Key Collisions on AES and Its Applications

ASIACRYPT2024@Dhono Dhanyo Auditorium

Kodai Taiyama [1], Kosei Sakamoto [2], Ryoma Ito [3], Kazuma Taka [1], Takanori Isobe [1]

1. University of Hyogo, Japan
2. Mitsubishi Electric Corporation, Japan
3. National Institute of Information and Communications Technology

Outline

① Overview

② Preliminaries

③ Automatic Tool for Key Collision

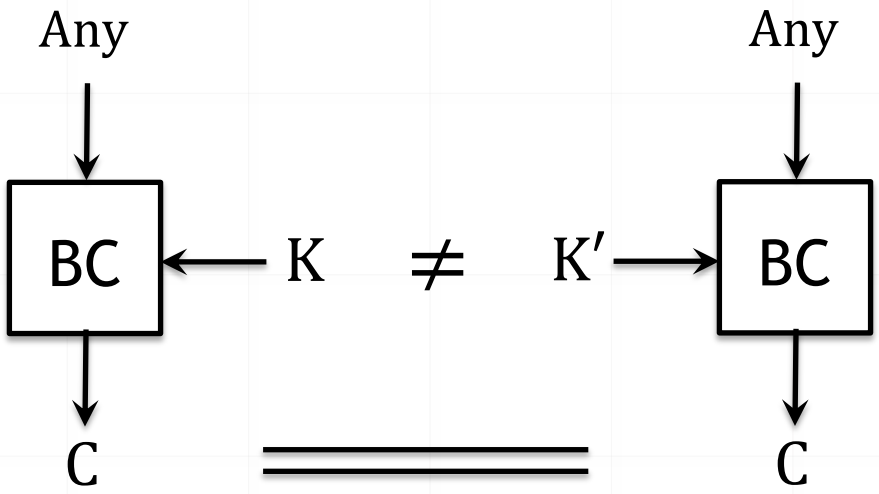
④ Key Collisions on AES256-DM

Outline

New variant of key collision

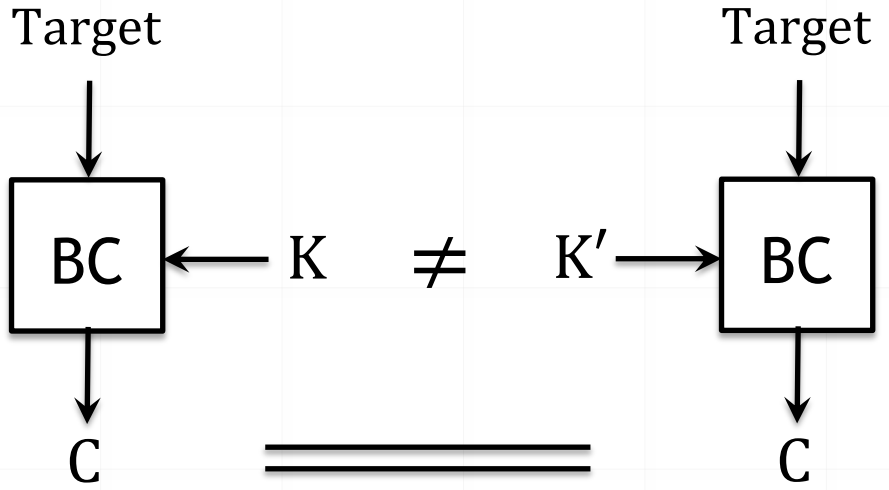
➢ Key Collision

- Two distinct keys produce identical subkeys through key scheduling function



➢ Target-plaintext key collision

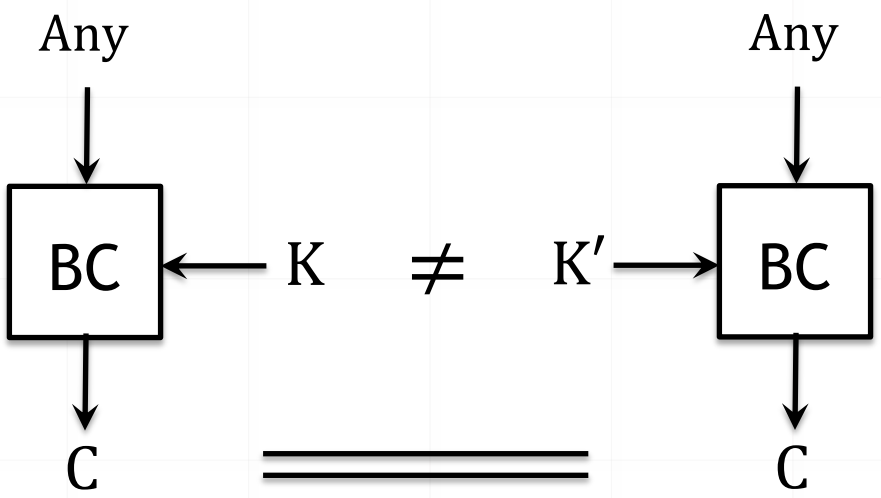
- Two distinct keys that generate the same ciphertext for a single target plaintext



New variant of key collision

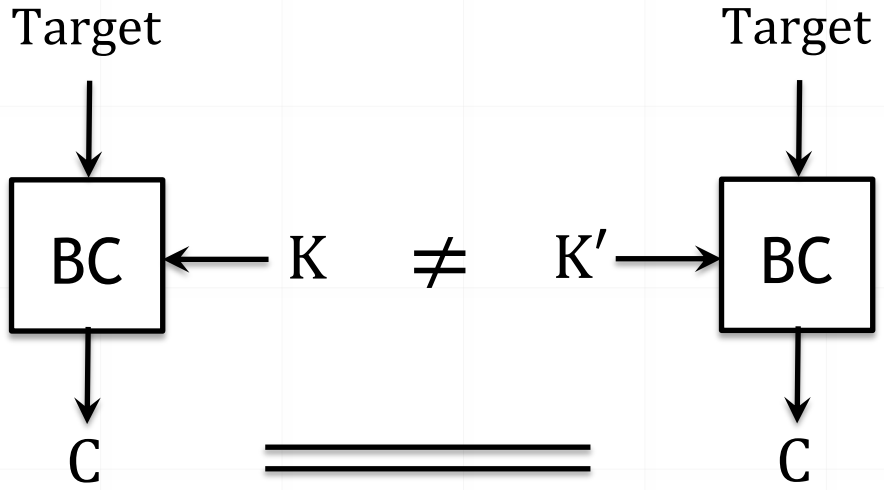
➢ Key Collision

- Two distinct keys produce identical subkeys through key scheduling function



➢ Target-plaintext key collision

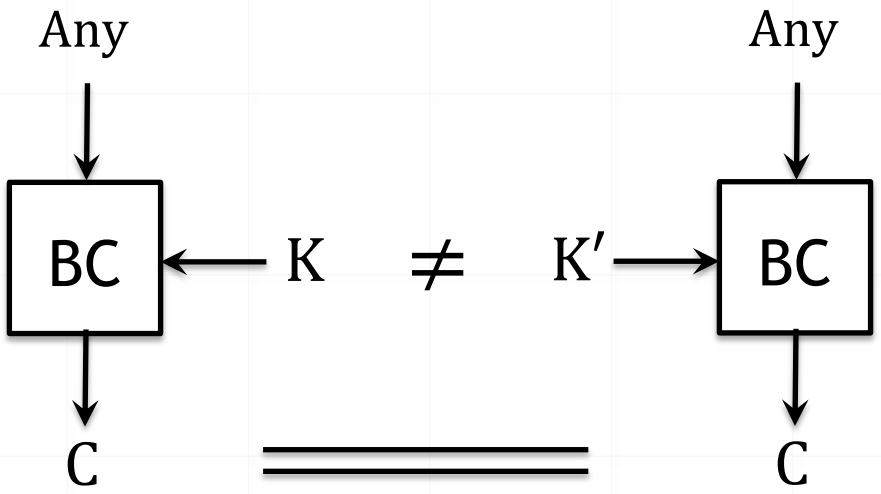
- Two distinct keys that generate the same ciphertext for a single target plaintext



New variant of key collision

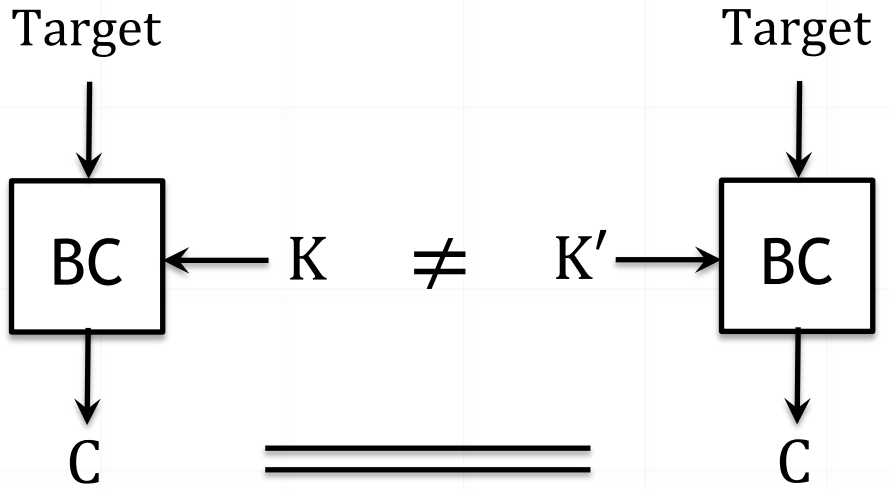
➢ Key Collision

- Two distinct keys produce identical subkeys through key scheduling function



➢ Target-plaintext key collision

- Two distinct keys that generate the same ciphertext for a single target plaintext

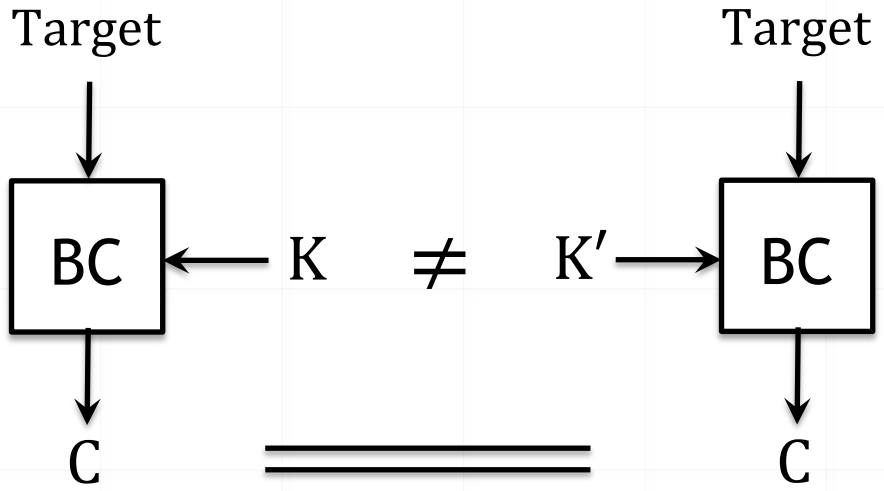


New variant of key collision

1. Fixed-target-plaintext
2. Free-target-plaintext

➢ **Target-plaintext key collision**

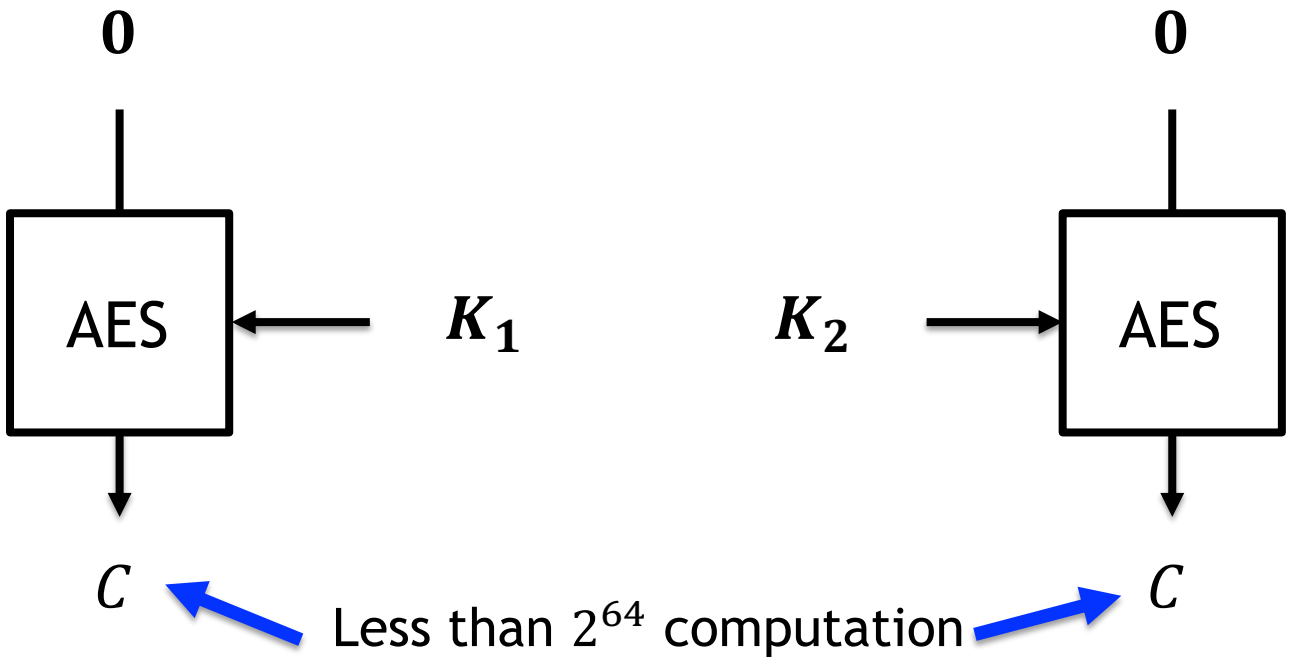
- Two distinct keys that generate the same ciphertext for a single target plaintext



Open problem in key committing security

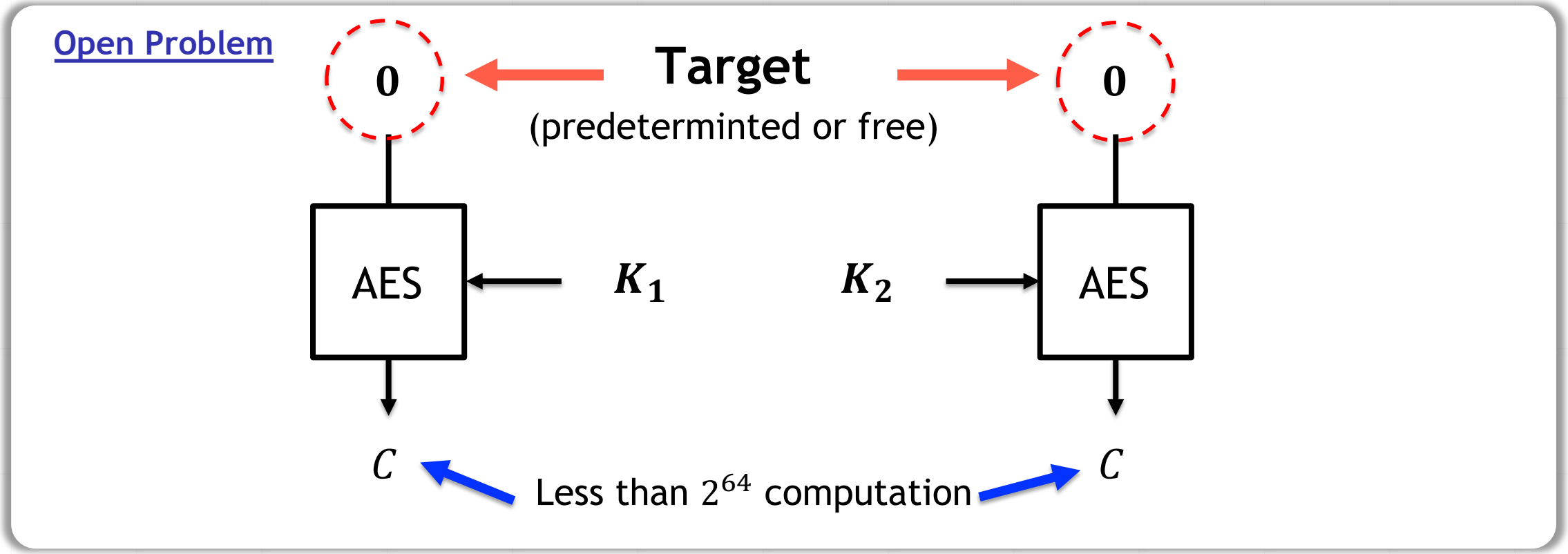
- **Albertini's research on AES-GCM and ChaCha20-Poly1305 (USENIX 2022) [1]**
  - Padding fix : $Enc(K, N, A, X||M)$

Open Problem



Open problem in key committing security

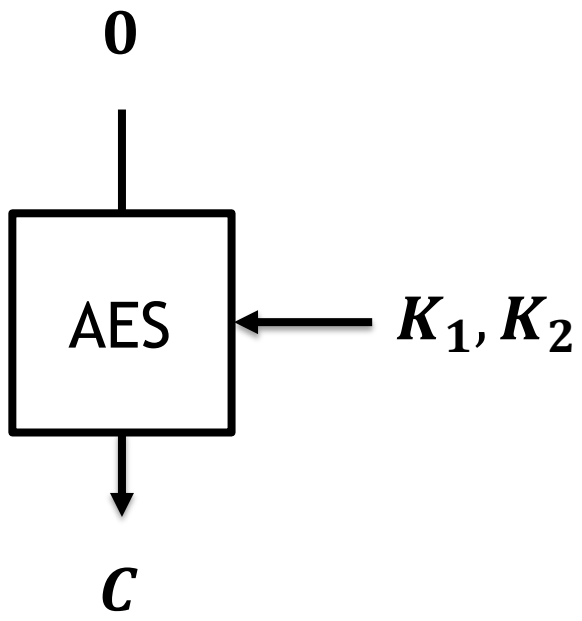
- **Albertini's research on AES-GCM and ChaCha20-Poly1305 (USENIX 2022) [1]**
  - Padding fix : $Enc(K, N, A, X||M)$



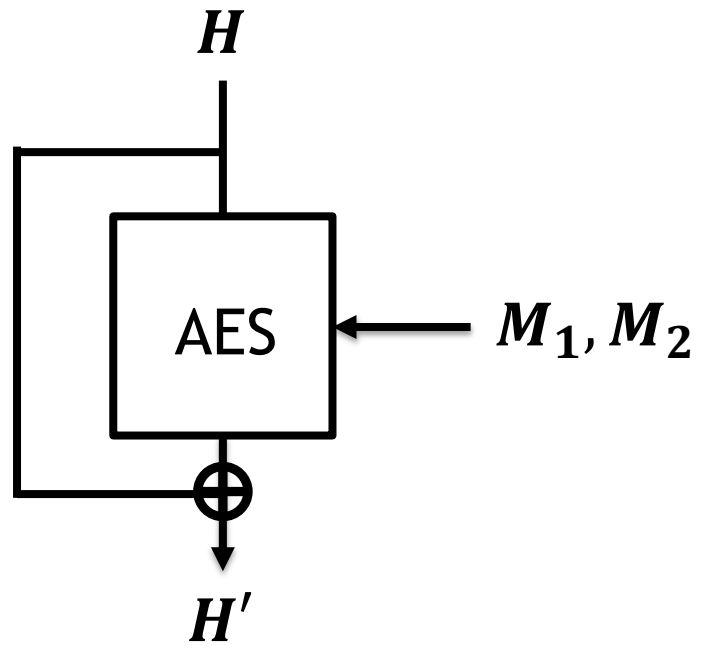
Application of new type of key collision

- Target-plaintext collision attack can be converted into **collision attack on Davies-Meyer(DM) hashing mode with AES**

Target-plaintext collision



Hash collision on DM-AES



Difficulties in target-plaintext key collision

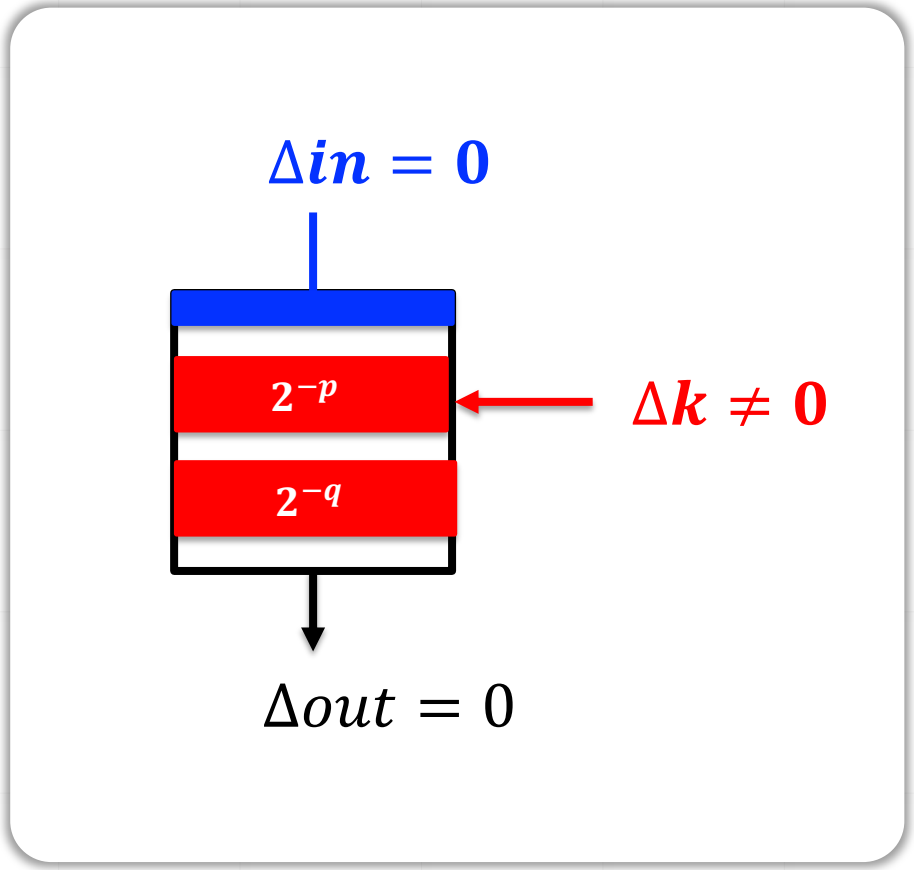
- Despite its significance, this type of attack has not yet been investigated for AES **over the past 20 years**

Problem 1.

Key differences should be canceled out by themselves

Problem 2.

Limitations in controlling plaintext values



Our approach

➢ Automatic tool to find target-plaintext key collision

- Utilizes bit-wise differential characteristics explored by SAT solver
- Automatically groups the internal state into independently computable components and evaluate the computational complexity

Results

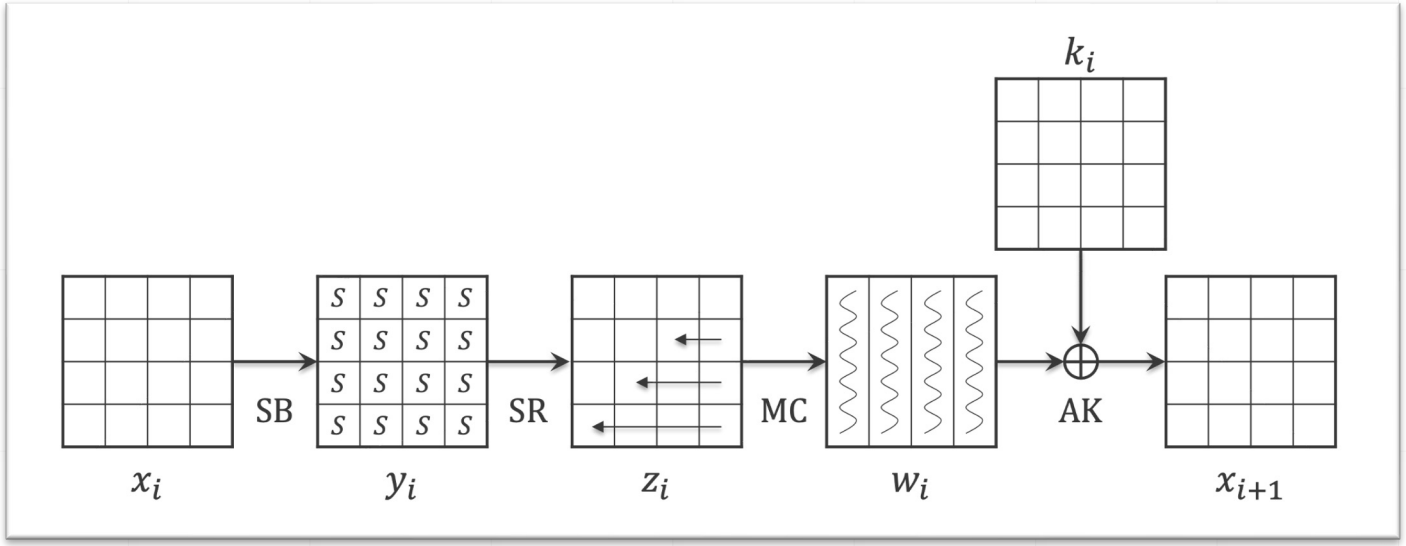
Target	Attack	Round	Time	Memory	Ref.
AES-128-DM	Collision	2/10	2^{49}	Negligible	[Ours]
	Collision*	3/10	2^{60}	2^{52}	[Ours]
	Semi-free-start	5/10	2^{57}	Negligible	[Ours]
	Free-start	5/10	2^{56}	2^{32}	[2]
	Free-start	6/10	2^{32}	2^{16}	[3]
AES-192-DM	Collision	5/12	2^{61}	Negligible	[Ours]
	Semi-free-start	7/12	2^{62}	Negligible	[Ours]
AES-256-DM	Collision	6/14	2^{61}	Negligible	[Ours]
	Collision*	9/14	2^{58}	2^{55}	[Ours]
	Semi-free-start	9/14	2^{30}	Negligible	[Ours]
	q pseudo-collision	14/14	$q \cdot 2^{67}$	Negligible	[4]

* it is a two-block collision

Outline

Specification of AES

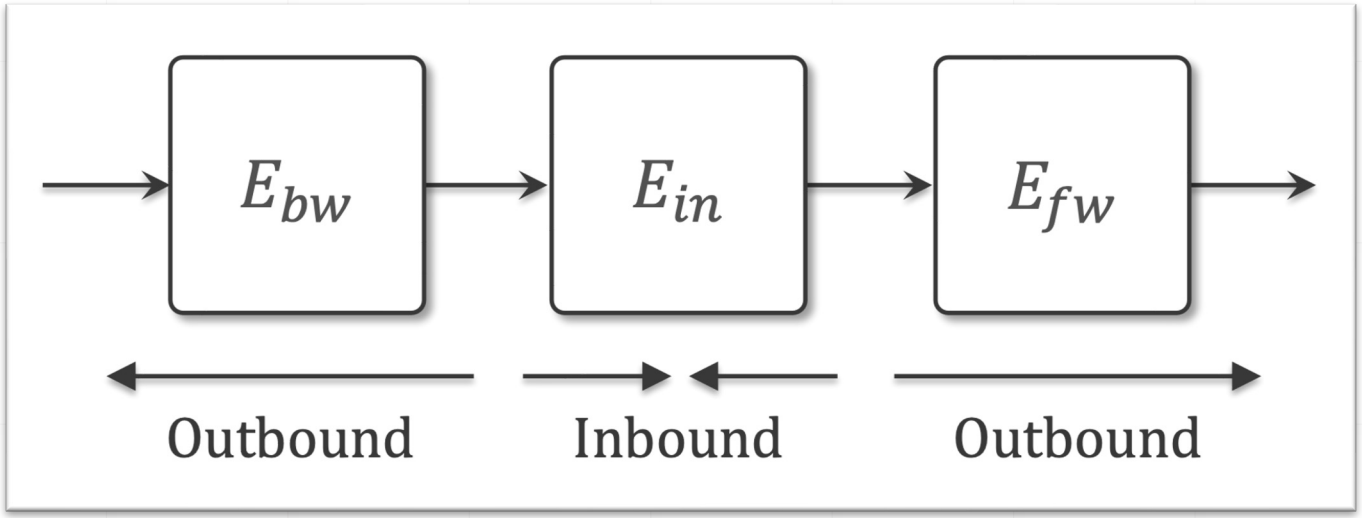
- **SubBytes(SB).** Parallel execution of 8-bit S-boxes.
- **ShiftRows(SR).** Row-wise shuffle operation.
- **MixColumns(MC).** Column-wise 4 × 4 matrix multiplication over the finite field with the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$
- **AddRoundKey(AK).** Application of the 128-bit round key.



Rebound Attack（Mendel et. al., FSE2009） [5]

- **Inbound phase** aims to find as many value pairs as possible in E_{in} where the differential probability is low.
  - DoF（Degrees of Freedom）: Value pairs satisfying the differential characteristic in E_{in}

- **Outbound phase** aims to obtain valid differential characteristic in both forward and backward direction through E_{fw} and E_{bw} to find desired collision.



Outline

Our Approach

Problem 1. Key differences should be canceled out by themselves without plaintext difference

Problem 2. Limitations in controlling plaintext values

Automatic tool to find Target-plaintext key collision

- Utilize bit-wise differential characteristics explored by SAT solver
- Automatically group the internal state into independently computable components and evaluate the computational complexity

Automatic tool for key collision

Step1.

Finding differential path for key collision

Step2.

converting bit-wise differential path into graphical expression

Step3.

Determining attack range

Step4.

Calculating DoF and check validity

Step5.

Grouping vertex and making DoF tree

Step6.

Calculating Attacking complexity

Automatic tool for key collision

Step1.

Finding differential path for key collision

Automatic tool for key collision

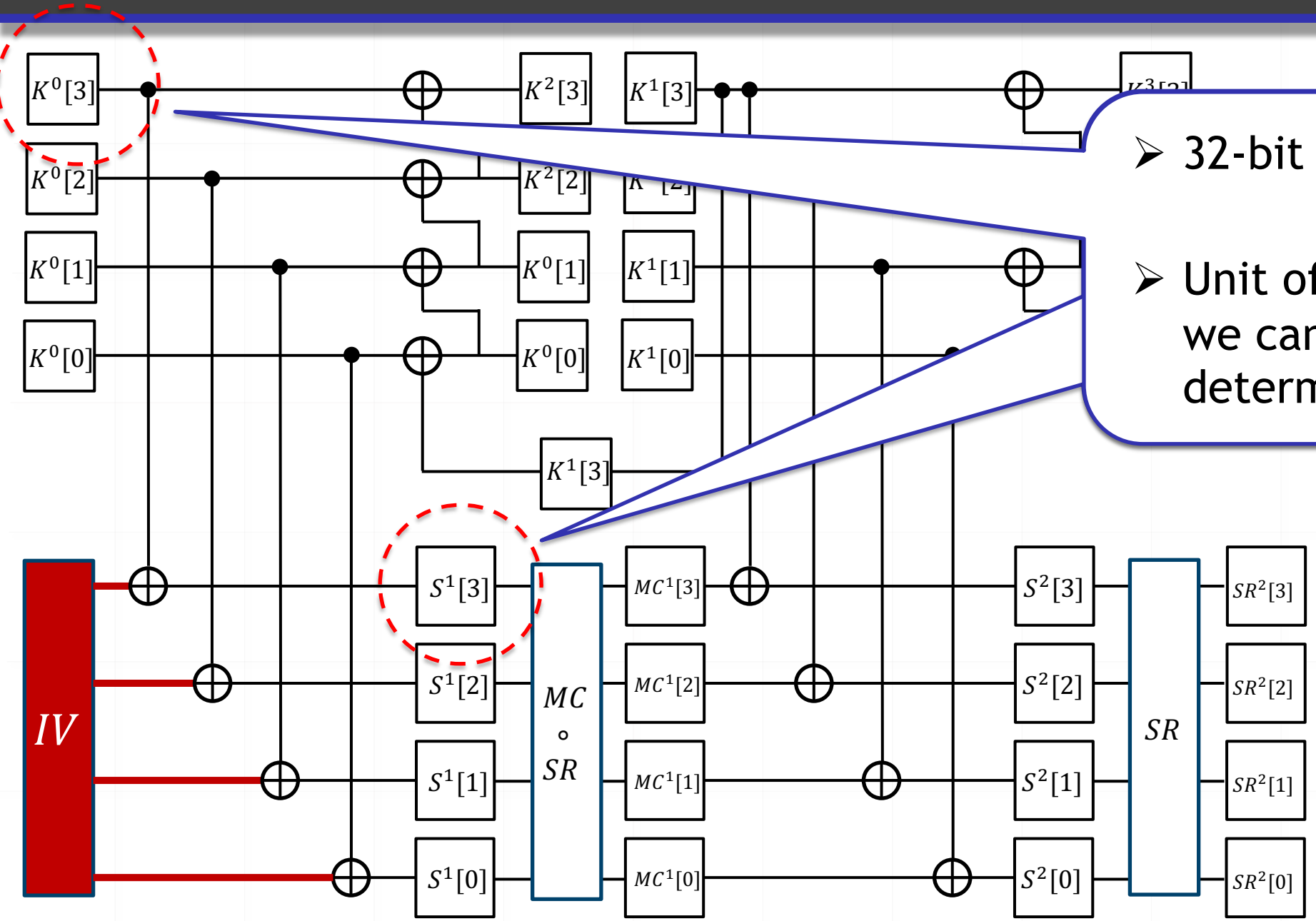
Step1.

Finding differential path for key collision

Step2.

converting bit-wise differential path
into graphical expression

Converting differential characteristics into graphical expression



- 32-bit vertex
- Unit of internal states that we can independently determine the values

Automatic tool for key collision

Step1.

Finding differential path for key collision

Step2.

converting bit-wise differential path into graphical expression

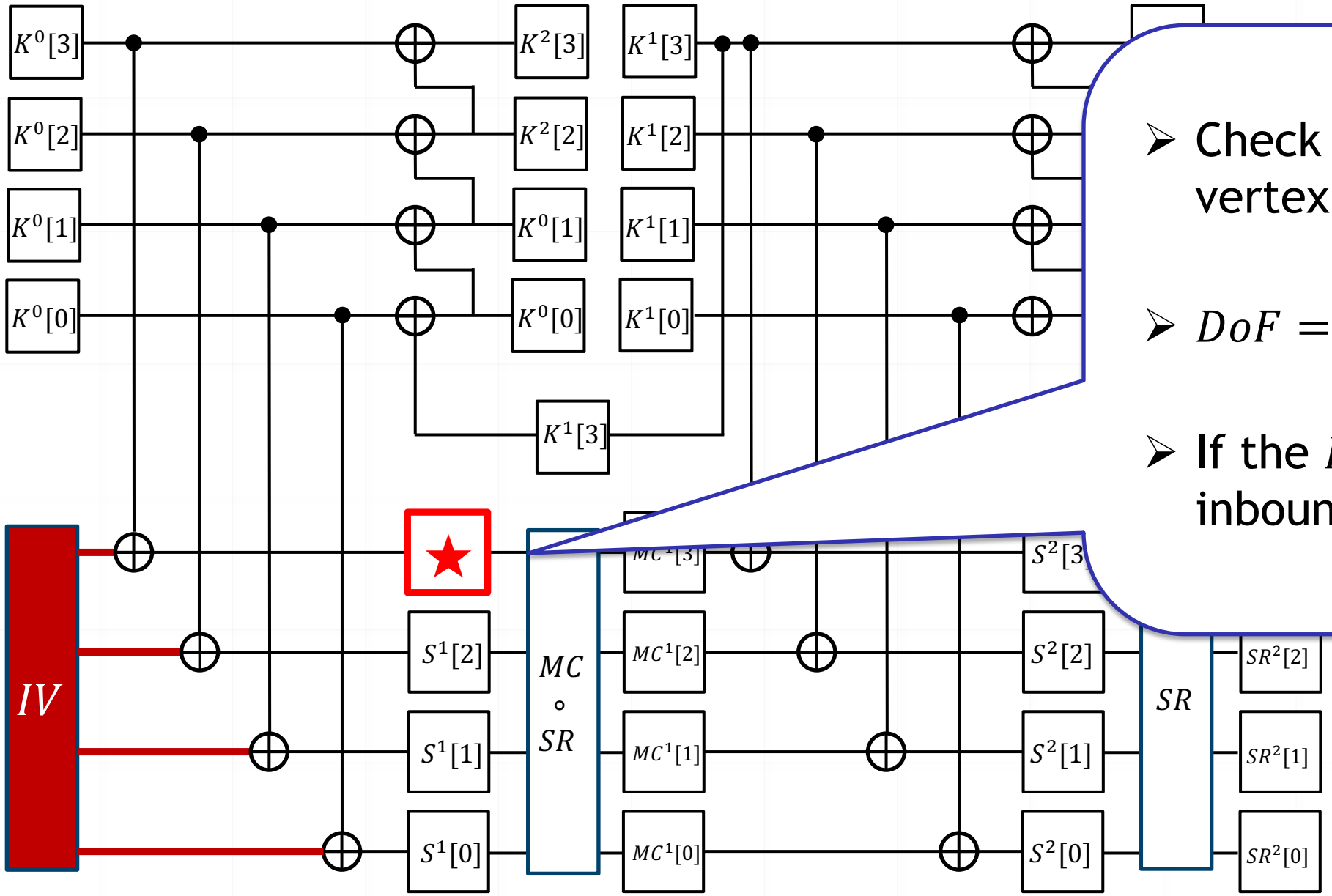
Step3.

Determining attack range

Step4.

Calculating DoF and check validity

Check validity of inbound vertex



- Check validity of inbound vertex
- $DoF = 2^{32} \cdot 2^{-p}$
- If the $DoF < 2^0$, the inbound phase is faild.

Automatic tool for key collision

Step1.

Finding differential path for key collision

Step2.

converting bit-wise differential path into graphical expression

Step3.

Determining attack range

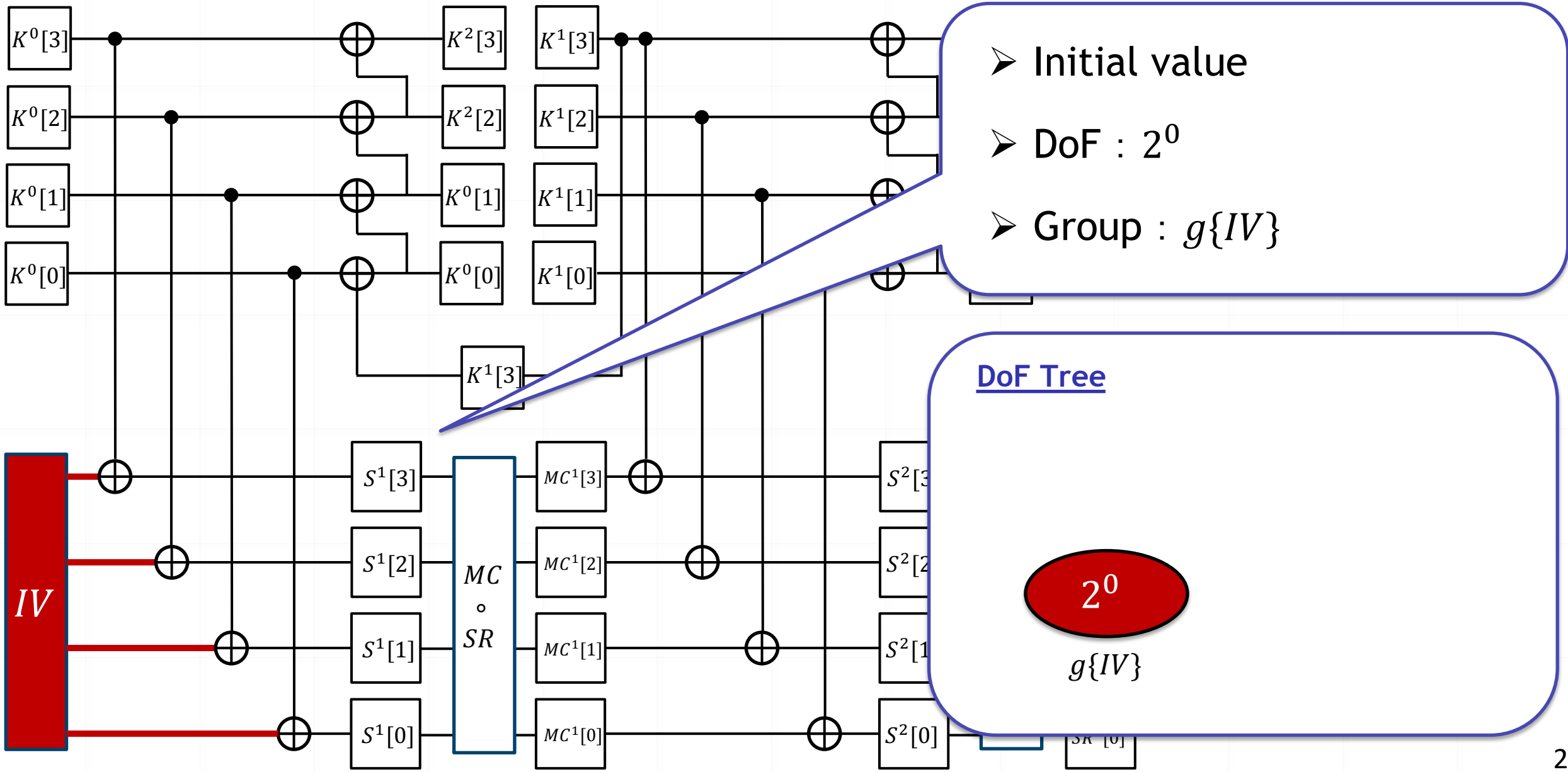
Step4.

Calculating DoF and check validity

Step5.

Grouping vertex and making DoF tree

Grouping vertices



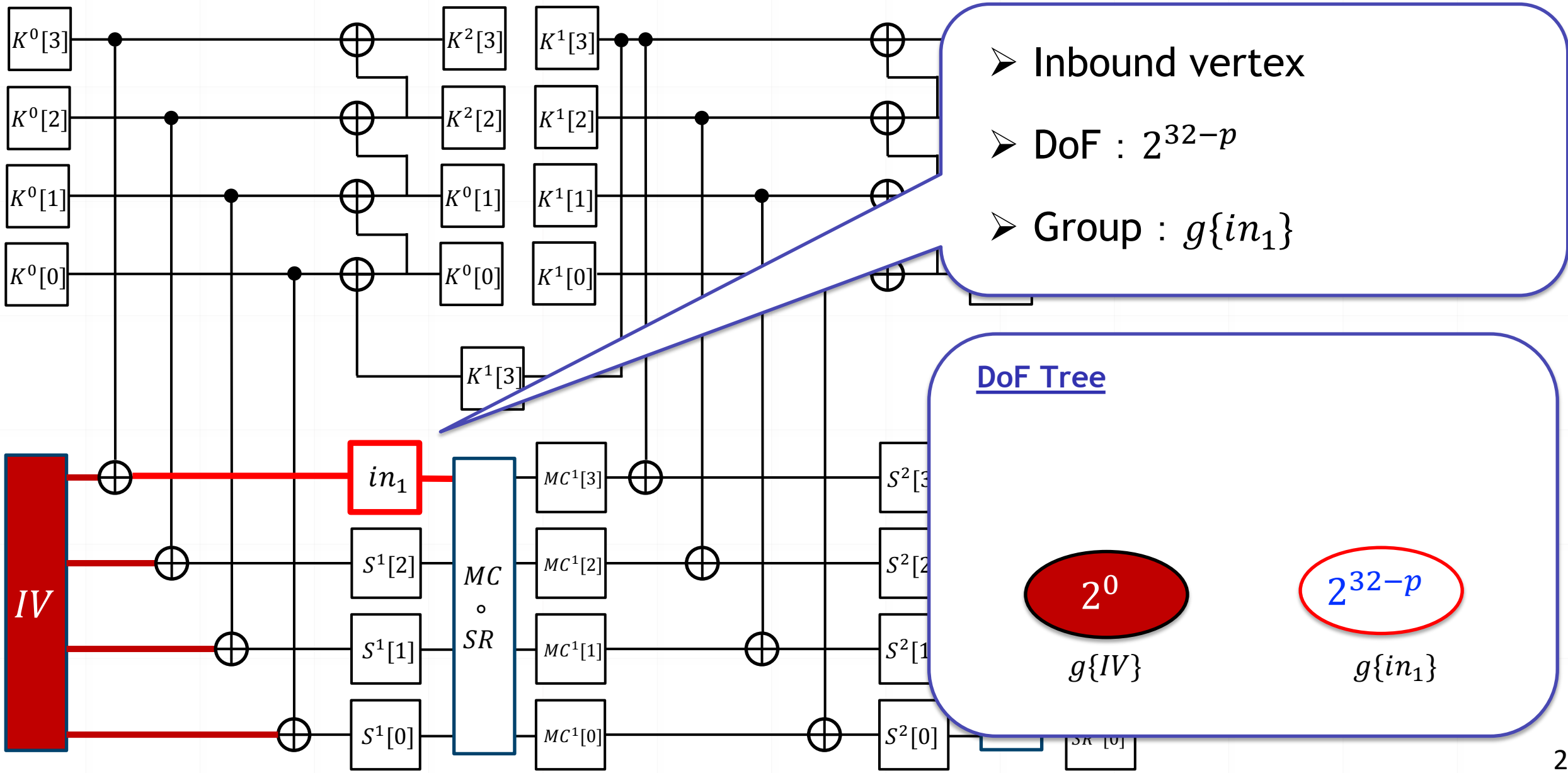
- Initial value
- DoF : 2^0
- Group : $g\{IV\}$

DoF Tree

2^0

$g\{IV\}$

Grouping vertices



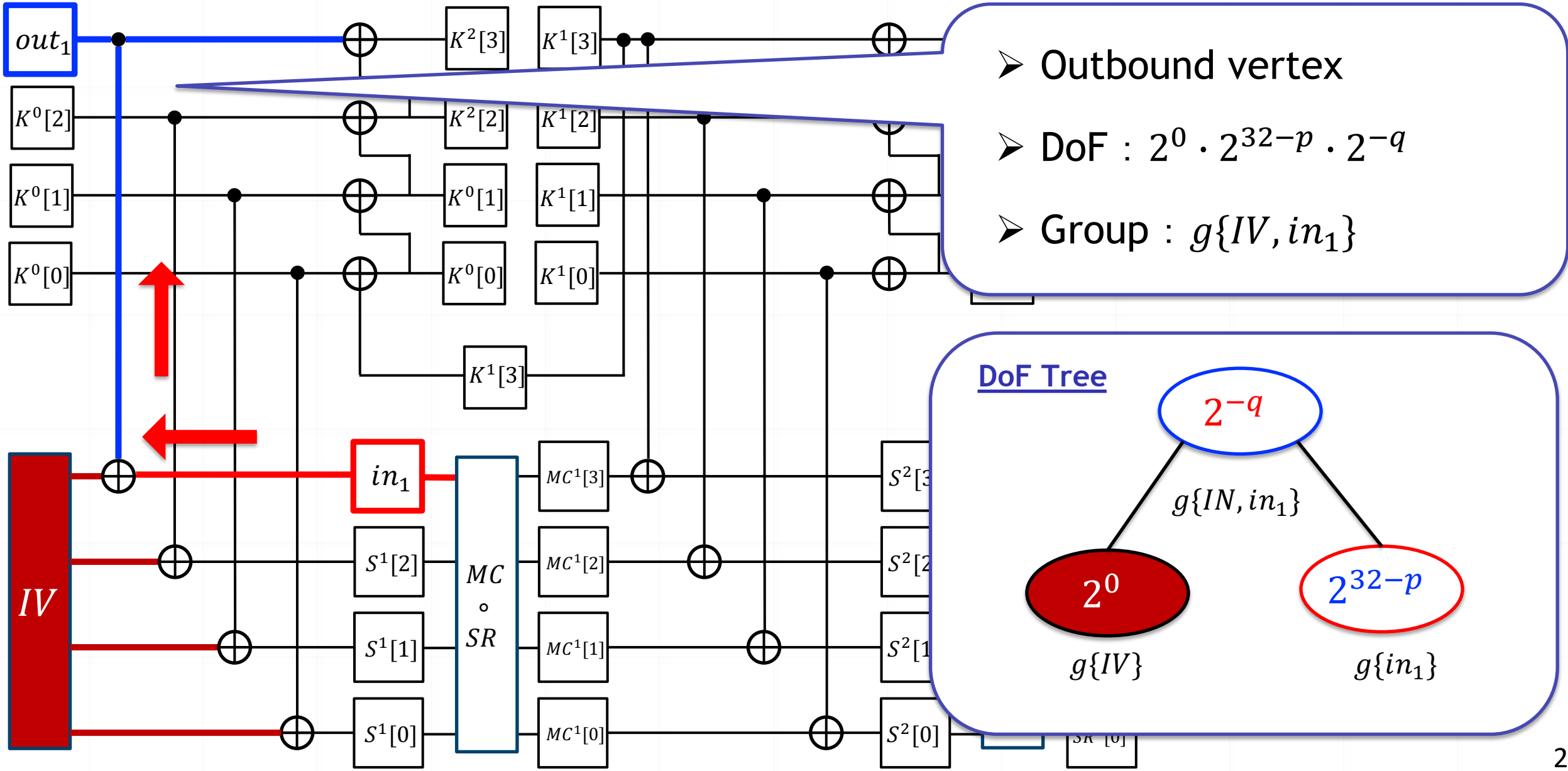
- Inbound vertex
- DoF : 2^{32-p}
- Group : $g\{in_1\}$

DoF Tree

2^0 — $g\{IV\}$

2^{32-p} — $g\{in_1\}$

Grouping vertices



Automatic tool for key collision

Step1.

Finding differential path for key collision

Step2.

converting bit-wise differential path into graphical expression

Step3.

Determining attack range

Step4.

Calculating DoF and check validity

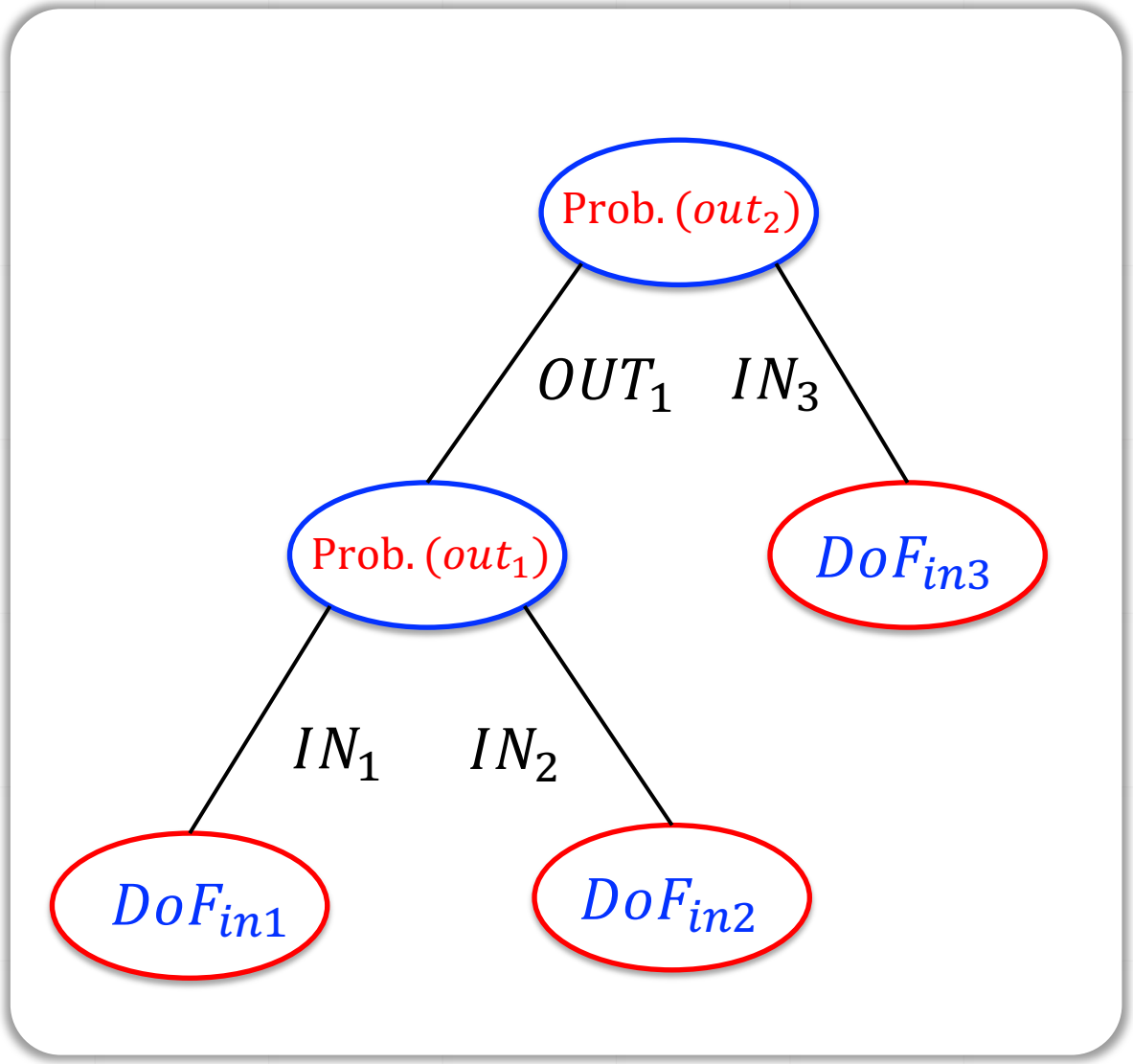
Step5.

Grouping vertex and making DoF tree

Step6.

Calculating Attacking complexity

Estimating attack complexity



- in_k, out_k

  → Group of inbound or outbound vertex

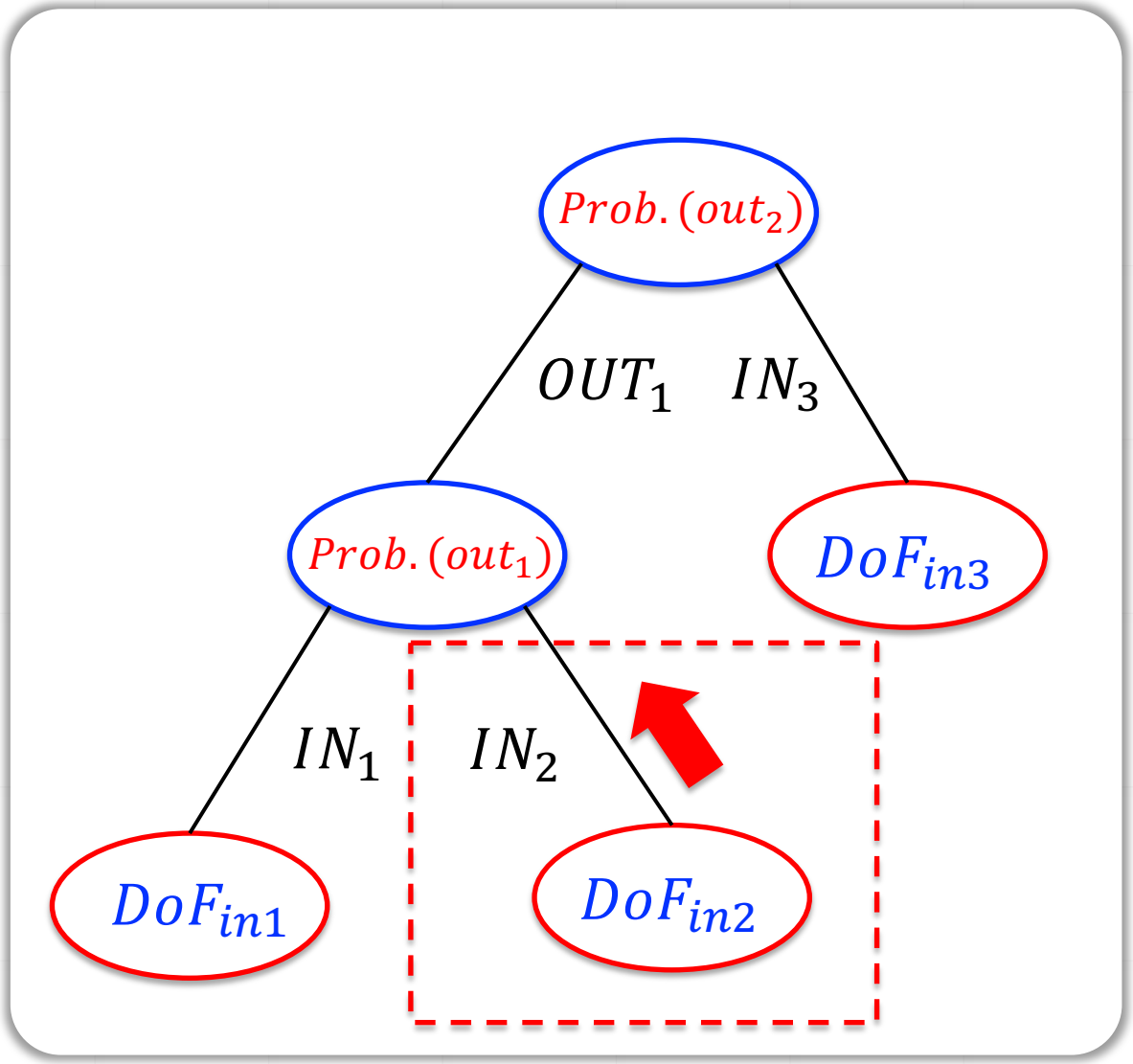
- $IN_k,\ OUT_k,\ DoF_{in_k},\ \text{Prob.}\,(out_k)$

  → Integer

- $DoF_{in_k},\ \text{Prob.}\,(out_k)$

  → The constant determined through grouping

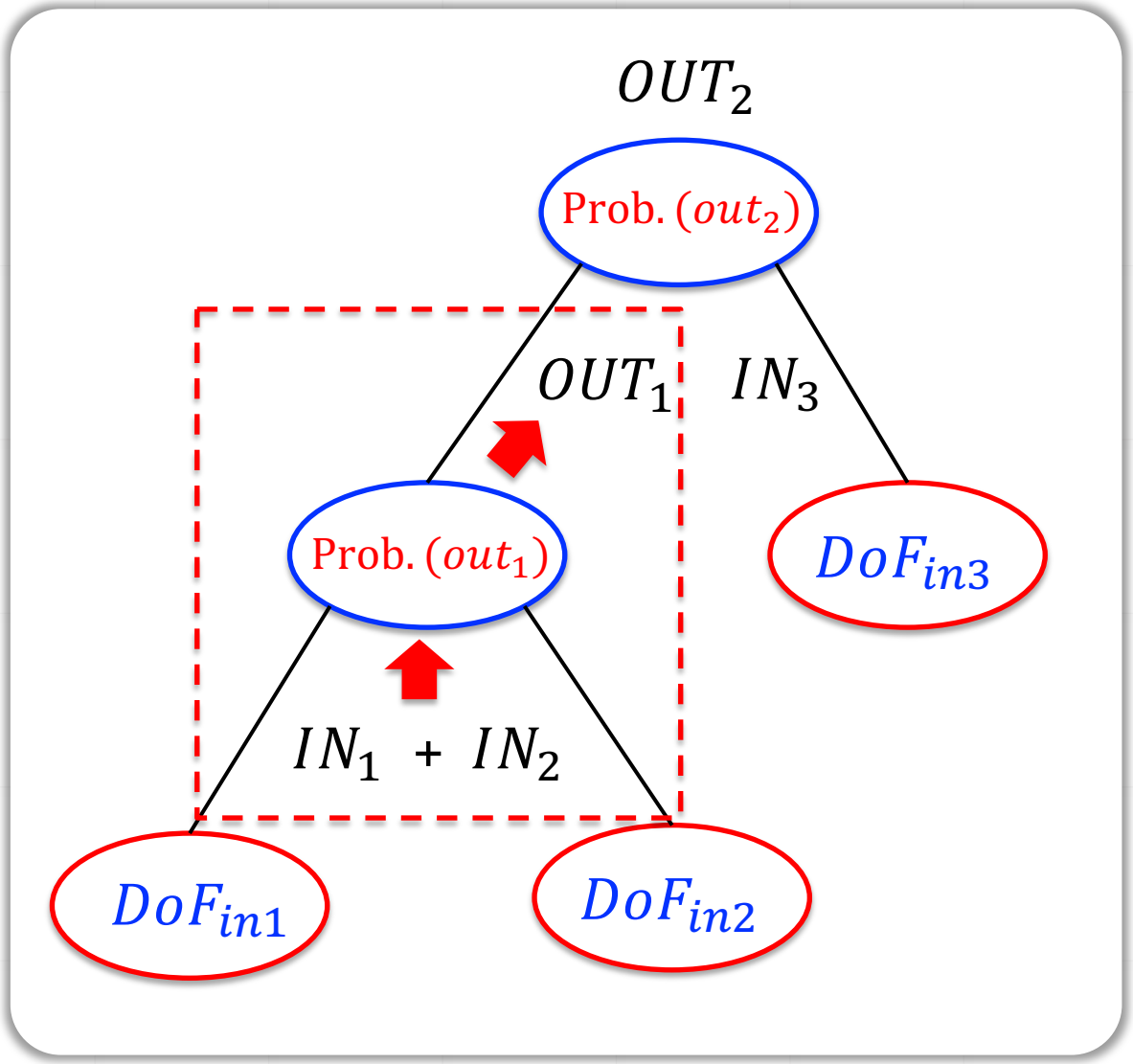
Estimating attack complexity



- The limitation on the DoF for inbound vertices
  - DoF_{in_k} : The maximum amount of values that can be used in group in_k
  - IN_k : The actual number of value pairs used in the attack

$$(1)\ \boldsymbol{IN_1 \leq DoF_{in_1}}$$
$$(2)\ \boldsymbol{IN_2 \leq DoF_{in_2}}$$
$$(3)\ \boldsymbol{IN_3 \leq DoF_{in_3}}$$

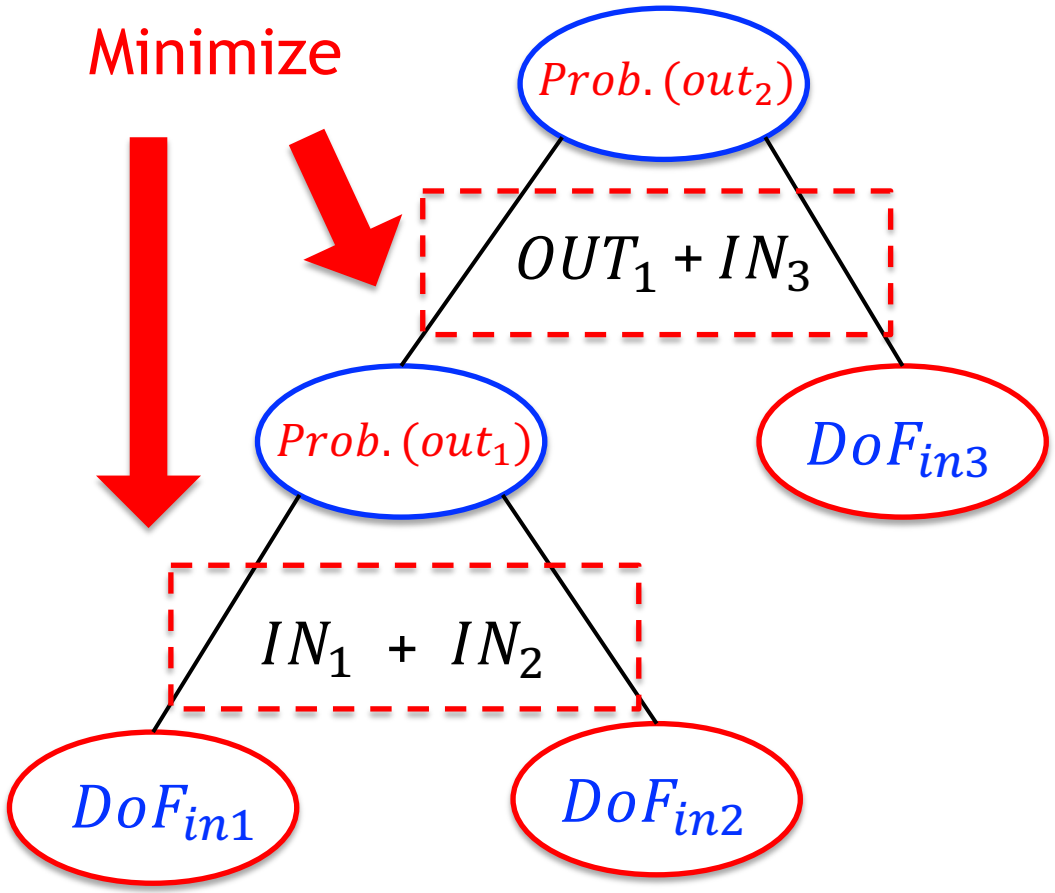
Estimating attack complexity



- Degrees of Freedom of outbound vertices
  - OUT_k : The actual number of value pairs used in the attack
  - $\mathrm{Prob.}(g)$: The total probability in group g

$$(3)\ \boldsymbol{OUT_1 = (IN_1 + IN_2) - Prob.(out_1)}$$
$$(4)\ \boldsymbol{OUT_2 = (OUT_1 + IN_3) - Prob.(out_2)}$$

Estimating attack complexity



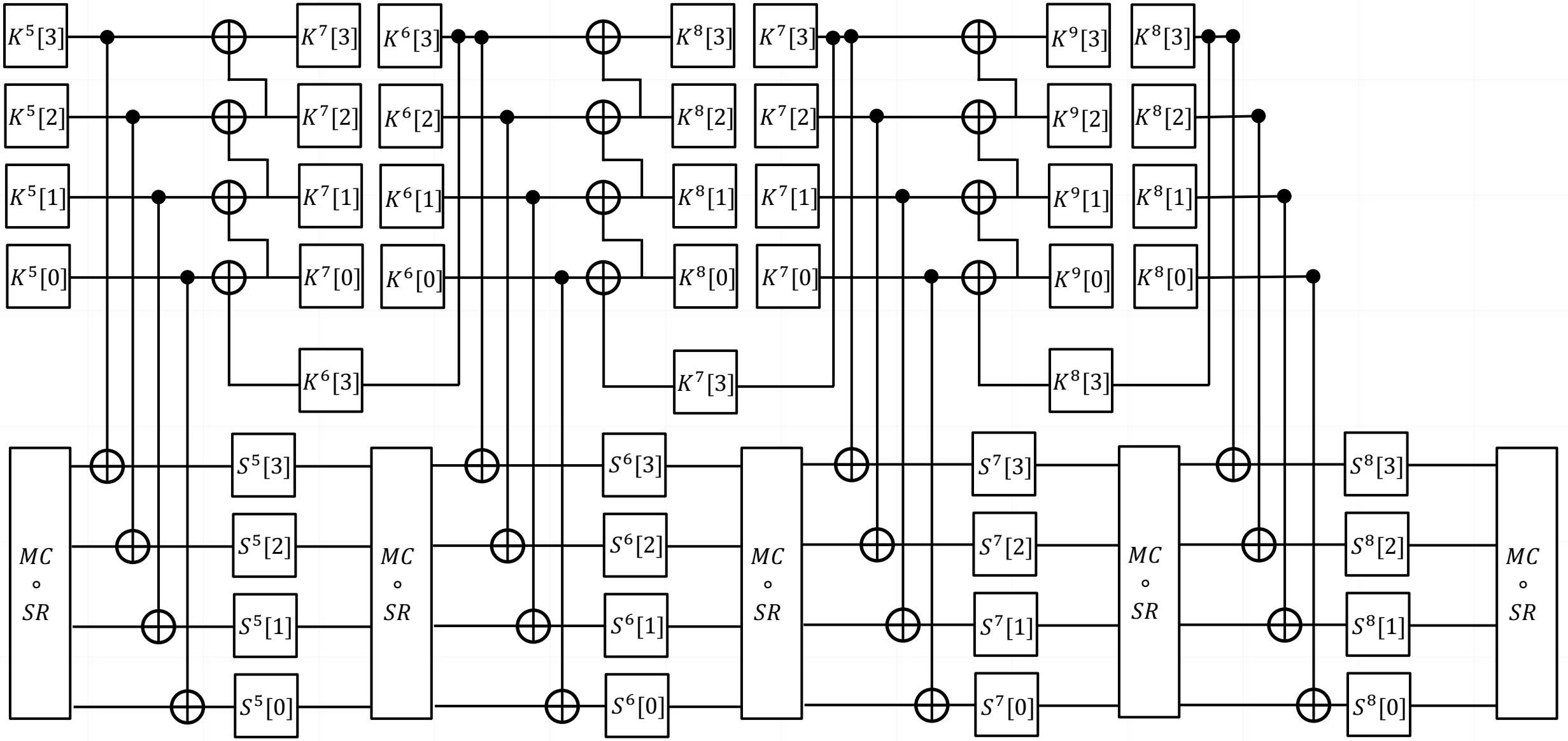
- Constraints of time complexity
  - T^i_{max} : The computational cost during outbound phase
  - T : The maximum value of T^i_{max}

$$(5)\ \boldsymbol{T^1_{max} \geq IN_1 + IN_2},$$
$$(6)\ \boldsymbol{T^2_{max} \geq OUT_1 + IN_3}$$

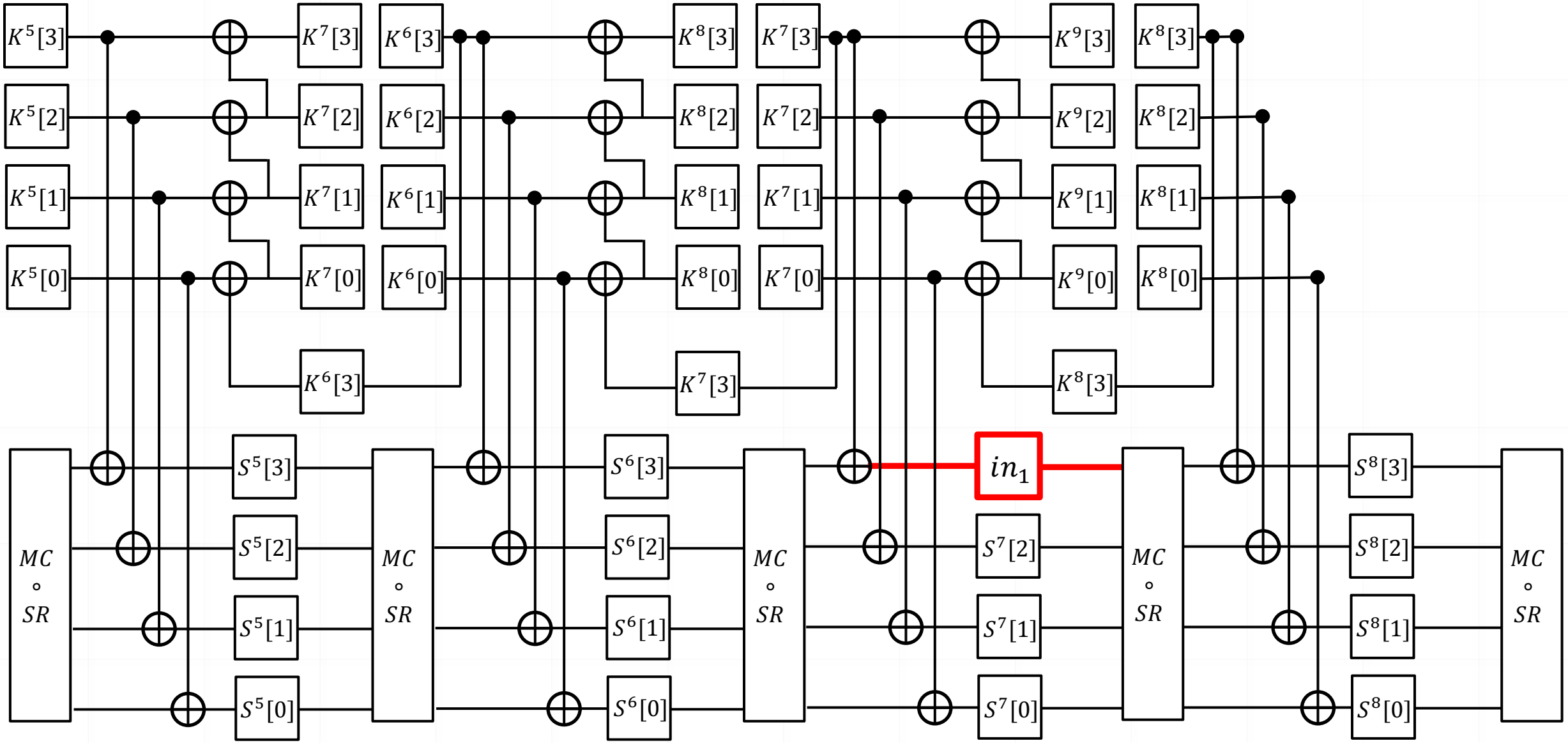
$$(7)\ \boldsymbol{T \geq T^1_{max}}$$
$$(8)\ \boldsymbol{T \geq T^2_{max}}$$

Outline

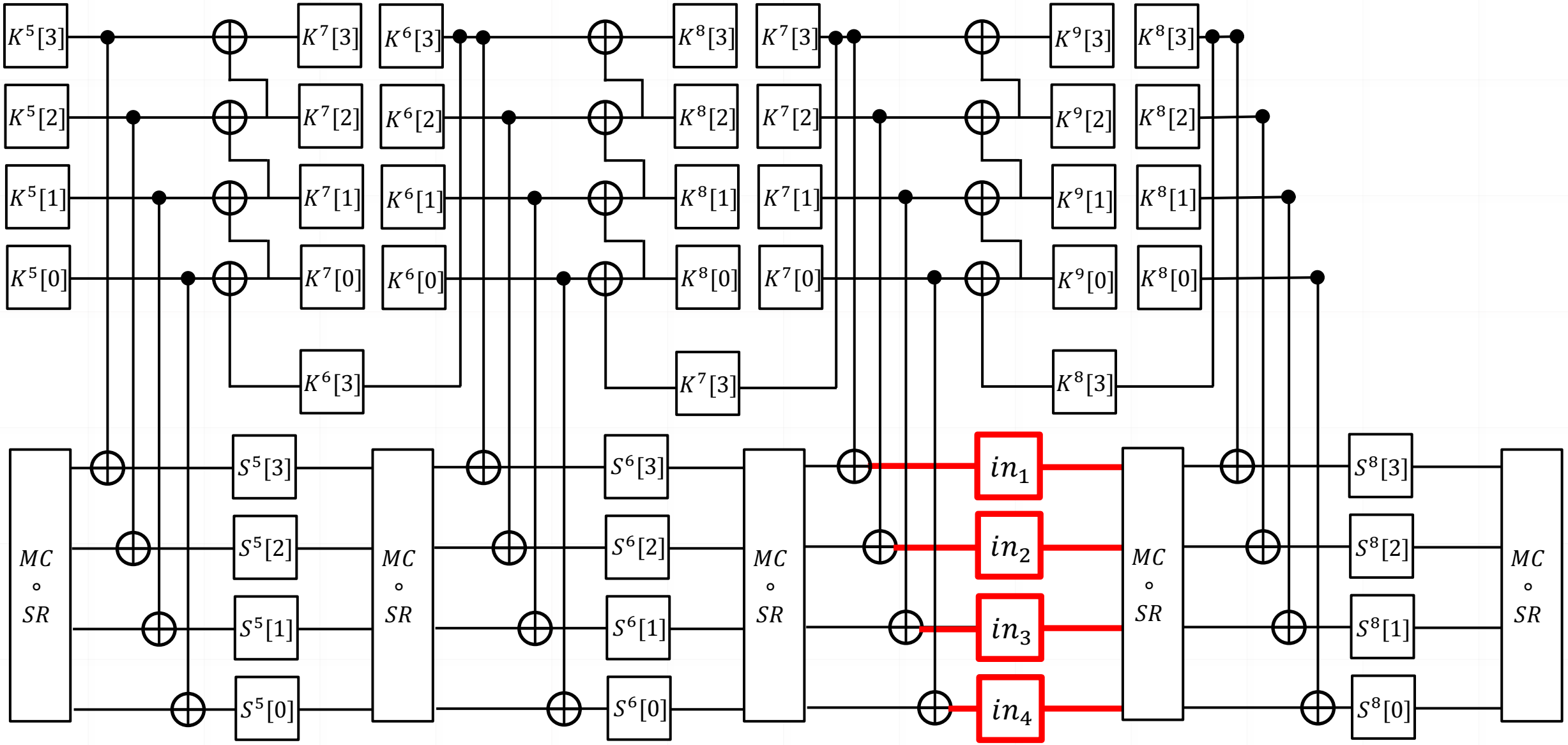
Application to AES256(Free-target-plaintext collision)



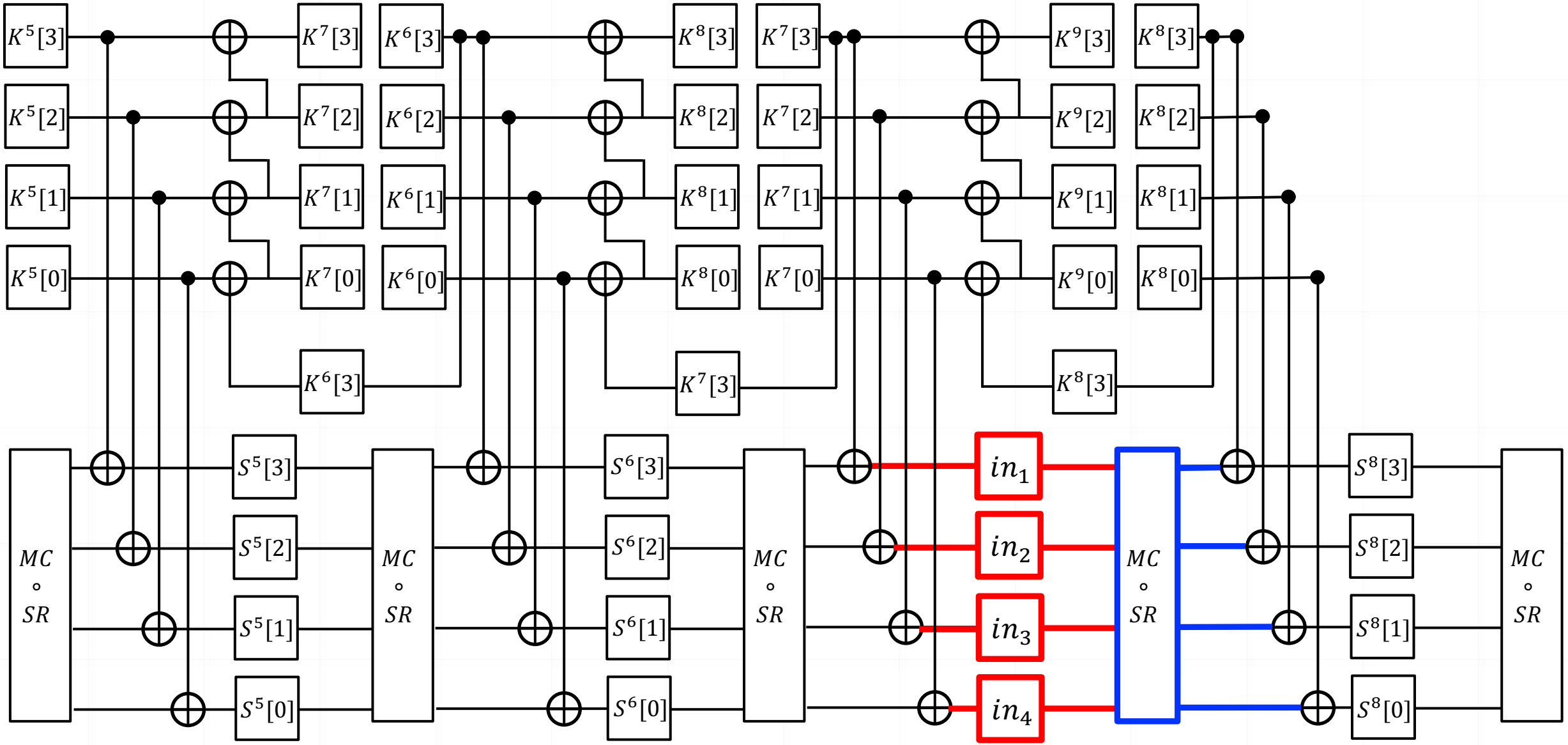
Application to AES256(Free-target-plaintext collision)



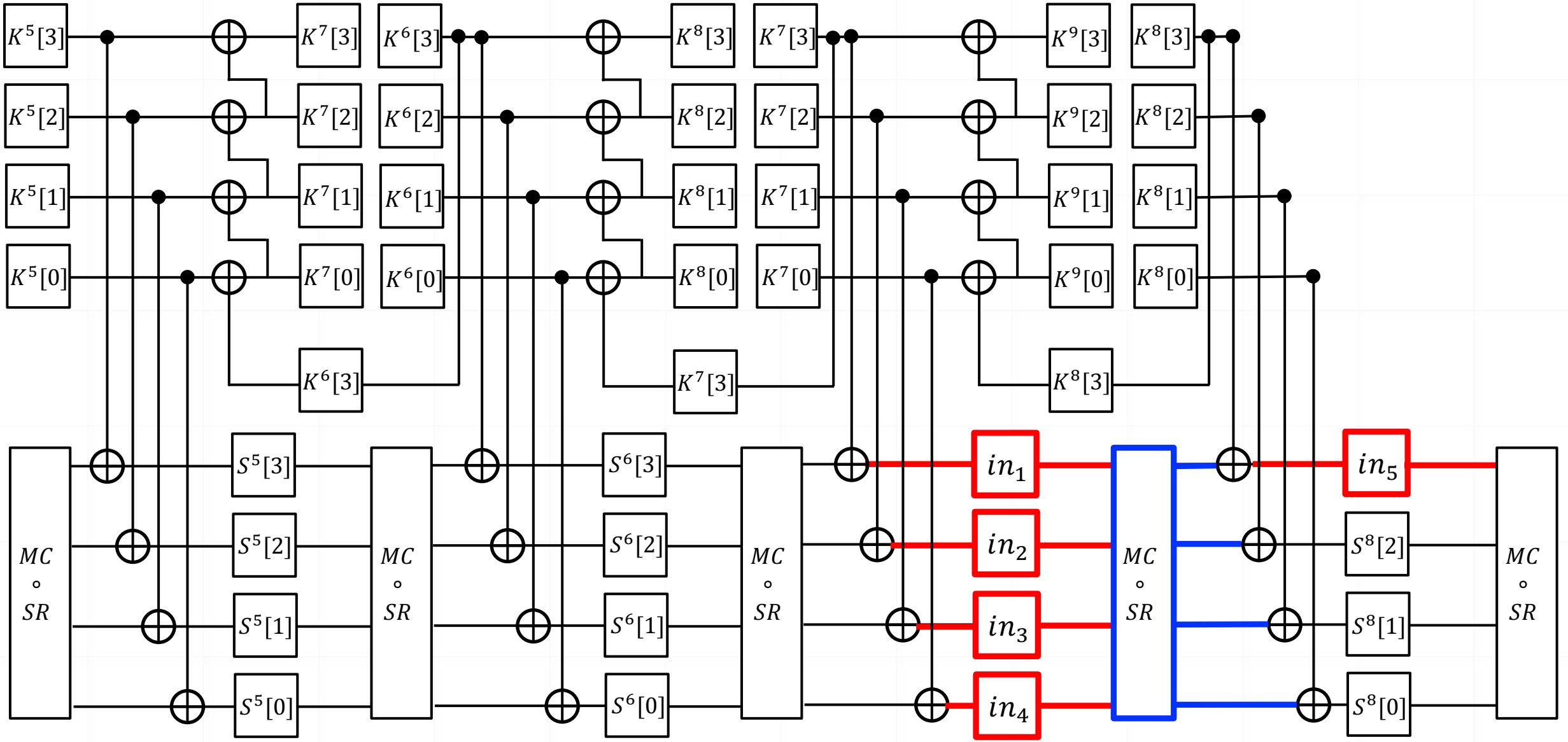
Application to AES256(Free-target-plaintext collision)



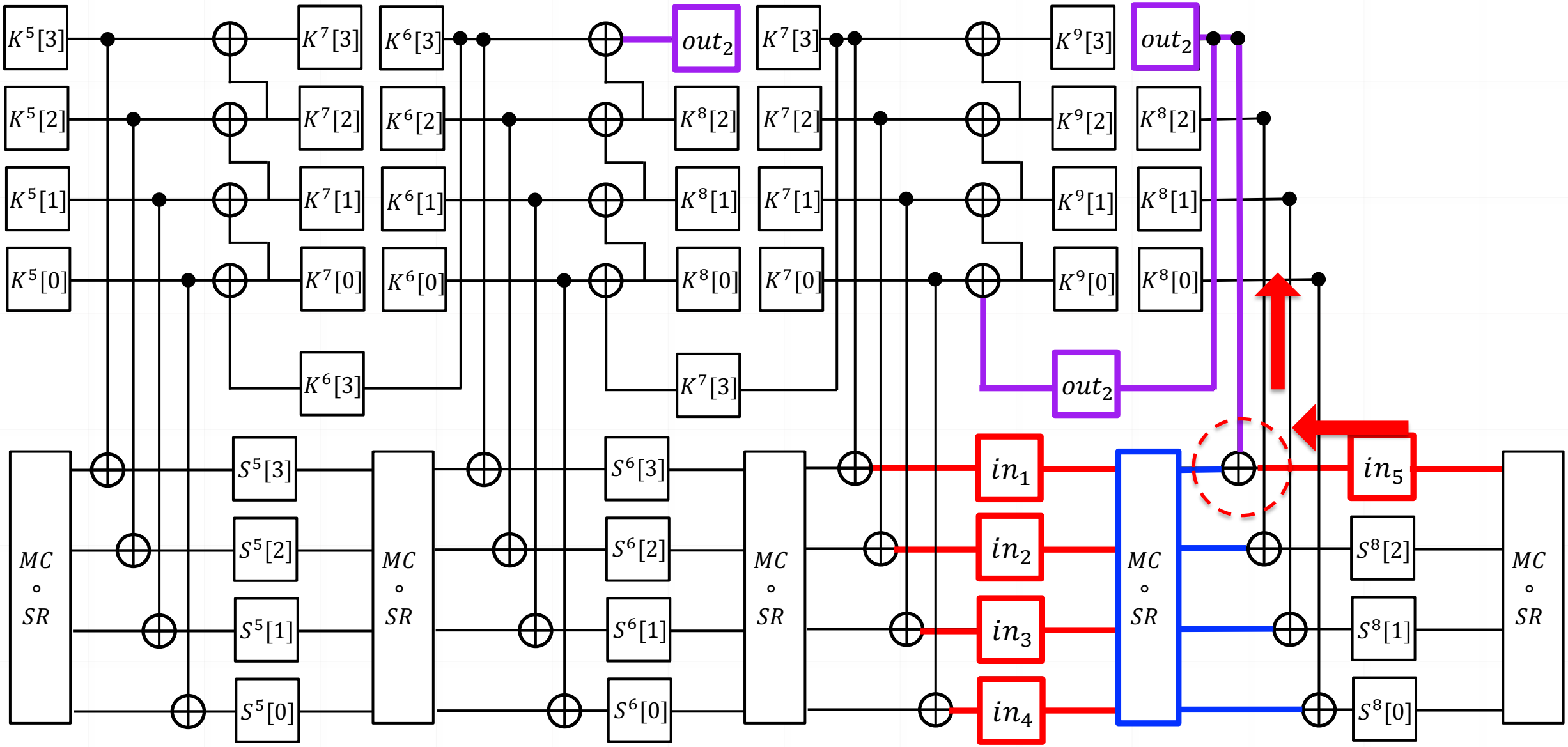
Application to AES256(Free-target-plaintext collision)



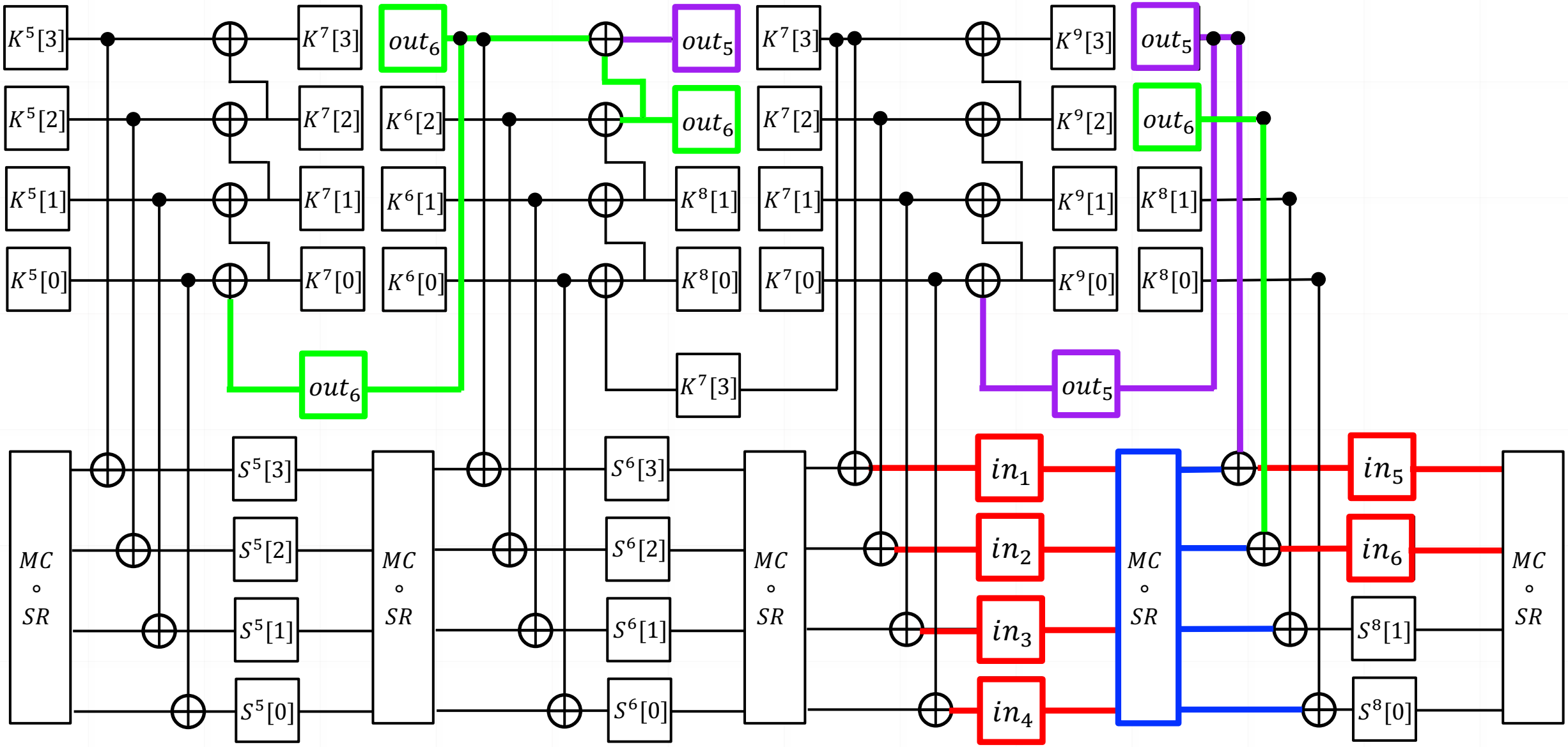
Application to AES256(Free-target-plaintext collision)



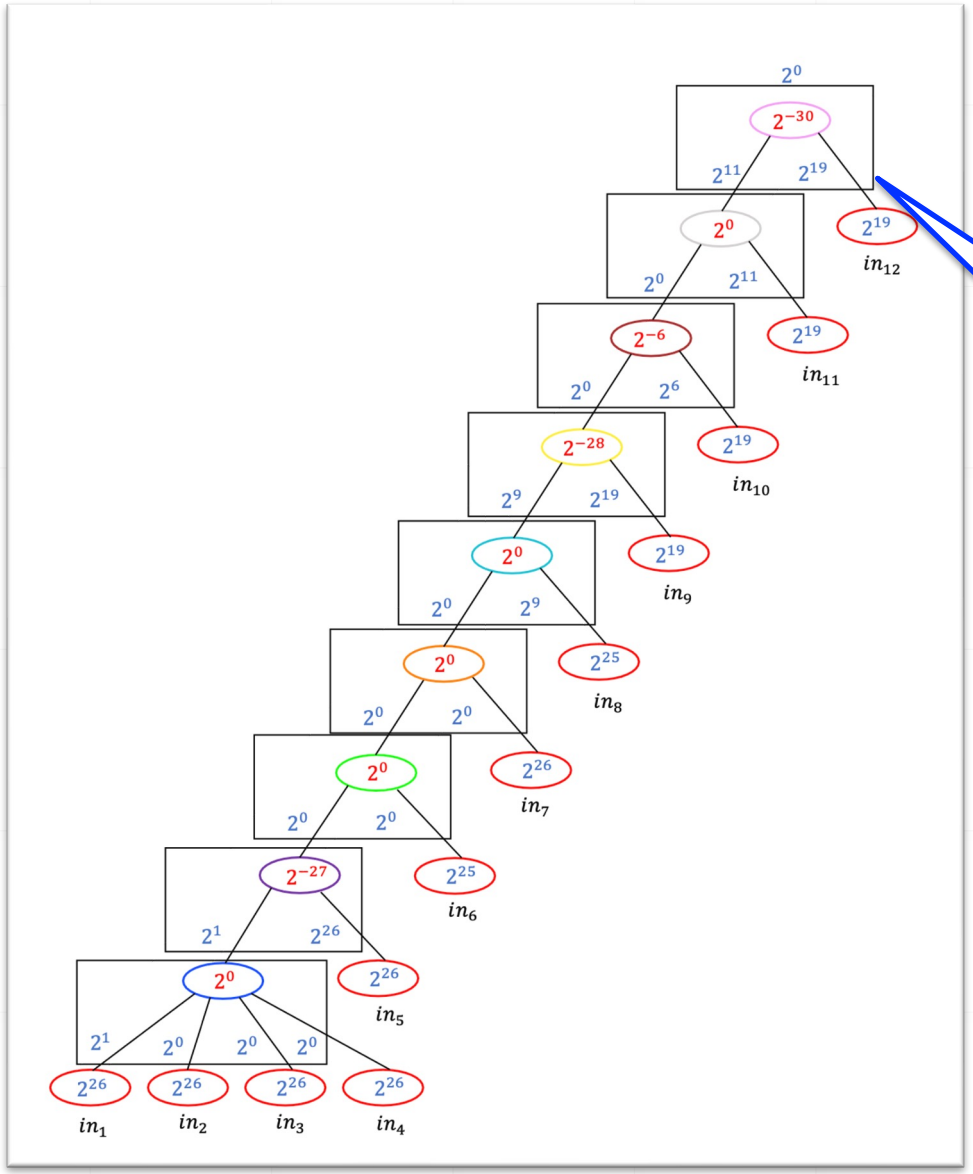
Application to AES256(Free-target-plaintext collision)



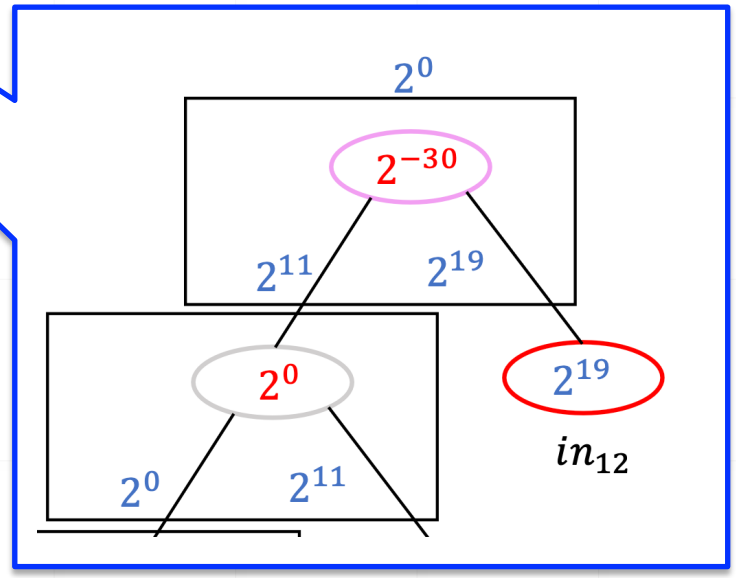
Application to AES256(Free-target-plaintext collision)



Application to AES256(Free-target-plaintext collision)



- Probability 2^{-30} is dominant
  - Time complexity : 2^{30}



Experimental verification

- Execution environment
  - AMD Ryzen Threadripper ™ PRO 5995WX @2.7GHz (64C/128T)
  - 512GB RAM

i	$Plaintext_i$	Key_i	$Ciphertext_i$
1	83 66 63 dc b1 bc 61 82 30 38 ab f7 14 c3 d4 6a	ca 45 20 ea 30 3c c2 06 7e 2f d9 46 96 2a 82 ef	7f ea d8 40 c0 59 30 d5 11 29 07 d0 39 08 5a 65
2	83 66 63 dc b1 bc 61 82 30 38 ab f7 14 c3 d4 6a	35 45 20 ea cf 3c c2 06 94 5a ac d9 7c 5f f7 70	7f ea d8 40 c0 59 30 d5 11 29 07 d0 39 08 5a 65

Summary

- We defined a new variant of key collision, called target-plaintext key collision
- We proposed an automatic tool for key collisions
  - It utilizes bit-wise differential characteristics explored by a SAT solver
  - It automatically groups the internal state into independently computable components. It then evaluates the computational complexity.
- Applied the tool to AES-128/192/256 and uncovered fixed-target-plaintext key collision and free-target-plaintext key collision

Rsults

Target	Attack	Rounds	Time	Memory	source
AES-128-DM	Collision	2/10	2^{49}	Negligible	[Ours]
	Collision*	3/10	2^{60}	2^{52}	[Ours]
	Semi-free-start	5/10	2^{57}	Negligible	[Ours]
	Free-start	5/10	2^{56}	2^{32}	[2]
	Free-start	6/10	2^{32}	2^{16}	[3]
AES-192-DM	Collision	5/12	2^{61}	Negligible	[Ours]
	Semi-free-start	7/12	2^{62}	Negligible	[Ours]
AES-256-DM	Collision	6/14	2^{61}	Negligible	[Ours]
	Collision*	9/14	2^{58}	2^{55}	[Ours]
	Semi-free-start	9/14	2^{30}	Negligible	[Ours]
	q pseudo-collision	14/14	$q \cdot 2^{67}$	Negligible	[4]

* Two-block collision

Reference(1)

[1] Albertini, A., Duong, T., Gueron, S., Kölbl, S., Luykx, A., Schmieg, S.:
How to abuse and fix authenticated encryption without key commitment.
In: Butler, K.R.B., Thomas, K. (eds.) 31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022. pp. 3291–3308. USENIX Association (2022)

[2] Mendel,F.,Peyrin,T.,Rechberger,C.,Schläffer,M.:
Improved cryptanalysis of the reduced grøstl compression function, ECHO permutation and AES block cipher.
In: Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 5867, pp. 16–35. Springer (2009)

Reference(2)

[3] Jean, J., Naya-Plasencia, M., Peyrin, T.
Multiple limited-birthday distinguishers and applications.
In: Selected Areas in Cryptography. Lecture Notes in Computer Science,vol. 8282, pp. 533–550. Springer (2013)

[4] Biryukov, A., Khovratovich, D., Nikolic, I.
Distinguisher and related-key attack on the full AES-256.
In: CRYPTO. Lecture Notes in Computer Science, vol. 5677, pp. 231–249. Springer (2009)

[5] Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.
The rebound attack: Cryptanalysis of reduced whirlpool and grøstl.
In: FSE.LectureNotesinComputerScience, vol. 5665, pp. 260–276. Springer (2009)