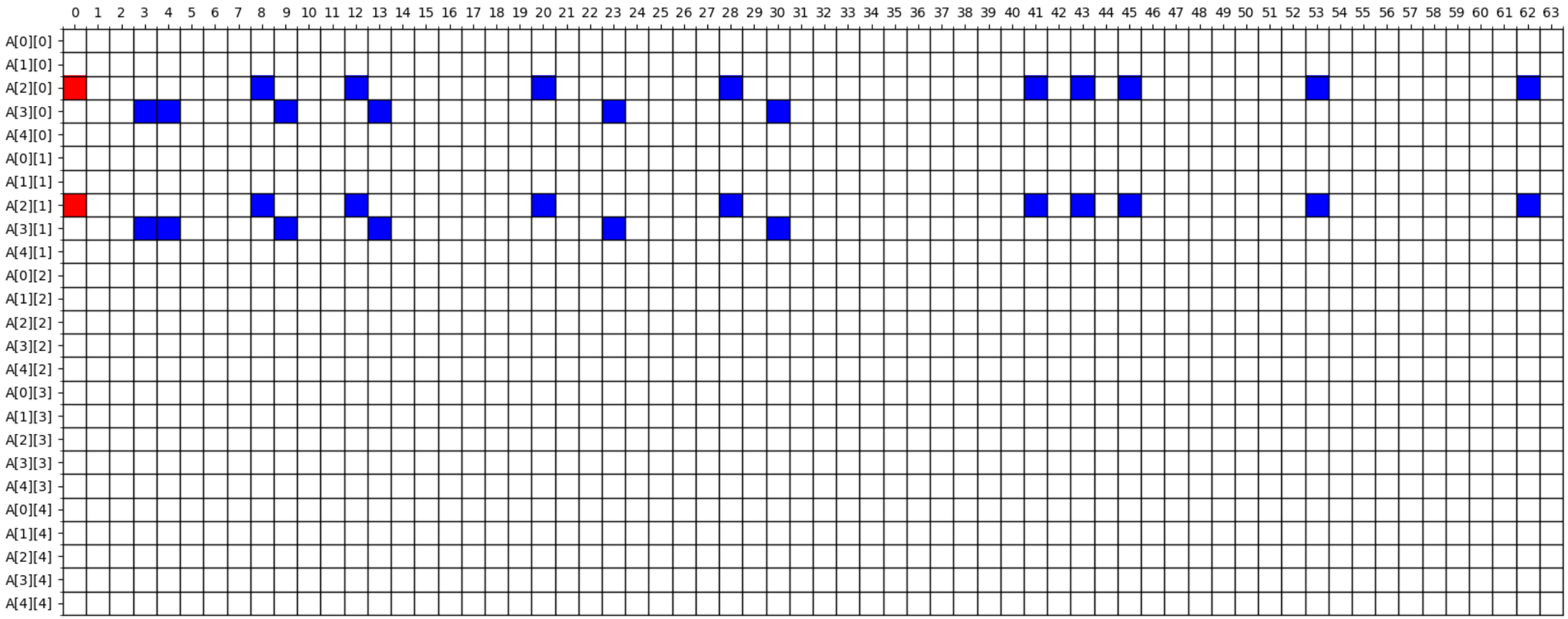
$\boldsymbol{\iota}$: $F[0][0] \leftarrow E[0][0] \oplus RC[i]$.

F (1600-bit)

Next, let us see the propagation of the remaining 15 ordinary cube variables $v_1, v_2, \ldots, v_{15}$.

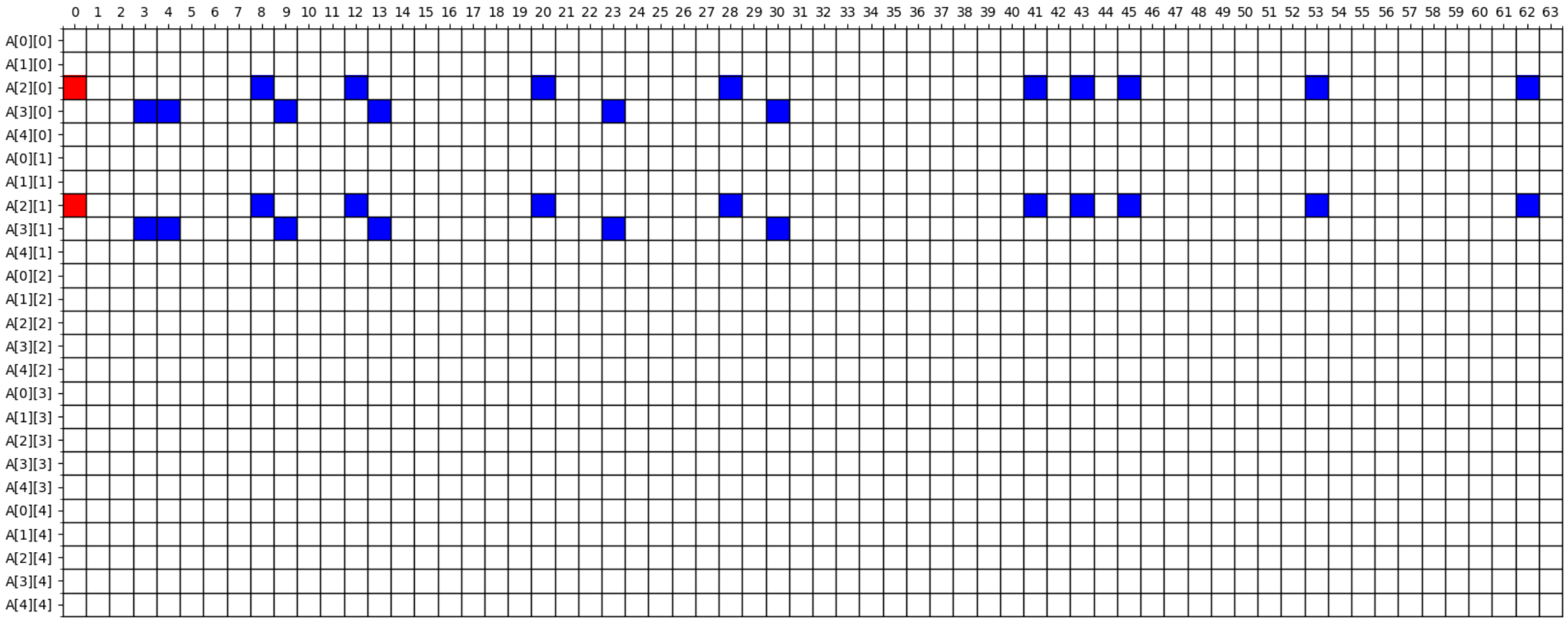
Ordinary Cube Variables	A[2][0][8]=A[2][1][8]=v_1, A[2][0][12]=A[2][1][12]=v_2, A[2][0][20]=A[2][1][20]=v_3, A[2][0][28]=A[2][1][28]=v_4, A[2][0][41]=A[2][1][41]=v_5, A[2][0][43]=A[2][1][43]=v_6, A[2][0][45]=A[2][1][45]=v_7, A[2][0][53]=A[2][1][53]=v_8, A[2][0][62]=A[2][1][62]=v_9, A[3][0][3]=A[3][1][3]=v_{10}, A[3][0][4]=A[3][1][4]=v_{11}, A[3][0][9]=A[3][1][9]=v_{12}, A[3][0][13]=A[3][1][13]=v_{13}, A[3][0][23]=A[3][1][23]=v_{14}, A[3][0][30]=A[3][1][30]=v_{15}
Conditional Cube Variables	A[2][0][0]=A[2][1][0]=v_0
Bit Conditions	A[4][0][44]=0, A[2][0][4]=k_5+k_{69}+A[0][1][5]+A[2][1][4]+1, A[2][0][59]=k_{60}+A[0][1][60]+A[2][1][59]+1, A[2][0][7]=A[4][0][6]+A[2][1][7]+A[3][1][7]
Guessed Key Bits	k_{60}, k_5+k_{69}

Initial State



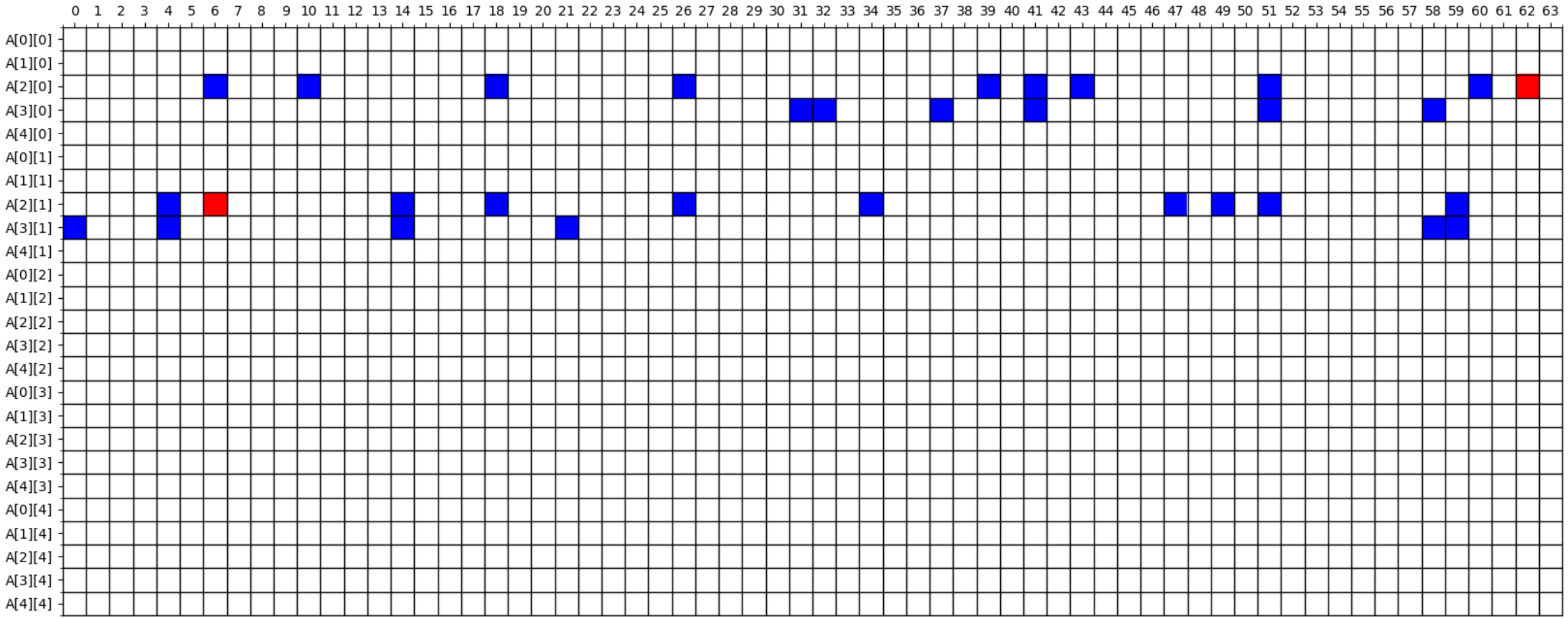
ROUND 1

$$\boldsymbol{\theta}: A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\left(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\right).$$



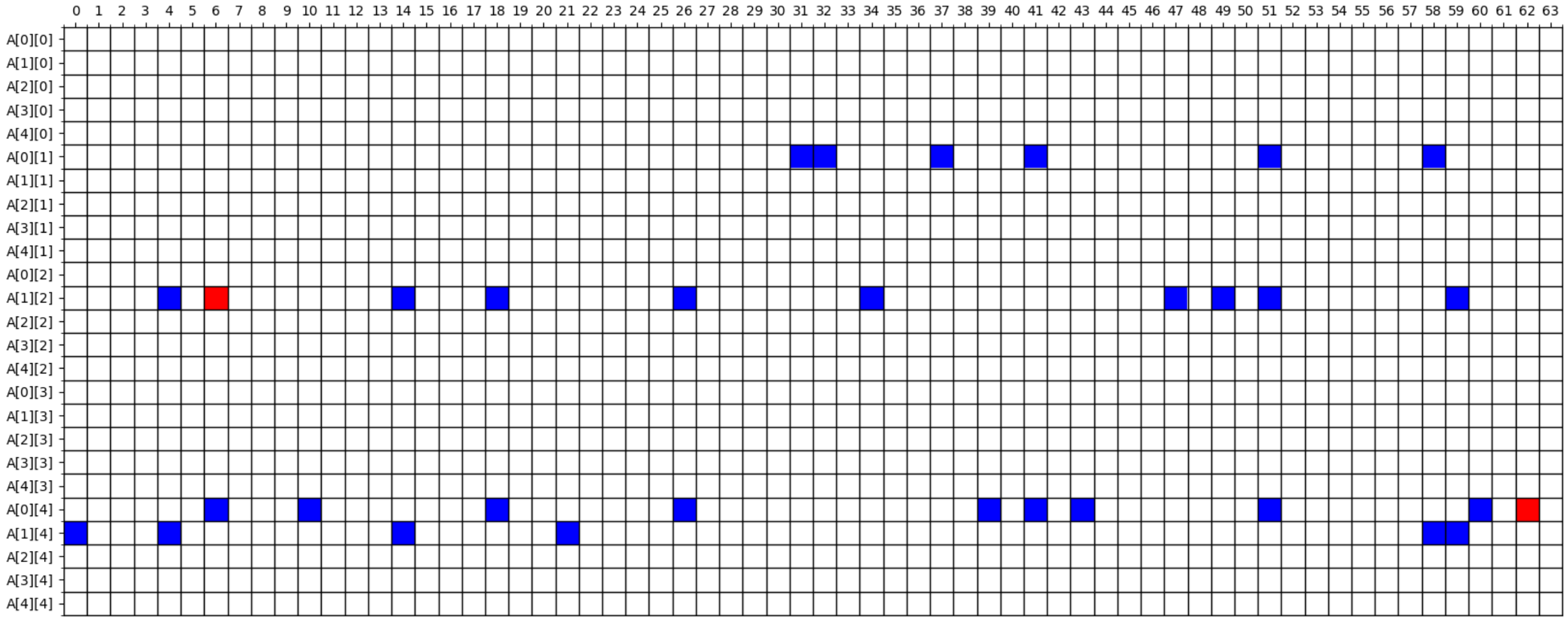
ROUND 1

$$\boldsymbol{\rho}: A[x][y] \leftarrow A[x][y] \ggg r[x][y].$$



ROUND 1

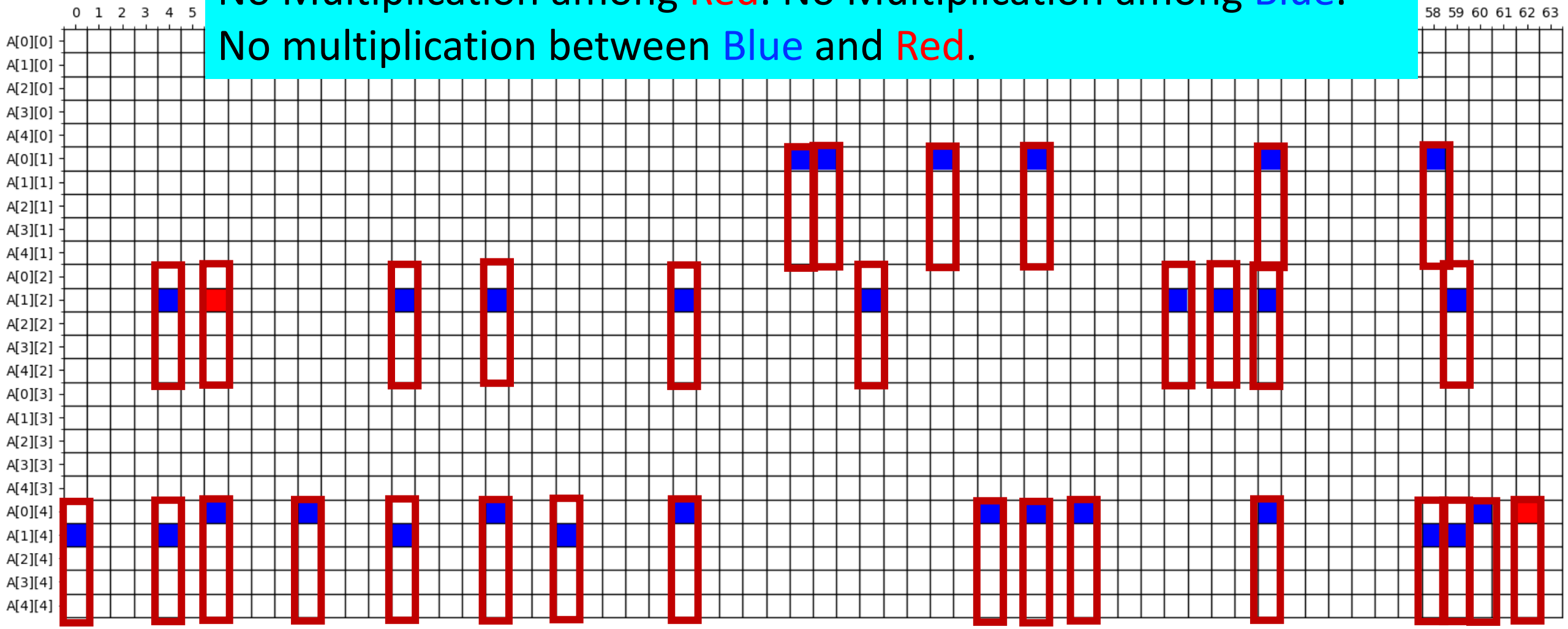
$$\pi: A[y][2x + 3y] \leftarrow A[x][y].$$



ROUND 1

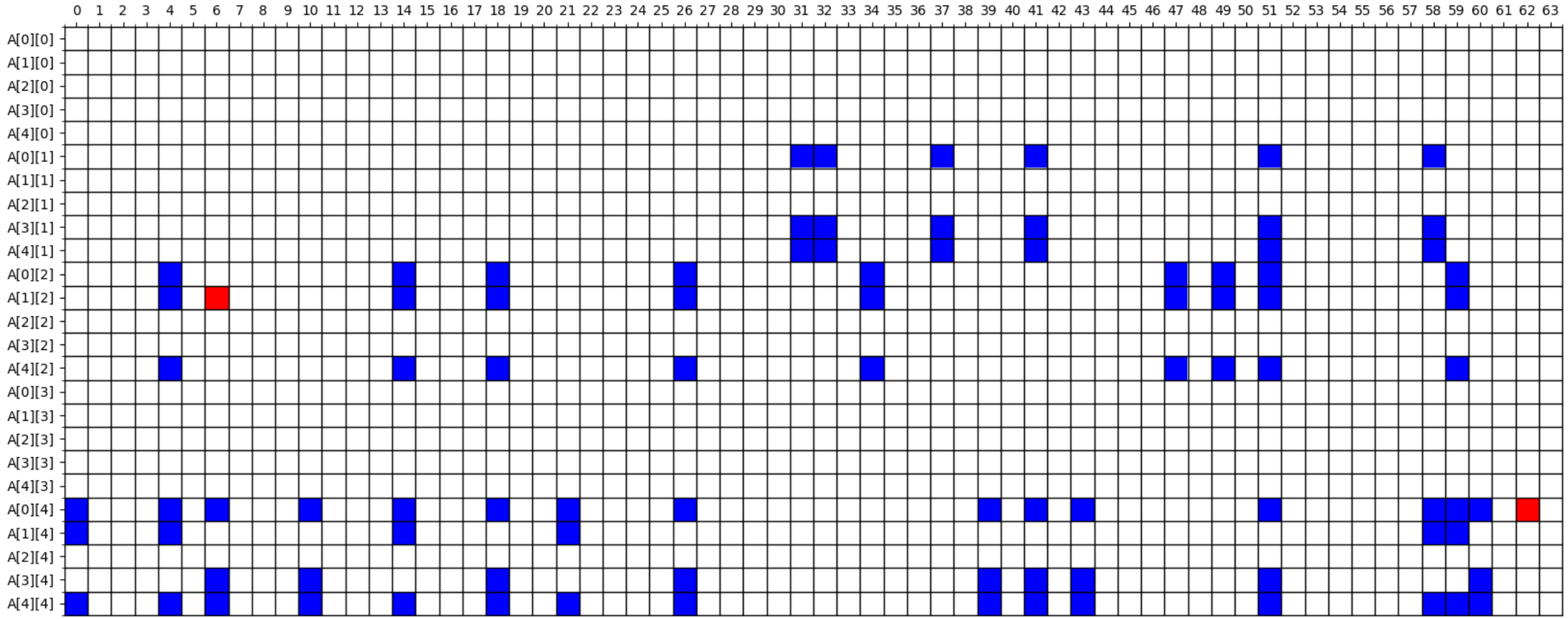
$\pi: A[y][2x + 3y] \leftarrow A[x][y].$

No Multiplication among Red. No Multiplication among Blue.
No multiplication between Blue and Red.



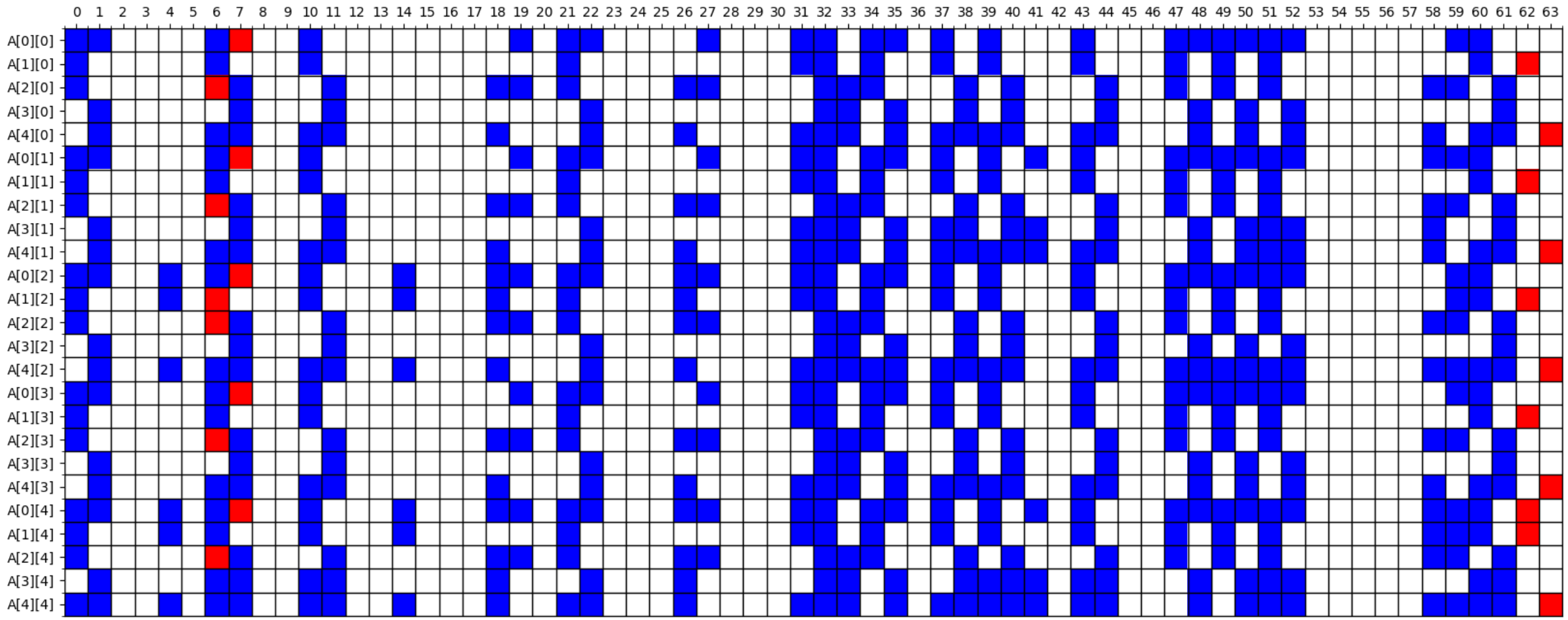
ROUND 1

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



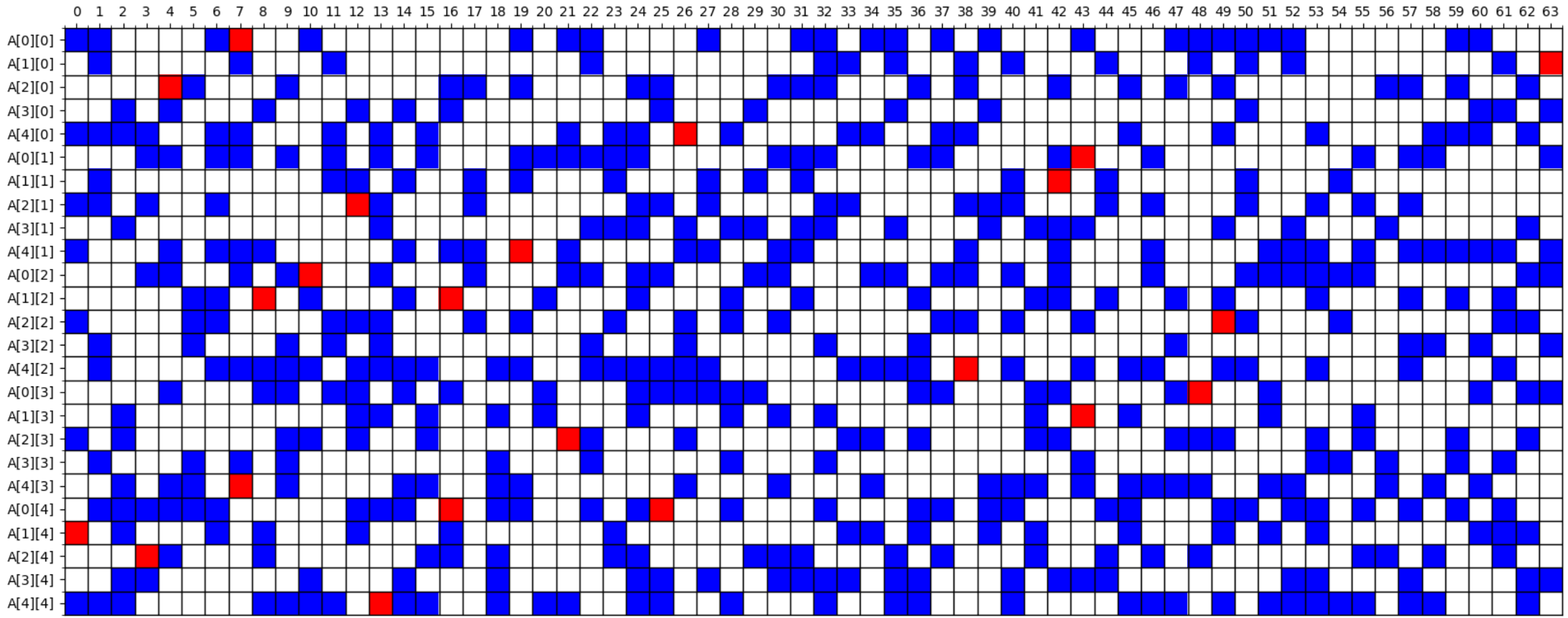
ROUND 2

$$\boldsymbol{\theta}: A[x][y] \leftarrow A[x][y] \oplus \sum_{j=0}^{4}\left(A[x-1][j] \oplus (A[x+1][j] \ggg 1)\right).$$



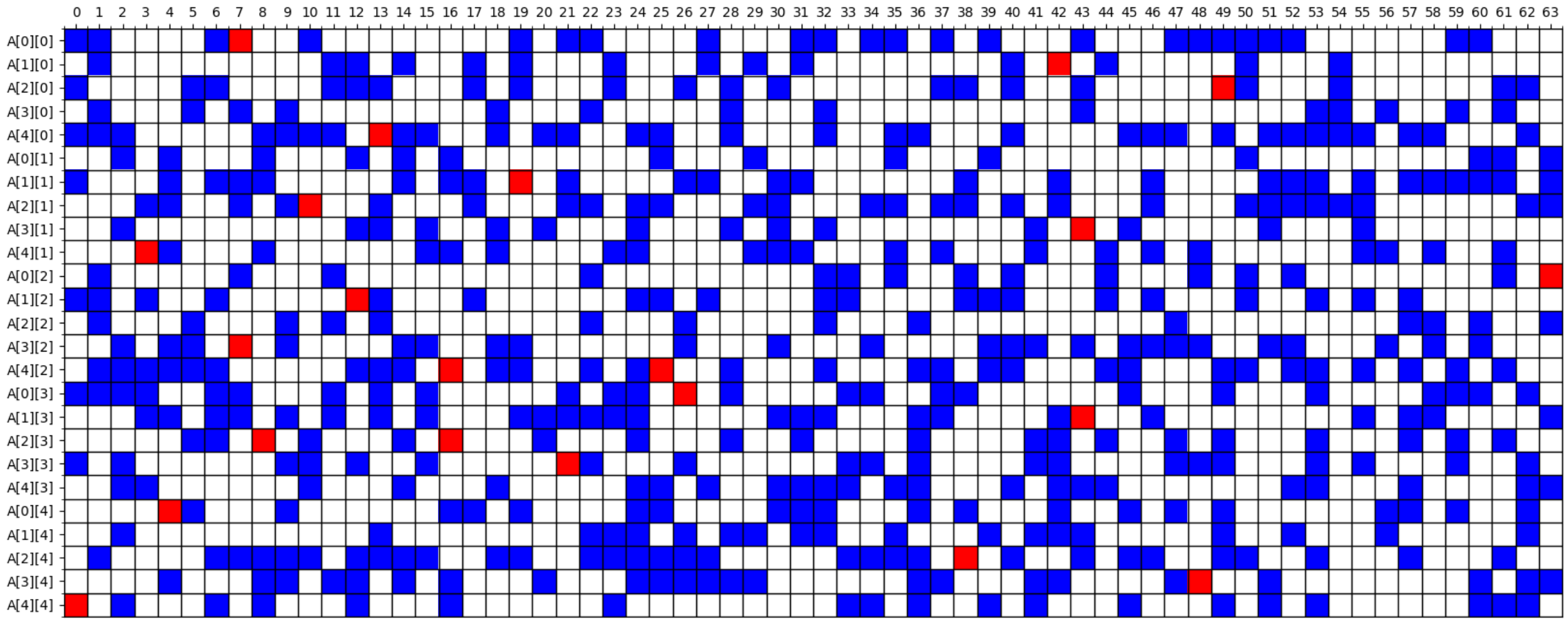
ROUND 2

$$\boldsymbol{\rho}: A[x][y] \leftarrow A[x][y] \ggg r[x][y].$$



ROUND 2

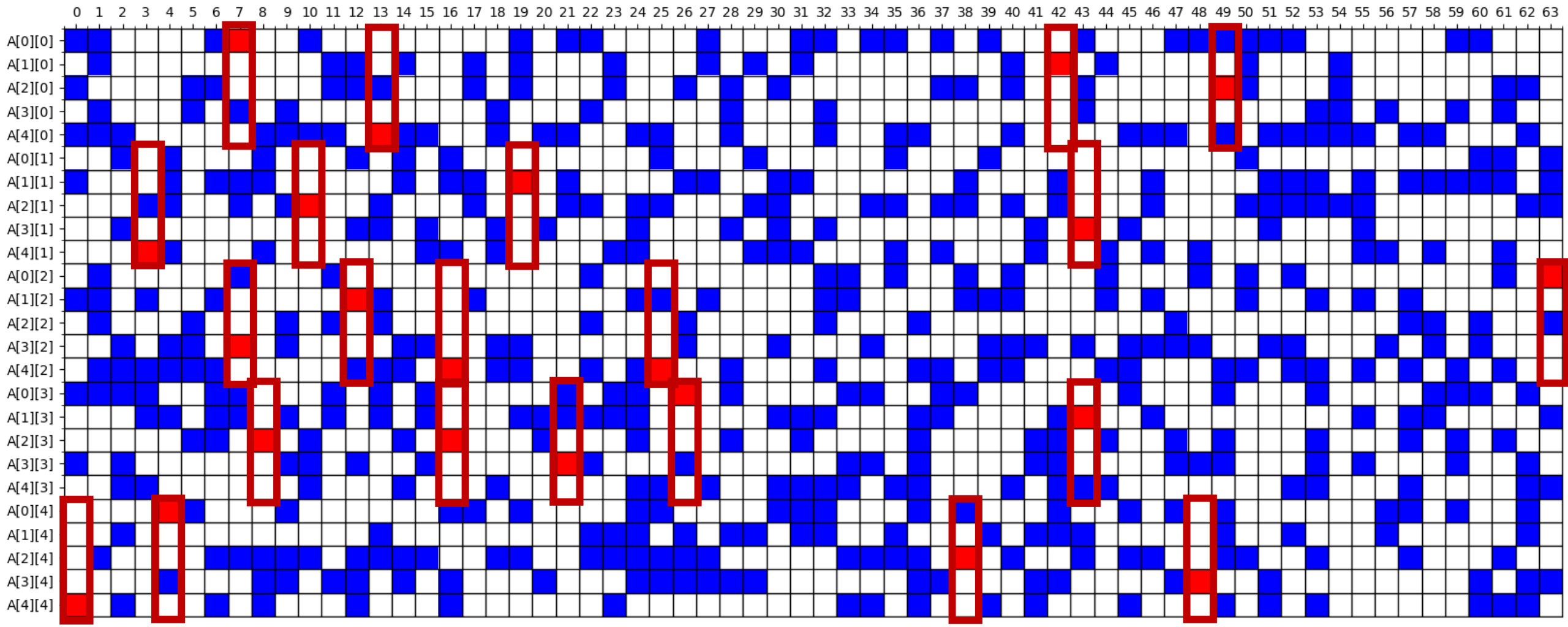
$$\pi: A[y][2x + 3y] \leftarrow A[x][y].$$



ROUND 2

$$\pi: A[y][2x + 3y] \leftarrow A[x][y].$$

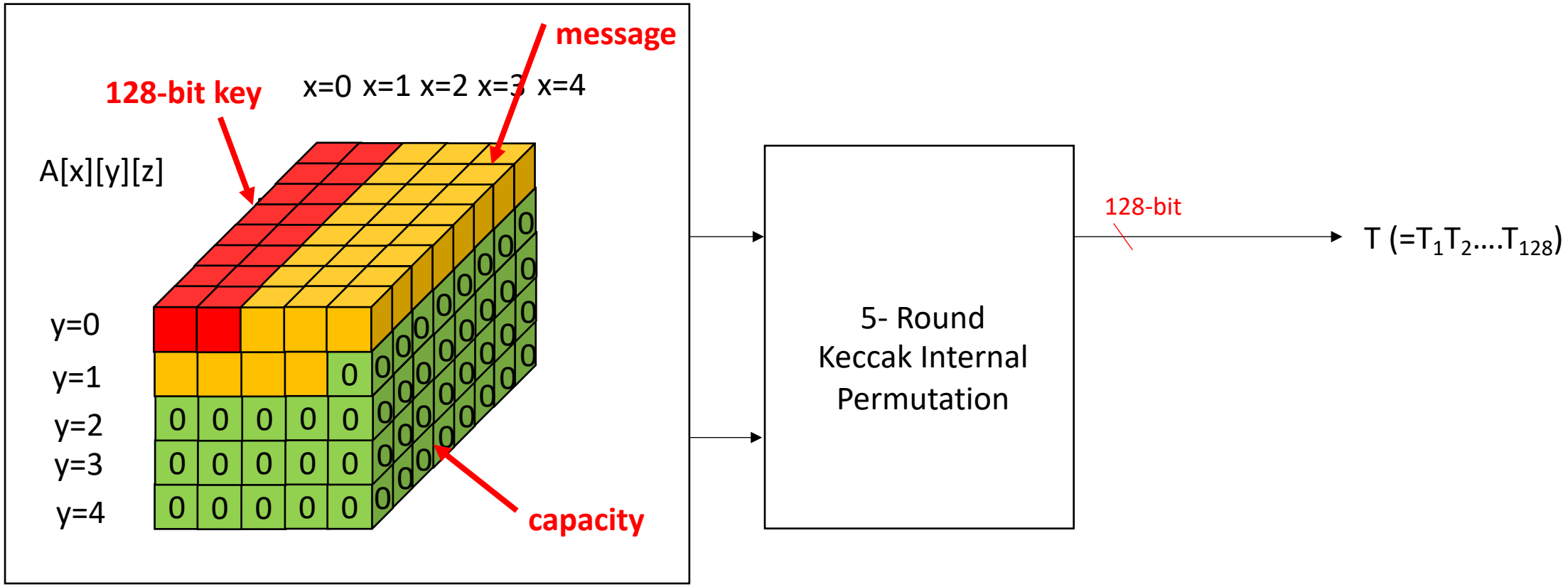
No multiplication between Blue and Red. No Multiplication among Red.



Therefore, according to Theorem 2,

- If the conditions are true, then the term $v_0v_1v_2...v_{15}$ will not appear in the output polynomials of **5**-round Keccak sponge function.
- By the cube-sum, we can determine whether the conditions (with some secret key bits) hold or not.

Setting of Input message M



Setting of Input message M

- These are the 4 conditions.

  ① A[4][0][44]=0

  ② A[2][0][4]=k_5+k_{69}+A[0][1][5]+A[2][1][4]+1

  ③ A[2][0][59]=k_{60}+A[0][1][60]+A[2][1][59]+1

  ④ A[2][0][7]=A[4][0][6]+A[2][1][7]+A[3][1][7]

- For each $(k_5+k_{69},k_{60})\in\{(0,0),(0,1),(1,0),(1,1)\}$,
  - According to the conditions, we choose A[4][0][44], A[2][0][4], A[0][1][5], A[2][1][4],A[2][0][59],A[0][1][60],A[2][1][59], A[2][0][7],A[4][0][6],A[2][1][7],A[3][1][7].
  - We have 16 cube variables in message part.
  - We arbitrary choose values of remaining message part.
  - Let us call this as input message **M**.

Online Phase of Attack on 5-Round Keccak-MAC-512

- For each $(k_5+k_{69},k_{60})\in\{(0,0),(0,1),(1,0),(1,1)\}$,
  - Define message M as explained in the previous slide.

  Attacker makes oracle $\mathrm{Keccak}-\mathrm{MAC}(K,\cdot)$-queries and compute the following online cube-sum:

$$\sum_{(v_0,v_1\ldots,v_{15})\in C_{v_0v_1\ldots v_{15}}} \mathrm{Keccak}{-}MAC(K,\mathbf{M}) \stackrel{?}{=} 0^{128}$$

  - Above cube-sum requires 2^{16} $\mathrm{Keccak}-\mathrm{MAC}$-oracle queries.
  - If the cube-sum is all zero, then the guessing (k_5+k_{69},k_{60}) is correct.
  - Otherwise, we continue to perform another cube-sum with different guessing of (k_5+k_{69},k_{60}).
- In total, in order to recover two-bit of key, (k_5+k_{69},k_{60}), we need 2^{18} $\mathrm{Keccak}-\mathrm{MAC}$-oracle queries.

Online Phase of Attack on 5-Round Keccak-MAC-512

- Till now, we showed only how to recover 2-bit of key, (k_5+k_{69},k_{60}).
- How can we find another key information? We consider any i ($0 \leq i \leq 63$) as follows: (This table is correct for all i's, which is verified by me and attached at the end of this talk.)

Ordinary Cube Variables	A[2][0][i+8]=A[2][1][i+8]=v_1, A[2][0][i+12]=A[2][1][i+12]=v_2, A[2][0][i+20]=A[2][1][i+20]=v_3, A[2][0][i+28]=A[2][1][i+28]=v_4, A[2][0][i+41]=A[2][1][i+41]=v_5,A[2][0][i+43]=A[2][1][i+43]=v_6, A[2][0][i+45]=A[2][1][i+45]=v_7, A[2][0][i+53]=A[2][1][i+53]=v_8,A[2][0][i+62]=A[2][1][i+62]=v_9, A[3][0][i+3]=A[3][1][i+3]=v_{10}, A[3][0][i+4]=A[3][1][i+4]=v_{11}, A[3][0][i+9]=A[3][1][i+9]=v_{12}, A[3][0][i+13]=A[3][1][i+13]=v_{13},A[3][0][i+23]=A[3][1][i+23]=v_{14},A[3][0][i+30]=A[3][1][i+30]=v_{15}
Conditional Cube Variables	A[2][0][i]=A[2][1][i]=v_0
Bit Conditions	① A[4][0][i+44]=0, ② A[2][0][i+4]=$k_{i+5 \bmod 64}$+$k_{64+(i+5 \bmod 64)}$+A[0][1][i+5]+A[2][1][i+4]+1, ③ A[2][0][i+59]=$k_{i+60 \bmod 64}$+A[0][1][i+60]+A[2][1][i+59]+1, ④ A[2][0][i+7]=A[4][0][i+6]+A[2][1][i+7]+A[3][1][i+7]
Guessed Key Bits	$k_{i+60 \bmod 64}$, $k_{i+5 \bmod 64}$+$k_{64+(i+5 \bmod 64)}$

Setting of Input message M

- These are the 4 conditions for each i ($0 \leq i \leq 63$).

  ① A[4][0][i+44]=0

  ② A[2][0][i+4]= $k_{i+5 \bmod 64}+k_{64+(i+5 \bmod 64)}$+A[0][1][i+5]+A[2][1][i+4]+1

  ③ A[2][0][i+59]=$k_{i+60 \bmod 64}$ +A[0][1][i+60]+A[2][1][i+59]+1

  ④ A[2][0][i+7]=A[4][0][i+6]+A[2][1][i+7]+A[3][1][i+7].

- For each ($k_{i+5 \bmod 64}+k_{64+(i+5 \bmod 64)}$, $k_{i+60 \bmod 64}$)$\in$\{(0,0),(0,1),(1,0),(1,1)\},
  - According to the conditions, we choose A[4][0][i+44], A[2][0][i+4], A[0][1][i+5], A[2][1][i+4],A[2][0][i+59],A[0][1][i+60],A[2][1][i+59], A[2][0][i+7],A[4][0][i+6],A[2][1][i+7],A[3][1][i+7].
  - We have 16 cube variables in message part.
  - We arbitrary choose values of remaining message part.
  - Let us call this as input message **M**.

Online Phase of Attack on 5-Round Keccak-MAC-512

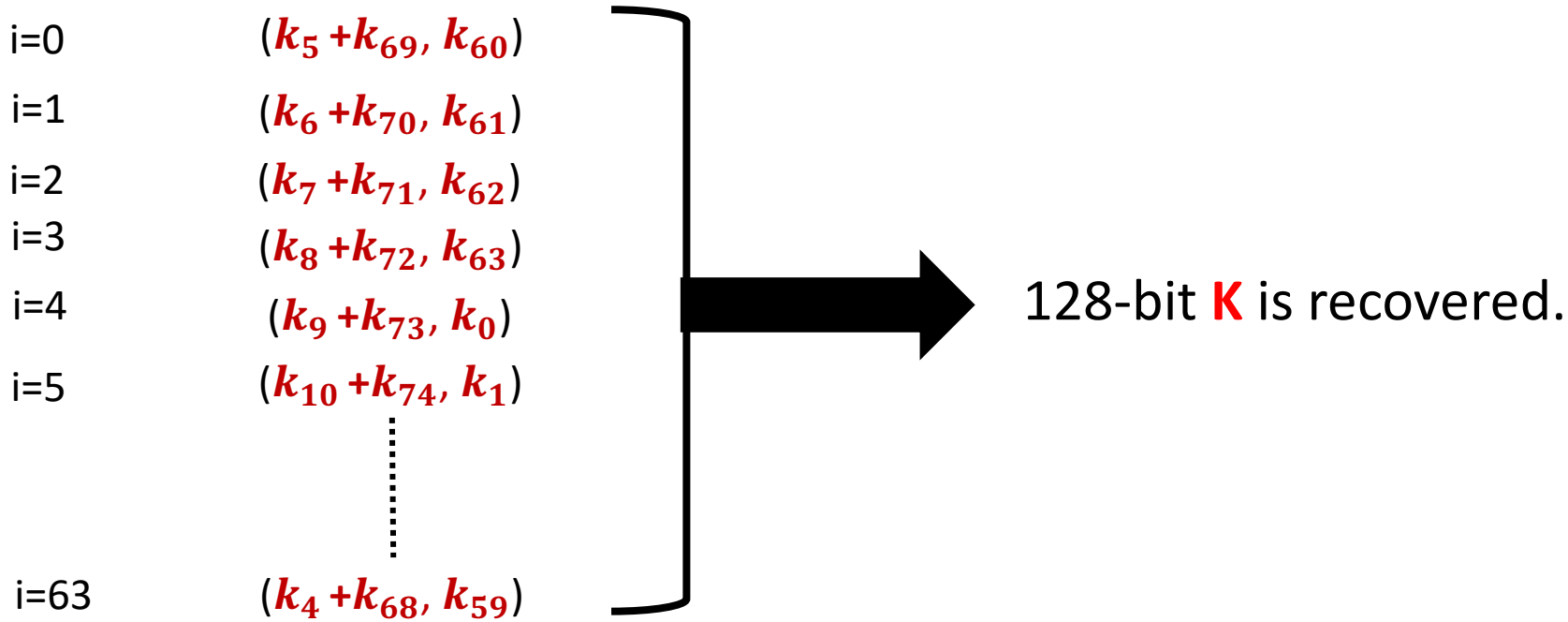
- For each $(k_{i+5 \bmod 64}+k_{64+(i+5 \bmod 64)}, k_{i+60 \bmod 64}) \in \{(0,0),(0,1),(1,0),(1,1)\}$,
  - Define message M as explained in the previous slide.
  - Attacker makes oracle $\mathrm{Keccak}-\mathrm{MAC}(K,\cdot)$-queries and compute the following online cube-sum:

$$\sum_{(v_0,v_1,\ldots,v_{15})\in C_{v_0 v_1 \ldots v_{15}}} \mathrm{Keccak}-MAC(K, M) \stackrel{?}{=} 0^{128}$$

  - Above cube-sum requires 2^{16} $\mathrm{Keccak}-\mathrm{MAC}$-oracle queries.
  - If the cube-sum is all zero, then the guessing $(k_{i+5 \bmod 64}+k_{64+(i+5 \bmod 64)}, k_{i+60 \bmod 64})$ is correct.
  - Otherwise, we continue to perform another cube-sum with different guessing of $(k_{i+5 \bmod 64}+k_{64+(i+5 \bmod 64)}, k_{i+60 \bmod 64})$.
- In total, for each i, in order to recover two-bit of key, $(k_{i+5 \bmod 64}+k_{64+(i+5 \bmod 64)}, k_{i+60 \bmod 64})$, we need 2^{18} $\mathrm{Keccak}-\mathrm{MAC}$-oracle queries.

Online Phase of Attack on 5-Round Keccak-MAC-512

- Therefore, we recovered all the 128-bit information of key **K** with 2^{24} $\mathrm{Keccak-MAC}$-oracle queries.
- where, 2^{24}=(64 i-cases)×(2-bit guess for each case)×(2^{16} for a cube-sum).



Conclusion

- We studied two major cube-attack approaches.
- I hope that you are motived to work more on cube attacks and try to analyze NIST LWC AEAD candidates using cube attacks.