暨南大學 JINAN UNIVERSITY


中國科學院信息工程研究所 INSTITUTE OF INFORMATION ENGINEERING,CAS

Improving the Rectangle Attack on GIFT-64

Yincen Chen, Nana Zhang, Xuanyu Liang, Ling Song,
Qianqian Yang, and Zhuohui Feng

Jinan University
IIE, Chinese Academy of Sciences

August 14, 2023
SAC 2023

Table of Contents

Table of Contents

GIFT is a lightwight **bit-wise** block cipher with Substitution-Permutation-Network, which Banik *et al.* proposed at CHES' 2017.

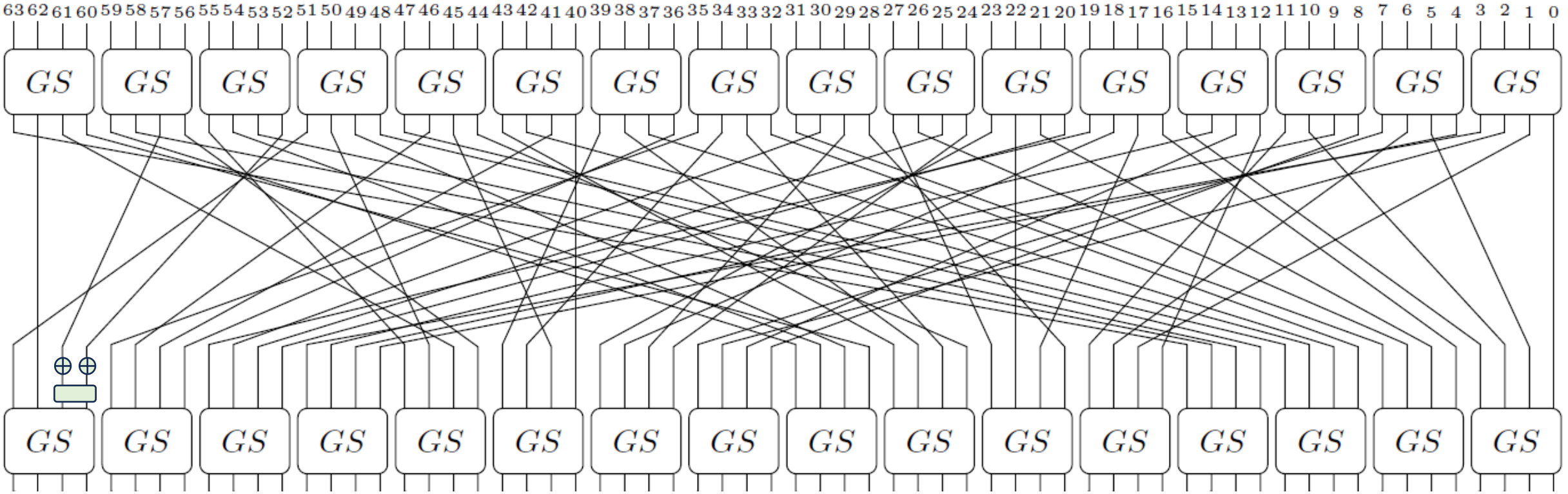


Fig. One round encryption of GIFT-64

- Bit Permutation, diffuse 4 bits in each cell to 4 cells in the next round.

$$P[i] = \left\{4\lfloor\frac{i}{16}\rfloor + 16\left(3\lfloor\frac{i \bmod 16}{4}\rfloor + (i \bmod 4) \bmod 4\right) + (i \bmod 4)\right\} \bmod 64.$$

- Key Schedule, rotation only, reused in 5 rounds.

$$k_7||k_6 \ldots||k_1||k_0\ (16\ bits * 8) \leftarrow K\ (128\ bits)$$

Subkey update

$$k_7||k_6|| \ldots ||\ k_1\ ||k_0 \leftarrow\ k_1 \ggg 2||k_0\ \ggg 12\ ||\ \ldots||k_3||k_2$$

Table of Contents

- The Differential attack [Biham *et al.* in 1991]:
  find $\Delta P \rightarrow \Delta C$ with high probability P (Hard to find long trails).

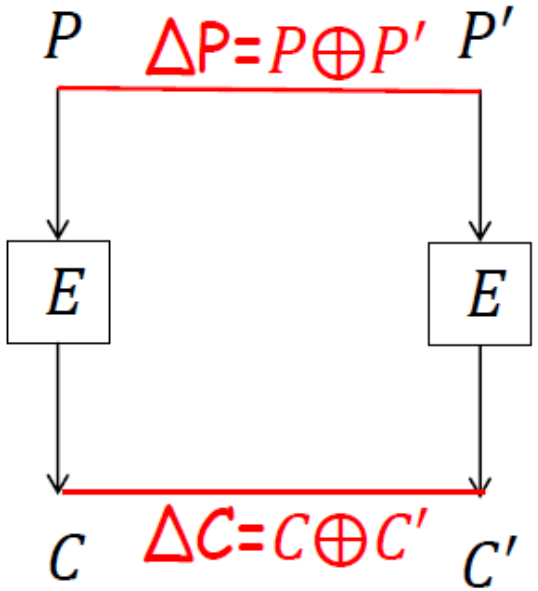


Fig. Differential

$$P = P_r\{C \oplus C' = \Delta C | P \oplus P' = \Delta P\}$$

- The Boomerang attack [Wagner in 1999]:

  Treat cipher E as $E_0 \circ E_1$, calculate the probability (p and q) and connect them.

◆Adaptive chosen plaintext/ciphertext attack.

$$\begin{cases} \alpha \to \beta & with\ probability\ \boldsymbol{p} \\ \gamma \to \delta & with\ probability\ \boldsymbol{q} \end{cases}$$

$$P_{boomerang} = \boldsymbol{p^2 q^2}$$

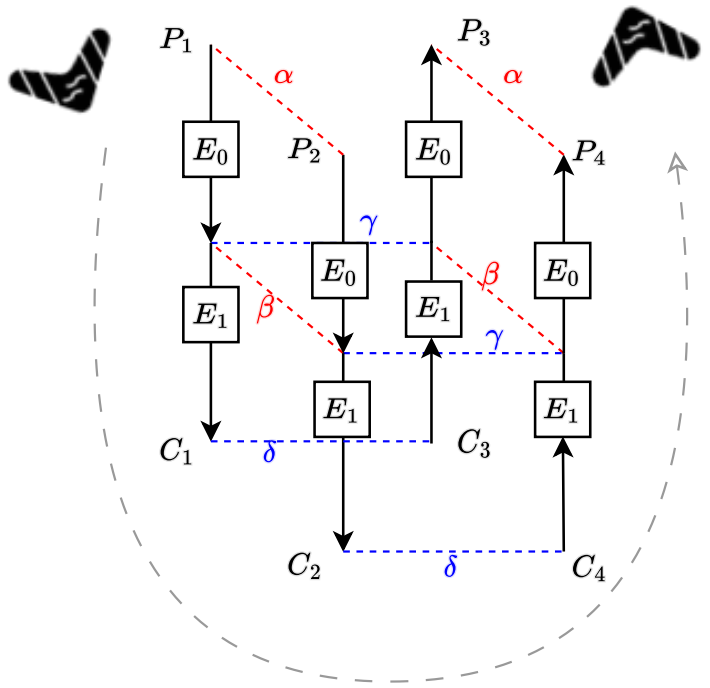


Fig. Boomerang attack

Boomerang attack

Amplified boomerang attack

[Kelsey *et al.* in 2000]

(choose plaintext/ciphertext)

Rectangle attack

[Biham *et al.* in 2001]

(choose plaintext/ciphertext)

$$\hat{p} = \sqrt{\sum_i P_r^2(\alpha \to \beta_i)}$$

$$\hat{q} = \sqrt{\sum_j P_r^2(\gamma_i \to \delta)}$$

$$P_{rectangle} = \mathbf{2^{-n}\hat{p}^2\hat{q}^2}$$

◆Steps:

- Choose plaintext $P_1 \oplus P_2 = P_3 \oplus P_4 = \alpha$.

$$P_1, P_2, P_3, P_4 \xrightarrow{E_k} C_1, C_2, C_3, C_4.$$

- If $C_1 \oplus C_3 = C_2 \oplus C_4 = \delta$, right quartet

$$\{(P_1, C_1), (P_2, C_2), (P_3, C_3), (P_4, C_4)\}.$$

- Song *et al.* proposed the most efficient and generic rectangle key recovery algorithm at ASIACRYPT 2022

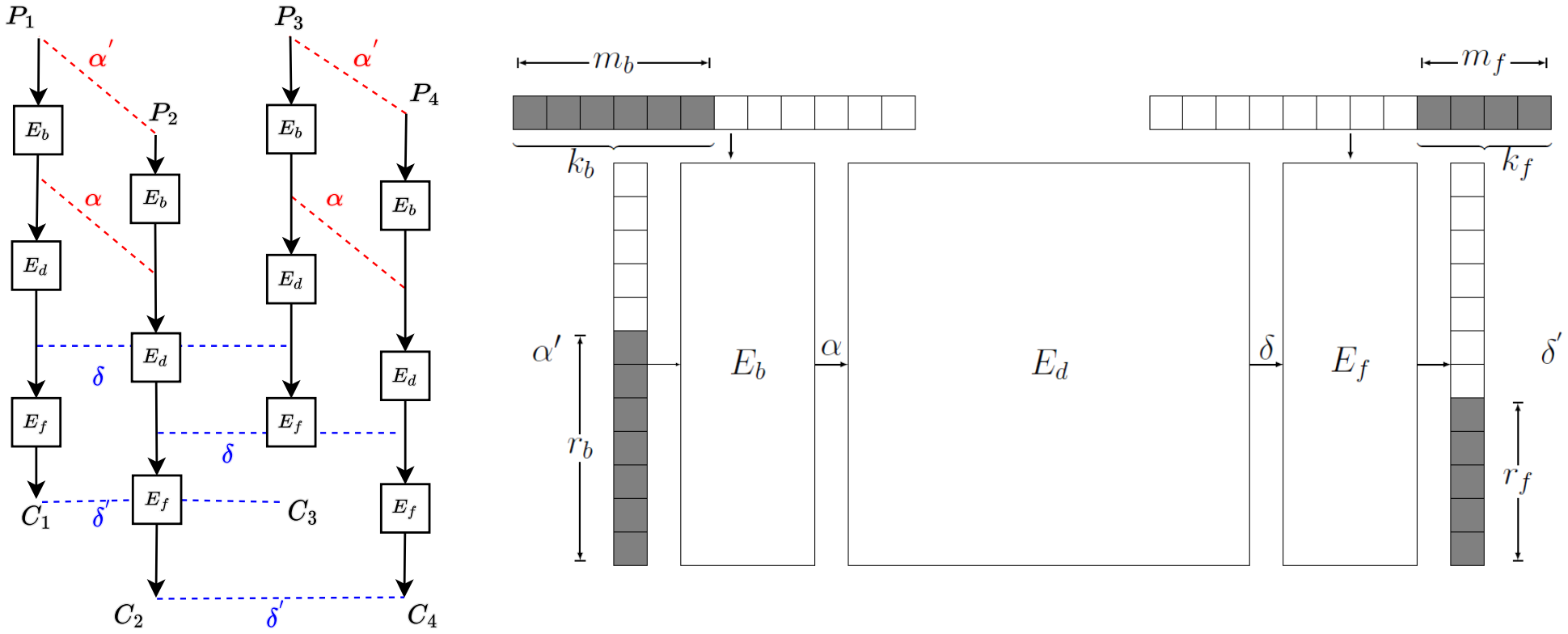


Fig. Outline of rectangle key recovery attack

Table of Contents

◆ Notation

$$\Delta Plaintext \xrightarrow{GS\ \&\ Bit\ permutation} \Delta Z_1 \xrightarrow{\boldsymbol{k_1}||\boldsymbol{k_0}} \Delta X_2 \xrightarrow{GS} \Delta Y_2 \cdots\cdots$$

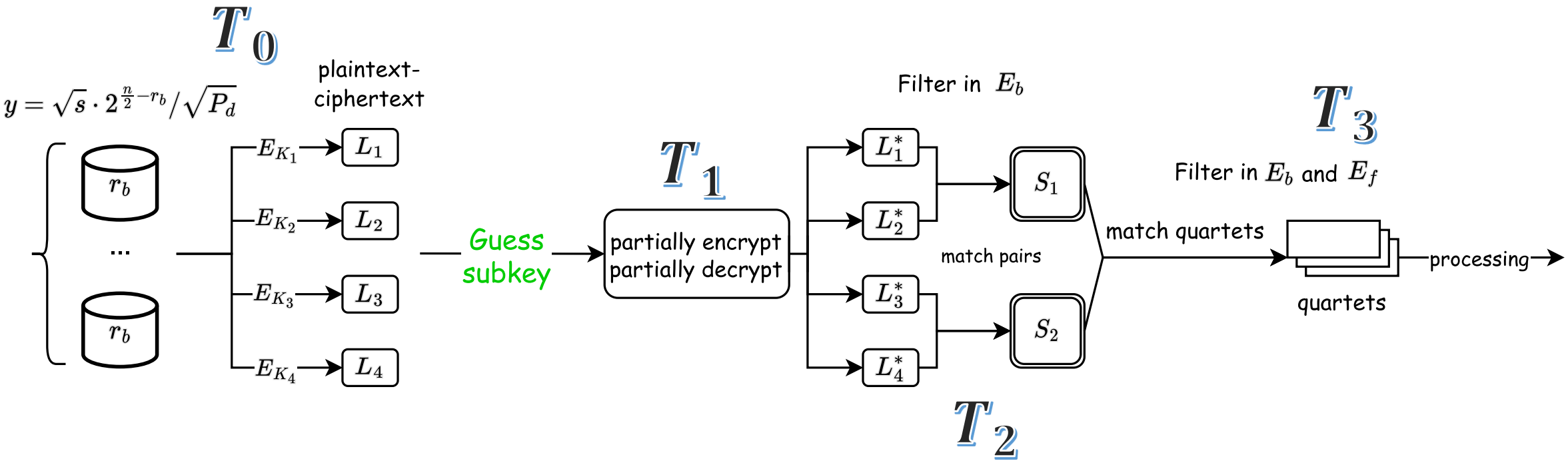
- **Based on the 20-round** rectangle distinguisher of GIFT-64 which proposed by Ji et al. in SAC 2020
- Extend **3** rounds forward and backward from the distinguisher
- The state: (backward i.e. E_b)

  $\boldsymbol{\Delta X}$: after ***subkey addition*** ⇔ before ***Sbox***;

  $\boldsymbol{\Delta Y}$: after ***Sbox*** ⇔ before ***bit permutation***;

  $\boldsymbol{\Delta Z}$: after ***bit permutation*** ⇔ before ***subkey addition***.

- The related-key rectangle key recovery attack on GIFT-64.



- Exhaustive search T_4

➢ Core idea:

Tradeoff the time complexity of each attack phase.

➢ Main question:

How to guess subkey bit and obtain the filtering bits?

➢ Basic rules:

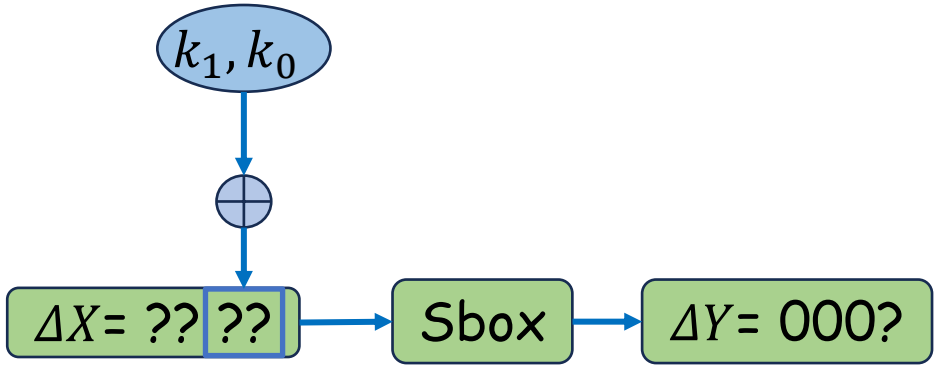
- ◆ Guessing fewer, acquiring more
- ◆ Balance the time complexity of
  - partially encryption and decryption. (T_1)
  - pair generation (T_2)
  - quartet generation and processing (T_3)

◆ How to get filtering bits by guessing subkey bits?

➢ Guessing m'_b $(resp. m'_f)$ subkey bits to obtain > $\mathbf{m'_b}$ **filtering bits** r'_b $(resp. r'_f)$

- Example:

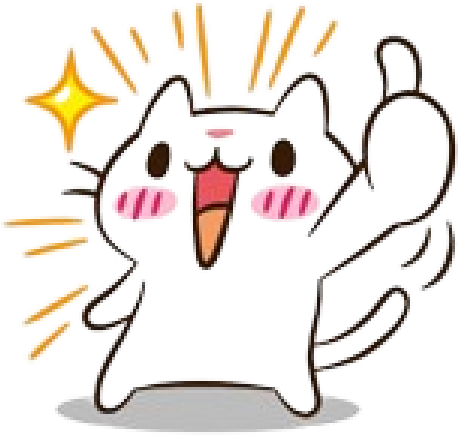
  3 bit fix diff. (000?) - 0 bit fix diff. (????) = get 3 filtering bits



➢ Advantage for time complexity: $\mathbf{2^2 \times 2^{-3} = 2^{-1}}$

More advantage with subkey reusing:

- Highlight green subkey bits can be used in round **1** and round **25**
- Highlight cyan subkey bits can be used in round **2** and round **26**
- Maximize $\frac{\textbf{\textit{guessed subkey bits}}}{\textbf{\textit{filtering bits}}}$ ⇔ Maximize advantage !!



ΔZ_1	????	????	????	????	0000	0000	0000	0000	11??	????	????	????	????	11??	????	????
$k_1 \| k_0$	15|15	14|14	13|13	12|12	11|11	10|10	9|9	8|8	7|7	6|6	5|5	4|4	3|3	2|2	1|1	0|0
ΔZ_2	????	0000	?1??	0000	0000	0000	0000	0000	0001	0000	0000	0000	0000	0000	0000	?1??
$k_3 \| k_2$	15|15	14|14	13|13	12|12	11|11	10|10	9|9	8|8	7|7	6|6	5|5	4|4	3|3	2|2	1|1	0|0
...	... ...															
ΔZ_{25}	??0?	??0?	??0?	??0?	???0	???0	???0	???0	0???	0???	0???	0???	?0??	?0??	?0??	?0??
$k_1 \| k_0$	11|7	10|6	9|5	8|4	7|3	6|2	5|1	4|0	3|15	2|14	1|13	0|12	15|11	14|10	13|9	12|8
ΔZ_{26}	????	????	????	????	????	????	????	????	????	????	????	????	????	????	????	????
$k_3 \| k_2$	11|7	10|6	9|5	8|4	7|3	6|2	5|1	4|0	3|15	2|14	1|13	0|12	15|11	14|10	13|9	12|8

Table. reused subkey bits

Table of Contents

◆ Variables:

- All involved subkey bits;
- All difference of involved state bits (fixed or unknown).

◆ Constraints

- The relation between guessing subkey bits and fixed state bits;
- The relation between {guessed subkey bits, fixed state bits} and time complexity.

◆ Objective function

- **The most balanced time complexity.**

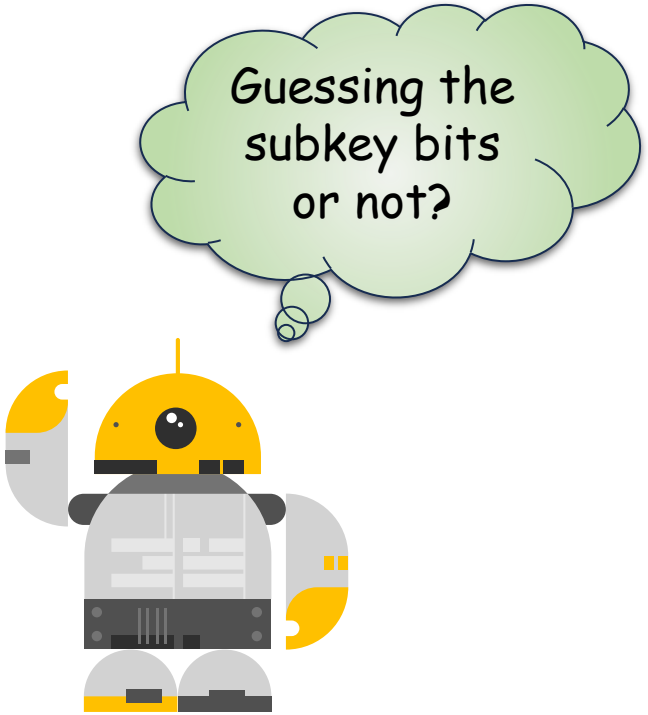
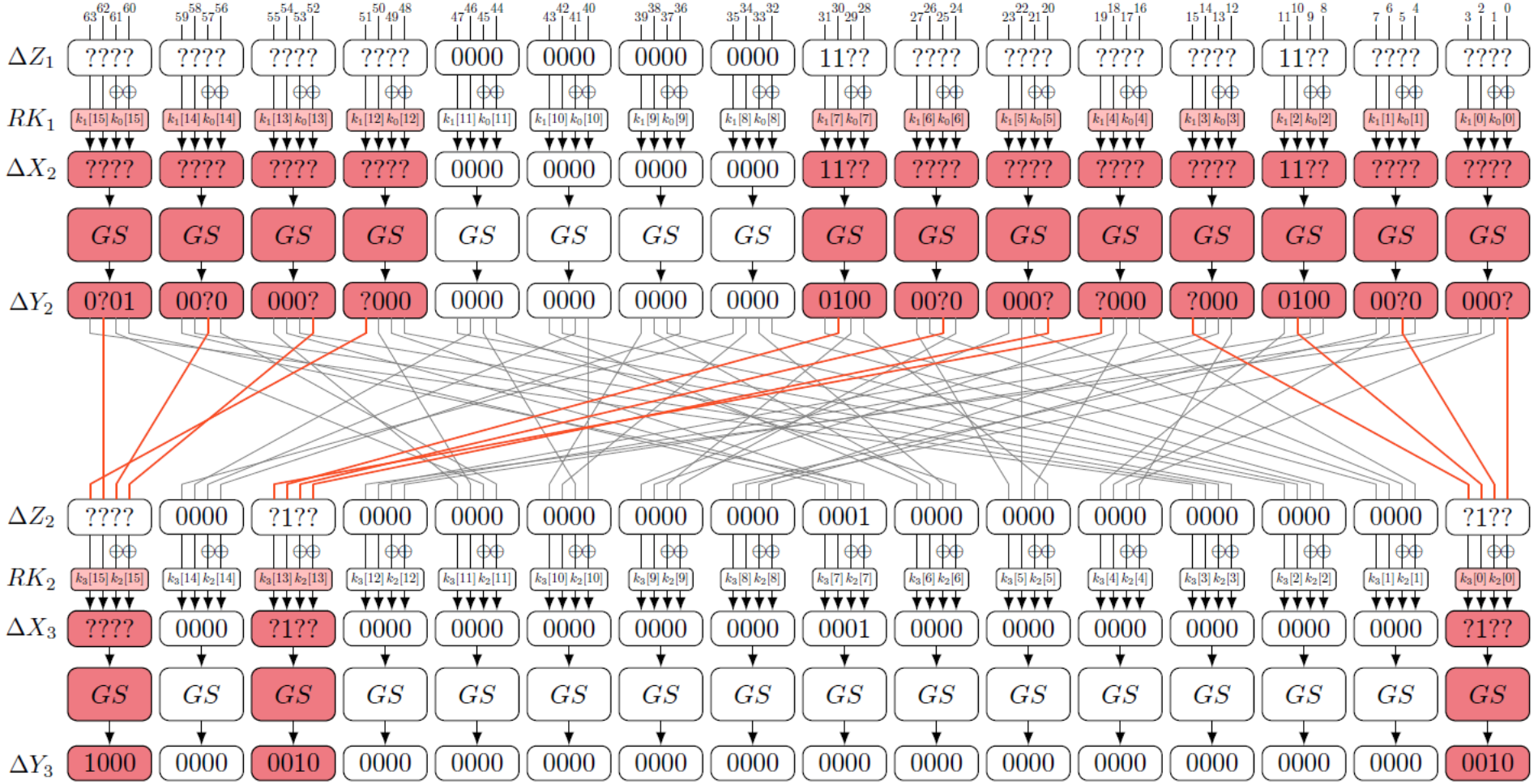


Table of Contents

Result and extensions

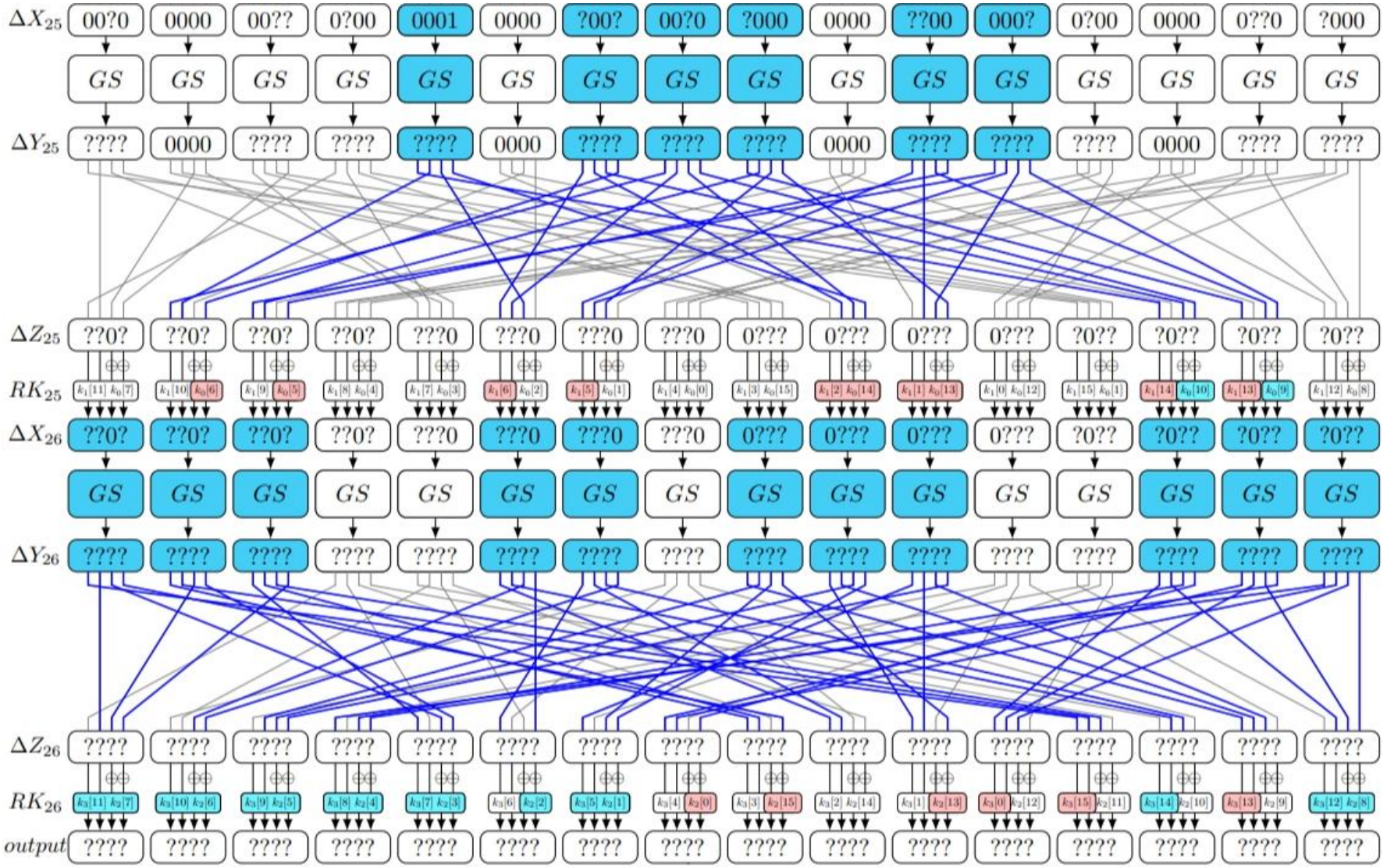
✓ Guessing strategy in E_b: Guessing m'_b=30 subkey bits to obtain r'_b=44 filtering bits.



Result and extensions

- ✓ Guessing strategy in E_f:

  Guessing m'_f=18 subkey bits to obtain r'_f=28 filtering bits *combine with reused subkey bits.*



Result and extensions

Choose $s = 2$ and $h = 20$

- ✓ E_b: $(m_b = 30,\ r_b = 44)$
  - m'_b=30, r'_b=44, r^*_b = 0
- ✓ E_f: $(m_f = 64,\ r_f = 64)$
  - m'_f=18, r'_f=28, r^*_f = 36

- $T_{total} = 2^{110.06}$ *enc.* & $2^{115.8}$ *m.a.*
- $D_{total} = 2^{63.78}$
- $M_{total} = 2^{64.36}$

$$D = y \cdot r_b$$

$$T_0 = 4 \cdot D = 2^{63.78}$$

$$T_1 = 2^{m'_b + m'_f} \cdot 4 \cdot D = 2^{111.78}$$

$$T_2 = 2^{m'_b + m'_f} \cdot 2 \cdot D \cdot 2^{r^*_b} = 2^{110.78}$$

$$T_3 = 2^{m'_b + m'_f} \cdot D^2 \cdot 2^{2r^*_b + 2r^*_f - 2n} \cdot \epsilon = 2^{115.56} \cdot \epsilon$$

$$T_4 = 2^{m'_b + m'_f + k - m'_b - m'_f - h} = 2^{k-h} = 2^{108}$$

$$D_{total} = 4 \cdot D = 2^{63.78}$$

$$\begin{aligned} M_{total} &= M_0 + M_1 + M_c \\ &= 4 \cdot D + 2 \cdot D \cdot 2^{r^*_b} + 2^{m^*_b + m^*_f} \\ &= 2^{63.78} + 2^{62.78} + 2^{36} \\ &\approx 2^{64.36} \end{aligned}$$

Result and extensions

➢ Strategy II

- ✓ E_b:
  - guessing **26** subkey bits, obtain **39** filtering bits;
- ✓ E_f:
  - guessing **22** subkey bits, obtian **34** filtering bits.

- $T_{total} = 2^{111.51}$ $enc.$ & $2^{115.78}$ $m.a.$
- $D_{total} = 2^{63.78}$
- $M_{total} = 2^{67.8}$

➢ Attack on 23-round GIFT-128

- ✓ E_b:
  - guessing **4** subkey bits, obtain **7** filtering bits;
- ✓ E_f:
  - guessing **0** subkey bits, obtian **0** filtering bits.

- $T_{total} = 2^{123.1}$ $enc.$ & $2^{126.31}$ $m.a.$
- $D_{total} = 2^{121.31}$
- $M_{total} = 2^{121.63}$

Result and extensions

Method (related-key)	Time	Data	Memory	Online
Differential	$2^{123.23}$ $enc.$	$2^{60.96}$	$2^{102.86}$	2021 ToSC
Rectangle	$2^{122.78}$ $enc.$	$2^{63.78}$	$2^{63.78}$	2022 E.C.
Rectangle	$2^{121.75}$ $enc.$	$2^{62.715}$	$2^{62.715}$	2022 Ins.C
Rectangle	$\mathbf{2^{110.06}}$ $\boldsymbol{enc.}$ & $\mathbf{2^{115.8}}$ $\boldsymbol{m.a.}$	$2^{63.78}$	$2^{64.36}$	This
Rectangle	$\mathbf{2^{111.51}}$ $\boldsymbol{enc.}$ & $\mathbf{2^{115.78}}$ $\boldsymbol{m.a.}$	$2^{63.78}$	$2^{67.8}$	This

Table. Results of 26-round attack on GIFT-64

Table of Contents

Summary

- The best attacks on GIFT-64 in terms of time complexity to date.
- An initial application of the generic rectangle key recovery algorithm for the bit-wise block ciphers
- The dedicated model of key recovery attack on GIFT-64.

Thank you!

Q & A

icsnow98@gmail.com