Integral Attack Goes More than Impossible Differential Attack for LBlock

Yu Sasaki[1], Lei Wang[2]

[1]: NTT Corporation

[2]: The University of Electro-Communications

NanYang Technological University

ASK 2012 (29/Aug/2012)

NTT

This talk contains;

- Re-announcement of our SAC 2012 paper
  - Meet-in-the-Middle Technique for Integral Attacks against Feistel Ciphers (Yu Sasaki and Lei Wang)
- Recent updates on integral analysis for LBlock

Status of SAC 2012 Paper

- Use MitM technique to reduce the complexity of integral attacks.
- Well-applied if the target is Feistel ciphers.

Target	Rounds	Data	Time	Ref.
LBlock	20	(Flawed)		[WZ11]
	20	$2^{63.6}$	$2^{39.6}$	**Ours**
HIGHT	22	2^{62}	$2^{118.71}$	[ZSL09]
	22	2^{62}	$2^{102.35}$	**Ours**
CLEFIA-128	12	$2^{115.7}$	$2^{116.7}$	[LWZ11]
	12	$2^{115.7}$	$2^{103.1}$	**Ours**

(#rounds are smaller than impossible diff. attacks.)

NTT

Contents

- Integral Attack and Partial-Sum Technique
- MitM technique for Integral Attacks
- Applications to LBlock, HIGHT, and CLEFIA-128
- Concluding Remarks

NTT

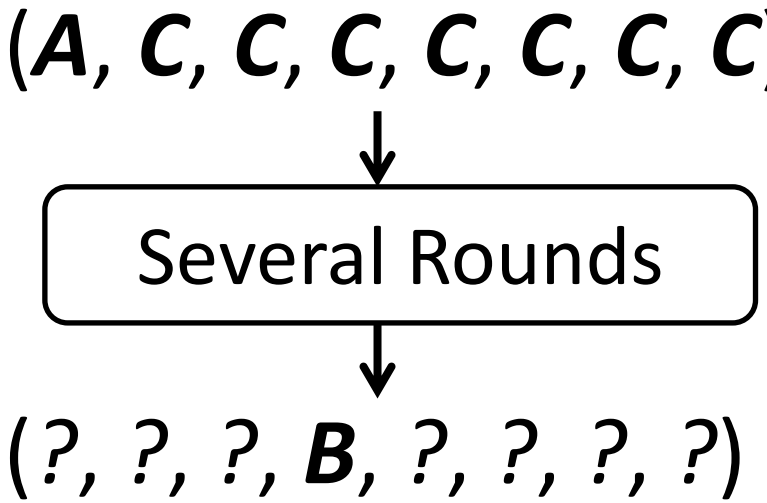
Introduction of Integral Attacks

- Introduced by Daemen et al. to evaluate the security of SQUARE cipher.

- Consisting of 2 parts;
  - Integral distinguisher
  - Key recovery phase

Integral Distinguisher

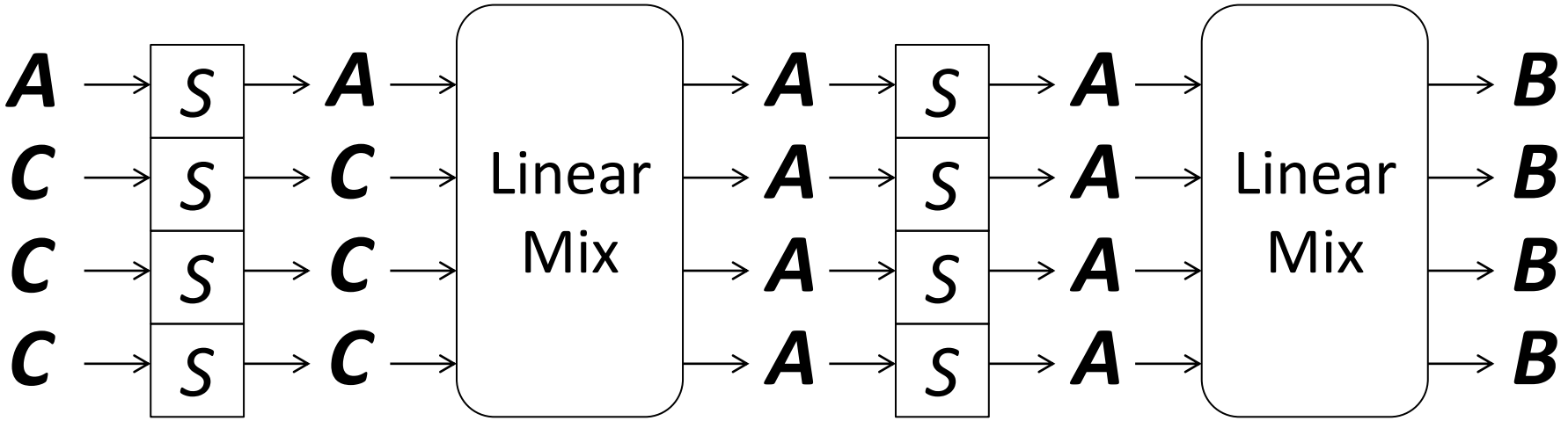
- Prepare a set of plaintexts which contains all possible values (***A***) for some bytes and has a constant value (***C***) for the other bytes.



- XOR of all texts in the set becomes 0 (***B***).

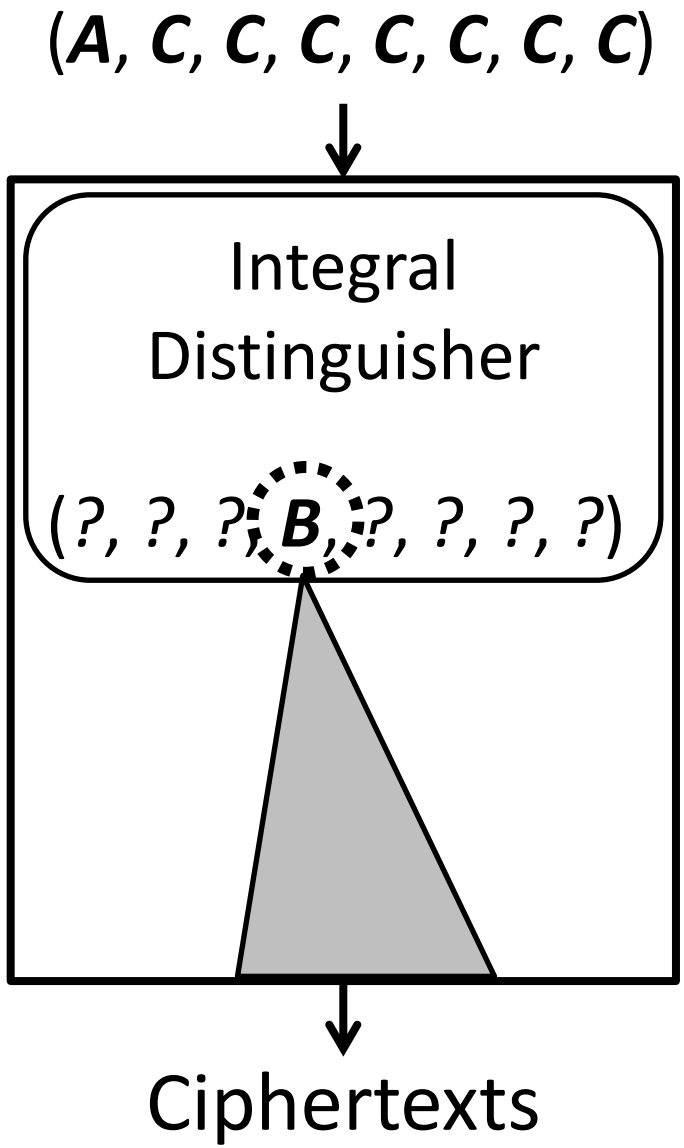
Logics in Behind

- ***A*** and ***C*** are preserved through S-box.
- Mixing two ***A*** states lose its property, but still keeps ***B*** property.
- Such property can be traced for a few rounds.



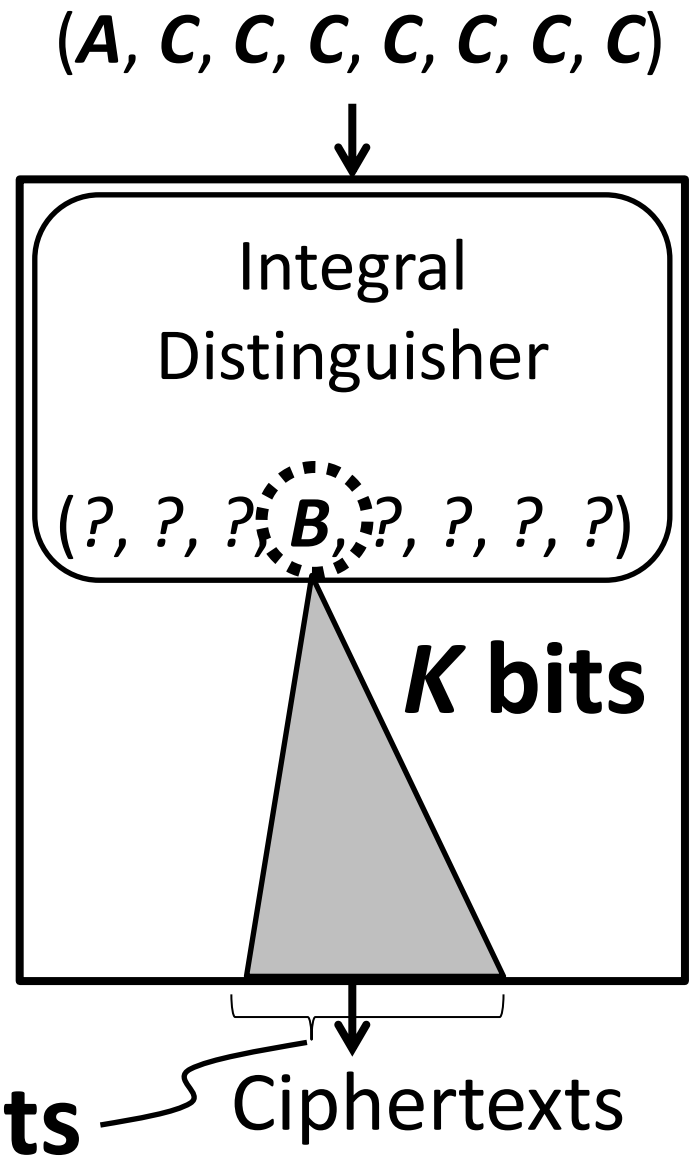
Key Recovery Phase

- Append several rounds to the distinguisher.
- Partially decrypt ciphertexts until the balanced state by partially guessing subkeys.
- If guess is correct, the sum of the results always becomes 0.

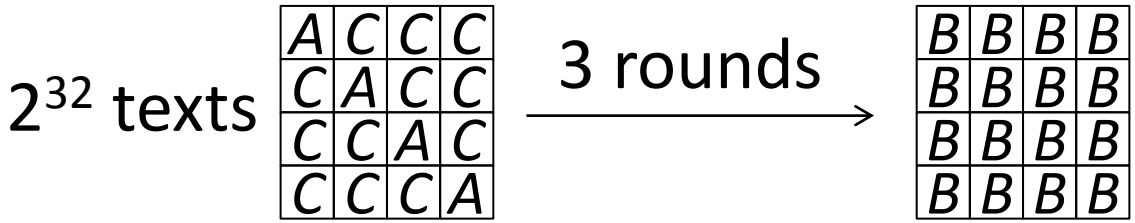


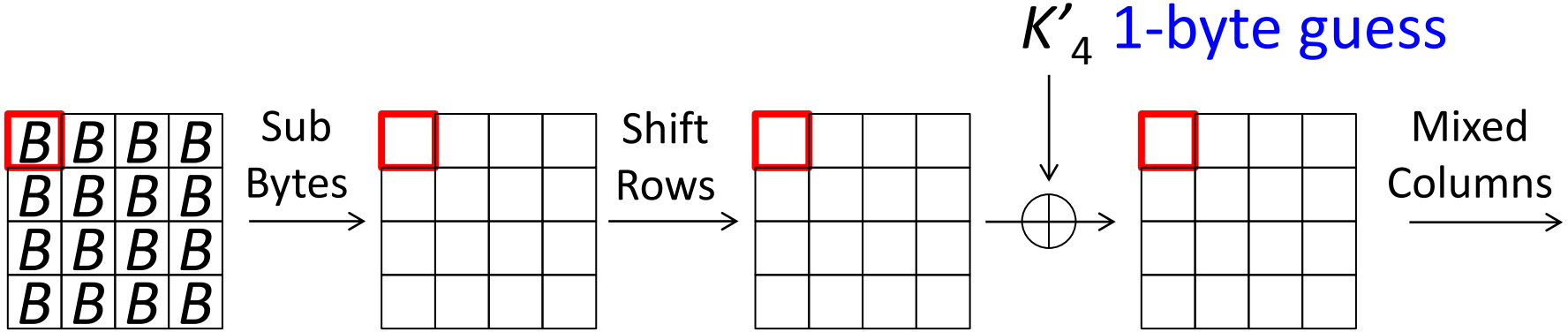
Complexity of Integral Attack

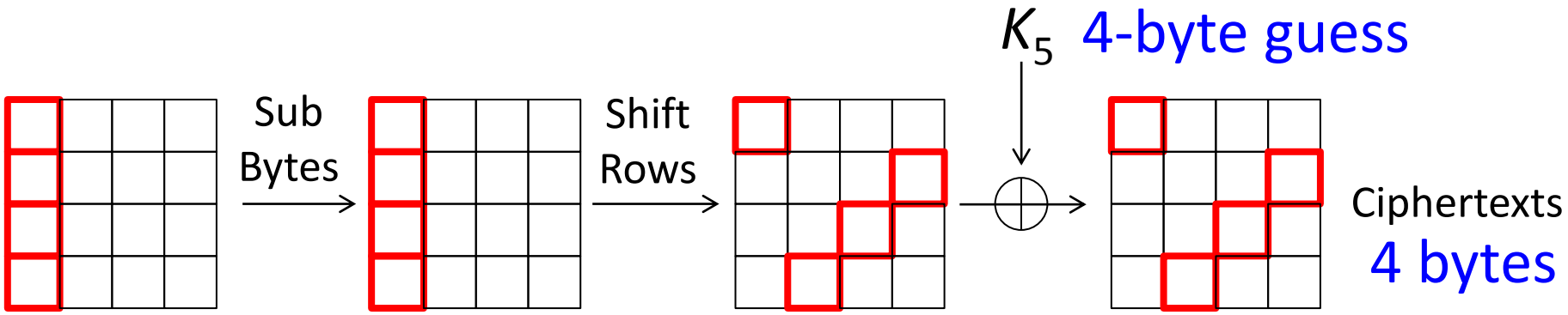
- Suppose that the partial decryption involves T bits of ciphertexts and K bits of keys.
- It requires 2^{T+K} partial decryption computations.
- Key space is reduced by $|B|$ bits per set.



Application to 5-round AES







$$\bigoplus_{n=1}^{2^{32}} \left[S_4 \left(S_0(c_{0,n} \oplus k_0) \oplus S_1(c_{1,n} \oplus k_1) \oplus S_2(c_{2,n} \oplus k_2) \oplus S_3(c_{3,n} \oplus k_3) \oplus k_4 \right) \right]$$

Application to 5-round AES

$$\bigoplus_{n=1}^{2^{32}} \left[S_4 \left(S_0(c_{0,n} \oplus k_0) \oplus S_1(c_{1,n} \oplus k_1) \oplus S_2(c_{2,n} \oplus k_2) \oplus S_3(c_{3,n} \oplus k_3) \oplus k_4 \right) \right]$$

Involves 4 ciphertext bytes and 5 key bytes.

- Straightforward: $2^{32} * 2^{40} = 2^{72}$ computations.
- Partial-sum: 2^{48} computations.

Guess each key byte one after another

Partial-Sum Technique

$$\bigoplus_{n=1}^{2^{32}} \left[S_4 \left(S_0(c_{0,n} \oplus k_0) \oplus S_1(c_{1,n} \oplus k_1) \oplus S_2(c_{2,n} \oplus k_2) \oplus S_3(c_{3,n} \oplus k_3) \oplus k_4 \right) \right]$$

x y z

- Computation starts from 2^{32} texts (c_0, c_1, c_2, c_3).
  - Guess two key bytes k_0, k_1.
  - For each guess, compute 2^{32} tuples (x, c_1, c_2).
  - Only pick (x, c_1, c_2) which appear odd times.
  - The size of the set is compressed into 2^{24}.
    - Guess one key byte k_2.
    - For each guess, compute 2^{24} tuples (y, c_2).
    - Only pick (y, c_2) which appear odd times.
    - The size of the set is compressed into 2^{16}.

Time: $2^{32}*2^{16}=2^{48}$

Time: $2^{24}*2^{16}*2^{8}=2^{48}$

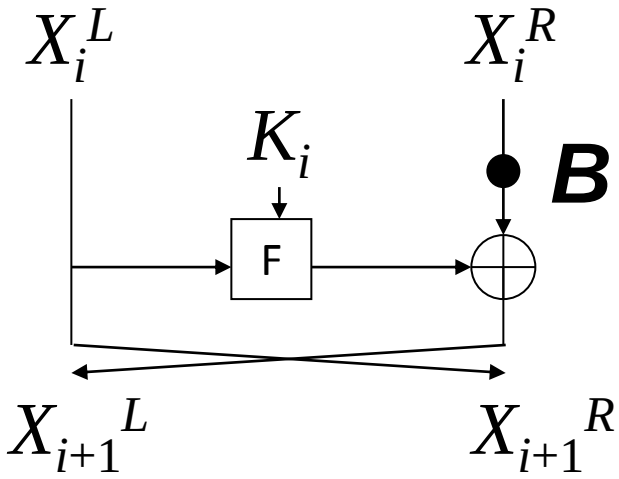
NTT

Summary of Partial-Sum

- A technique to reduce the complexity of the key recovery phase for integral attacks.
- Whether or not it can be applied depends on the structure of the attack target.
- Our technique (MitM) can be combined with the partial-sum technique.

Integral Analysis for Feistel Ciphers

- The property of ***B*** is always broken from the right-hand side.

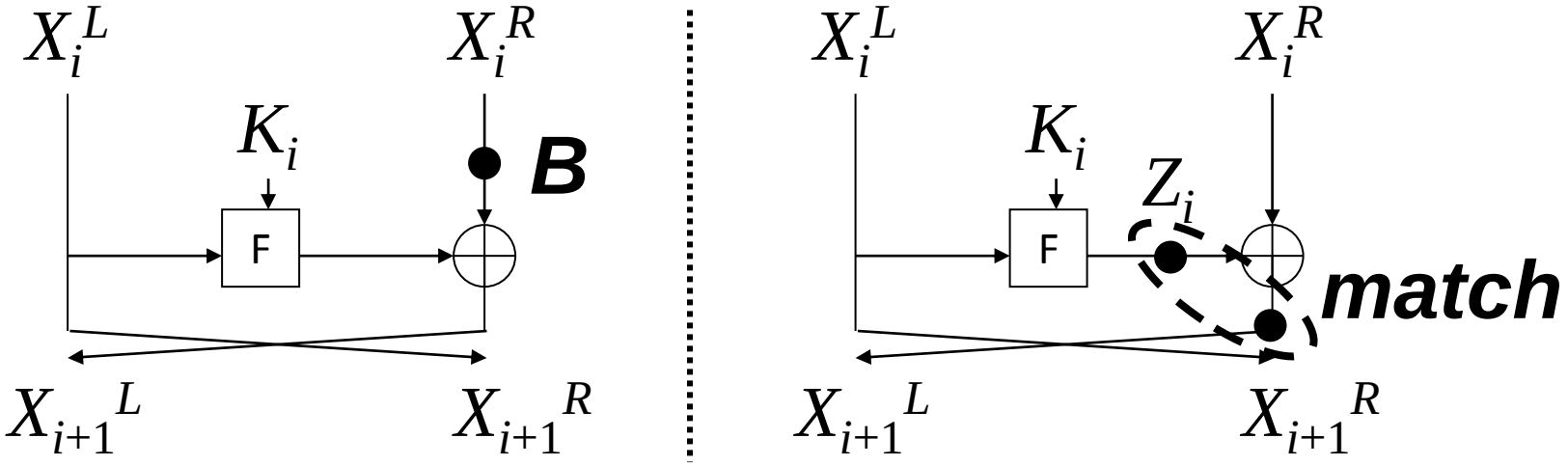


- Let #$K(X)$ and #$T(X)$ be the number of key bits and ciphertext bits to compute X.
- The complexity is $2^{\#K(X)+\#T(X)}$.

MitM technique for Integral Attacks

The same approach was used in the designers' evaluation of TWINE.

Introduction of MitM Approach

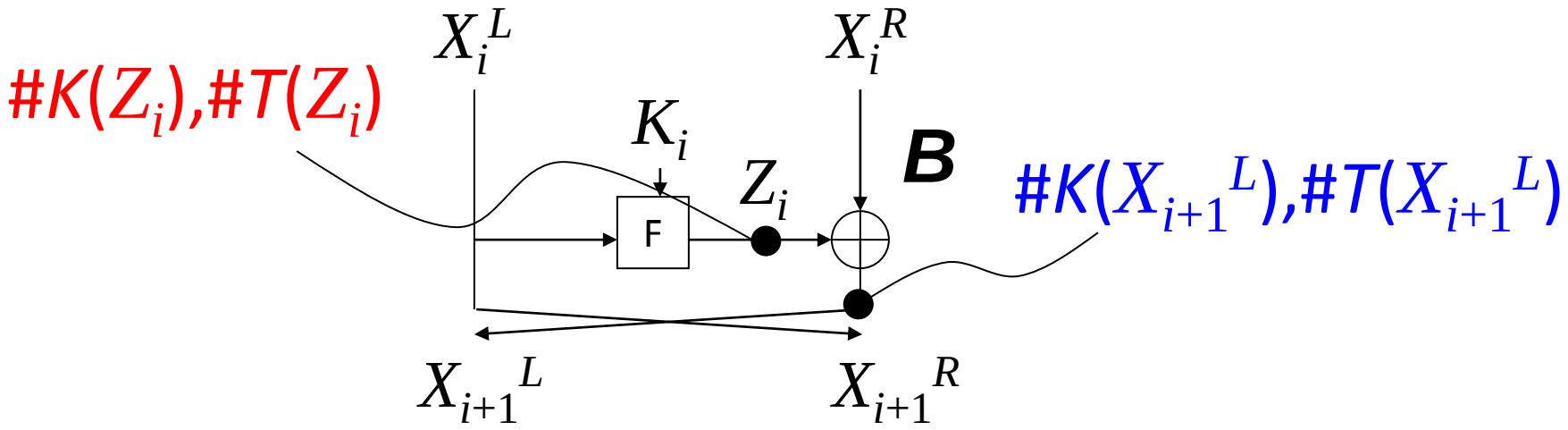


- $\bigoplus X_i^R = 0 \Rightarrow \bigoplus (Z_i \oplus X_{i+1}^L) = 0$

$$\Rightarrow \bigoplus Z_i = \bigoplus X_{i+1}^L$$

- Two terms can be computed independently. Right key candidates are identified by checking the match of two lists.

Complexity of the MitM Approach



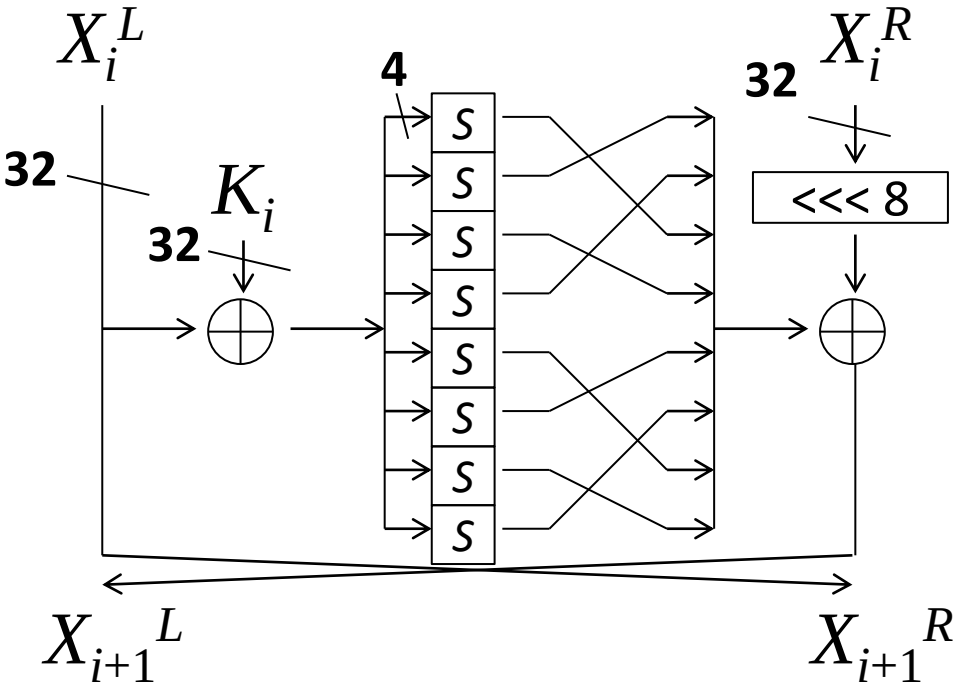
- Red part is always more expensive than blue part.
- The complexity is $2^{\#K(Z_i)+\#T(Z_i)}$.
- If $\#K(Z_i)$ and $\#K(X_{i+1}{}^L)$ share some bits in common, memory complexity can be saved as standard meet-in-the-middle attacks.

NTT

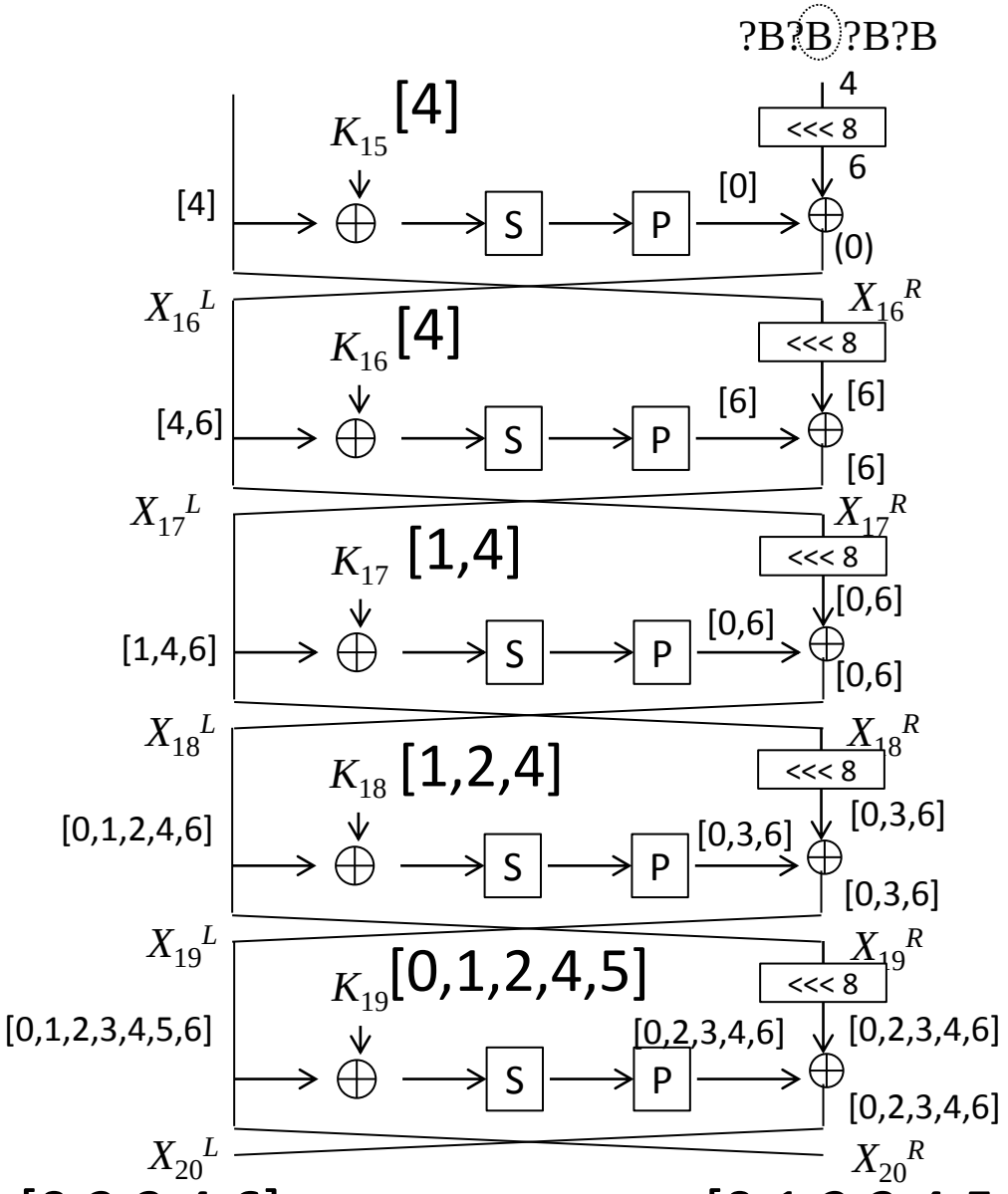
Applications

LBlock

- Proposed by Wu and Zhang at ACNS 2011.
- 64-bit block, 80-bit key.
- Modified Feistel structure with 32 rounds.
- 15-round distinguisher is known.

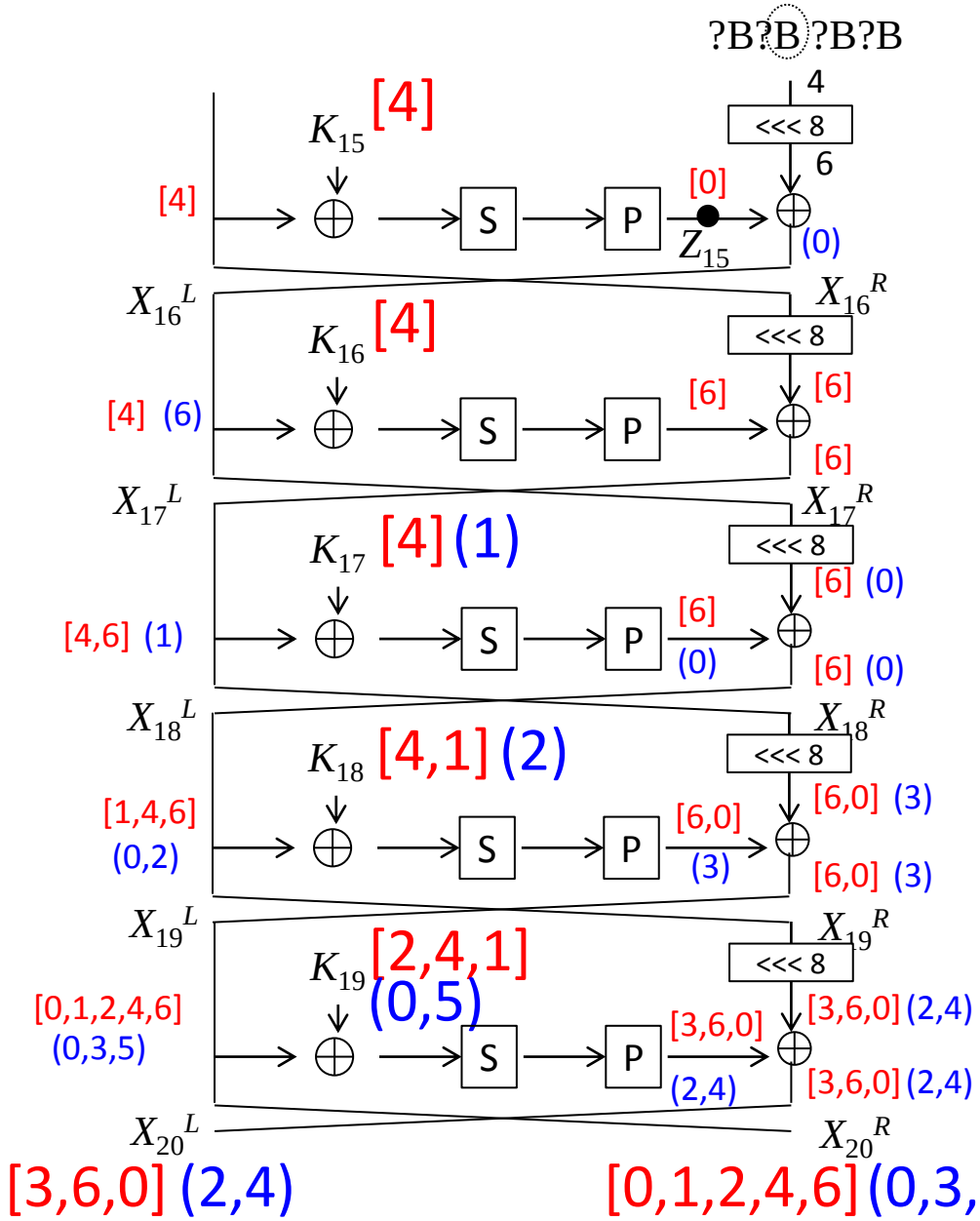


Flaw of Previous 20-Round Attack



- #$K(X_{15}[4])$=48
- #$T(X_{15}[4])$=48
- The complexity is $2^{48+48} = 2^{96}$, worse than the brute force.
- [WZ11] did not count #$T(X_{15}[4])$.

Our 20-Round Attack



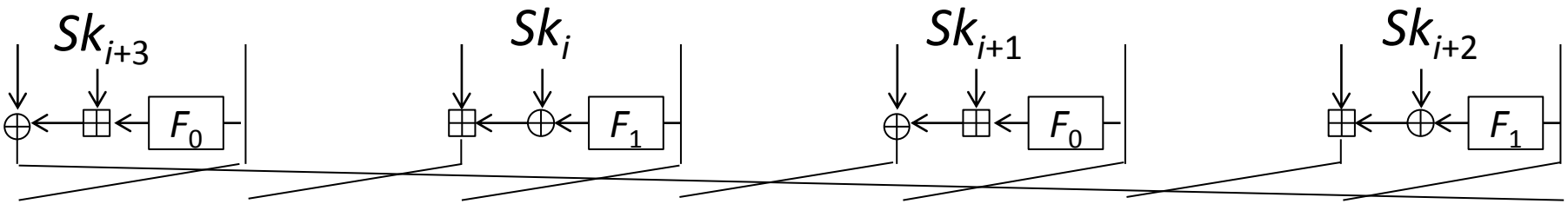
- #$K(Z_{15})$=32
- #$T(Z_{15})$=32
- The complexity is $2^{32+32} = 2^{64}$.

  Valid Attack !!
- Further optimization by the ***partial-sum***:

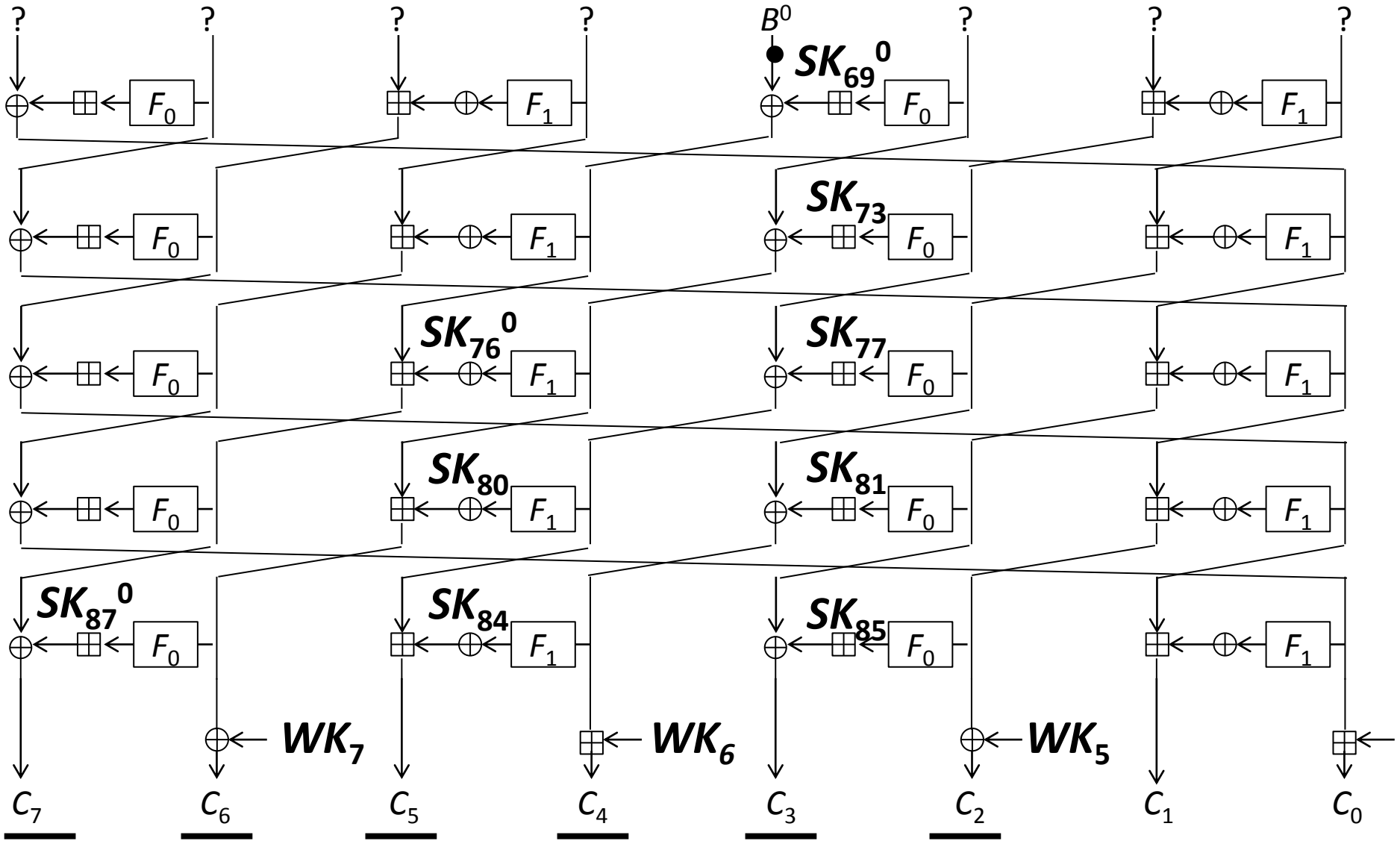
  Complexity is $\mathbf{2^{36}}$.

HIGHT

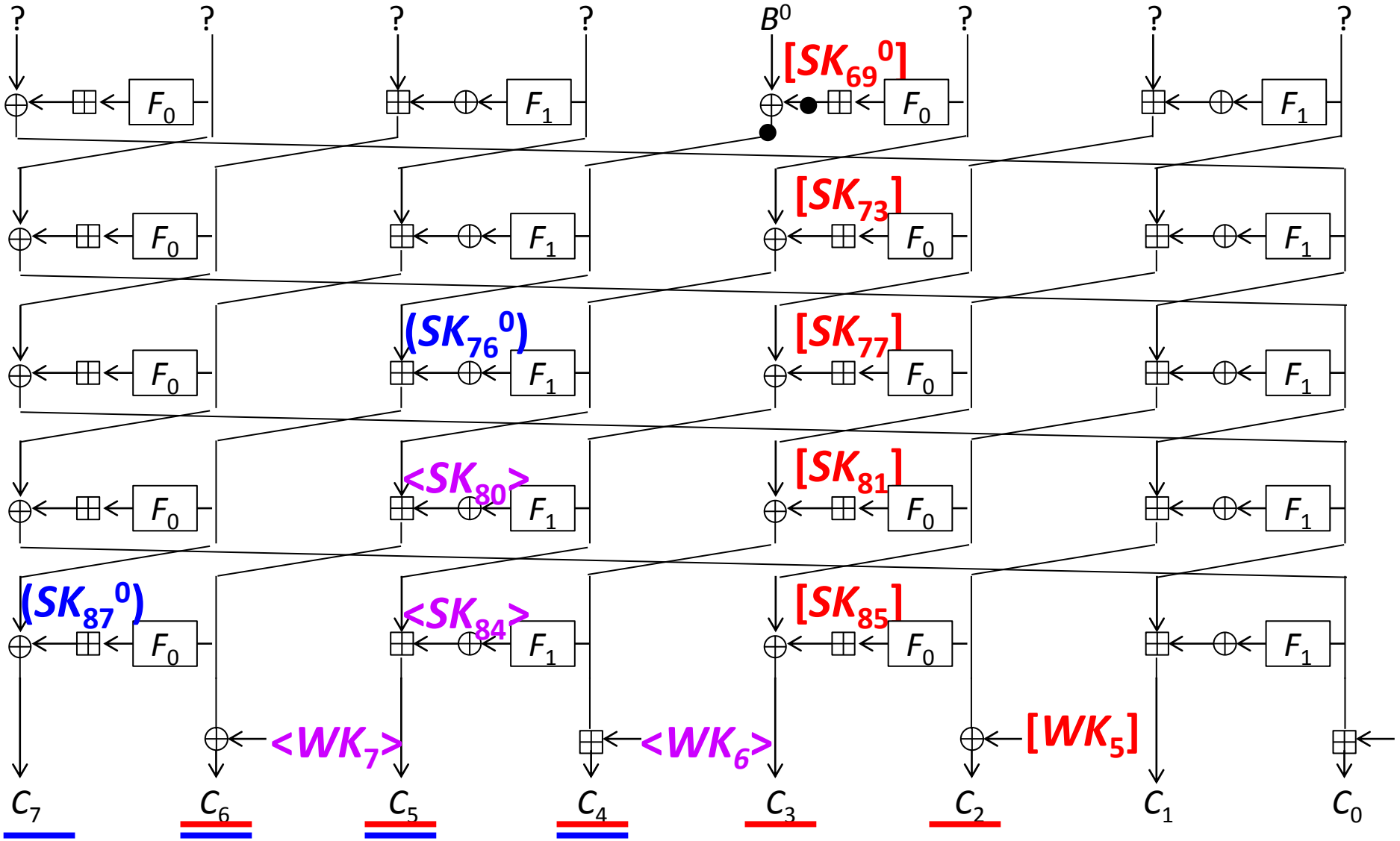
- Proposed by Hong et al. at CHES 2006.
- 64-bit block, 128-bit key.
- Generalized Feistel network with 8 branches, in total 32 rounds.
- 17-round distinguisher is known.



22-Round Attack on HIGHT



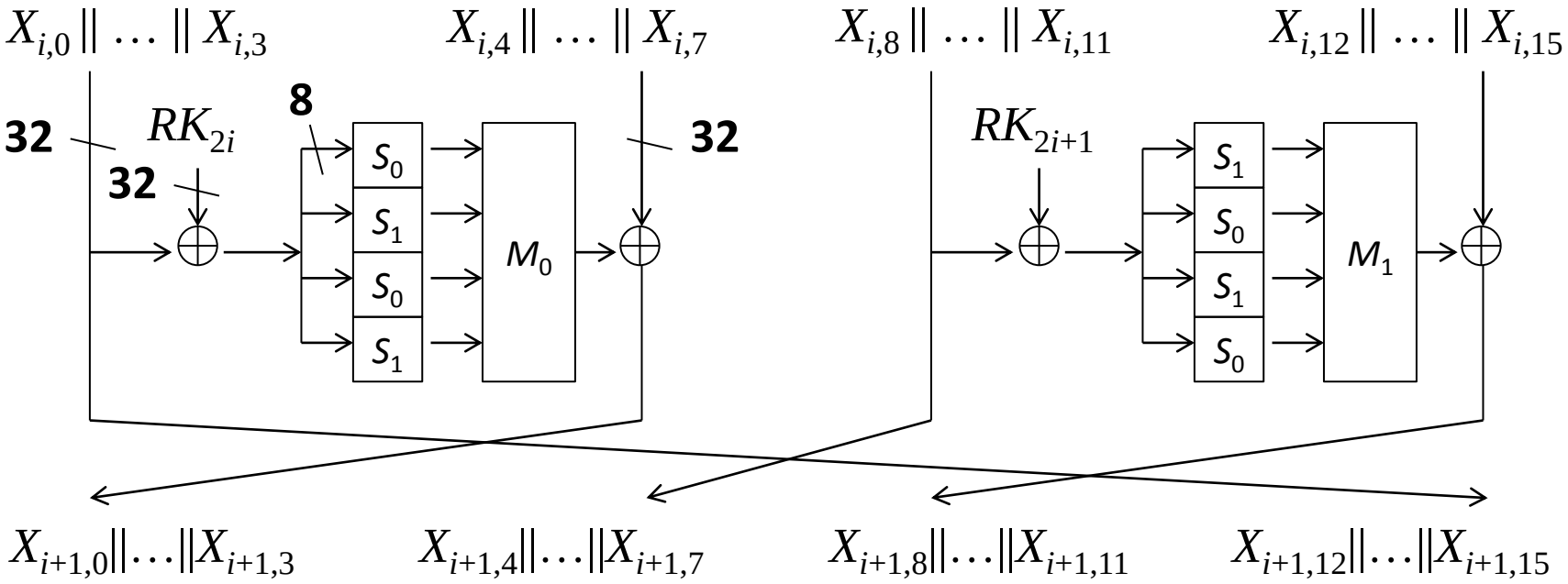
22-Round Attack on HIGHT



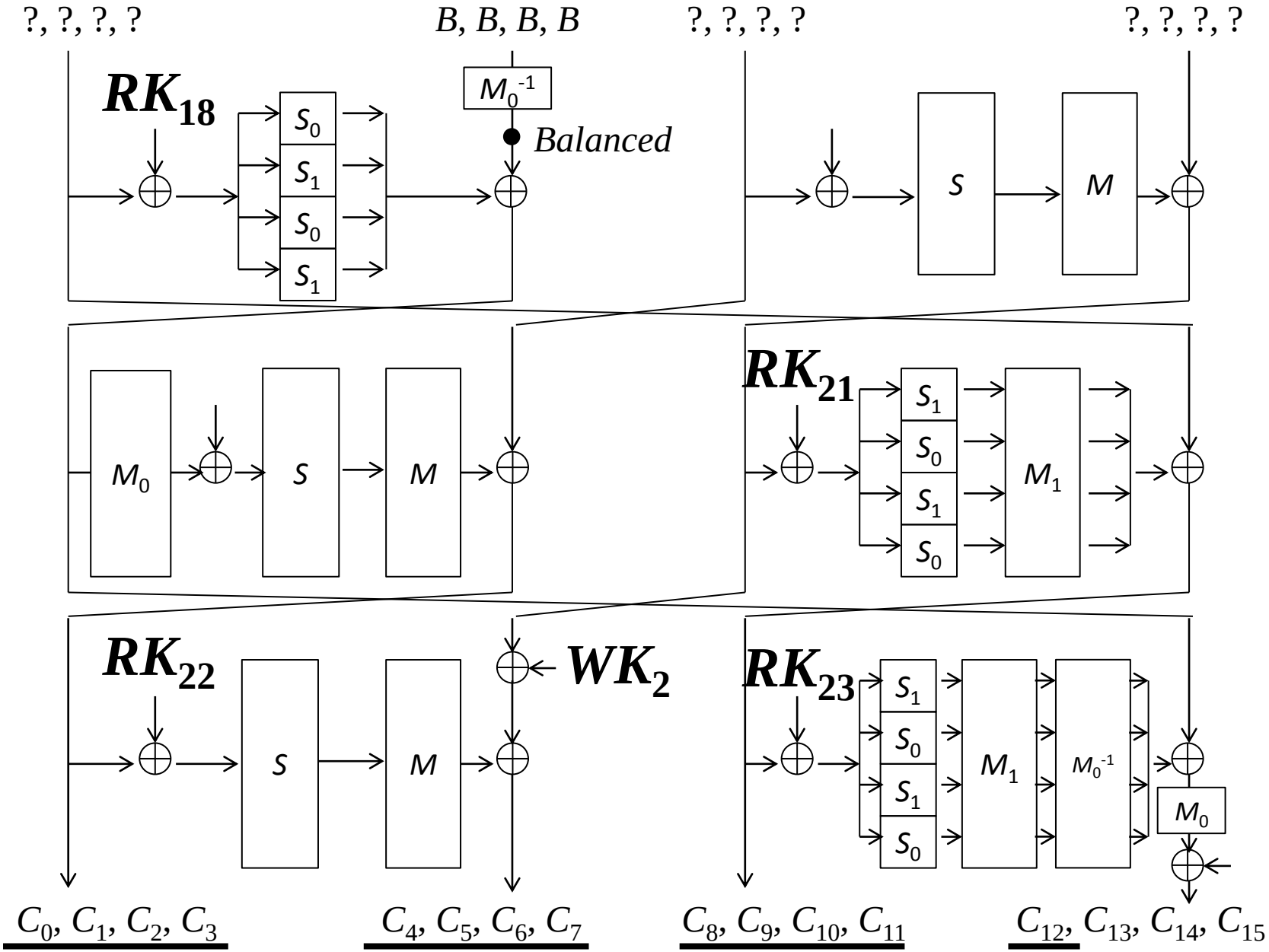
Our improvement contains other small observations.

CLEFIA-128

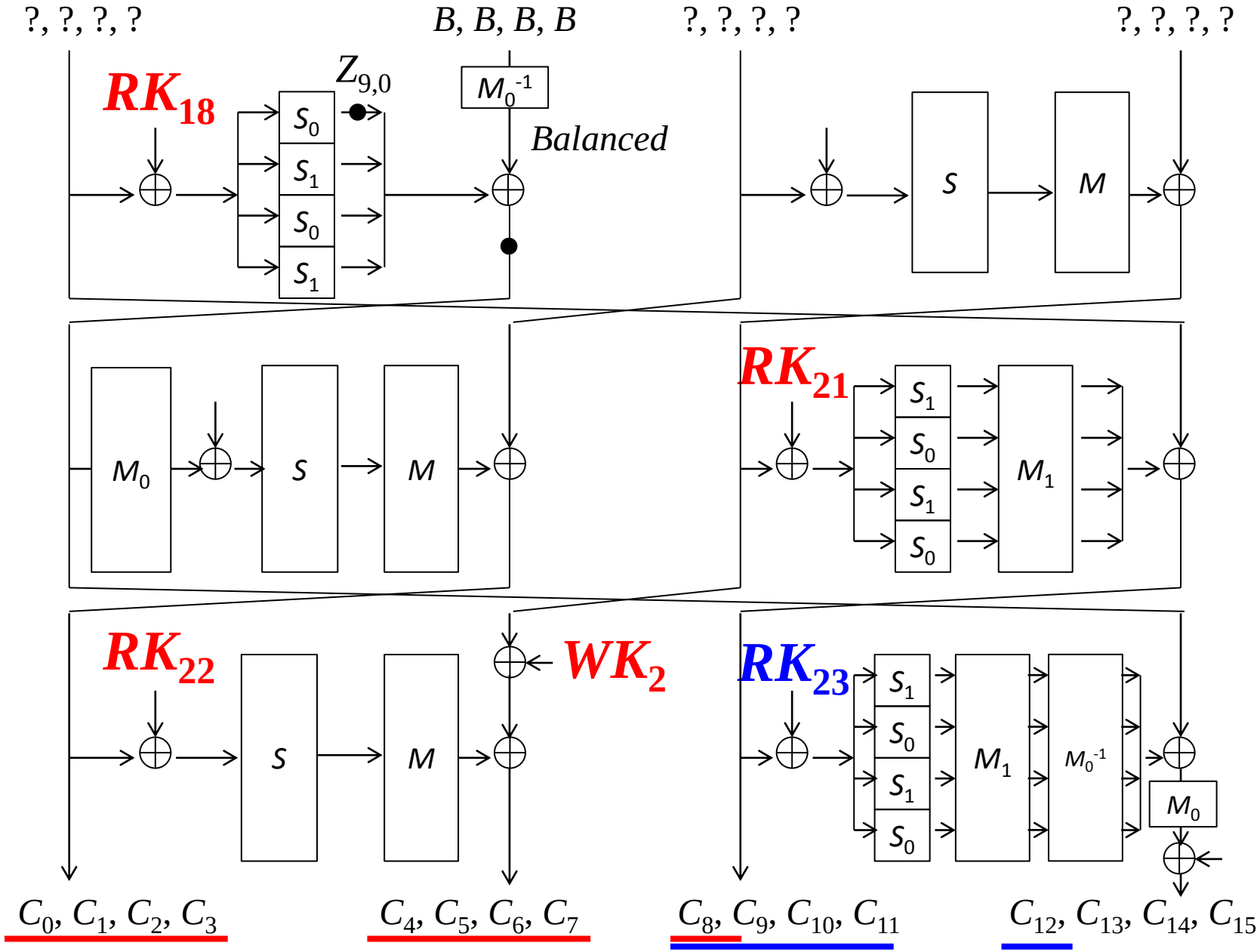
- Proposed by Shirai et al. at FSE 2007.
- 128-bit block, 128-bit key.
- Generalized Feistel network with 4 branches, in total 18 rounds.
- 9-round distinguisher is known.



12-Round Attack on CLEFIA-128



12-Round Attack on CLEFIA-128



Partial-Sum for Feistel Ciphers

- We also improve the partial-sum on CLEFIA, which is applied to Feistel ciphers in generic.

$$\bigoplus \Big[S_0 \Big(\underbrace{S_1(C_8 \oplus RK_{21,0}) \oplus 08 \cdot S_0(C_9 \oplus RK_{21,1}) \oplus}_{X} \\ 02 \cdot S_1(C_{10} \oplus RK_{21,2}) \oplus 0a \cdot S_0(C_{11} \oplus RK_{21,3}) \oplus C_{12} \oplus RK'_{18,0} \Big) \Big] \\ = \bigoplus C', \tag{8}$$

- Previous: Guess 2 key bytes, and then compress.
-

Previous: $2^{40+16}=2^{56}$

Partial-Sum for AES [reuse]

$$\bigoplus_{n=1}^{2^{32}} \left[S_4 \left(S_0(c_{0,n} \oplus k_0) \oplus S_1(c_{1,n} \oplus k_1) \oplus S_2(c_{2,n} \oplus k_2) \oplus S_3(c_{3,n} \oplus k_3) \oplus k_4 \right) \right]$$

x y z

- Computation starts from 2^{32} texts (c_0, c_1, c_2, c_3).
  - Guess two key bytes k_0, k_1.
  - For each guess, compute 2^{32} tuples (x, c_1, c_2).
  - Only pick (x, c_1, c_2) which appear odd times.
  - The size of the set is compressed into 2^{24}.
    - Guess one key byte k_2.
    - For each guess, compute 2^{24} tuples (y, c_2).
    - Only pick (y, c_2) which appear odd times.
    - The size of the set is compressed into 2^{16}.

Time: $2^{32}*2^{16}=2^{48}$

Time: $2^{24}*2^{16}*2^{8}=2^{48}$

Partial-Sum for Feistel Ciphers

- We also improve the partial-sum on CLEFIA, which is applied to Feistel ciphers in generic.

$$\bigoplus \Big[S_0 \Big(\underbrace{S_1(C_8 \oplus RK_{21,0}) \oplus 08 \cdot S_0(C_9 \oplus RK_{21,1}) \oplus}_{x} \\ 02 \cdot S_1(C_{10} \oplus RK_{21,2}) \oplus 0a \cdot S_0(C_{11} \oplus RK_{21,3}) \oplus \underbrace{C_{12} \oplus RK'_{18,0}}_{y} \Big) \Big] \\ = \bigoplus C', \qquad (8)$$

- Previous: Guess 2 key bytes, and then compress.
- Ours: Feistel ciphers usually includes the term which only consists of ciphertext.

  Guess only 1 byte, and then compress.

Previous: $2^{40+16}=2^{56}$ Ours: $2^{40+8}=2^{48}$

NTT

Summary of Our SAC Paper

- Use MitM approach for integral attacks.
- Well-applied to Feistel ciphers.
- Applied it to LBlock, HIGHT, and CLEFIA-128 together with other improvements.

NTT

Integral Attack Goes More than Impossible Differential Attack for LBlock

NTT

Life is so HARD!

Target	Rounds	Data	Time	Ref.
Imp. Diff.	21	$2^{62.5}$	$2^{73.7}$	[LG++11]
Integral	20	$2^{63.6}$	$2^{39.6}$	[SW12]
	21	$2^{61.6}$	$2^{54.2}$	**Ours**
	22	2^{61}	$2^{70.0}$	**Ours**
Biclique	32	$2^{52.7}$	$2^{78.4}$	[WW++12]

NTT

Life is so HARD!

Target	Rounds	Data	Time	Ref.
Imp. Diff.	21	$2^{62.5}$	$2^{73.7}$	[LG++11]
Integral	20	$2^{63.6}$	$2^{39.6}$	[SW12]
	21	$2^{61.6}$	$2^{54.2}$	**Ours**
	22	2^{61}	$2^{70.0}$	**Ours**
	22	$2^{61.6}$	$2^{71.2}$	[L12]*
Biclique	32	$2^{52.7}$	$2^{78.4}$	[WW++12]

*: Yanjun Li. Integral Cryptanalysis on Block Ciphers (in Chinese): [D]. Beijing: Institute of Software, Chinese Academy of Sciences, 2012.

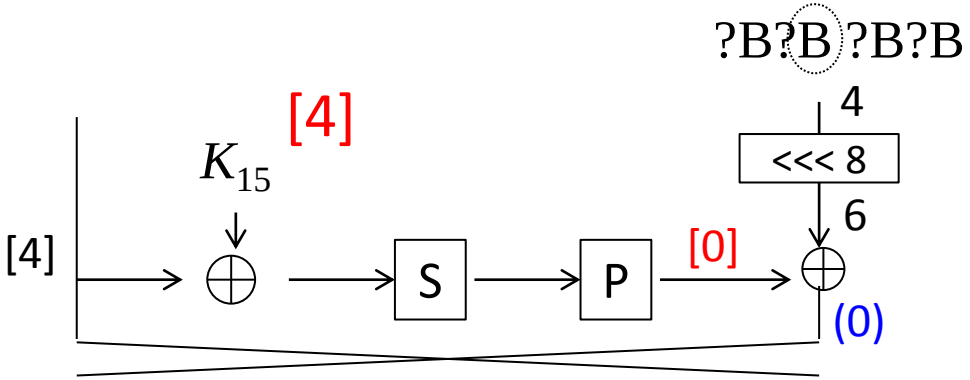
NTT

We knew that

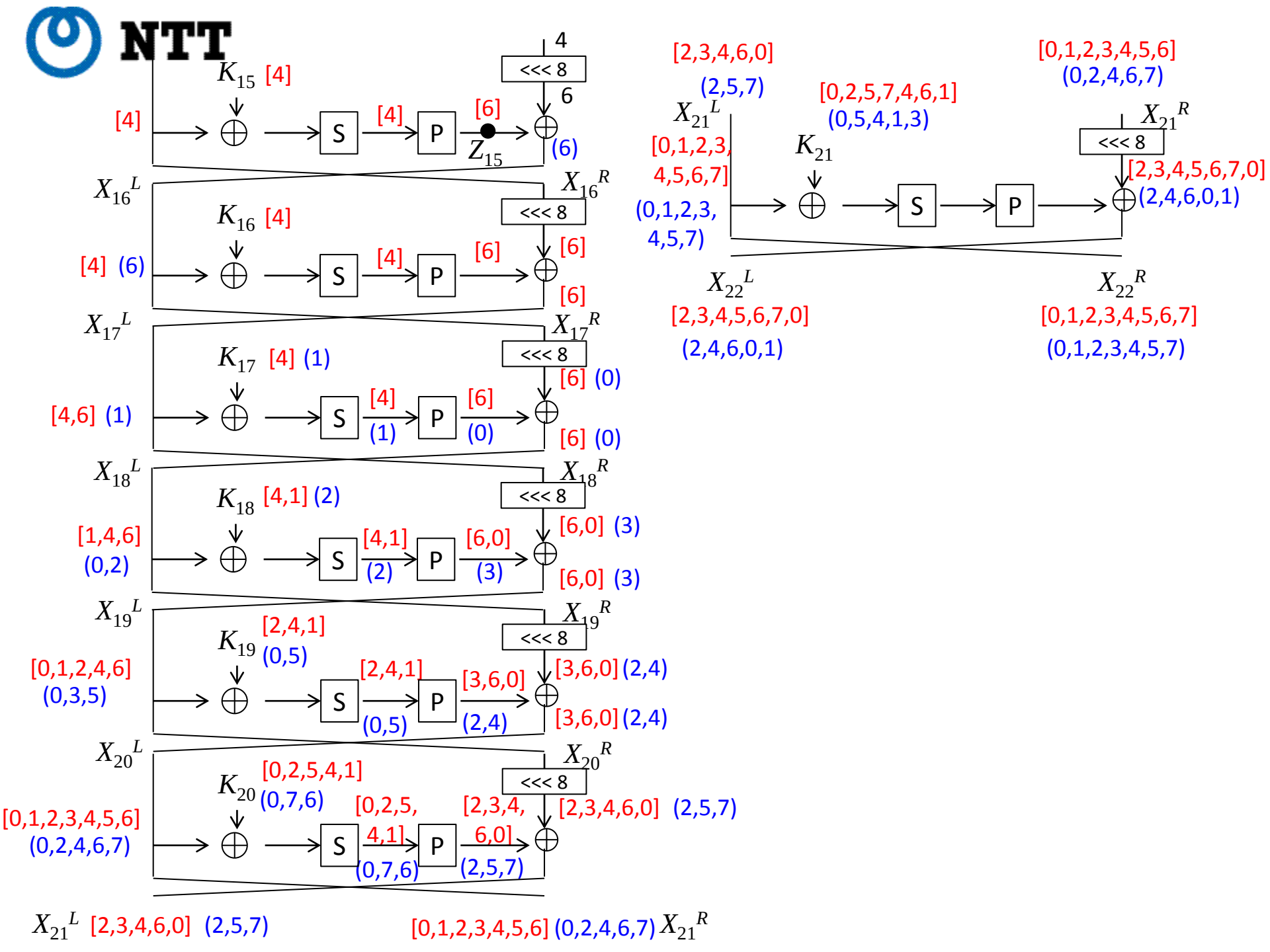
Integral Attack Goes More than Impossible Differential Attack for LBlock

The contents in this talk are completely independent of the results in [L12].

Comprehensive Analysis



- Try all possible balanced byte positions.
- Meet-in-the-Middle technique
- Partial-sum technique
- Subkey relations
- Combining exhaustive search


NTT
K_{15} [4]
[4]
S
[4]
P
[6]
Z_{15}
<<< 8
4
6
(6)
$X_{16}{}^L$
$X_{16}{}^R$
K_{16} [4]
[4] (6)
[6]
[6]
$X_{17}{}^L$
$X_{17}{}^R$
K_{17} [4] (1)
[6] (0)
[4,6] (1)
[4] (1)
[6] (0)
[6] (0)
$X_{18}{}^L$
$X_{18}{}^R$
K_{18} [4,1] (2)
[6,0] (3)
[1,4,6] (0,2)
[4,1] (2)
[6,0] (3)
[6,0] (3)
$X_{19}{}^L$
$X_{19}{}^R$
K_{19} [2,4,1] (0,5)
[3,6,0] (2,4)
[0,1,2,4,6] (0,3,5)
[2,4,1] (0,5)
[3,6,0] (2,4)
[3,6,0] (2,4)
$X_{20}{}^L$
$X_{20}{}^R$
K_{20} [0,2,5,4,1] (0,7,6)
[2,3,4,6,0] (2,5,7)
[0,1,2,3,4,5,6] (0,2,4,6,7)
[0,2,5, 4,1] (0,7,6)
[2,3,4, 6,0] (2,5,7)
$X_{21}{}^L$ [2,3,4,6,0] (2,5,7)
[0,1,2,3,4,5,6] (0,2,4,6,7) $X_{21}{}^R$
[2,3,4,6,0] (2,5,7)
$X_{21}{}^L$
[0,2,5,7,4,6,1] (0,5,4,1,3)
K_{21}
[0,1,2,3, 4,5,6,7] (0,1,2,3, 4,5,7)
S
P
<<< 8
[0,1,2,3,4,5,6] (0,2,4,6,7)
$X_{21}{}^R$
[2,3,4,5,6,7,0] (2,4,6,0,1)
$X_{22}{}^L$
[2,3,4,5,6,7,0] (2,4,6,0,1)
$X_{22}{}^R$
[0,1,2,3,4,5,6,7] (0,1,2,3,4,5,7)

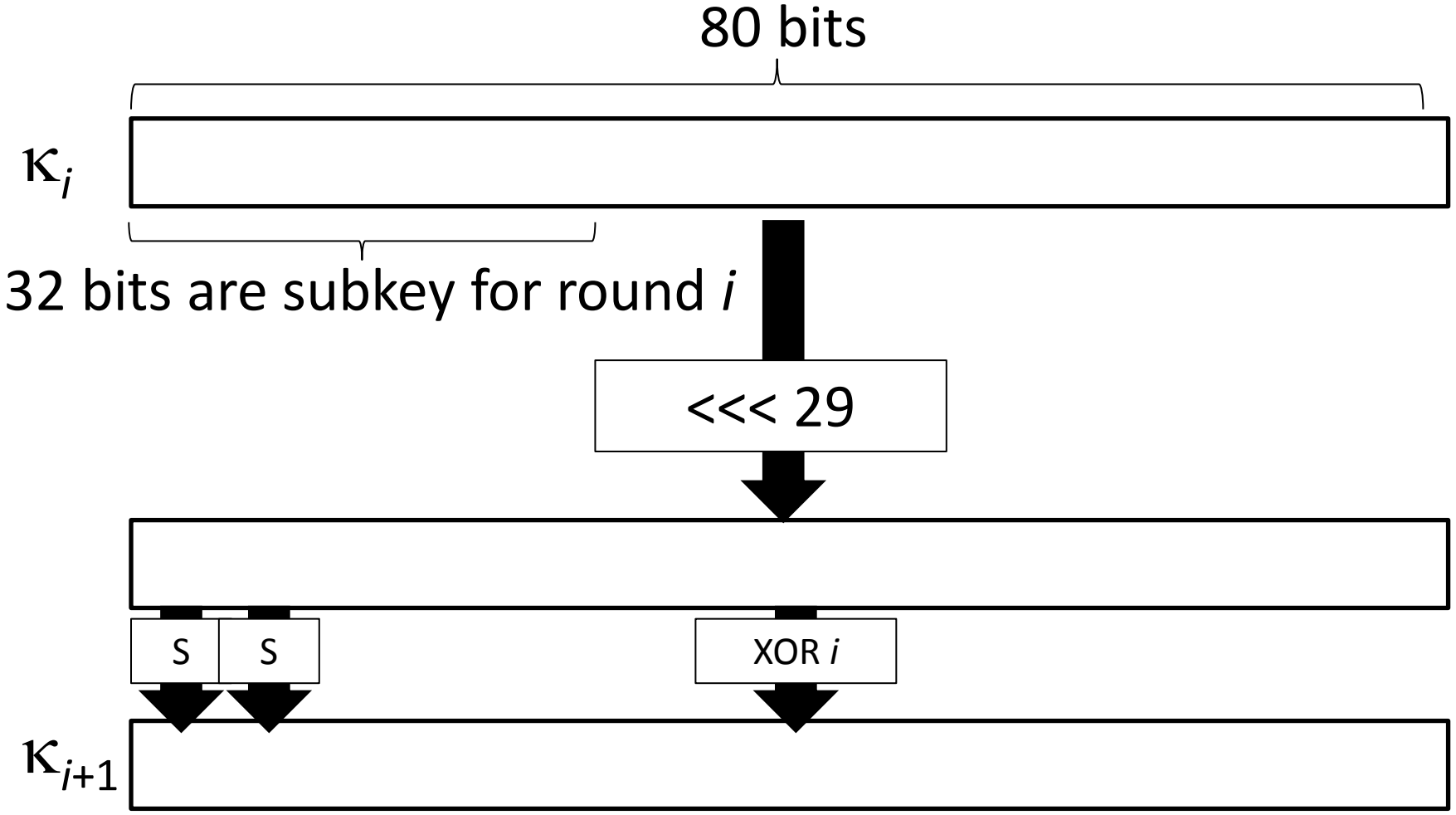
NTT

Overview

- #Guessed key bytes becomes 32 bytes (20 bytes for the red part, 12 bytes for the blue part)
- #related ciphertext bytes are also many (15 bytes for the red part and 12 bytes for the blue part).
- Using the partial-sum is necessary, but still not enough to be a valid attack.
- Relations between subkeys must be considered.

Considering the key schedule

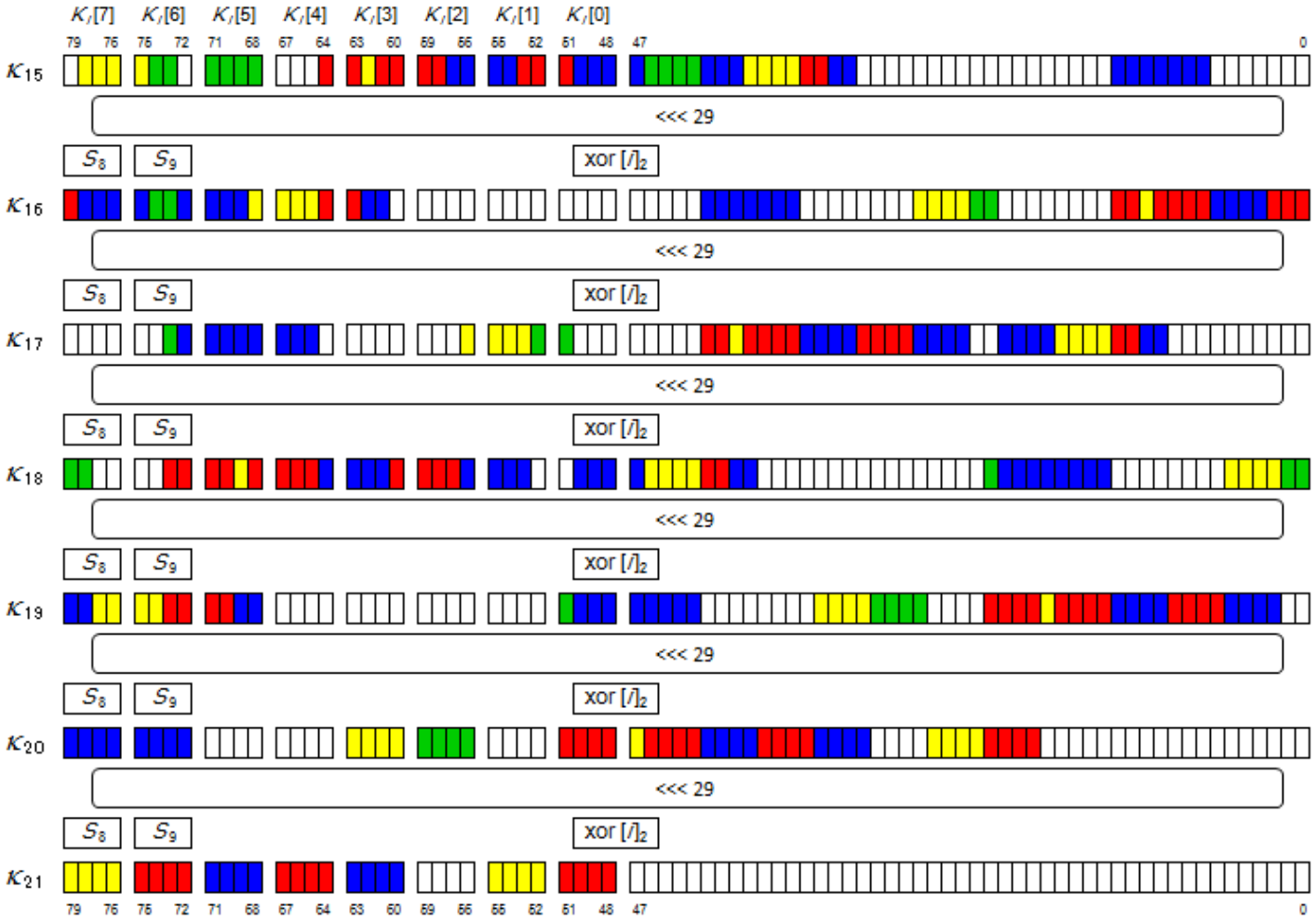
- Master key is loaded into the 80-bit key state κ_0.



Key Schedule Analysis

#overlapped bits depends on guessed key-byte positions.

The analysis must be iterated for all balanced-byte positions.



NTT

Identifying best balanced-byte position

Exhaustively checked which balanced-byte position is the best to mount an integral attack.

	0th	2nd	4th	6th
21R red	50	44	47	42
21R both	63	61	63	57
22R red	62	55	63	65
22R both	75	69	75	77

Summary

- We did comprehensive analysis on LBlock. Optimize the attack with all exiting techniques.
- The number of attacked rounds is extended.
- First example that the integral analysis could beat the impossible differential analysis.

Target	Rounds	Data	Time	Ref.
Imp. Diff.	21	$2^{62.5}$	$2^{73.7}$	[LG++11]
Integral	22	2^{61}	$2^{70.0}$	**Ours**
	22	$2^{61.6}$	$2^{71.2}$	[L12]*

NTT

Thanks for your attention !!