NTT

Cryptanalysis of MORUS

(Initially discussed at Lorentz center in Mar 2018)

Tomer Ashur	imec-COSIC KU Leuven
Maria Eichlseder	Graz University of Technology
Martin M. Lauridsen	
Gäetan Leurent	Inria
Brice Minaud,	Royal Holloway University of London
Yann Rotella	Inria
Yu Sasaki	NTT Secure Platform Laboratories
Benoît Viguier	Radbond University

13 Nov 2018 @ ASK2018

Outline

- Background and MORUS specification
- MiniMORUS and its linear trails
- Extension to Full MORUS (omit details)
- Observations for Initialization and Finalization
- Conclusion

Remarks: Paper Title Collision

After the galley-proof of our paper submission, we realized the following paper.

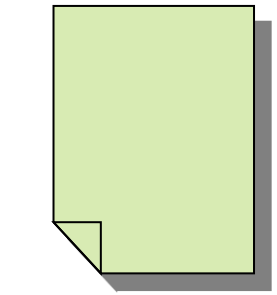
Yanbin Li and Meiqin Wang. "*Cryptanalysis of MORUS*". Designs, Codes and Cryptography, pages 1—24, First Online: 09 June 2018

(Our paper was submitted to ePrint on 17 May 2018)

MILP-aided search for reduced MORUS.

- Integral distinguishers for 6.5 steps of MORUS-640.
- Differential distinguishers for 4.5 steps of MORUS-1280.

Authenticated Encryption (AE)

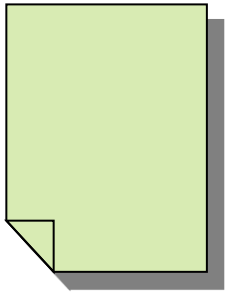


message

privacy → Encryption

authenticity → MAC

independently computed



message

privacy + authenticity → Authentication Encryption

all-in-one

- Simple security discussion
- Higher performance

History of CAESAR

- Competition to determine portfolio of authenticated encryption (AE) schemes

R1: From March 2014 with 58 candidates

R2: From July 2015 with 29 candidates

R3: From August 2016 with 15 candidates

RF: From March 2018 with 7 candidates

CAESAR Finalists

Innovative R&D by NTT

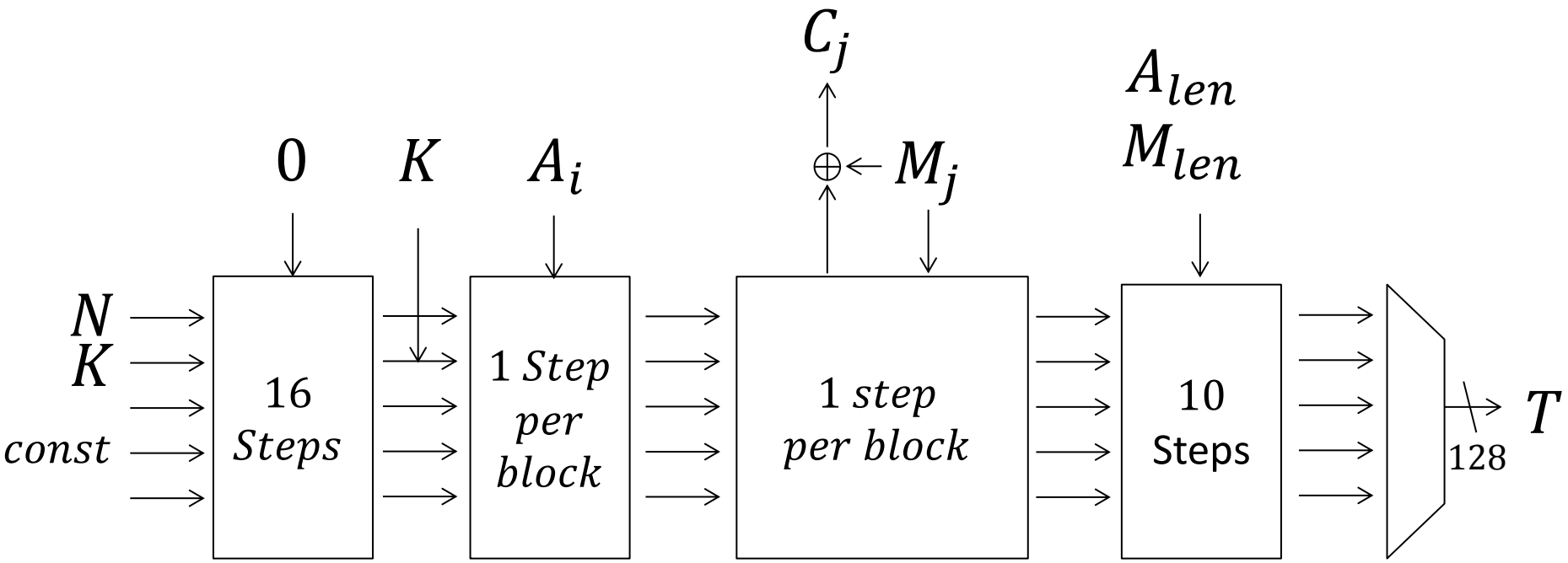
Low-end	High-end	Security
ACORN (dedicated)	**AEGIS** (dedicated)	COLM (online AE)
ASCON (sponge)	**MORUS** (dedicated)	Deoxys-II (robust AE)
	OCB (parallelizable)	

MORUS

MORUS

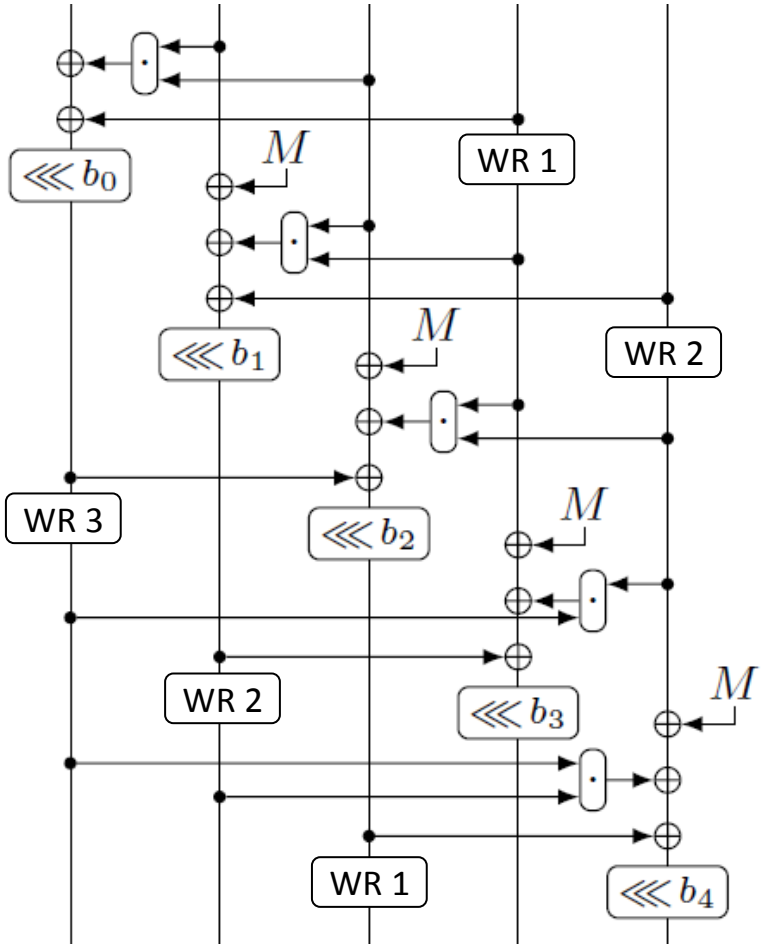
- Designed by Hongjun Wu and Tao Huang
- Suitable for SIMD instructions
- Stream-cipher like design
  - A big state (640 or 1280 bits) is initialized from nonce N and key K (heavy operation).
  - Encryption part is light.
- MORUS-640 for 128-bit key
- MOUS-1280 for 128- or 256-bit key

Overall Structure of MORUS

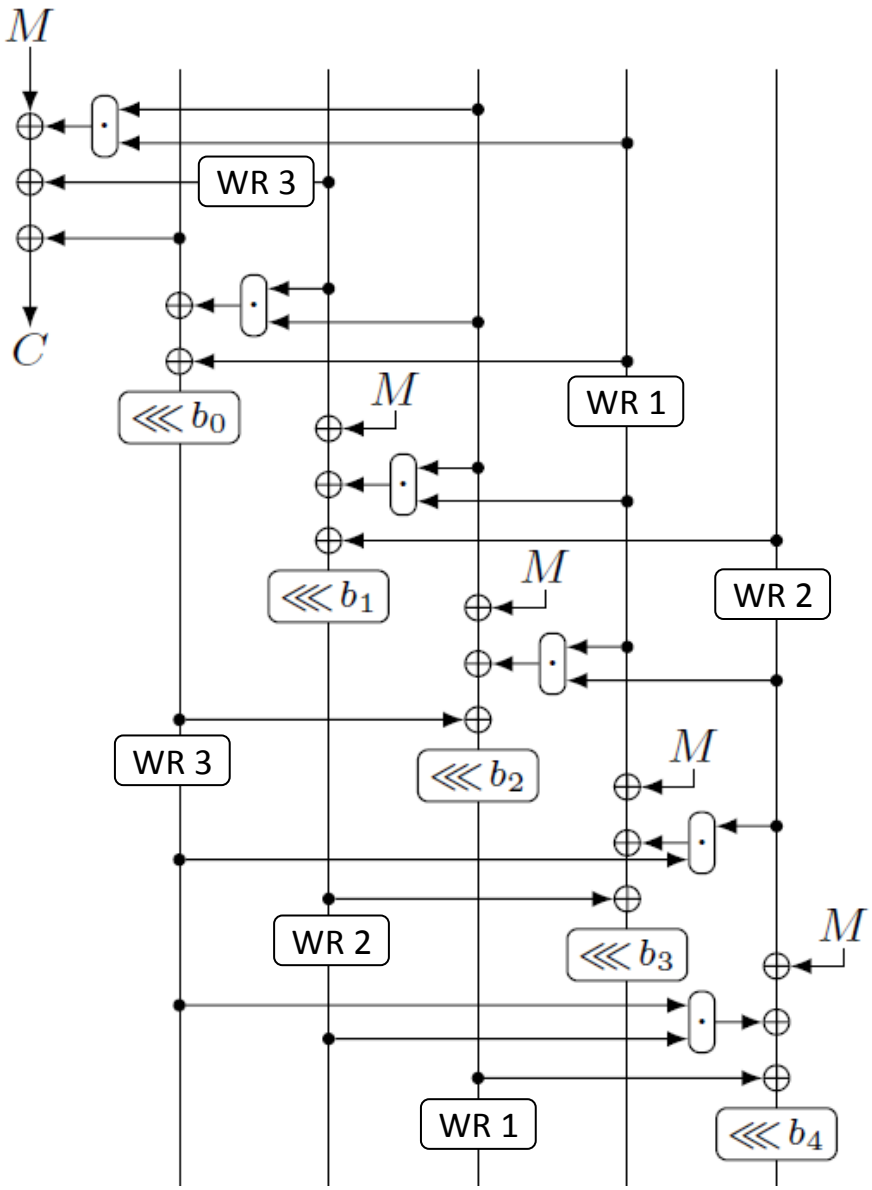


- Allow lines: $4w$-bit register, $w = 32$ and 64 for MORUS640 / MORUS1280
- Each register consists of 4 words of w bits.

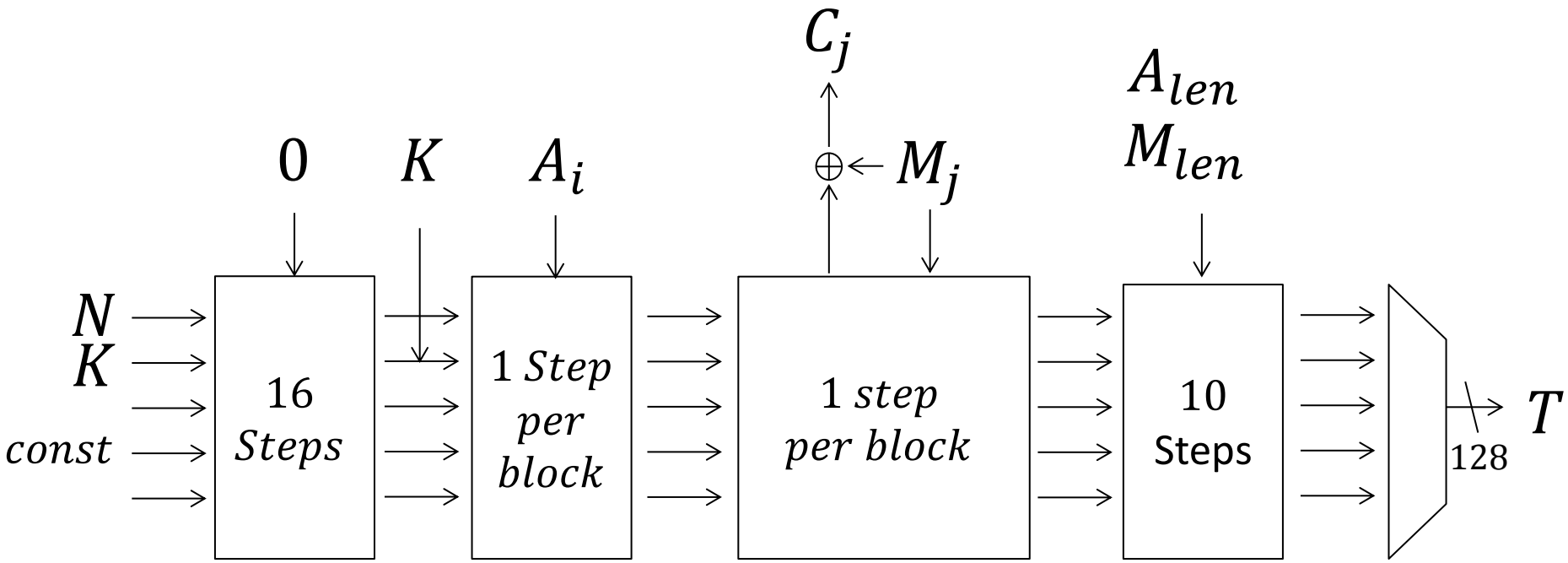
Step Function of MORUS



Step Function for Encryption



Aim to Analyze All Parts of MORUS



Reduced-round key-recovery in nonce respect

full brake of confidentiality

Reduced-round forgery

Security Claim

- Nonce respect security

	Confidentiality (bits)	Integrity (bits)
MORUS-640-128	128	128
MORUS-1280-128	128	128
MORUS-1280-256	256	128

Confidentiality of MORUS-1280-256 can be broken after 2^{152} encryptions.

Bias of Key Stream Generated by Encryption

Basics of Linear Cryptanalysis

- An event E with probability $\Pr(E) = \frac{1}{2} \pm \epsilon$ has bias ϵ.
- Correlation: $Cor(E) = 2\Pr(E) - 1 = 2\epsilon$
- Weight: $\text{weight}(E) = -\log_2 Cor(E)$

Piling-Up Lemma:

The correlation (resp. weight) of an XOR of independent variables is equal to their product (resp. sum).

Linear approximation of AND:

$\Pr(a \cdot b) = 0 \; or \; 1$ (weight 1)

$\Pr(a \cdot b) = a \; or \; b$ (weight 1)

$\Pr(a \cdot b) = a \oplus b$ (weight 1)

$\Pr(E) = 3/4 . \quad \epsilon. = 2^{-2}.$

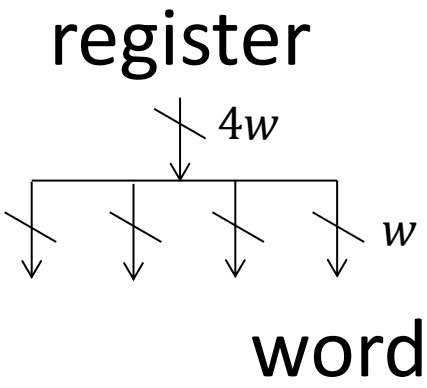
$Cor(E). = 2^{-1}.$

$weight(E) = 1$

Effect of E is detected by processing $2^{2 \cdot weight(E)}$ inputs.

Rotation-Invariant of Step Function

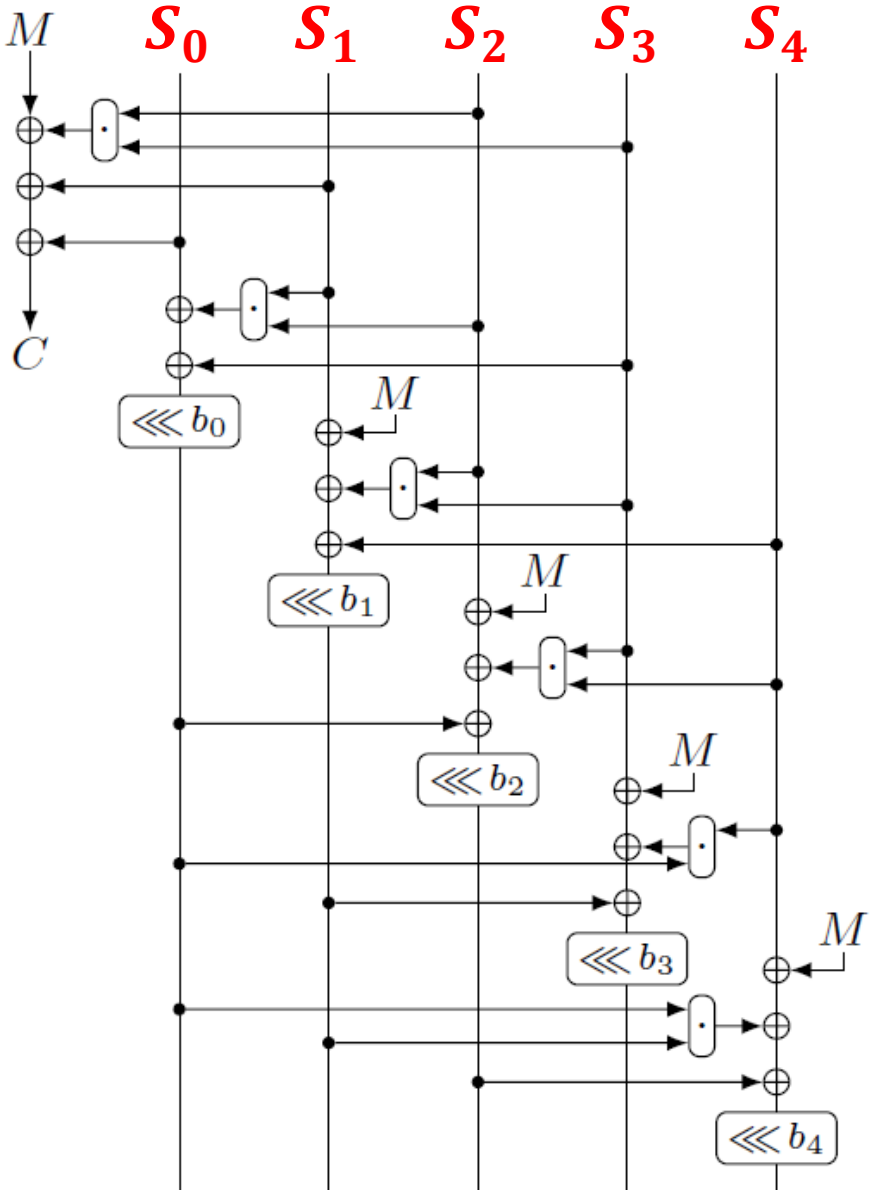
- Each register has 4 words and different registers are rotated by different word numbers (complex).
- linearly approximate 4 bits in positions $i, i+w, i+2w, i+3w$.
- 4 iterations of the same linear trail
  → compress the register to w bits.



MiniMORUS

- A linear trail with weight X for MiniMORUS
  → A linear trail with weight $4X$ for MORUS.

Diagram of MiniMORUS



Overview of Liner Trails

We combine the following five trail fragments;

α_i: approximate 1 bit of S_0 from ciphertext bit.

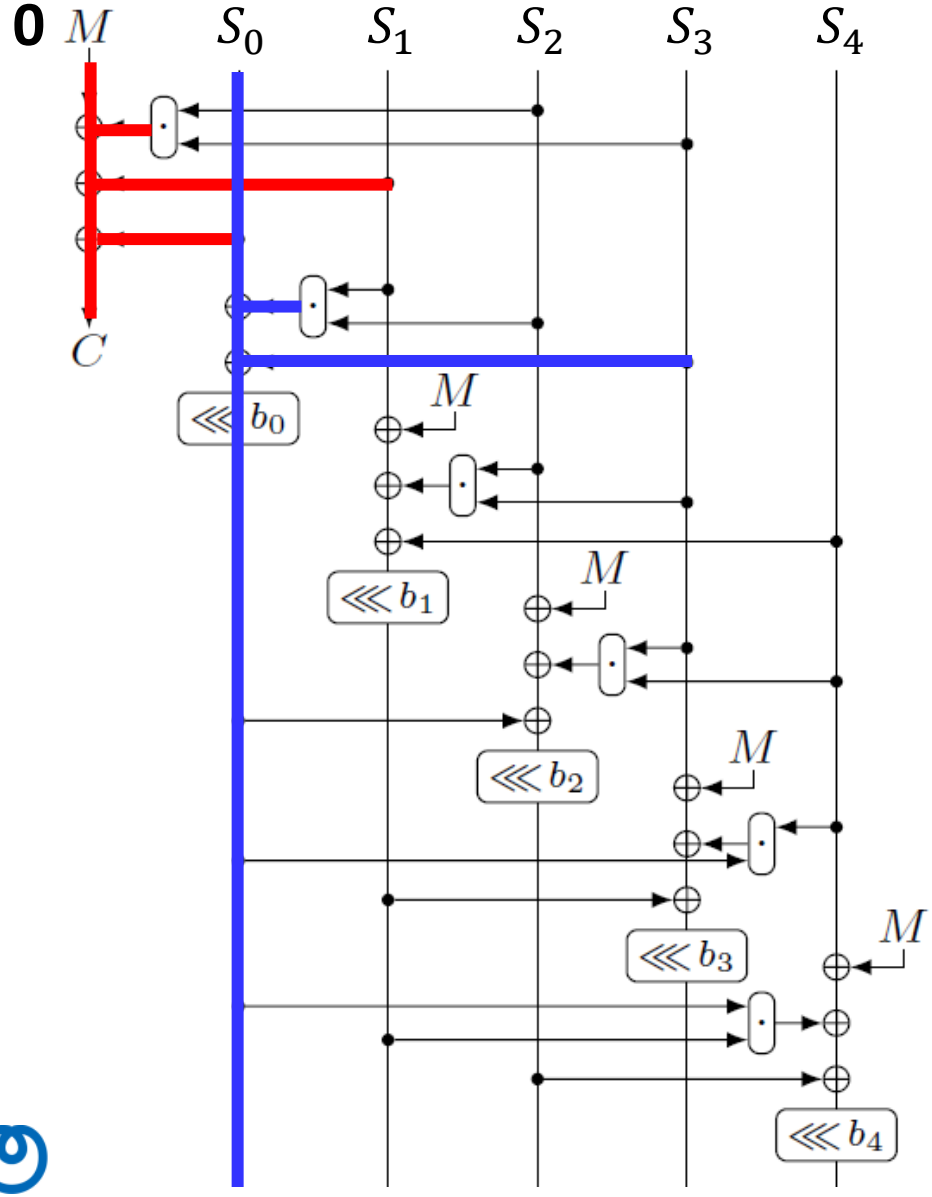
β_i: approximate 1 bit of S_1 from S_0 and ctxt bit.

γ_i: approximate 1 bit of S_4 from 2 bits of S_1.

δ_i: approximate 1 bit of S_2 from 2 bits of S_4.

ϵ_i: approximate 1 bit of S_0 from 2 bits of S_2.

α_i: from ciphertext bit to S_0



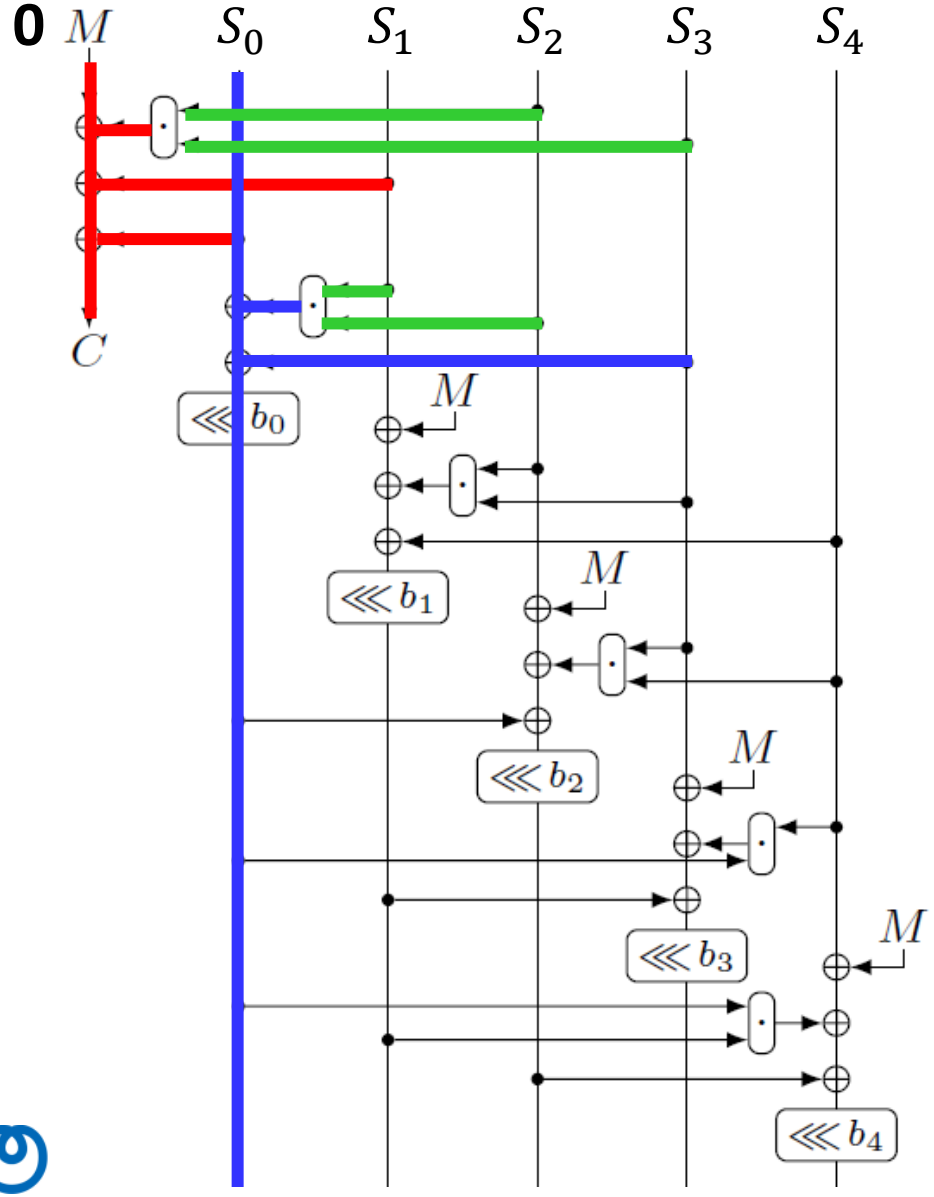
$$C^i = (S_2^i \cdot S_3^i) \oplus S_1^i \oplus S_0^i$$

$$S_0^{i+b_0} = (S_1^i \cdot S_2^i) \oplus S_3^i \oplus S_0^i$$

Combine ➡ $C^i = S_0^{i+b_0}$

(weight: 2)

α_i: from ciphertext bit to S_0



$$S_3^i \quad S_2^i \oplus S_3^i$$

$$C^i = (S_2^i \cdot S_3^i) \oplus S_1^i \oplus S_0^i$$

$$S_0^{i+b_0} = (S_1^i \cdot S_2^i) \oplus S_3^i \oplus S_0^i$$

$$S_1^i \quad S_1^i \oplus S_2^i$$

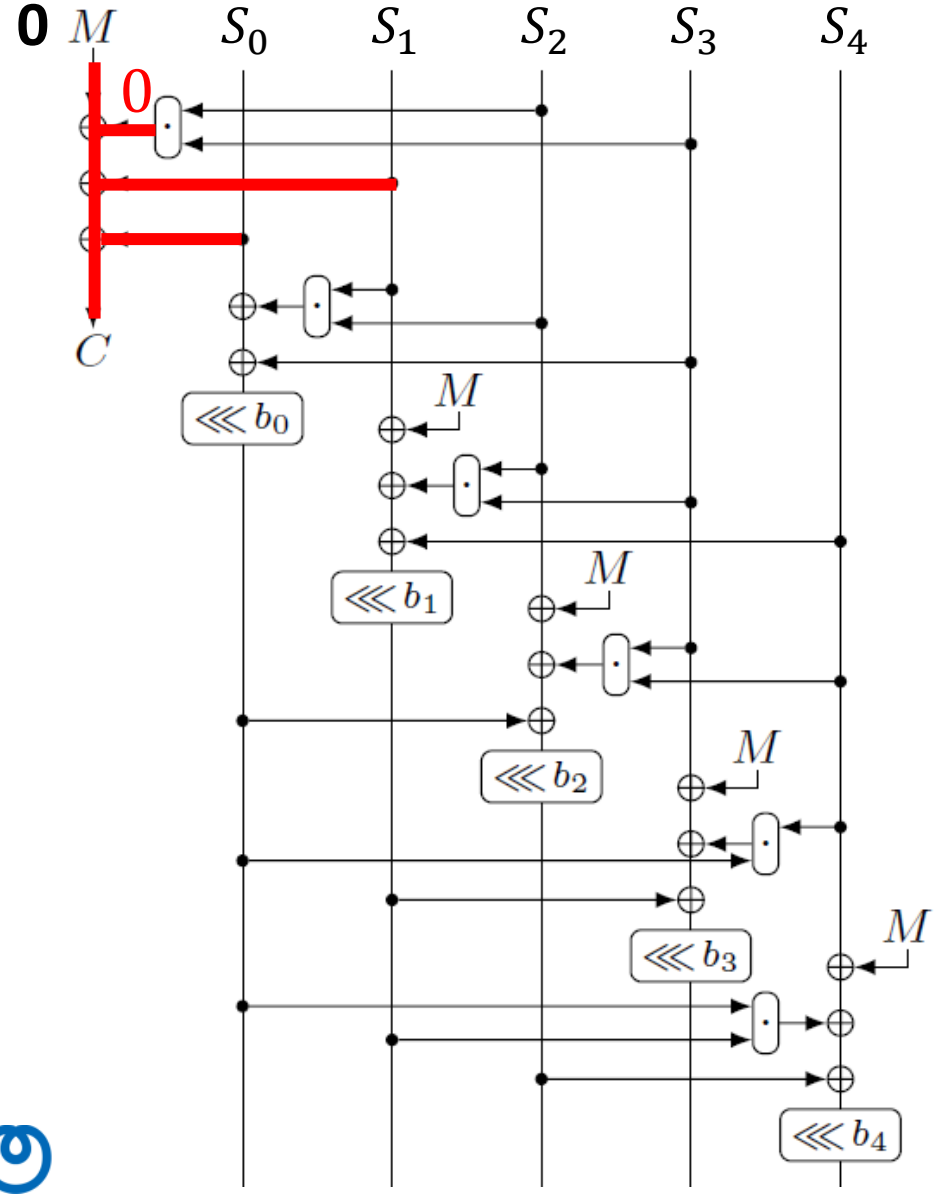
Combine ➡ $C^i = S_0^{i+b_0}$

(weight: 2)

Linear Hull :

weight: 2 → 1

β_i: from S_0 and C to S_1



$$\beta_i\colon\ C^i = \left(S_2^i \cdot S_3^i\right) \oplus S_1^i \oplus S_0^i$$

(weight: 1)

Combine α_i and β^{i+b_0}

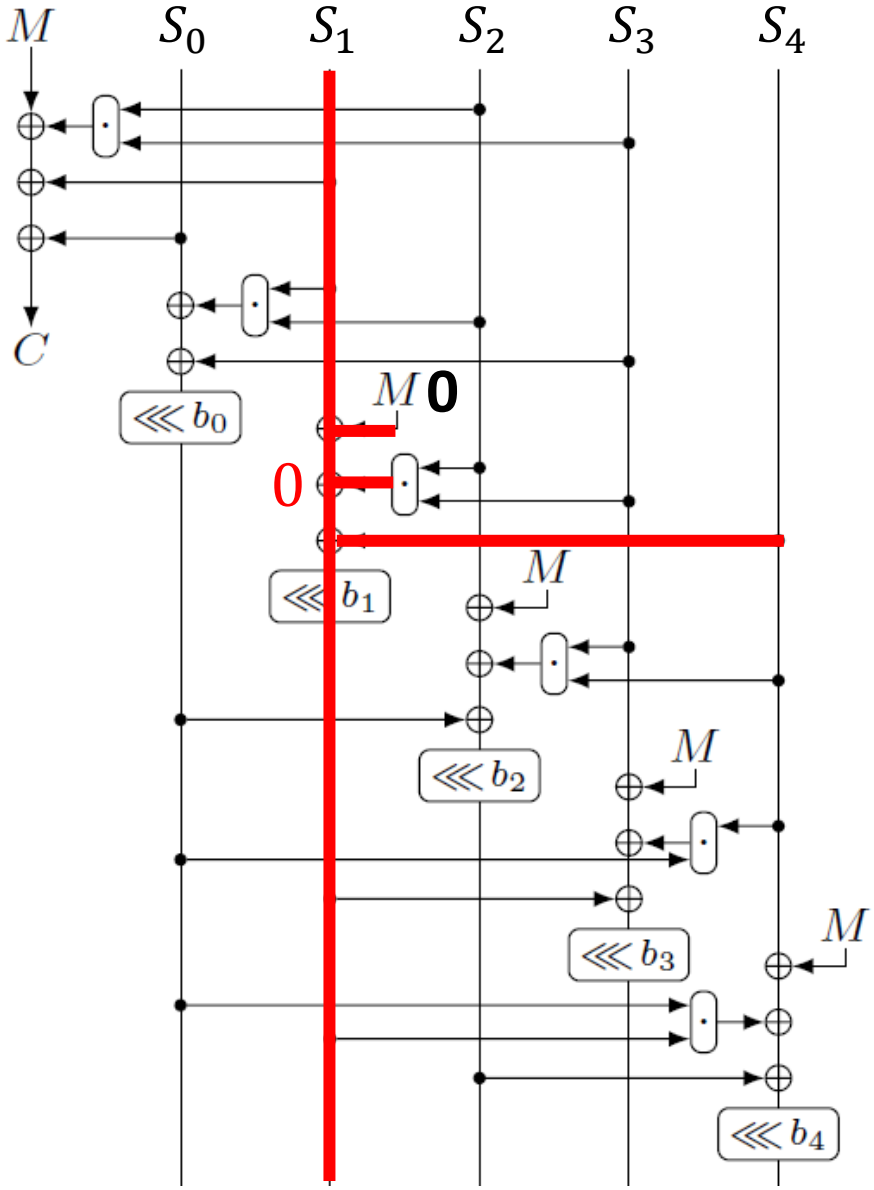
$$\alpha_i\colon\quad C^i = S_0^{i+b_0}$$

$$\beta^{i+b_0}\colon\ C^{i+b_0} = S_1^{i+b_0} \oplus S_0^{i+b_0}$$

$$\bigoplus_i C^i = S_1^j$$

(weight: 2)

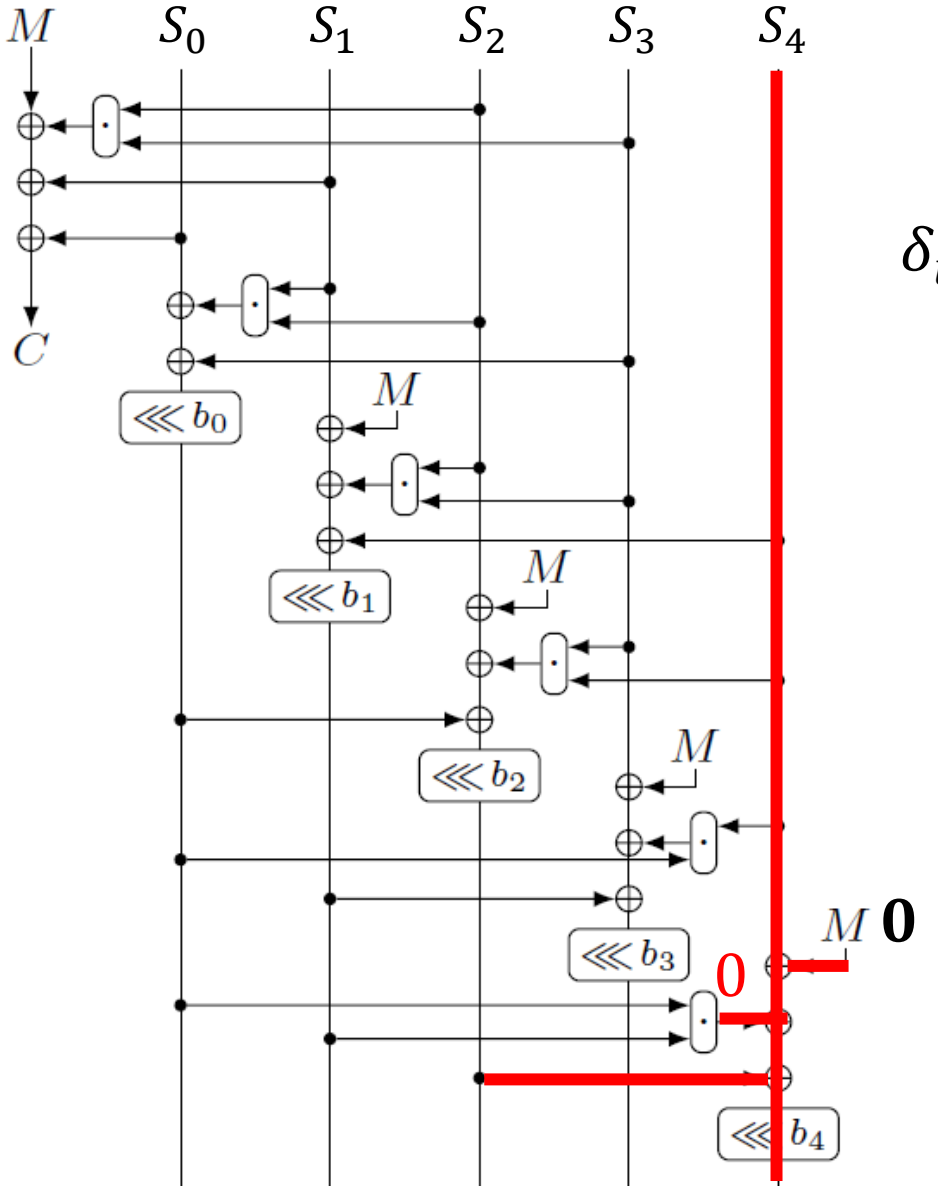
γ_i: from two bits of S_1 to S_4



$$\gamma_i\colon S_1^{i+b_1} = \left(S_2^i \cdot S_3^i\right) \oplus S_4^i \oplus S_1^i$$

(weight: 1)

δ_i: from two bits of S_4 to S_2

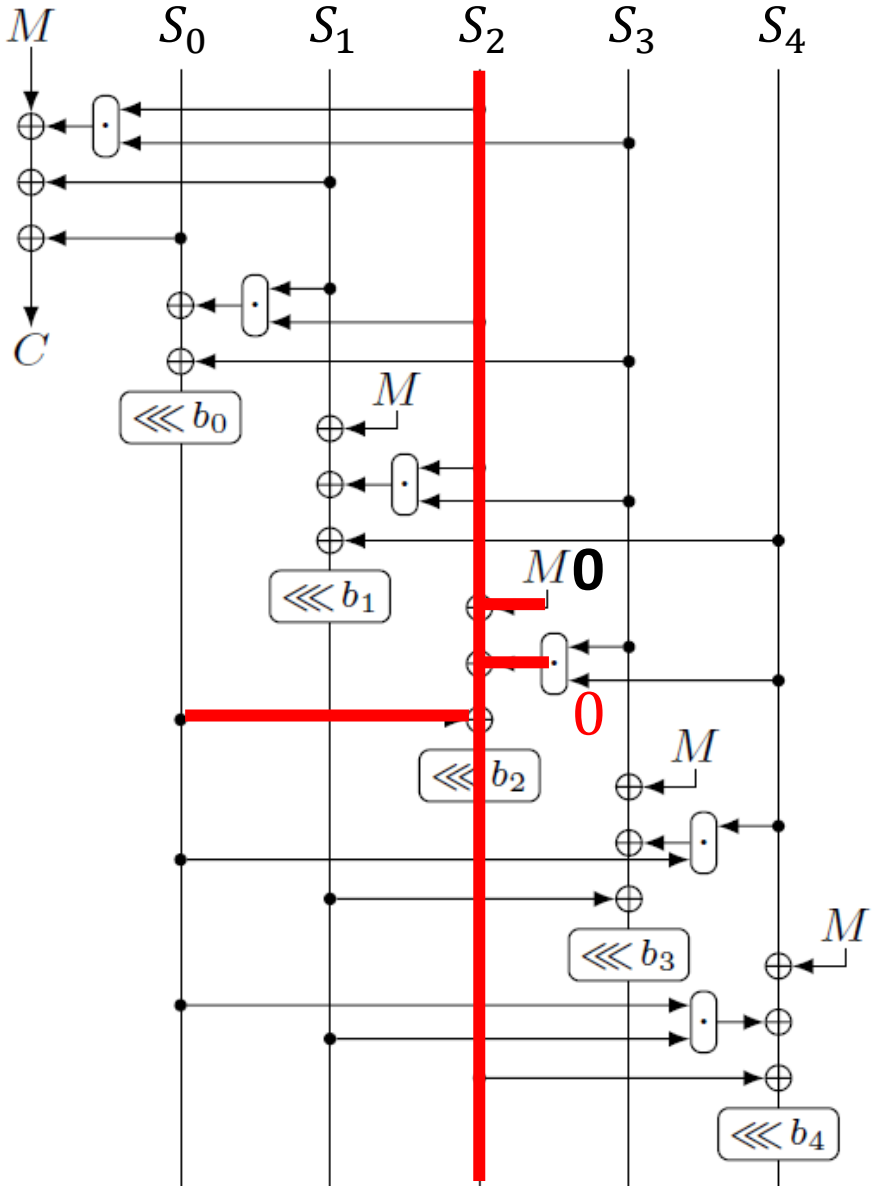


$$\delta_i\colon S_4^{i+b_4} = \left(S_0^i \cdot S_1^i\right) \oplus S_2^i \oplus S_4^i$$

0

(weight: 1)

ϵ_i: from two bits of S_4 to S_2

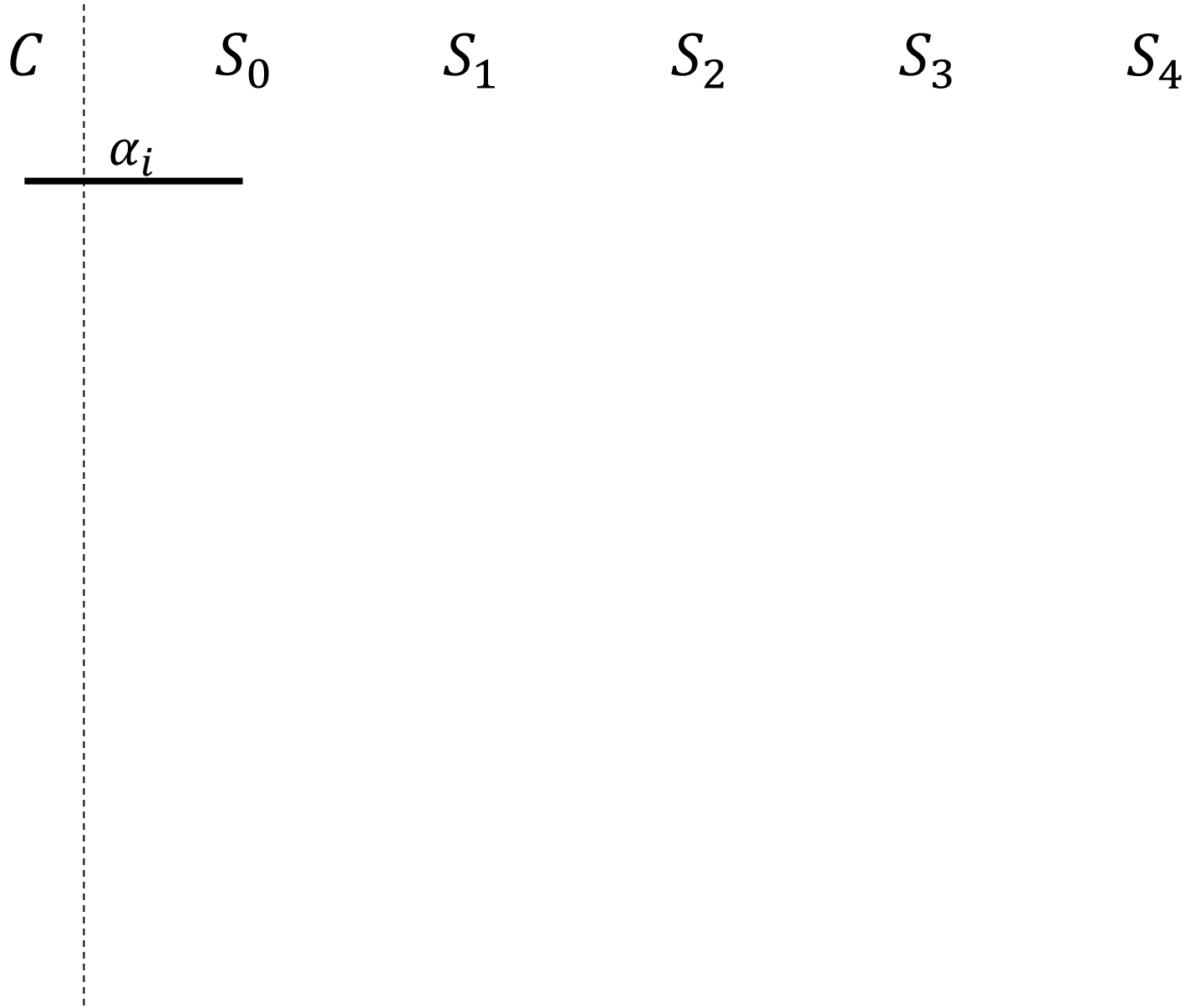


0

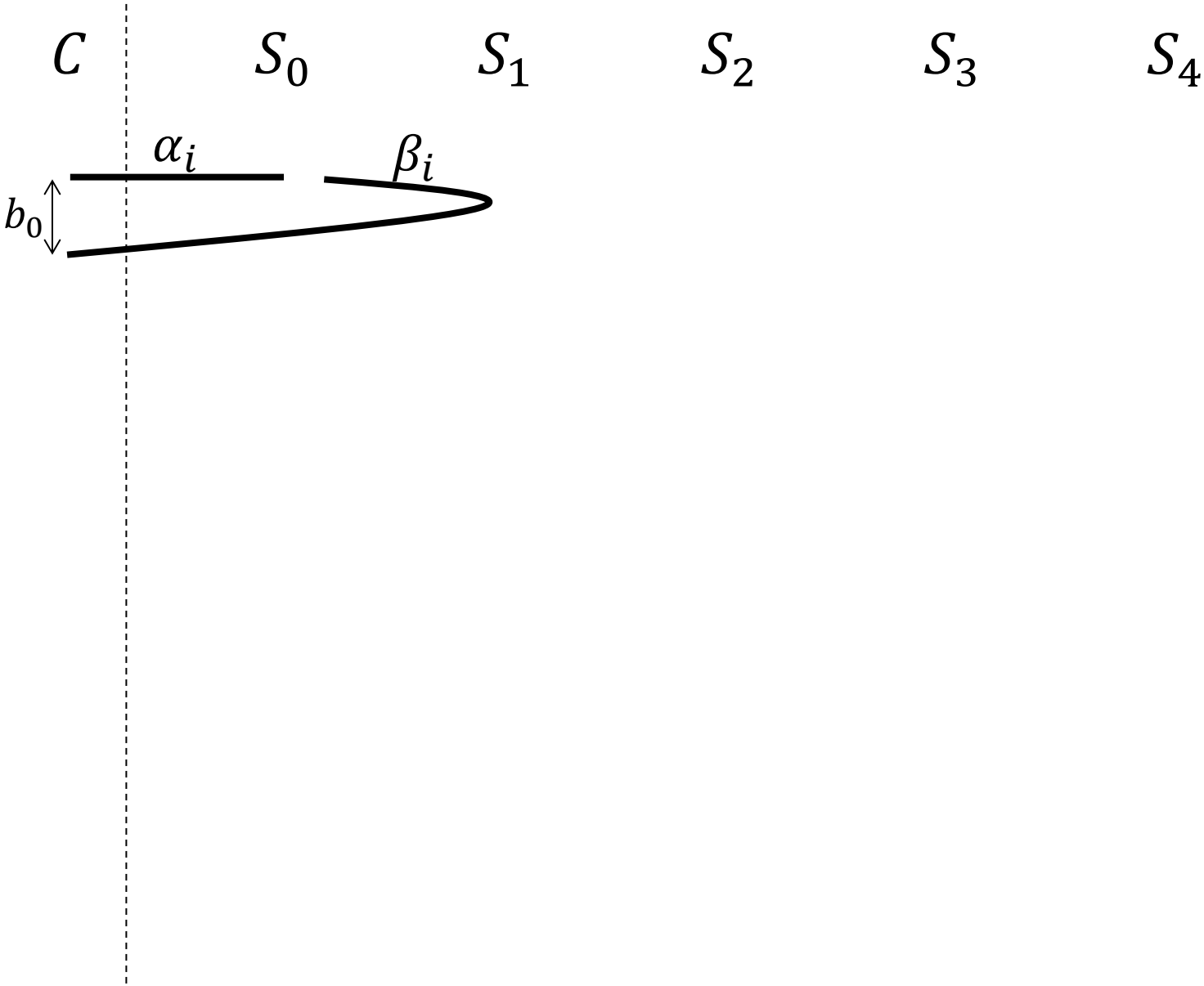
$$\epsilon_i\colon S_2^{i+b_2} = \left(S_3^i \cdot S_4^i\right) \oplus S_0^i \oplus S_2^i$$

(weight: 1)

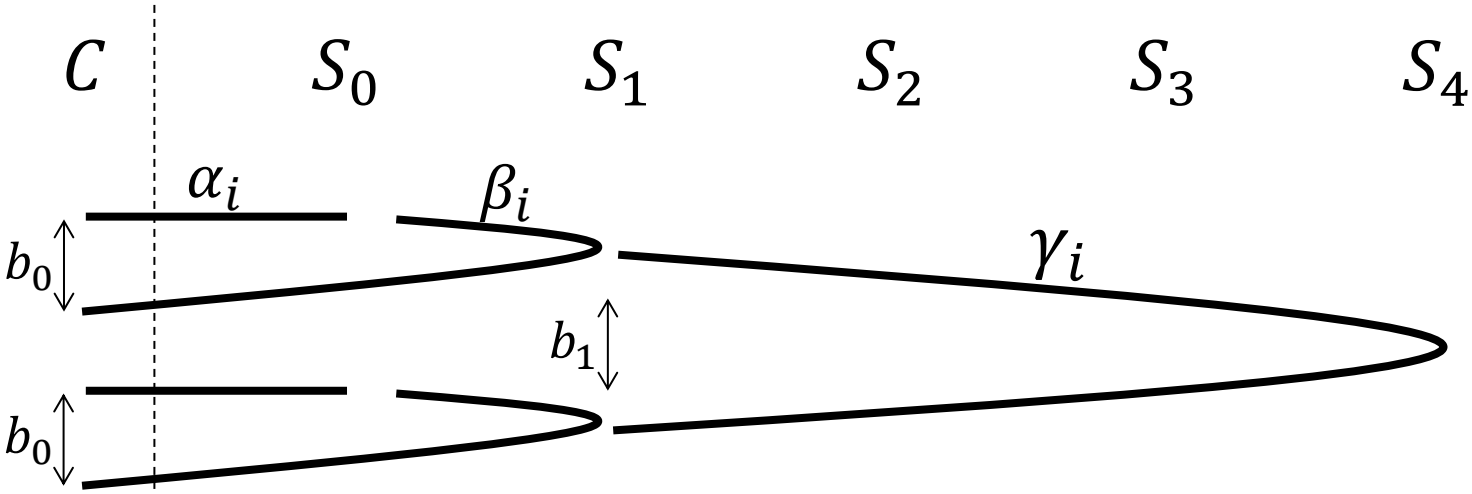
$$C^i = S_0^j$$



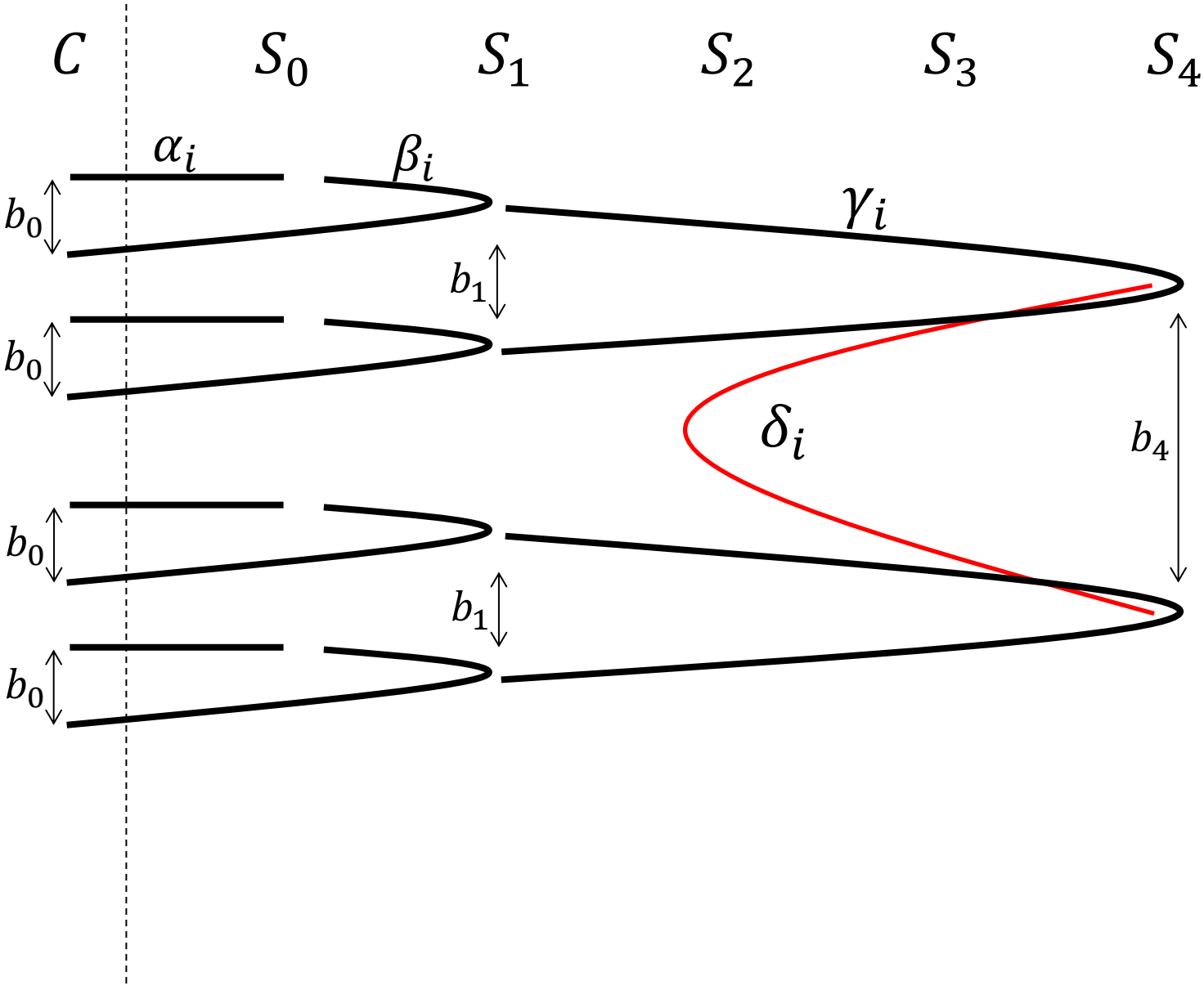
$$\bigoplus_i C^i = S_1^j$$



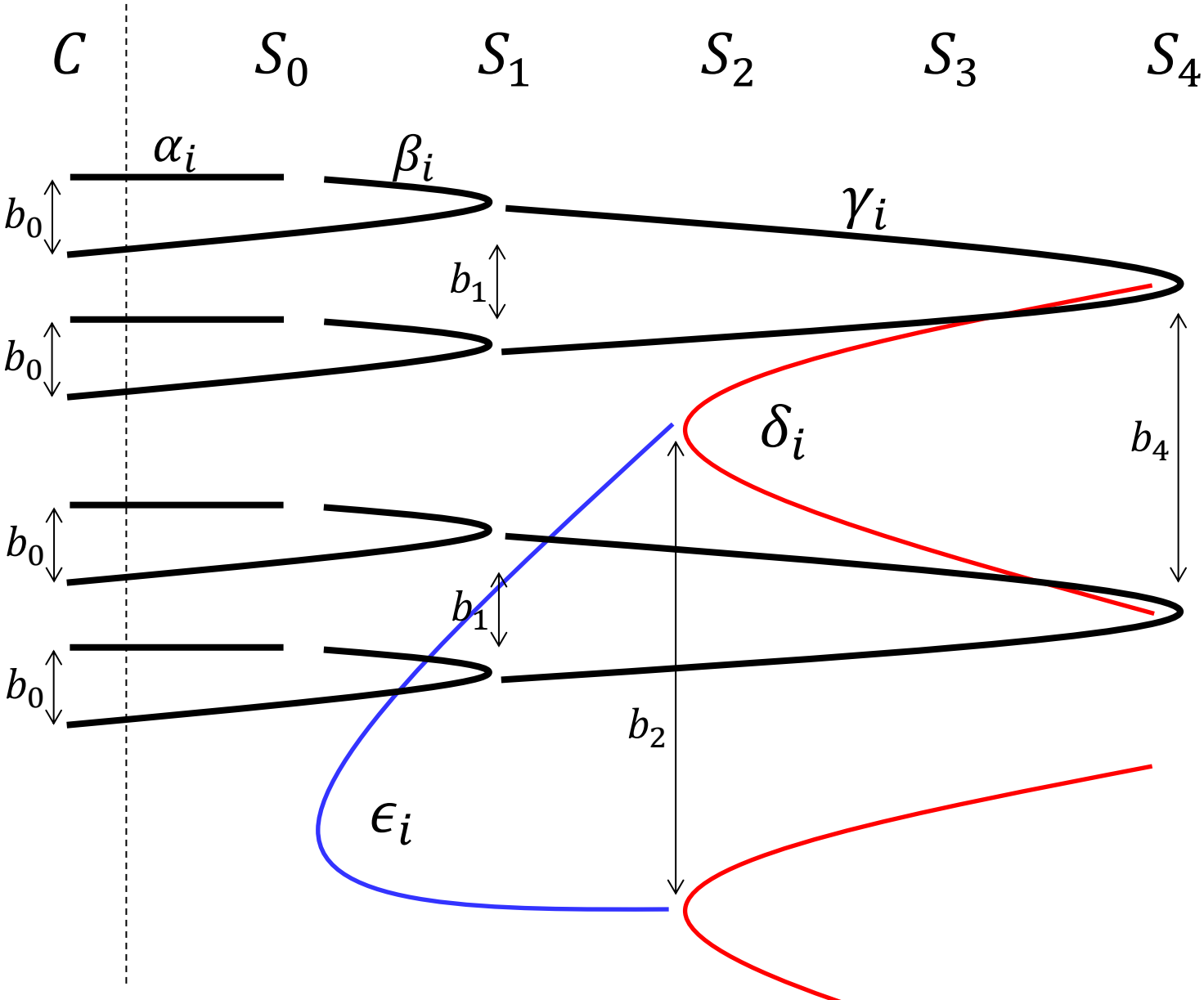
$$\bigoplus_i C^i = S_4^j$$



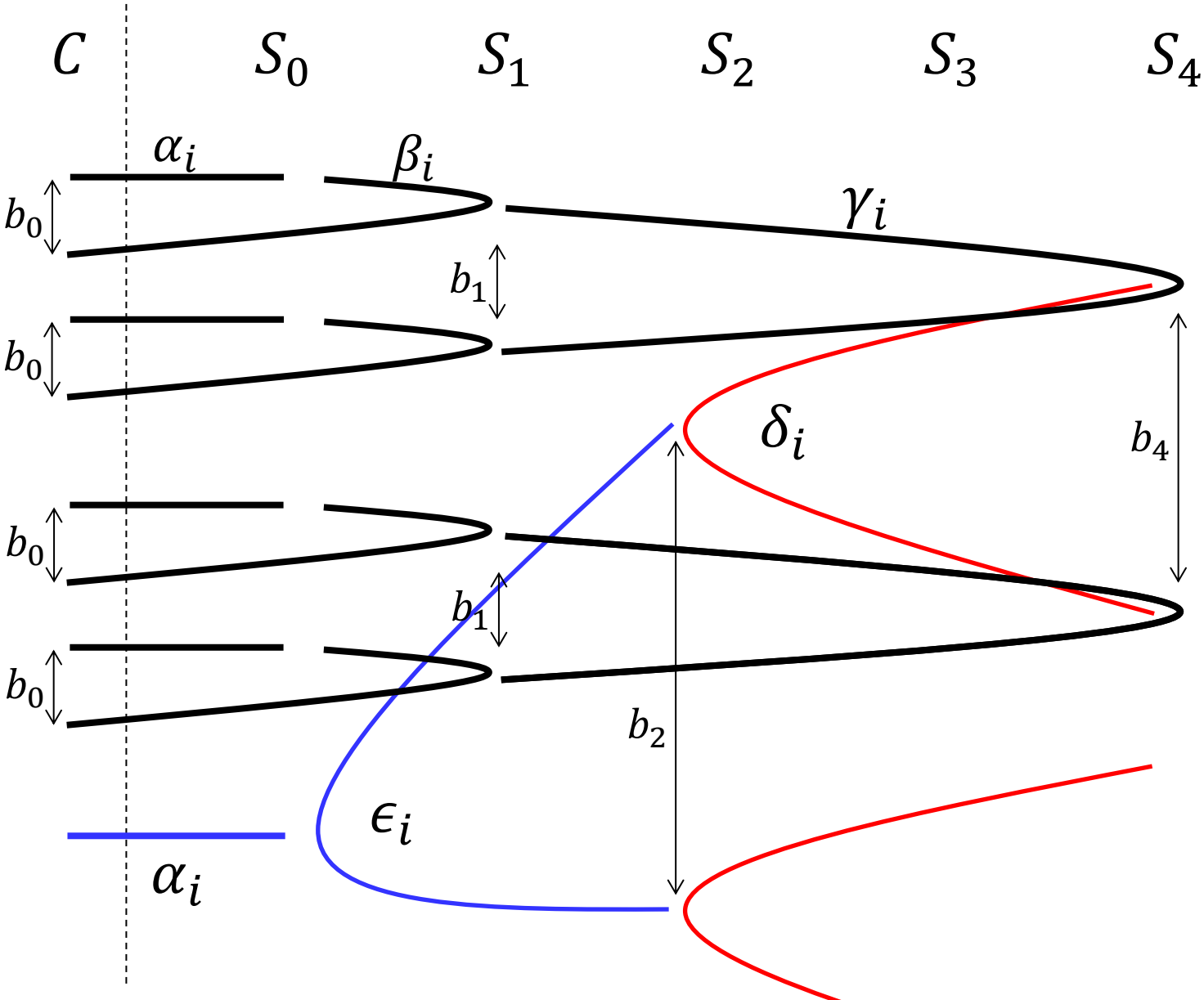
$$\bigoplus_i C^i = S_2^j$$

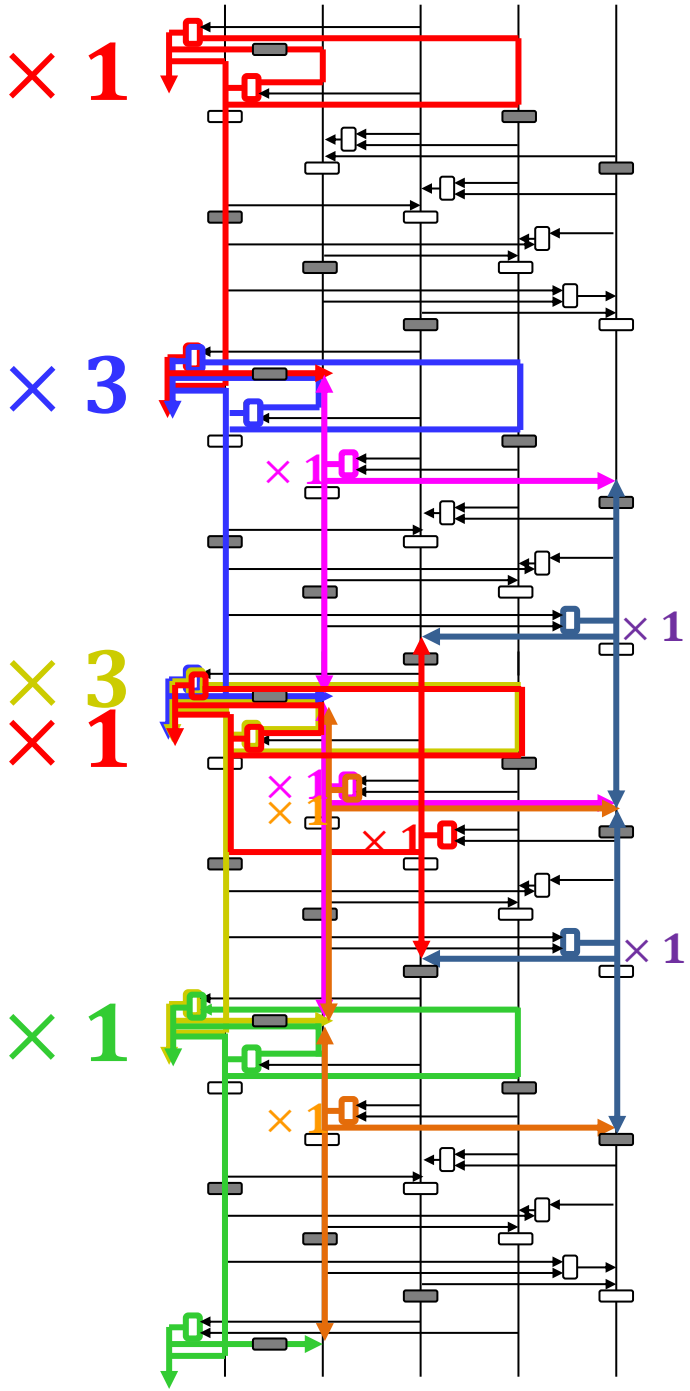


$$\oplus_i C^i = S_0^j$$



$$\oplus_i C^i = 0$$


C
S_0
S_1
S_2
S_3
S_4
α_i
β_i
γ_i
δ_i
ϵ_i
α_i
b_0
b_0
b_1
b_0
b_0
b_1
b_2
b_4


× 1
× 3
× 1
× 3
× 1
× 1
× 1
× 1
× 1
× 1
× 1

Result of Combination for MiniMORUS

- $\oplus C^i = 0$ with weight 24

Dependency between β_i and γ_i (occur 4 times):

- β_i: $C^i = (S_2^i \cdot S_3^i) \oplus S_1^i \oplus S_0^i$ (weight: 1)
- γ_i: $S_1^{i+b_1} = (S_2^i \cdot S_3^i) \oplus S_4^i \oplus S_1^i$ (weight: 1)

0

- No need to approximate, saves weight 8 for 4 (α_i, β_i).

- $\oplus C^i = 0$ with weight 16.
  Experimentally verified with 2^{32} ciphertexts.

- works for different choices of rotation numbers

Extension to MORUS: Overview

- w-bit state of MiniMORUS can be extended to $4w$-bit state of MORUS by making 4 copies of the linear trail.
  - Linear trail for MiniMORUS: weight 16
  - Linear trail for MORUS: weight 64 ??

- Most of the part, the evaluation is true, however saving weight by dependency cannot be used.
- A few more issued on overlap of the bit position in independent linear approximation.
- In the end, the weight is 76, which is verified if you have 2^{152} ciphertexts.

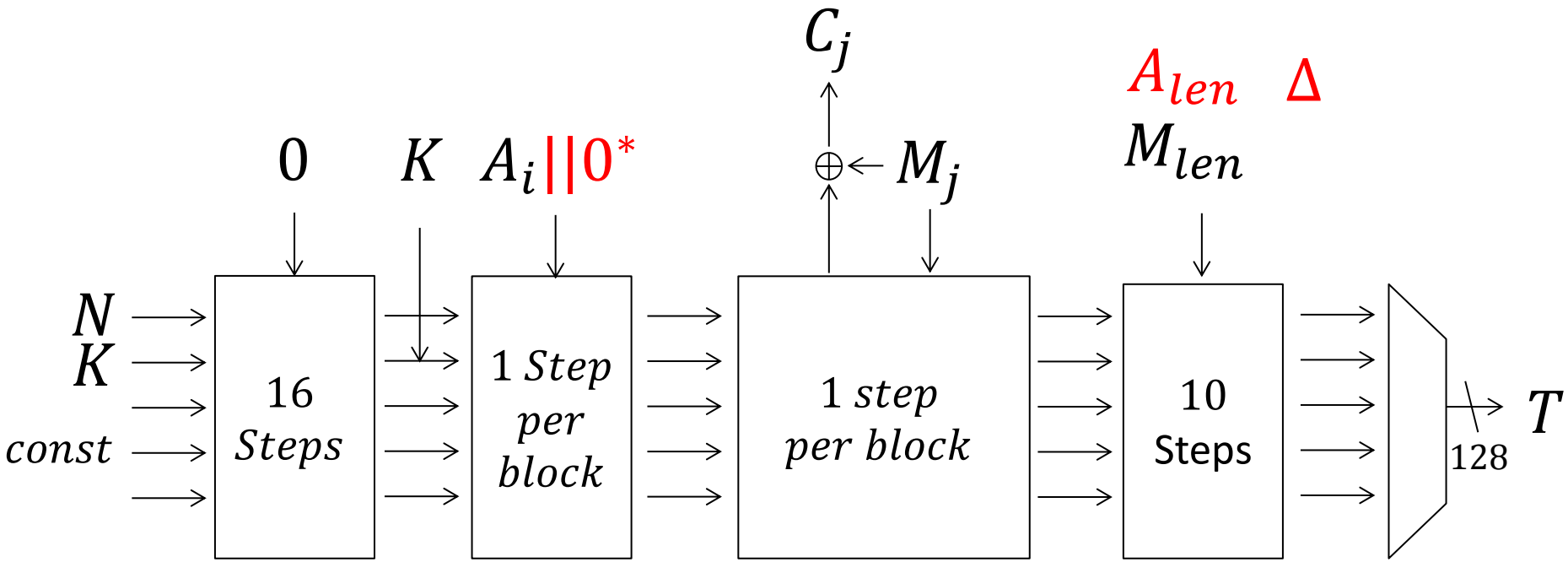
Attack Impact

- Detected bias is absolute. The attack does not have any limitation for the choice of K and N. (cannot be prevented by key management)

- The attack works in the broadcast setting. Some protocol fixes the first message block to some sensitive information (e.g. user authentication token in HTTP). Correlation of key steam may be exploited to recover it

Analysis on Finalization of MORUS-1280-256

Attack Strategy



Padding for associated data is 0-padding.

High prob. diff. trail

- A and $A||0$ lead to the same state value.
- $\mathrm{HW}(\Delta A_{len})$ can be set to 1.
- Expect that diffusion is slow in finalization. Differential trail with prob $> 2^{-128}$ can immediately be used for forgery.

Fast Diffusion

Diffusion is fast. Only works for 3 out of 10 steps.

There exists ΔT that can be observed with $DP = 2^{-88}$ after 3 steps and the tag generation.

#rounds	1	2	3	4	5	6	7	8
Log(DP)	0	0	0	0	-1	-3	-6	-10

Step 1 Step 2

#rounds	9	10	11	12	13	14	15	tag
Log(DP)	-14	-20	-28	-39	-53	-69		-88

Step 3

Remarks

- The last operation in Step function:

$$S_4 \leftarrow S_4 \oplus (S_0 \cdot S_1) \oplus S_2 \oplus (A_{len} || M_{len}) \ll b_4$$

- The tag generation function:

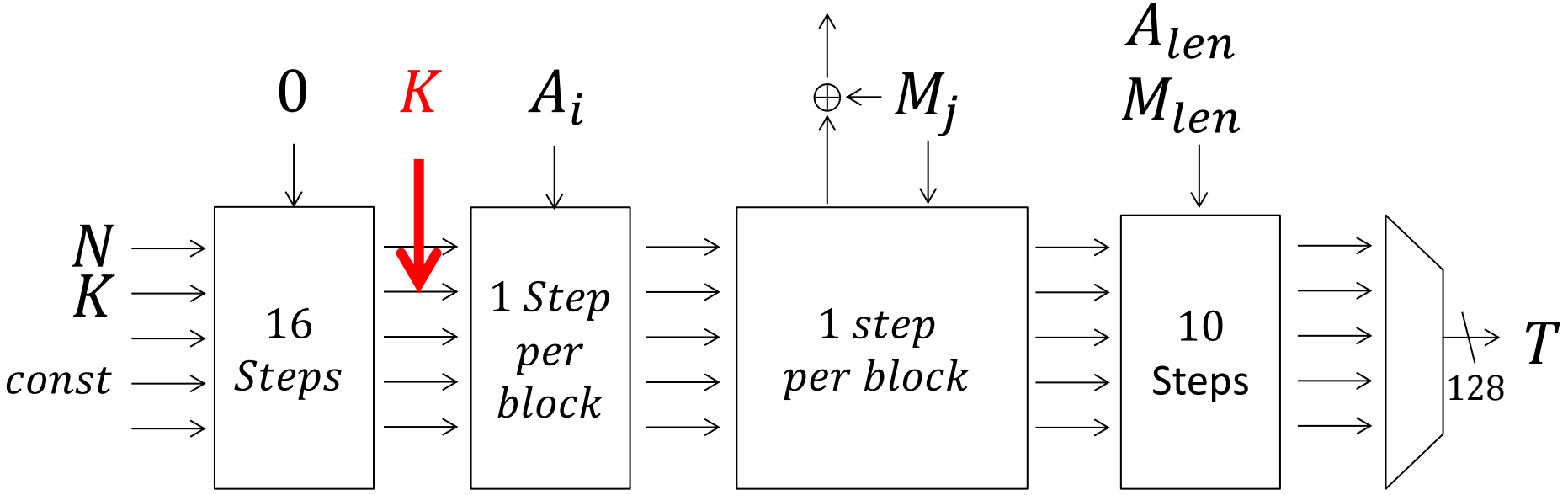
$$T \leftarrow S_0 \oplus (S_2 \cdot S_3) \oplus (S_1 \ll b_t)$$

- The last updated value S_4 is not used to generate tag. (waste of computation resource)

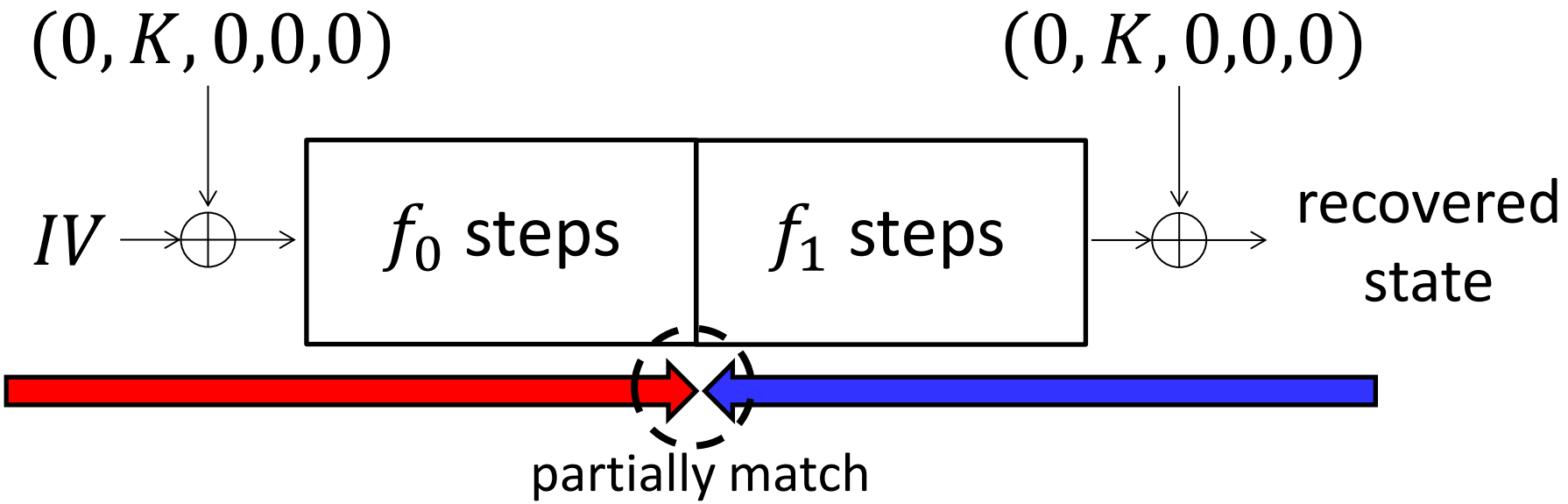
Analysis on initialization of MORUS-1280-256

Previous Work

- State-recovery and universal forgery in nonce-repeating using 2^5 queries [KEM2017].
- Cannot recover the key due to feed-forward.
- Our goal is to recover K for reduced rounds.



Overview: (3-Subset) MitM Attack



- G_0: a set of key bits guessed for the forward function
- G_1: a set of key bits guessed for the backward function
- G_2: a set of key bits in the intersection of G_0 and G_1
- m : the number of bits matching in the middle

$$\max(2^{|G_0|}, 2^{|G_1|}, 2^{|G_0|+|G_1|-|G_2|-x})$$

Attack Results

$$\max(2^{|G_0|}, 2^{|G_1|}, 2^{|G_0|+|G_1|-|G_2|-x})$$

- Set $|G_0| = |G_1| = 127$ and $|G_2| = 126$.
- Search for such separation that ensures $x \geq 1$.

- For $(f_0, f_1) = (4,6)$, we found a configuration achieving $x = 4$.
- 128-bit key K is recovered with 2^{127} computations against 10 out of 16 steps.

- MORUS was designed to have fast diffusion in forwards. The attack exploits slow backward diffusion.

Concluding Remarks

Concluding Remarks

- Breaking 256-bit confidentiality of MORUS-1280-256 with 2^{152} encryptions.)
- First full break of CAESAR finalists.
- Combination of AND operations allows the efficient construction of linear trail.
- The same attack was known on AEGIS by [Minaud SAC2014]
- Initialization/Finalization were investigated.

Thank you for your attention!!