清華大學
Tsinghua University

On Solving Challenges in the KECCAK Crunchy Crypto Contest

- Xiaoen Lin[1], Zhengrong Lu[1], Yantian Shen[1], Chongxu Ren[1], Hongbo Yu[1,2,3*]

- [1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
- [2]Zhongguancun Laboratory, Beijing, China
- [3]State Key Laboratory of Cryptography and Digital Economy Security, Tsinghua University, Beijing, China

- May 4, 2025

Contents

- Background
  - KECCAK
  - Crunchy Contest
- Some Results
  - KECCAK[r = 640, c = 160, n_r = 4]
    - Based on (non) linear structures (complexity around 2^{61})
  - KECCAK[r = 40, c = 160, n_r = 2]
    - Based on (non) linear structures (complexity around 2^{63})
    - Based on linear structures and symmetries (complexity around 2^{49})
  - KECCAK[r = 240, c = 160, n_r = 4]
    - Based on internal differential cryptanalysis (complexity around 2^{59})
  - KECCAK[r = 640, c = 160, n_r = 5] (unsolved yet)
    - Based on internal differential cryptanalysis (complexity around 2^{62})

The Sunway TaihuLight supercomputer

Background

- KECCAK
  - Sponge construction
  - Operate on a state of $b = r + c$ bits.
    - The state can be described as 5×5 w-bit lanes.
  - KECCAK-f permutation
    - Consist of $12 + 2log_2(w)$ rounds.
    - 5 steps for each round.
- The KECCAK Crunchy Crypto Collision and Pre-image Contest
  - b is in $\{200, 400, 800, 1600\}$.
  - $c = 160$ (output size is 80-bit for pre-image)

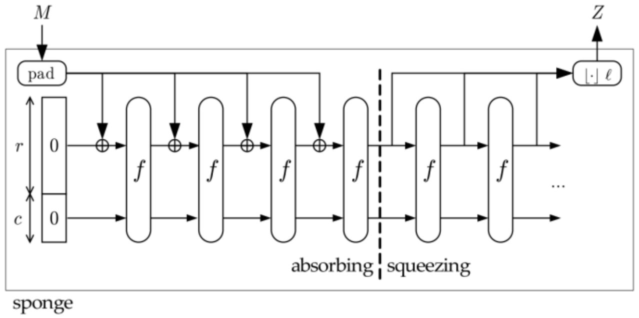


Figure 1: The sponge construction [BDPA11a].

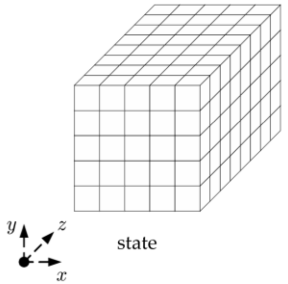


Figure 2: The Keccak-f state [BDH+b].

$$\theta : A_{x,y,z} = A_{x,y,z} \oplus \bigoplus_{i=0\sim 4}(A_{x-1,i,z} \oplus A_{x+1,i,z-1})$$
$$\rho : A_{x,y,z} = A_{x,y,(z-r_{x,y})}$$
$$\pi : A_{x,y,z} = A_{x+3y,x,z}$$
$$\chi : A_{x,y,z} = A_{x,y,z} \oplus (A_{x+1,y,z} \oplus 1) \cdot A_{x+2,y,z}$$
$$\iota : A_{0,0,z} = A_{0,0,z} \oplus RC_z$$

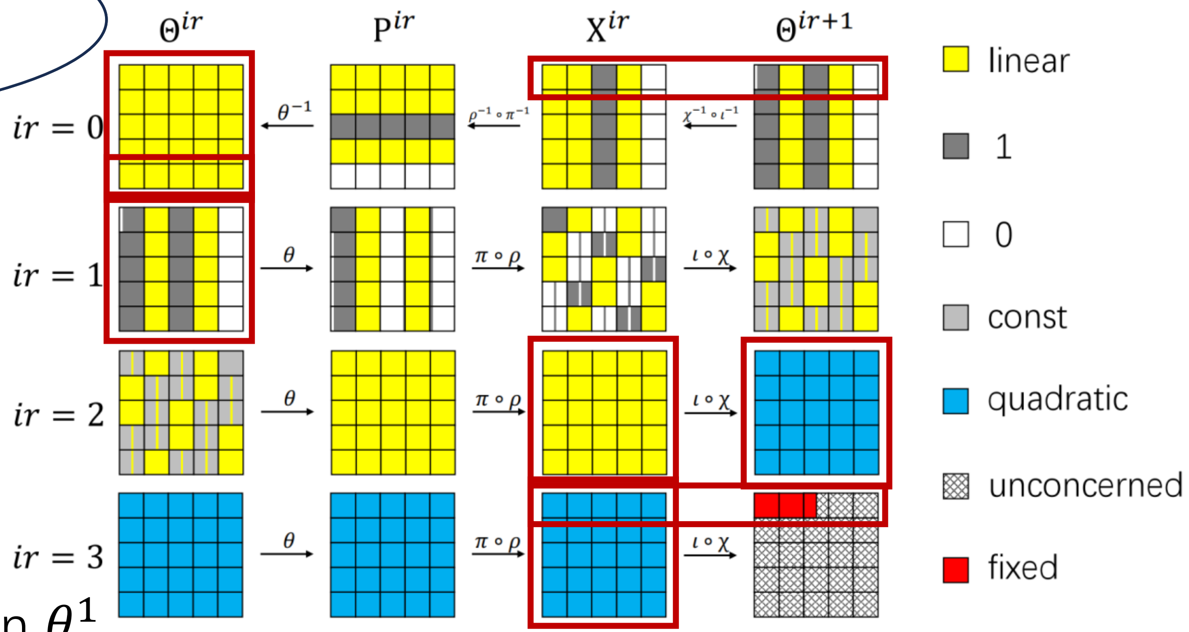
KECCAK[r = 640, c = 160, n_r = 4]

- Related previous techniques
  - linear structure [GLS16]
    - Maintain linear expressions of initial variables through rounds.
    - Enable pre-image recovery via solving linear systems repeatedly.
  - allocating approach [LS19]
    - Apply better linear structure.
    - Use two-block model with trade-off.
  - (non)linear structure [Raj19, LIMY21, WWF+21]
    - Allow quadratic bits in linear structure.
    - Solve quadratic equation systems.
  - zero coefficient [HLY21]
    - Determine column sums carefully.
    - Obtain more linear dependent bit-pairs.
  - and so on ……

Keccak[$r = 640, c = 160, n_r = 4$]

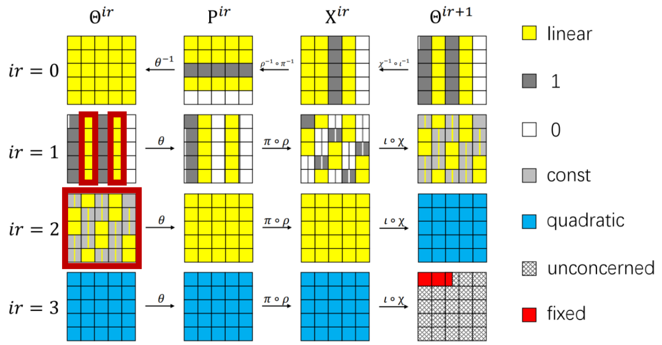
Why select these 10 lanes.
Why use these specific values for the column sums.

- Overview
  - Select 10 lanes on θ^1 as variables.
  - Backward:
    - $[x, x, 1, x+y, 0] \xleftarrow{\chi^{-1}} [1, x, 1, y, 0]$
    - Starting state is still linear.
    - Add equations to match capacity part.
  - Forward:
    - Add equations to control column sums on θ^1.
    - Add equations to restrict some linear bits on X^2 constant bits.
    - Some bits on θ^3 will be linearized, so that some bits on X^3 will become linear.
    - Add equations to promote the probability of matching the digest.

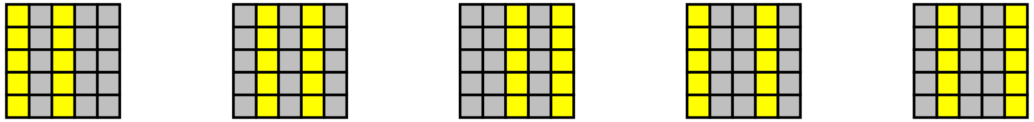


How to determine the bits and equations for linearization?

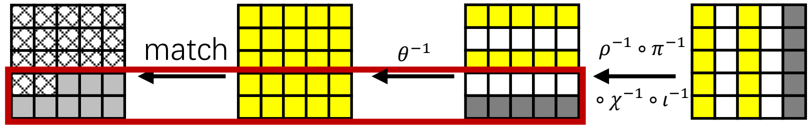
KECCAK[r = 640, c = 160, n_r = 4]



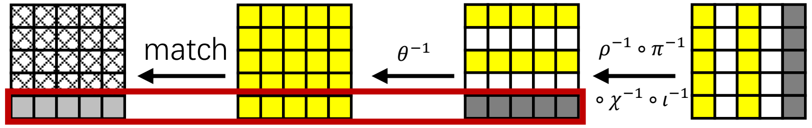
- Why selecting these 10 lanes on θ^1 as variables?
  - There are 5 choices.



  - Previous attacks on round-reduced KECCAK-224/256 select the first type.
    - Many extra matching gains

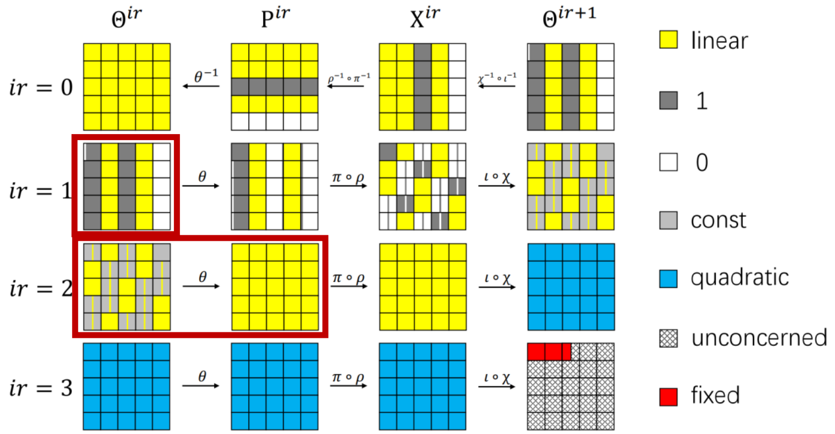


  - Only 2^1 (for padding) extra matching gains for KECCAK[r=640, c=160].

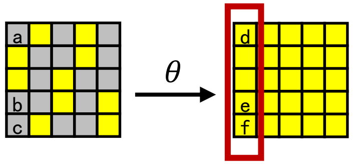


  - Different choices affect the distribution of linear/constant bits on θ^2.
    - Select the best choice for subsequent linearization according to given digest.

KECCAK[r = 640, c = 160, n_r = 4]

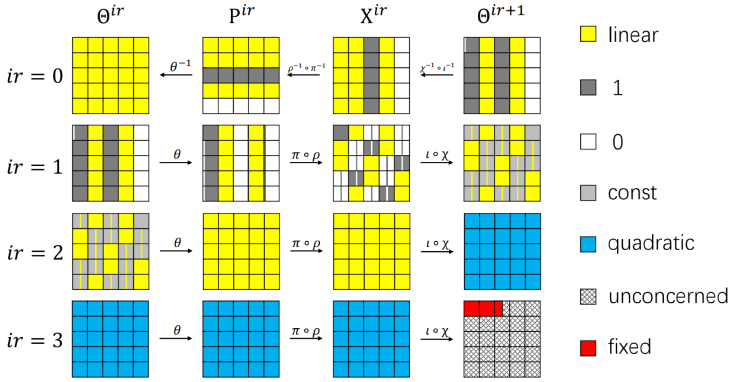


- The specific column sums on θ^1
  - The property for θ operation
    - $a \oplus b = const = d \oplus e$
    - $a \oplus c = const = d \oplus f$



  - When restrict d a constant bit, e and f will be constant simultaneously.
    - Try to make constant bits on θ^2 as many as possible.
  - Use specific column sums on θ^1 so that there are 3 constant bits for most rows on θ^2.
    - $[x, 0, y, 0, 1] \xrightarrow{\chi} [x + y, 0, y + 1, 0, 1]$
    - Accordingly, there are 3 constant bits for most columns on θ^2
  - Restrict less bits on X^2 and linearize more bits on θ^3.

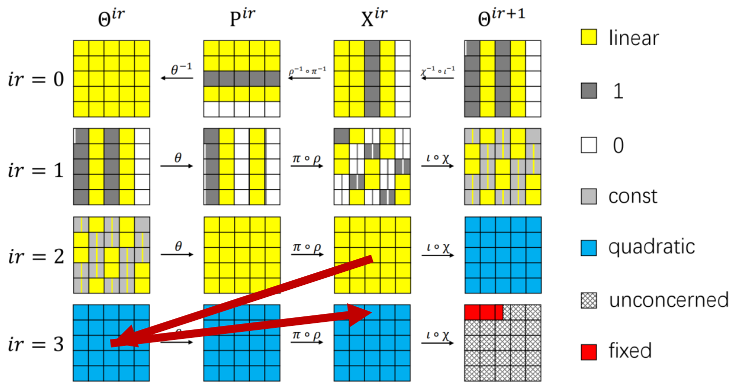
KECCAK[r = 640, c = 160, n_r = 4]



- How to determine the strategy of linearizing the last two rounds?
  - The linearization
    - Remain $320 - 63 - 1 - 158 = 98$ degrees of freedom on X^2.
      - $\boxed{63}$ for column sums on θ^1 (-1 because inherent linear dependence)
      - 1 for padding rule
      - $\boxed{158}$ for matching starting state (-1 because inherent linear dependence, another -1 because two capacity candidates)
    - Spend some to restrict some linear bits on X^2 constant bits.
      - Every equation can restrict three bits on X^2 constant bits.
    - Some bits on X^3 can be linearized.
      - Every constant bit on X^2 will linearize two bits on θ^3.
      - Every specific 11 linearized bits on θ^3 will linearize a bit on X^3.
    - Equations can be added on X^3 to promote probability of digest matching.
      - Precompute a table of optimal matching probabilities under various conditions.
  - MILP model.

KECCAK[$r = 640, c = 160, n_r = 4$]

- Details of MILP model (first part, linearize last two rounds)
  - Use $800 + 800 + 160 = 1760$ Boolean variables.
    - Whether add an equation on $X^2_{x,y,z}$.
    - Whether the bit $\theta^3_{x,y,z}$ is linearized.
    - Whether the bit $X^3_{x,0,z}$ is linearized.



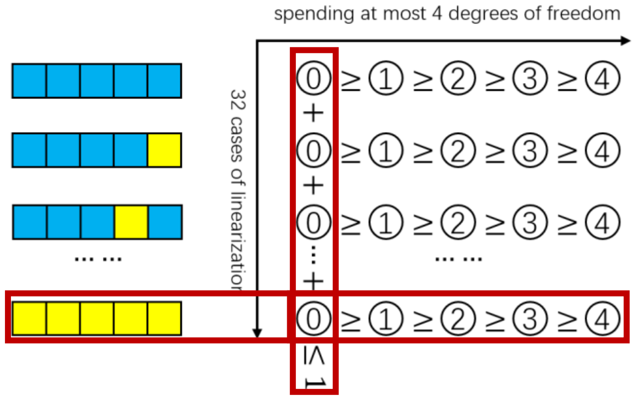
  - Add 800 equations for θ^3.
    - For example, $a_{\Theta^3_{4,0,0}} \le a_{X^2_{0,0,0}} + a_{X^2_{1,0,0}} + a_{X^2_{4,2,18}} + a_{X^2_{3,4,9}} + a_{X^2_{3,1,1}} + a_{X^2_{2,3,30}}$ means: $\theta^3_{4,0,0}$ will be linearized when an equation is added to restrict any of these six bits on X^2.
  - Add $11 \times 160 = 1760$ equations for X^3.
    - For example,

$a_{X^3_{0,0,0}} \le a_{\Theta^3_{0,0,0}}$	$a_{X^3_{0,0,0}} \le a_{\Theta^3_{1,0,31}}$	$a_{X^3_{0,0,0}} \le a_{\Theta^3_{4,0,0}}$	$a_{X^3_{0,0,0}} \le a_{\Theta^3_{1,1,31}}$
$a_{X^3_{0,0,0}} \le a_{\Theta^3_{4,1,0}}$	$a_{X^3_{0,0,0}} \le a_{\Theta^3_{1,2,31}}$	$a_{X^3_{0,0,0}} \le a_{\Theta^3_{4,2,0}}$	$a_{X^3_{0,0,0}} \le a_{\Theta^3_{1,3,31}}$
$a_{X^3_{0,0,0}} \le a_{\Theta^3_{4,3,0}}$	$a_{X^3_{0,0,0}} \le a_{\Theta^3_{1,4,31}}$	$a_{X^3_{0,0,0}} \le a_{\Theta^3_{4,4,0}}$	

means: $X^3_{0,0,0}$ will be linearized when the 11 bits on θ^3 are all linearized.

KECCAK[r = 640, c = 160, n_r = 4]

- Details of MILP model (second part, modeling an S-box (a row))
  - Use $32 \times 5 = 160$ Boolean variables (for z^{th} row).
    - Transform to 32 cases. For example:
      - Case $22 = (10110)_2$ means 2^{nd}, 3^{rd}, and 5^{th} bits are linearized.
    - 5 variables for each case.
      - The first k variables are 1 means adding $(k-1)$ equations.
  - Add an equation: $\Sigma_{i=0\sim 31} a^z_{i,0} \leq 1$
    - The linearization circumstance must belong to only one case.
  - For each case (32 cases in total):
    - Add at most 5 equations (Case $22 = (10110)_2$ for example):
      - May belong to case 22 when the three bits on X^3 are all linearized.
    - Add 4 equations: $a^z_{i,1} \leq a^z_{i,0}$ $a^z_{i,2} \leq a^z_{i,1}$ $a^z_{i,3} \leq a^z_{i,2}$ $a^z_{i,4} \leq a^z_{i,3}$
      - The restrictions will be added one by one.



$a^z_{22,0} \leq a_{X^3_{1,0,z}}$

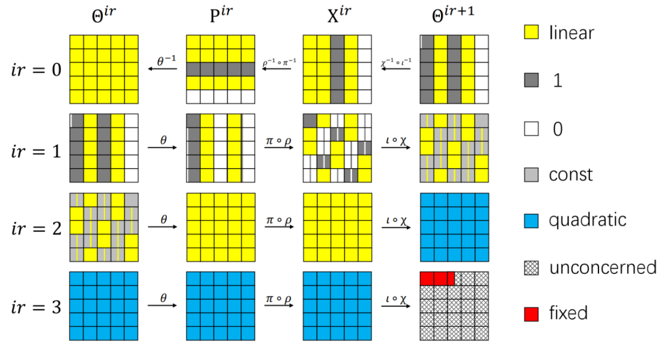
$a^z_{22,0} \leq a_{X^3_{2,0,z}}$

$a^z_{22,0} \leq a_{X^3_{4,0,z}}$

KECCAK[r = 640, c = 160, n_r = 4]

- Details of MILP model (third part, global constraint and objective)
  - Add an equation: $(\Sigma_{x=0\sim4,y=0\sim4,z=0\sim31} a_{X^2_{x,y,z}}) + (\Sigma_{i=0\sim31,j=0\sim4,z=0\sim31} a^z_{i,j}) + 8 \leq 98$
    - The number of used degrees of freedom is limited by 98.
    - Leave around 7 degrees of freedom because quadratic equation system with 34 equations on 7 variables can be solved linearly.
    - Leaving one more degree of freedom does not affect the result of MILP model while it speeds up the solving time.
  - The objective: $Maximizing : \Sigma_{z=0\sim31,i=0\sim31,j=1\sim4}(log_2(p_{z,i,j}/p_{z,i,j-1}) \times a^z_{i,j})$
    - The gain (probability calculated by log_2) of adding one more equation (when $a^z_{i,j} = 1$).
    - $p_{z,i,j}$ is a precomputed table recording the best probability under following conditions:
      - z^{th} row
      - i^{th} case of linearization (which bits on X^3 are linearized and can be controlled)
      - j added equations

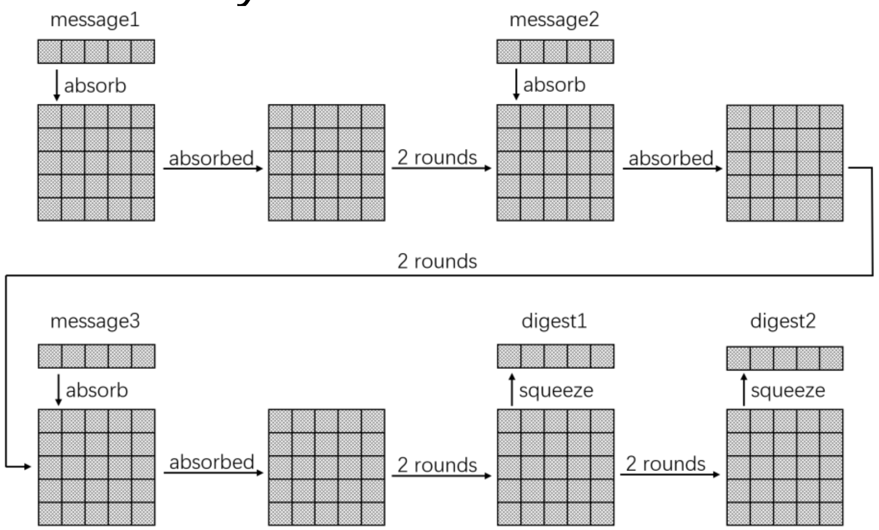
KECCAK[r = 640, c = 160, n_r = 4]



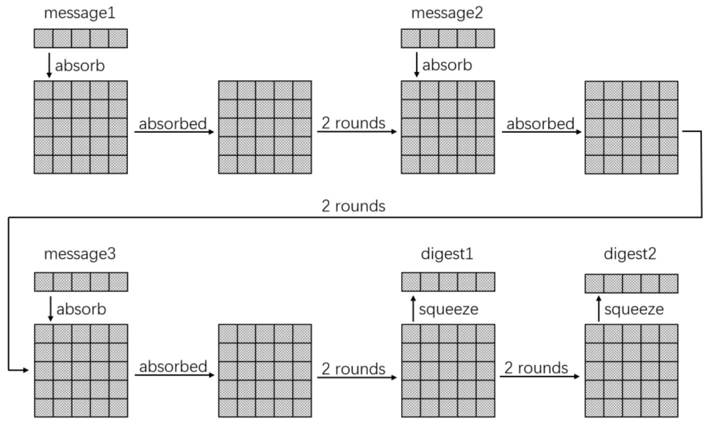
- Complexity
  - There are 320 variables on θ^1.
  - Add 158 equations to match one of capacity parts. One more for padding.
  - Add 63 equations to control column sums on θ^1.
  - Add 71 equations to restrict some linear bits on X^2 constant bits.
    - 19 bits on X^3 will become linear.
  - Add 19 equations to promote the probability of matching the digest.
    - Bring gains of $\sim 2^{16.5}$.
  - Solve quadratic equation systems with 34 quadratic equations on 8 variables.
  - Try $\sim 2^{56.5}$ different guesses for 71 equations on X^2.
    - It is expected to have one solution.

KECCAK[r = 40, c = 160, n_r = 2]

- First method (earlier attempt)
  - By solving quadratic equation systems.
  - Reduce the number of quadratic terms.
  - Although the complexity is relatively high, it is independent of round constants.
- Second method
  - Based on combination of linear structures and symmetries.
  - Round constant is zero for $ir = 3$.
  - Solve in backward direction.

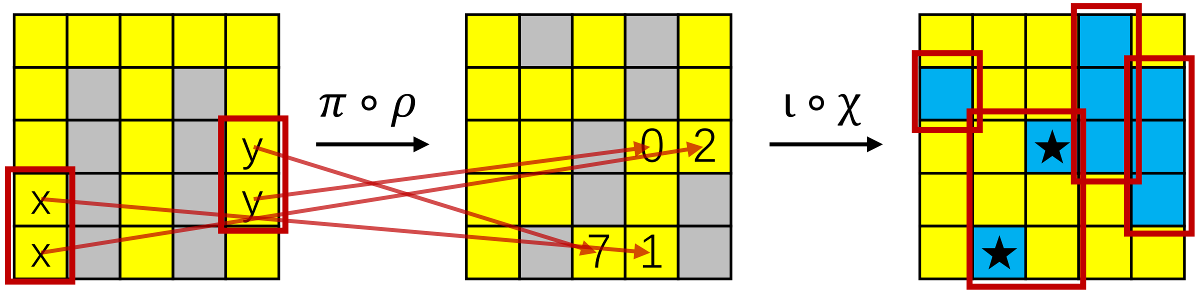


KECCAK[$r = 40, c = 160, n_r = 2$]



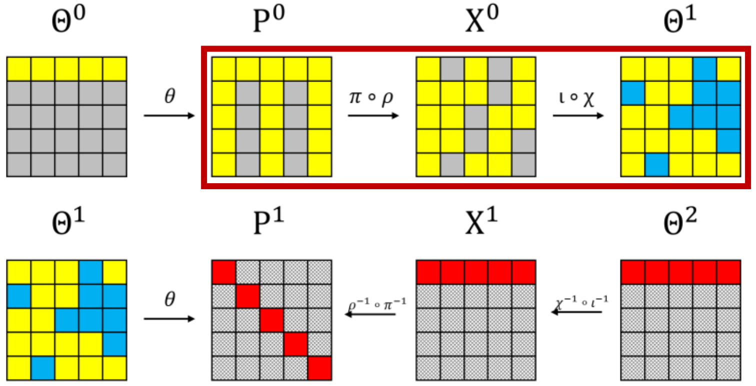
- Overview of the first method
  - Solve the first 40-bit.
  - Let the remaining 40-bit digest matched randomly.
- Use quadratic structure and solve equation systems
  - Leave as many degrees of freedom as possible.
  - Produce as few quadratic terms as possible.

KECCAK[$r = 40$, $c = 160$, $n_r = 2$]

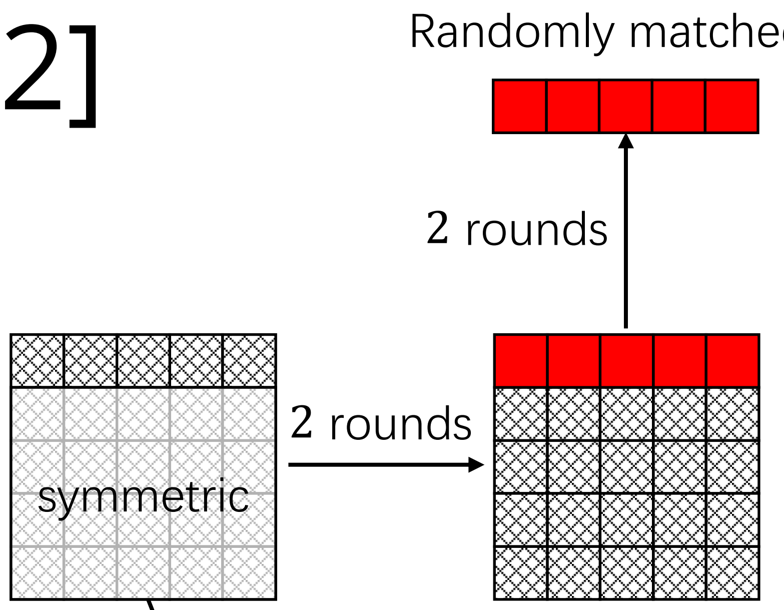


$4 \times 8 = 32$ different quadratic terms

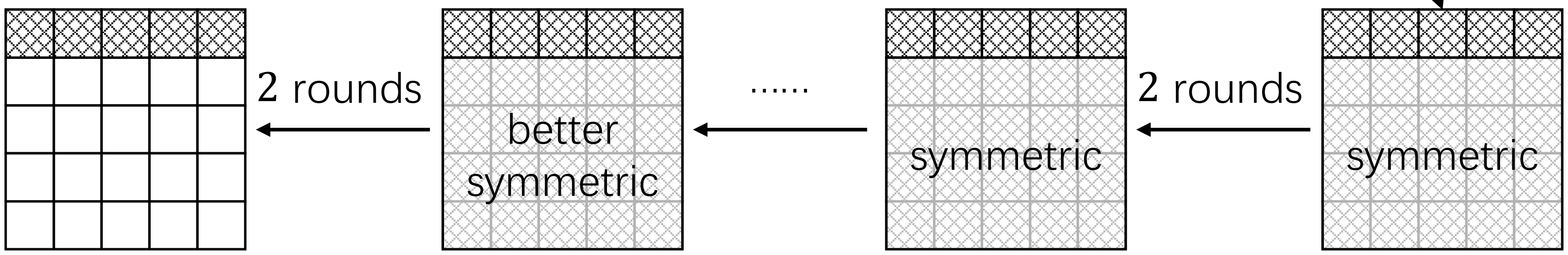
- The quadratic structure
  - $5 \times 8 - 1 = 39$ variables on θ^0
  - $2 \times 8 = 16$ equations restricting column sums
  - Rest 23 degrees of freedom.
  - 40 equations for (first 40-bit) digest matching
    - 8 linear equations, and 32 quadratic equations
  - Guess four more bits to solve the quadratic equation system linearly.
    - Rest around 19 degrees of freedom, and require guessing around 2^{21} times for matching the first 40-bit digest.
    - Randomly match the rest 40-bit digest.



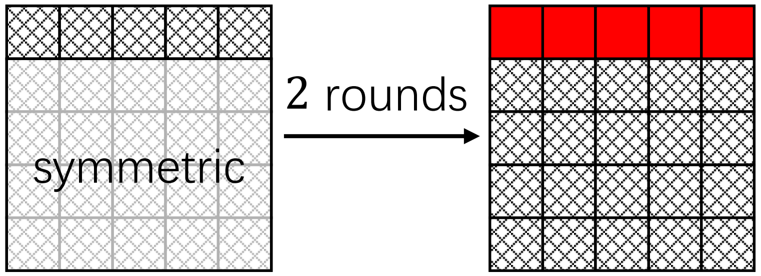
KECCAK[$r = 40, c = 160, n_r = 2$]

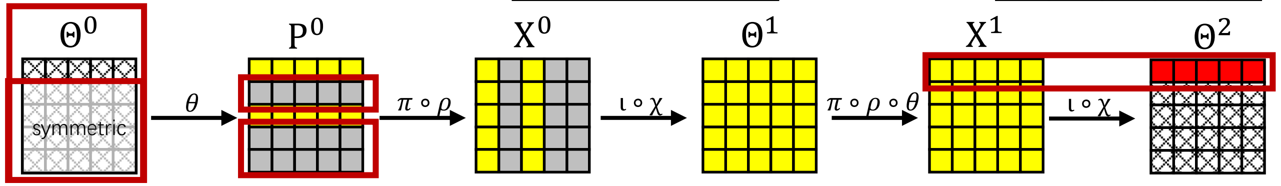


- Overview of the second method
  - Use linear structure produce the last block:
    - Capacity part of starting state is symmetric.
    - Match the first 40-bit digest with 1 probability.
  - Repeat 2^{40} times, and match the whole digest.
  - Try different symmetric message blocks and compute backward.
    - Match the all zero IV.



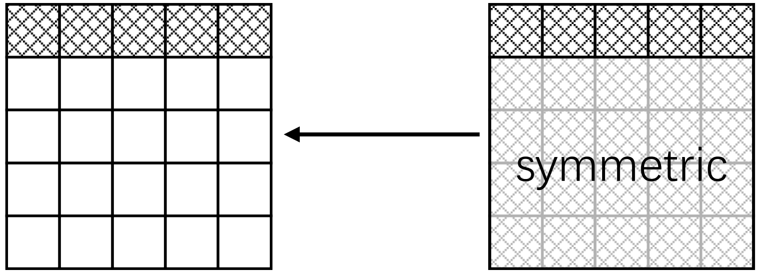
Keccak[$r = 40, c = 160, n_r = 2$]



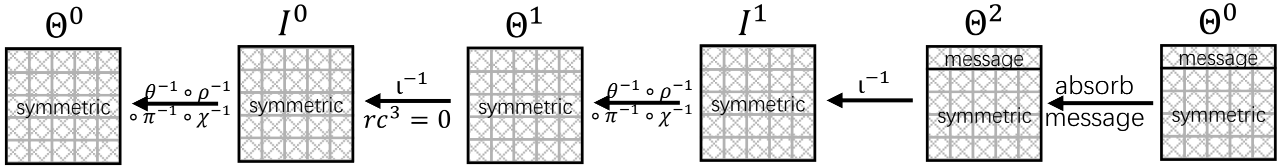


- Get the last message block
  - $25 \times 8 = 200$ variables on Θ^0
  - Each lane on the capacity part of Θ^0 is symmetric (e.g. $0\text{x}11$).
    - Add $20 \times 4 = 80$ equations.
  - The bits on 2^{nd}, 4^{th}, and 5^{th} planes of P^0 are constant bits.
    - Add $15 \times 8 = 120$ equations.
  - Match the first 40-bit digest on Θ^2.
    - Add 40 equations.
  - 41 equations among them are linear-dependent.
    - The number of degrees of freedom is enough.
  - Try different constant settings on P^0, until the last 40-bit digest satisfied.

KECCAK[$r = 40$, $c = 160$, $n_r = 2$]



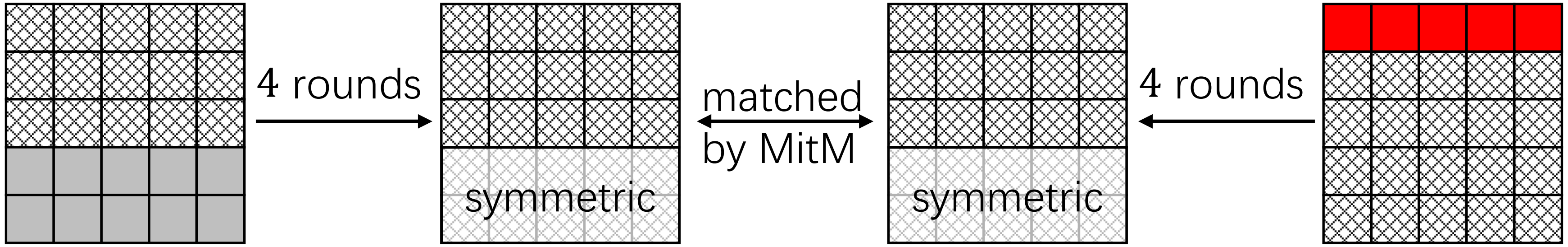
- The backward process
  - Start from a symmetric capacity.
  - Select different message blocks, so that the state is symmetric before inverse ι.
  - For the property of $\theta^{-1} \circ \rho^{-1} \circ \pi^{-1} \circ \chi^{-1}$ operations, state stays symmetric.
  - Select the start round index $ir = 3$, where the round constant is zero for the first round ($rc^{ir=3} = 0$).The property of symmetry will always hold.
  - The property of symmetry may be better and better.
    - period $\mathrm{i} = 4$ to period $\mathrm{i} = 2$ to period $\mathrm{i} = 1$ to all 0 (e.g. $0\text{x}11 \rightarrow 0\text{x}aa \rightarrow 0\text{x}ff \rightarrow 0\text{x}00$).
      00010001 10101010 11111111 00000000



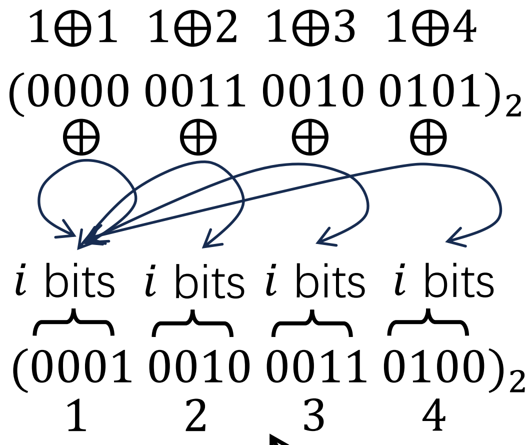
ff	00	00	00	00		8b	00	ff	ff	00		de	aa	aa	00	00		a9	aa	33	00	88
00	00	00	00	00	$p = 2^{-20}$	00	ff	ff	ff	00	$p = 2^{-20}$	00	55	aa	00	aa	$p = 2^{-40}$	33	66	ee	88	44
00	00	00	00	00	$\leftarrow$	00	00	00	00	ff	$\leftarrow$	00	55	00	00	55	$\leftarrow$	22	ee	ee	cc	dd
00	00	00	00	00		ff	ff	ff	00	ff		ff	00	00	aa	00		33	ff	cc	99	dd
00	00	00	00	00		00	00	00	ff	ff		aa	00	55	55	55		bb	66	55	88	dd

KECCAK[r = 240, c = 160, n_r = 4]

- Overview
  - The first stage
    - Use the target internal difference algorithm (TIDA) based on previous work [DDS13, ZHL23, ZHL24].
    - From constant starting state, produce around 2^{32} states with symmetric capacity part.
  - The second stage
    - Use a new technique based on internal differential cryptanalysis.
    - Produce around 2^{49} states with symmetric capacity part and match the digest.
  - The symmetric states produced by two stages all lie within a set of size 2^{80}.
    - With time-memory trade-off, find collisions between two sets produced by two stages.

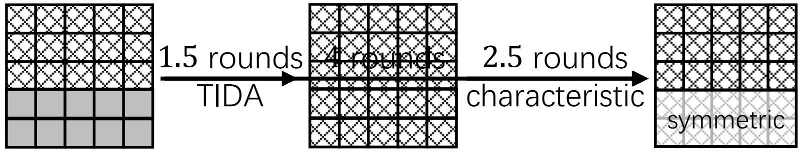


KECCAK[r = 240, c = 160, n_r = 4]

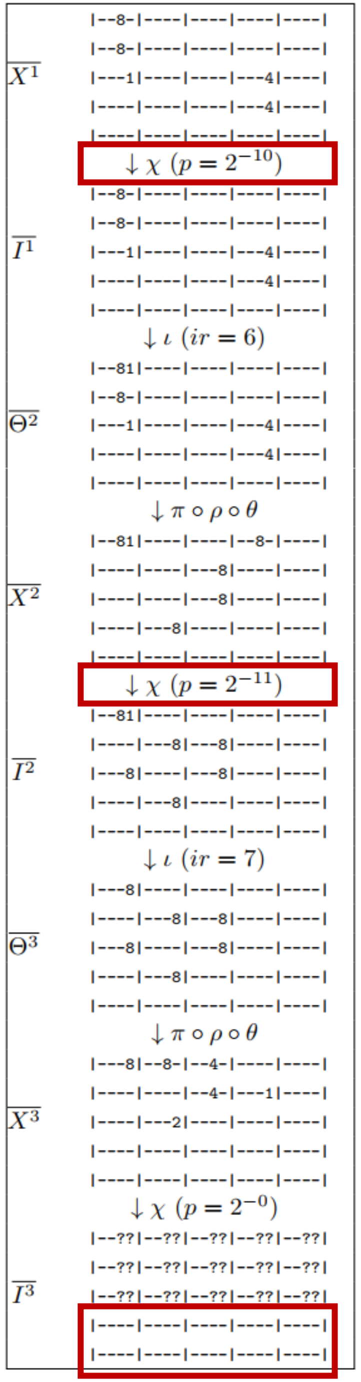


- Preliminary concepts
  - Internal difference
    - Period i
    - Symmetric state (each lane consists of w/i repetitions of i-bit segments)
    - Internal difference of a state (XOR the segments for each lane)
      - For example, let lane size $w = 16$, period $i = 4$, then consider the actual value of a lane $0\text{x}1234$.
      - The internal difference of this lane will be $0\text{x}0325$.
    - Given internal difference of state A, internal difference of $\theta(A)$, $\rho(A)$, $\pi(A)$, and $\iota(A)$ can be directly derived with 1 probability. However, internal difference of $\chi(A)$ may propagate to different cases according to the actual value, unless A is a symmetric state.
  - Internal differential characteristic
    - Exhibits the internal difference propagating through a few rounds.
    - Contains a holding probability for χ operation of each round.
    - The characteristics also hold in a reversed direction because the operations are invertible.

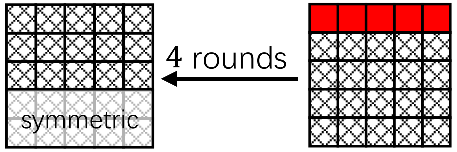
KECCAK[$r = 240, c = 160, n_r = 4$]



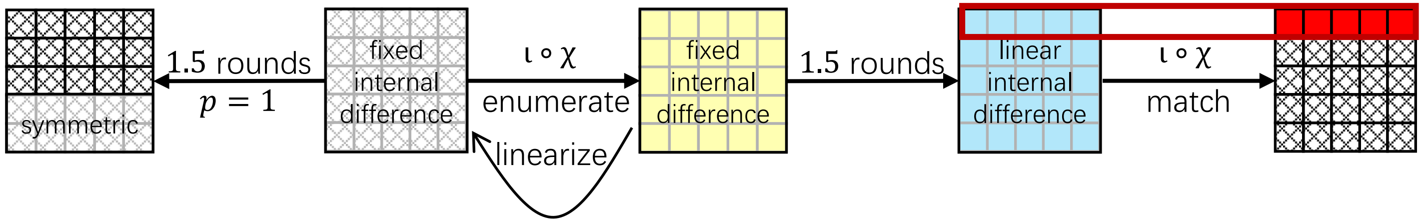
- The first stage
  - 2.5-round internal differential characteristic
    - Period $i = 8$ (half of lane size $w = 16$).
    - Leads to symmetric (all zero internal difference) capacity part.
    - Holding probability $2^{-10-11} = 2^{-21}$.
  - TIDA [DDS13, ZHL23, ZHL24]
    - Linking constant starting state and fix internal difference on Θ^1.
    - Difference phase
      - Prepare an equation system describe internal difference on first round.
      - Add equations restricting internal difference on capacity part.
      - Select an affine subset for internal difference of each row on X^1, and add equations.
    - Value phase
      - Enumerate every solution of equation system in the difference phase.
      - Prepare another equation system describe actual value on first round.
      - Add equations restricting actual value on capacity part.
      - Select an affine subset for actual value of each row on X^1, and add equations.
  - Enumerate every solution of equation system in the value phase.
    - Check whether 2.5-round internal differential characteristic is passed or not.



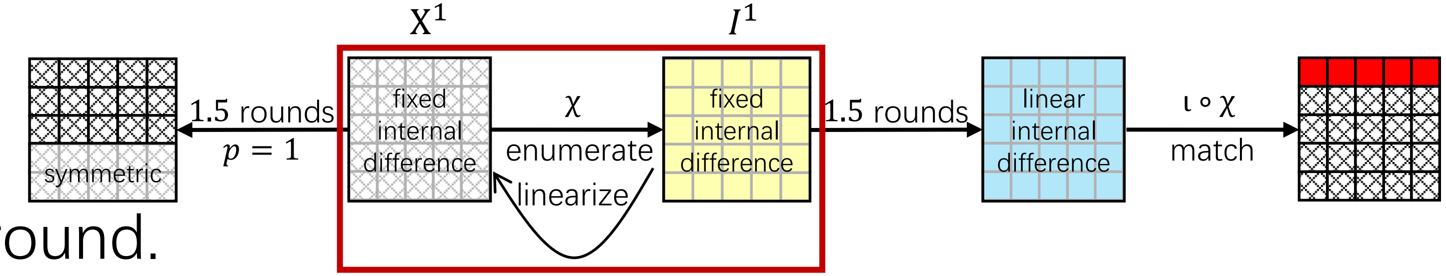
KECCAK[$r = 240, c = 160, n_r = 4$]



- The second stage (overview)
  - Determine a 1.5-round inversed internal differential characteristic ($\mathrm{X}^1 \rightarrow \Theta^0$).
  - Guess different internal difference after the χ operation.
  - Derive the linear strategy for the χ^{-1} operation.
  - After 1.5-round, add linear restrictions on internal difference.
  - Linearize some other bits (actual value).
    - The strategy is determined by MILP model.
  - Let the rest bits satisfied randomly.
  - Compute backward and get starting state.

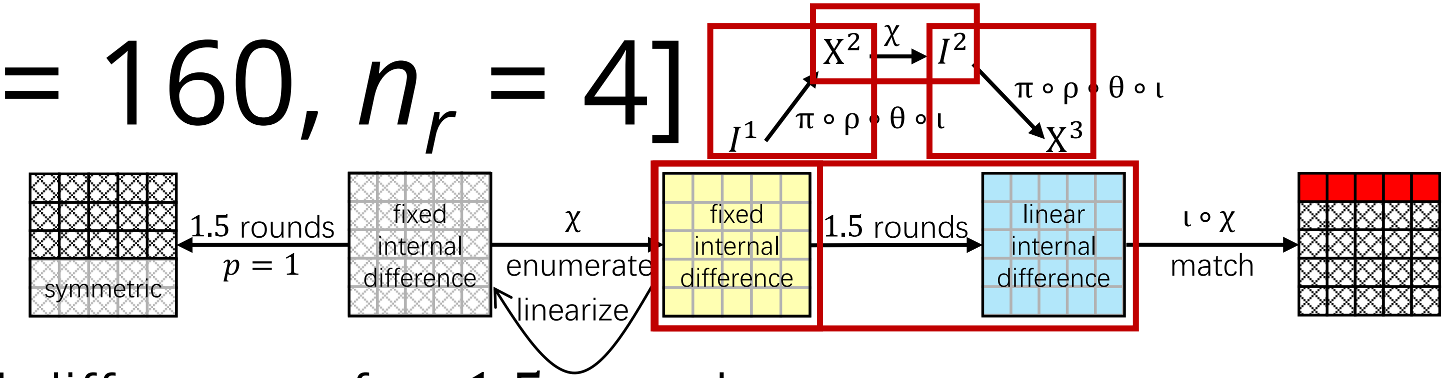


KECCAK[$r = 240$, $c = 160$, $n_r = 4$]



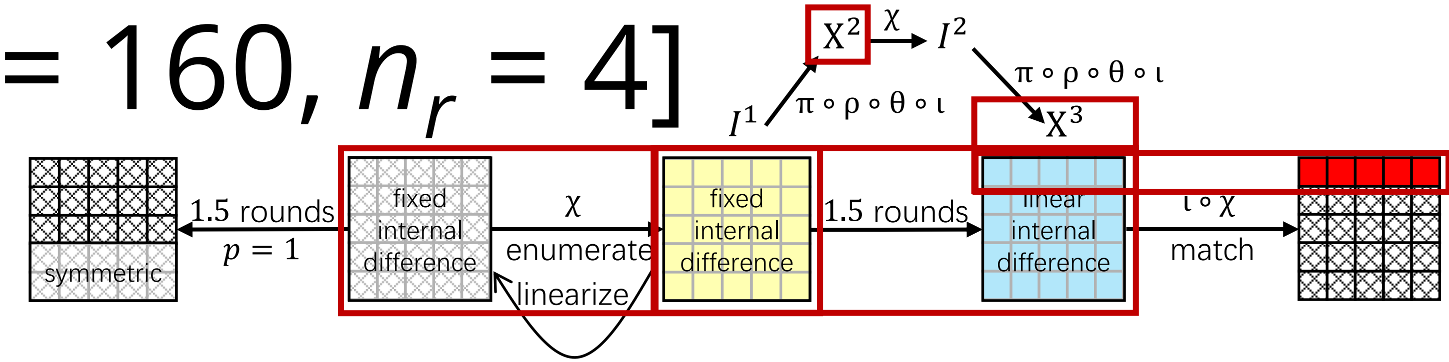
- The second stage
  - Deal with χ in the second round.
    - Internal difference of X^1 is known according to internal differential characteristic.
    - For each row, guess possible internal difference on I^1.
    - Then determine the actual value.
      - The variables are selected on I^1.
      - Add equations on these variables so that the restricted actual values ensure that internal difference of each row can be transformed backward to X^1 with 1 probability.
        - For non-active S-box (all zero row), no equations are required.
        - For each row with $DDT = 8$, 3 equations are required (there are two kinds of 3 equations).
        - For each row with $DDT = 4$, 3 equations are required.
        - For each row with $DDT = 2$, 4 equations are required (there is no row with $DDT = 2$ here).
      - Degrees of freedom
        - There are 9 active rows → using 27 degrees of freedom

KECCAK[$r = 240, c = 160, n_r = 4$]



- The second stage
  - linear restrictions on internal difference after 1.5 rounds
    - The state I^1: the internal difference is fixed constant, and the actual value is all linear.
      - For each lane: high $i = 8$ bits are variables, while the remaining $w - i = 8$ bits either exactly match these variable bits or differ by a constant 1.
    - After ι, θ, ρ, and π, the internal difference is still constant and the actual value is still linear.
    - After χ, the internal difference will be linear although the actual value becomes quadratic.
      - Suppose three consecutive bits on two corresponding rows are x, y, z and $x \oplus c_x, y \oplus c_y, z \oplus c_z$.
      - After χ, the first bits on two rows will be $x \oplus z \oplus yz$ and $(x \oplus c_x) \oplus (z \oplus c_z) \oplus (y \oplus c_y)(z \oplus c_z)$.
      - Although the two bits are both quadratic, the difference $(c_x \oplus c_z \oplus c_z y \oplus c_y z \oplus c_y c_z)$ is linear.
    - After ι and θ, ρ, π of next round, the internal difference is still linear.
      - Add restrictions so that the digest is matched when high i bits on each lane were matched.

KECCAK[r = 240, c = 160, n_r = 4]

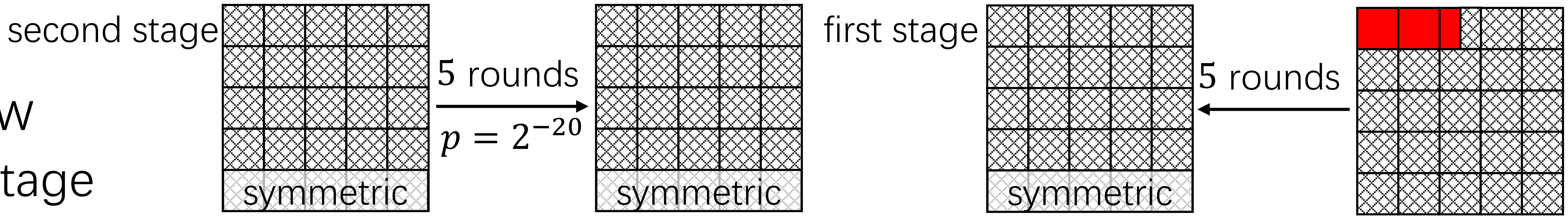


- The second stage
  - Linearize some other bits
    - Use MILP model, and similar to cryptanalysis on $b = 800$.
    - Spend $87 + 31 = 118$ degrees of freedom restrict 31 more bits on X^3.
  - Complexity
    - There are 400 degrees of freedom on I^1.
    - 200 equations are added to restrict the internal difference on I^1.
    - 27 equations are added to restrict the internal difference on X^1 (restrict the actual value for 9 active rows).
    - 40 equations are added to restrict the internal difference on X^3.
    - 87 equations are added to restrict some bits on X^2 constant bits.
    - 31 equations are added to restrict the corresponding required bits on X^3.
    - Solve linear equation system and verify the around $2^{400-200-27-40-87-31} = 2^{15}$ solutions.
    - For one try, the probability of matching the digest will be $2^{-(80-40-31)} = 2^{-9}$.
    - Collect around 2^{49} symmetric starting states that also lead to required digest.

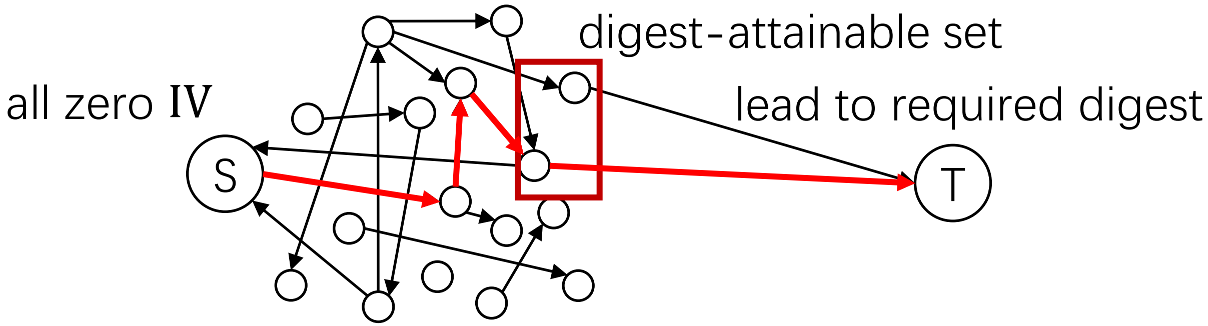
KECCAK[r = 640, c = 160, n_r = 5]

- Unsolved yet (the complexity is around 2^{62})
- Differences from cryptanalysis on 4-round $b = 400$
  - Increasing the round number leads to the difficulty of the first stage.
    - Instead, cancel the first stage and exploit the symmetry property of all zero IV.
  - Without MitM, higher symmetry is required to reduce the complexity.
    - We select period $i = 8$ ($w = 32$, and 4 repetitions for each lane).
  - Other modifications such as characteristic, linearization and so on.

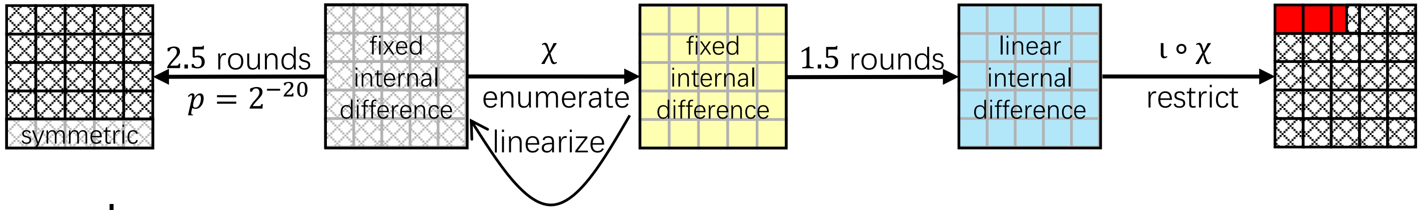
KECCAK[r = 640, c = 160, n_r = 5]



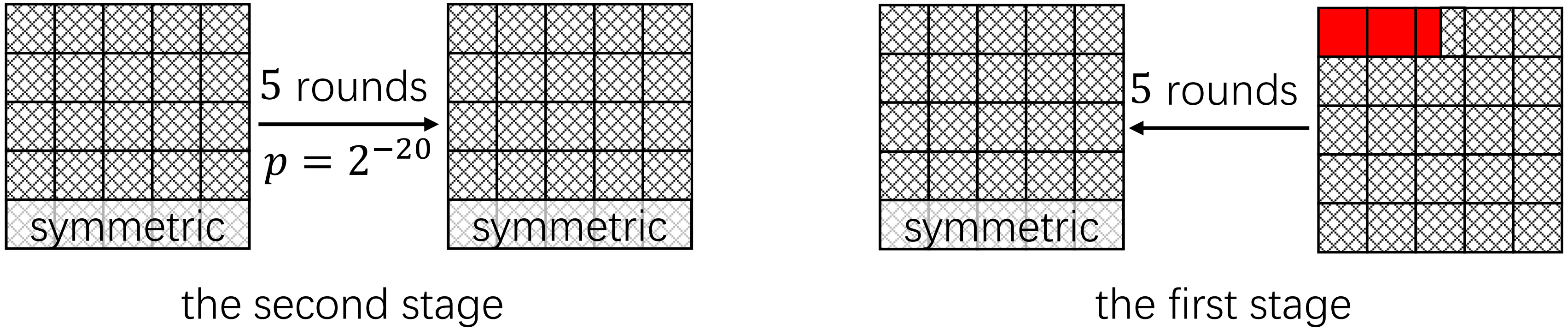
- Overview
  - First stage
    - From given digest, generate some applicable starting states with symmetric capacity part.
  - Second stage
    - Generate a large number of directed edge from a symmetric capacity part to another.
    - The pre-image can be found when the node representing the all zero IV becomes connected to any node in digest-attainable set.



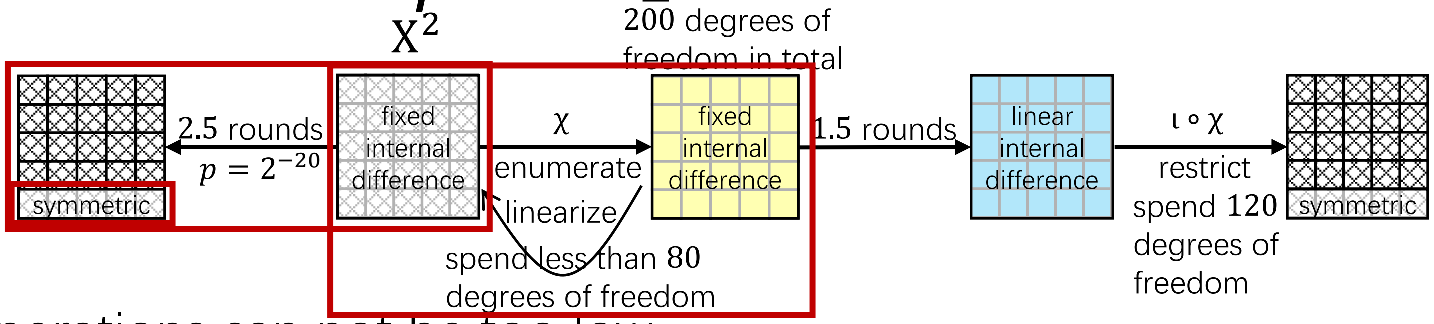
KECCAK[r = 640, c = 160, n_r = 5]



- The first stage
  - Almost the same with the second stage.
  - Only difference:
    - When matching digest, restrictions are on the first plane instead of the last plane.
    - The number of restrictions are fewer.



KECCAK[r = 640, c = 160, n_r = 5]

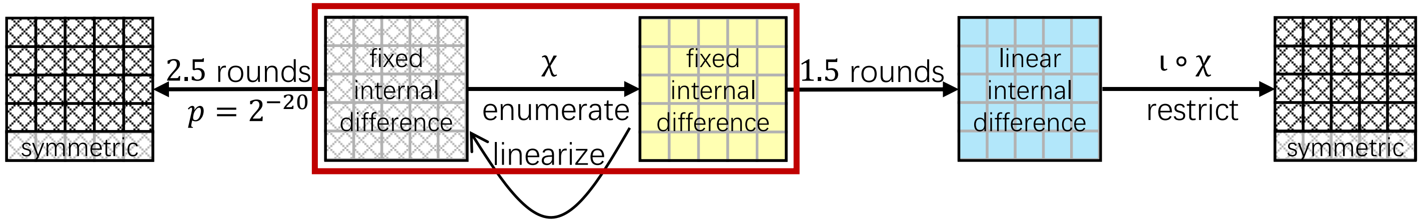


- The second stage
  - Find the characteristic.
    - The probability of passing χ operations can not be too low.
      - The probability should be at least 2^{-20}.
      - The sum of numbers of active rows for first two χ should be less than 10.
    - Require more non-active rows on X^2 due to insufficient degrees of freedom.
      - The number of degrees of freedom used for linearization of the third χ should be less than 80.
    - There is no difference bit on the last plane of the starting state.
      - The capacity part should be symmetric.
    - We employ condition-guided search instead of using the MILP model.
      - The MILP model with basic implementation does not provide desired characteristic quickly.
      - We believe the MILP model with better modeling can also find the good characteristic.

KECCAK[r = 640, c = 160, n_r = 5]

- The second stage $\boxed{\Theta^0 \xleftarrow{\theta^{-1}\circ\rho^{-1}\circ\pi^{-1}} X^0 \xleftarrow{\chi^{-1}}}\boxed{I^0}\boxed{\xleftarrow{\iota^{-1}} \Theta^1} \xrightarrow{\pi\circ\rho\circ\theta} \boxed{X^1 \xrightarrow{\chi} I^1} \xrightarrow{\iota} \Theta^2 \xrightarrow{\pi\circ\rho\circ\theta} X^2$
  - The way of searching characteristic:
    - Determine a start round index ($ir = 5$ shows the best result).
    - Enumerate three or four bit-pairs on Θ^1 (even parity for each column).
    - First prune:
      - Some bits on I^0 should be cancelled out by the first-round constant ($I^0 \xleftarrow{\iota^{-1}} \Theta^1$).
    - Second prune:
      - Assume the second χ ($X^1 \xrightarrow{\chi} I^1$) can extend any additional bits on each active row.
      - Then, for the most ideal case with as many even-parity columns as possible on Θ^2, the number of required degrees of freedom on X^2 should not exceed the limit.
    - Third prune:
      - Assume the first χ^{-1} ($X^0 \xleftarrow{\chi^{-1}} I^0$) can extend any additional bits on each active row.
      - Then, for the most ideal case, all the columns on X^0 should be even-parity columns.
    - For the rare cases after pruning, enumerate all the possible propagations for each χ, and check whether the characteristic meet the requirements.

KECCAK[r = 640, c = 160, n_r = 5]



- The second stage
  - Deal with χ in the third round.
    - For more complex characteristic and different ratio ($\frac{w}{i} = 4$) of lane size w and period i, the linearization is similar but with minor differences.
    - We should consider four rows at the same time (with stride $i = 8$).
      - If there is no active row:
        - No equations are required.
      - If there is only one active row:
        - For the case of $DDT = 8$, 3 equations are required (there are two kinds of 3 equations).
          - Or 2 (or 1) equations with 0.75(or 0.5) probability.
        - For the case of $DDT = 4$, 3 equations are required.
        - For the case of $DDT = 2$, 4 equations are required.
      - If there are at least two active rows:
        - Enumerate all the 32 cases for actual value, and record the cases that satisfy the propagation for the four rows at the same time.
        - If there are two available cases, 4 equations are required.
        - If there is only one available case, 5 equations are required to fully determine the actual value.

KECCAK[$r = 640$, $c = 160$, $n_r = 5$]

- The second stage (supplements)
  - The internal differential characteristic for the second stage.
    - Require 72 equations to restrict the χ in the third round ($72 + 120 < 200$).
    - Probability of passing the first two rounds is $2^{-(8+12)} = 2^{-20}$ (require $\sim 2^{41}$ states).

$\overline{\Theta^0}$

```
|--------|--------|-------2|--------|------2-|
|--------|--------|--------|--------|--------|
|--------|--------|-------2|--------|------2-|
|--------|--------|--------|--------|--------|
|--------|--------|--------|--------|--------|
```

$\uparrow \theta^{-1} \circ \rho^{-1} \circ \pi^{-1}$

$\overline{X^0}$

```
|--------|--------|----1---|--------|--------|
|--------|--------|--------|--------|--------|
|--------|--------|--------|--------|--------|
|-------1|--------|--------|--------|--------|
|--8-8-8-|--------|----1---|--------|--------|
```

$\uparrow \chi^{-1}$ $(p = 2^{-8})$

$\overline{I^0}$

```
|--------|--------|----1---|--------|--------|
|--------|--------|--------|--------|--------|
|--------|--------|--------|--------|--------|
|-------1|--------|--------|--------|--------|
|--8-8-8-|--------|----1---|--------|--------|
```

$\uparrow \iota^{-1}$ $(ir = 5)$

$\overline{\Theta^1}$

```
|--8-8-81|--------|----1---|--------|--------|
|--------|--------|--------|--------|--------|
|--------|--------|--------|--------|--------|
|-------1|--------|--------|--------|--------|
|--8-8-8-|--------|----1---|--------|--------|
```

$\uparrow \theta^{-1} \circ \rho^{-1} \circ \pi^{-1}$

$\overline{X^1}$

```
|--8-8-81|--------|--------|--------|--------|
|--------|--------|--------|--------|-----2--|
|--------|--------|--------|--------|---2----|
|--------|--------|--------|--------|--------|
|-----4--|--------|--------|-----2--|--------|
```

$\uparrow \chi^{-1}$ $(p = 2^{-12})$

$\overline{I^1}$

```
|--8-8-81|--------|--------|--------|--------|
|--------|--------|--------|-----2--|-----2--|
|--------|--------|--------|--------|---2----|
|--------|--------|--------|--------|--------|
|-----4--|--------|-----2--|-----2--|--------|
```

$\uparrow \iota^{-1}$ $(ir = 6)$

$\overline{\Theta^2}$

```
|----8-8-|--------|--------|--------|--------|
|--------|--------|--------|-----2--|-----2--|
|--------|--------|--------|--------|---2----|
|--------|--------|--------|--------|--------|
|-----4--|--------|-----2--|-----2--|--------|
```

$\uparrow \theta^{-1} \circ \rho^{-1} \circ \pi^{-1}$

$\overline{X^2}$

```
|---2828-|-----8-8|--------|--c-c-4-|---24242|
|----4-6-|--b-b-a-|--1-1---|----1-1-|------4-|
|---1-1--|--------|-----8-c|---8-1-1|--18181-|
|-----848|--2-2---|-----2-2|--------|-----4-4|
|--------|-----2-2|--8581-1|-----4-4|---2-2--|
```

References

- [BDPA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Crypto-graphic sponge functions, 2011. http://sponge.noekeon.org/CSF-0.1.pdf.
- [BDH+] Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. The keccak-f state. https://keccak.team/figures.html.
- [GLS16] Jian Guo, Meicheng Liu, and Ling Song. Linear structures: Applications to cryptanalysis of round-reduced Keccak. In Jung Hee Cheon and Tsuyoshi Takagi, editors, ASIACRYPT 2016, Part I, volume 10031 of LNCS, pages249–274. Springer, Berlin, Heidelberg, December 2016. https://doi.org/10.1007/978-3-662-53887-6_9.
- [LS19] Ting Li and Yao Sun. Preimage attacks on round-reduced Keccak-224/256via an allocating approach. In Yuval Ishai and Vincent Rijmen, editors, EUROCRYPT 2019, Part III, volume 11478 of LNCS, pages 556–584. Springer, Cham, May 2019. https://doi.org/10.1007/978-3-030-17659-4_19.
- [Raj19] Mahesh Sreekumar Rajasree. Cryptanalysis of round-reduced KECCAK using non-linear structures. In Feng Hao, Sushmita Ruj, and Sourav Sen Gupta, editors, INDOCRYPT 2019, volume 11898 of LNCS, pages 175–192. Springer, Cham, December 2019. https://doi.org/10.1007/978-3-030-35423-7_9.
- [LIMY21] Fukang Liu, Takanori Isobe, Willi Meier, and Zhonghao Yang. Algebraic attacks on round-reduced Keccak. In Joonsang Baek and Sushmita Ruj, editors, ACISP 21, volume 13083 of LNCS, pages 91–110. Springer, Cham, December 2021. https://doi.org/10.1007/978-3-030-90567-5_5.
- [WWF+21] Congming Wei, Chenhao Wu, Ximing Fu, Xiaoyang Dong, Kai He, JueHong, and Xiaoyun Wang. Preimage attacks on 4-round keccak by solving multivariate quadratic systems. In Jong Hwan Park and Seung-Hyun Seo, editors, ICISC 21, volume 13218 of LNCS, pages 195–216. Springer, Cham, December 2021. https://doi.org/10.1007/978-3-031-08896-4_10.
- [HLY21] Le He, Xiaoen Lin, and Hongbo Yu. Improved preimage attacks on 4-roundKeccak-224/256.IACR Trans. Symm. Cryptol., 2021(1):217–238, 2021. https://doi.org/10.46586/tosc.v2021.i1.217-238.
- [DDS13] Dinur, I., Dunkelman, O., Shamir, A.: Collision Attacks on Up to 5 Rounds of SHA-3 Using Generalized Internal Differentials. In: Moriai, S. (eds) Fast Software Encryption. FSE 2013, LNCS vol. 8424, pp. 219–240. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-662-43933-3_12
- [ZHL23] Zhang, Z., Hou, C., Liu, M.: Collision Attacks on Round-Reduced SHA-3 Using Conditional Internal Differentials. In: Hazay, C., Stam, M. (eds) Advances in Cryptology – EUROCRYPT 2023, LNCS vol 14007, pp. 220–251. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30634-1_8
- [ZHL24] Zhang, Z., Hou, C., Liu, M.: Probabilistic Linearization: Internal Differential Collisions in up to 6 Rounds of SHA-3. In: Reyzin, L., Stebila, D. (eds) Advances in Cryptology – CRYPTO 2024, LNCS, vol 14923, pp. 241–272. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-68385-5_8


清華大學
Tsinghua University
THANK YOU!