Asiacrypt 2023

清華大學
Tsinghua University

# Differential-Linear Approximation Semi-Unconstrained Searching and Partition Tree: Application to LEA and Speck

Yi Chen[1], Zhenzhen Bao[2,4], Hongbo Yu[3,4]

[1] Institute for Advanced Study, Tsinghua University
[2] Institute for Network Sciences and Cyberspace, BNRist, Tsinghua University
[3] Department of Computer Science and Technology, Tsinghua University
[4] Zhongguancun Laboratory

# Our Contributions



## Methods:

### 1. Differential-linear approximation (DLA) semi-unconstrained searching algorithms

a) Iterative search for short DLAs b) Meet-in-the-middle search for long DLAs

| Three-stage search [5,16,28] (before 2023) | Our algorithms | MIQCP/MILP models [6, 23] (reported in 2023) |
|---|---|---|
| The Hamming weight of linear masks is limited | 1. Have no constraints on the Hamming weight of linear masks<br>2. Apply to large-state ciphers | Fully automated but currently slow, i.e., not applying to large state ciphers |

### 2. Partition tree

a) A general tool for building partitions for various encryption functions, which breaks the barrier of applying the partitioning technique and partition-based key recovery attacks

# Our Contributions



## Applications:

### 3. Best-known or better distinguishers of LEA and Speck

| Cipher | Type | Round | Cor / Pr | Source |
|---|---|---|---|---|
| LEA | Boomerang | 16 (previous best) | $\mathrm{Pr} = 2^{-117.11}$ | [19] |
| | Differential-linear | 17 | $\mathrm{Cor} = -2^{-59.04}$ | This paper |
| Speck48 | Differential-linear | 11 (previous best) | $\mathrm{Cor} = -2^{-17.55}$ | [23] |
| | Differential-linear | 11 | $\mathrm{Cor} = -2^{-17.40}$ | This paper |

| Method | Speck32 | Speck48 | Speck64 | Speck96 | Speck128 |
|---|---|---|---|---|---|
| MIQCP/MILP [6] | $A_{10}(-12.0)$ | × | × | × | × |
| MIQCP/MILP [23] | $A_{10}(-11.58)$ | $A_{11}(-17.55)$ | $A_{12}(-26.93)$ | × | × |
| Ours | $A_{10}(-12.2)$ | $A_{11}(-17.40)$ | $\mathbf{A_{13}(-28.15)}$ | $A_{15}(-41.72)$ | $A_{18}(-55.81)$ |

× : not reported. $A_r(X)$ : an r-round DLA with an absolute correlation $2^X$.

# Our Contributions

Tsinghua University

## Applications:

### 4. Best-known key recovery attacks on all the members of LEA

| Variant | R.A./T.R. | Type | Time | Data (CP) | Source |
|---|---|---|---|---|---|
| LEA-128 | 14 / 24 | Differential | $2^{124.79}$ | $2^{124.79}$ | [30] |
| | **17 / 24** | **Differential-Linear** | $\mathbf{2^{82.9}}$ | $\mathbf{2^{70.9}}$ | **This paper** |
| LEA-192 | 14 / 28 | Differential | $2^{124.79}$ | $2^{124.79}$ | [30] |
| | **17 / 28** | **Differential-linear** | $\mathbf{2^{82.9}}$ | $\mathbf{2^{70.9}}$ | **This paper** |
| | **18 / 28** | **Differential-linear** | $\mathbf{2^{189.63}}$ | $\mathbf{2^{126.63}}$ | **This paper** |
| LEA-256 | 15 / 32 | Differential | $2^{252.79}$ | $2^{124.79}$ | [30] |
| | **17 / 32** | **Differential-linear** | $\mathbf{2^{82.9}}$ | $\mathbf{2^{70.9}}$ | **This paper** |
| | **18 / 32** | **Differential-linear** | $\mathbf{2^{189.63}}$ | $\mathbf{2^{126.63}}$ | **This paper** |

Our attacks are based on newly found distinguishers and the partitioning technique.

# DLA Semi-Unconstrained Searching



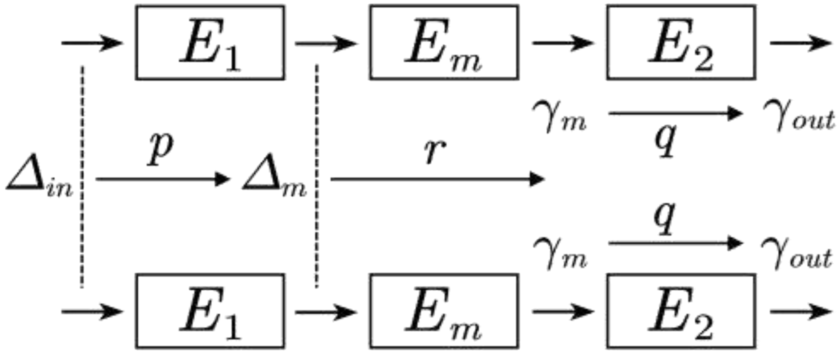



**Fig. 1.** The latest structure of differential-linear distinguishers.

**Before 2023, three-stage searching [5,16,28]:**

1.Verify short DLAs $\Delta_{\mathrm{m}} \xrightarrow{E_m} \gamma_{\mathrm{m}}$ $\boxed{/* \text{ for a difference } \Delta_{\mathrm{m}}, \gamma_{\mathrm{m}} = [i] \text{ or } [i, i+1] */}$

2. Search $\Delta_{\mathrm{in}} \xrightarrow{E_1} \Delta_{\mathrm{m}}$ and $\gamma_{\mathrm{m}} \xrightarrow{E_2} \gamma_{\mathrm{out}}$ under fixed $\Delta_{\mathrm{m}}$ and $\gamma_{\mathrm{m}}$

3. Connect three short distinguishers

# DLA Semi-Unconstrained Searching



## Core ideas:

1. **generate a new DLA from two known ones by the XOR operation.**

$$\Delta \to \gamma_1,\ \Delta \to \gamma_2 \implies \Delta \to \gamma_3 = \gamma_1 \oplus \gamma_2$$

$$\Delta \to \gamma_4,\ \Delta \to \gamma_4 \implies \Delta \to \gamma_6 = \gamma_4 \oplus \gamma_5$$

2. **preferentially verify the correlation of DLA generated from two DLAs with high absolute correlation.**

$$\text{If } |G_{i_1}| > c \text{ for } i_1 \in \{1,2\},\ |G_{i_2}| \leqslant c \text{ for } i_2 \in \{4,5\},$$

$$\text{we regard } \Delta \to \gamma_3 \text{ as a better choice.}$$

$$/ * G_i = \mathrm{Cor}(\Delta \to \gamma_i):\ \text{the correlation of } \Delta \to \gamma_i\ * /$$

# DLA Semi-Unconstrained Searching



## Motivation:

$$\Delta \xrightarrow{E} \gamma \qquad\qquad z_\gamma = \langle E(P) \oplus E(P \oplus \Delta), \gamma \rangle$$

**under the assumption that $z_{\gamma_i}$ are independent, we have**

$$\Delta \to \gamma_3 = \gamma_1 \oplus \gamma_2 \;\Rightarrow\; G_3 = G_1 \times G_2 \qquad \Delta \to \gamma_6 = \gamma_4 \oplus \gamma_5 \;\Rightarrow\; G_6 = G_4 \times G_5$$

**Since** $|G_{i_1}| > c$ for $i_1 \in \{1, 2\}$, $|G_{i_2}| \leqslant c$ for $i_2 \in \{4, 5\}$, **then**

$$|G_3| > |G_6|$$

## Heuristic conclusion:

**Compared with two DLAs with a low absolute correlation, two ones with a high absolute correlation would be more likely to generate another relatively good DLA.**

# DLA Semi-Unconstrained Searching



## Iterative Search:

$$\Delta_{\mathrm{m}} \xrightarrow{E_{\mathrm{m}}} \gamma_{\mathrm{m}}$$

**1. Initialization phase: preset a difference $\Delta_{\mathrm{m}}$ and threshold $c$, select $t$ DLAs $\Delta_{\mathrm{m}} \rightarrow \gamma_i$**

$$\left|\mathrm{Cor}\left(\Delta_{\mathrm{m}} \xrightarrow{E_{\mathrm{m}}} \gamma_i\right)\right| > c \qquad \mathcal{P} = \{\gamma_1, \cdots, \gamma_t\}$$

**2. Iterative phase: generate DLAs with a high correlation, repeat several iterations**

$$\gamma_i \oplus \gamma_j \notin \mathcal{P} \text{ where } \gamma_i, \gamma_j \in \mathcal{P}$$

$$\left|\mathrm{Cor}\left(\Delta_{\mathrm{m}} \xrightarrow{E_{\mathrm{m}}} \gamma_i \oplus \gamma_j\right)\right| > c \longrightarrow \mathcal{P} \leftarrow \mathcal{P} + \mathcal{Q} \longrightarrow \text{next iteration}$$

$$\mathcal{Q} \leftarrow \mathcal{Q} + \{\gamma_i \oplus \gamma_j\}$$

# DLA Semi-Unconstrained Searching



**Iterative Search:**

$$\Delta_{\mathrm{m}} \xrightarrow{E_{\mathrm{m}}} \gamma_{\mathrm{m}}$$

**1. Initialization phase: preset a difference $\Delta_{\mathrm{m}}$ and threshold $c$, select $t$ DLAs $\Delta_{\mathrm{m}} \rightarrow \gamma_i$**

$$\left|\mathrm{Cor}\left(\Delta_{\mathrm{m}} \xrightarrow{E_{\mathrm{m}}} \gamma_i\right)\right| > c \qquad \mathcal{P} = \{\gamma_1, \cdots, \gamma_t\}$$

**Strong (Weak) Unbalanced bit:**

$$\left|\mathrm{Cor}\left(\Delta_{\mathrm{m}} \xrightarrow{E_{\mathrm{m}}} [i]\right)\right| \geqslant c$$

$$\Downarrow$$

**Bit $i$ is a strong unbalanced bit (SUB)**

$$\left|\mathrm{Cor}\left(\Delta_{\mathrm{m}} \xrightarrow{E_{\mathrm{m}}} [i]\right)\right| < c$$

$$\Downarrow$$

**Bit $i$ is a weak unbalanced bit (WUB)**

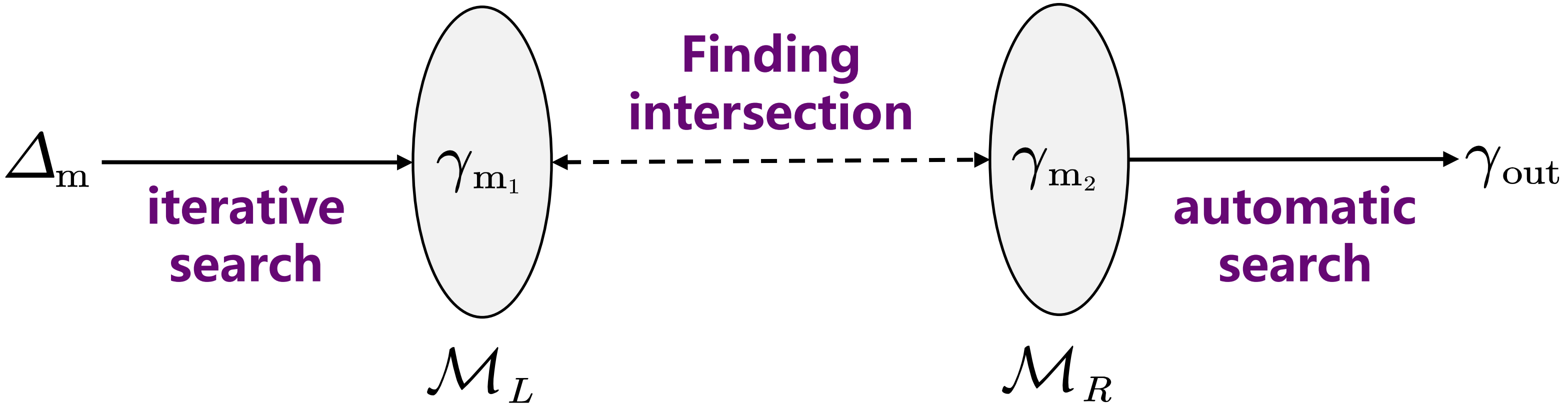
# DLA Semi-Unconstrained Searching



## Meet-in-the-Middle Search:

$$\Delta_{\mathrm{m}} \xrightarrow{E_{\mathrm{m}}} \gamma_{\mathrm{m}} \xrightarrow{E_2} \gamma_{\mathrm{out}}$$

## High-level idea




## Implementation

1. Search $\{\gamma_{\mathrm{m}} \xrightarrow{E_2} \gamma_{\mathrm{out}} | \gamma_{\mathrm{m}}[i] = 0 \text{ for } i \notin \mathcal{B}_S\}$

/* $\mathcal{B}_S$: the set of strong unbalanced bits*/

2. Verify the correlation of $\Delta_{\mathrm{m}} \xrightarrow{E_{\mathrm{m}}} \gamma_{\mathrm{m}}$

# DLA Semi-Unconstrained Searching



## Support experiment for the heuristic conclusion:

$$\mathcal{X}_1 = \{\Delta_{\mathrm{m}} \xrightarrow{E_{\mathrm{m}}} \gamma_{\mathrm{m}} \mid 0 < HW(\gamma_{\mathrm{m}}) \leqslant d\},$$
$$\mathcal{X}_2 = \{\Delta_{\mathrm{m}} \xrightarrow{E_{\mathrm{m}}} \gamma_{\mathrm{m}} \mid 0 < HW(\gamma_{\mathrm{m}}) \leqslant d; \gamma_{\mathrm{m}}[i] = 0 \text{ for } i \notin \mathcal{B}_{\mathcal{S}}\}\}$$

/* $HW(\gamma_{\mathrm{m}})$:Hamming weight of $\gamma_{\mathrm{m}}$; $\mathcal{B}_{\mathcal{S}}$:strong unbalanced bit set*/

$\mathcal{G} \subset \mathcal{X}_1$:the set of DLAs with an absolute correlation higher than a threshold $c$

$$\frac{|\mathcal{G}|}{|\mathcal{X}_1|} \text{ vs } \frac{|\mathcal{G} \cap \mathcal{X}_2|}{|\mathcal{X}_2|}$$

# DLA Semi-Unconstrained Searching





## Results of the support experiment :

**Table 3.** Comparison of differential-linear approximations in two spaces.

| $E_m$ | n | $\Delta_m$ | c | $\lvert\mathcal{B}_S\rvert$ | d | $\lvert\mathcal{X}_1\rvert$ | $\lvert\mathcal{X}_2\rvert$ | $\lvert\mathcal{G}\rvert$ | $\lvert\mathcal{G}\cap\mathcal{X}_2\rvert$ | $\frac{\lvert\mathcal{G}\rvert}{\lvert\mathcal{X}_1\rvert}$ | $\frac{\lvert\mathcal{G}\cap\mathcal{X}_2\rvert}{\lvert\mathcal{X}_2\rvert}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 8-round LEA | 128 | [31] | $2^{-4}$ | 14 | 2 | 8256 | 105 | 72 | 43 | **0.0087** | **0.4095** |
| 5-round Speck32 | 32 | [22] | $2^{-4}$ | 10 | 4 | 41448 | 385 | 785 | 311 | **0.0189** | **0.8078** |
| | | | | | 3 | 5488 | 175 | 250 | 146 | **0.0456** | **0.8343** |
| 5-round PRESENT | 64 | [56] | $2^{-4}$ | 16 | 2 | 2080 | 136 | 46 | 31 | **0.0221** | **0.2279** |
| 4-round DES | 64 | [6] | $2^{-4}$ | 11 | 2 | 2080 | 66 | 31 | 22 | **0.0149** | **0.3333** |

| | 72 | 785 | 250 | 46 | 31 |
|---|---|---|---|---|---|
| Type 0 (only contains **weak** unbalanced bits) | 15 | 27 | 10 | 8 | 3 |
| Type 1 (only contains **strong** unbalanced bits) | 43 | 311 | 146 | 31 | 22 |
| Type 2 (contains WUBs and SUBs) | 14 | 447 | 94 | 7 | 6 |

# Partition Tree



**Partition-based Differential-Linear Attack [5,4]:**

**Build partitions for the function $F$ containing no keys:**

1. Partition conditions: $b_i$ for $i \in \{1, \cdots, s\}$

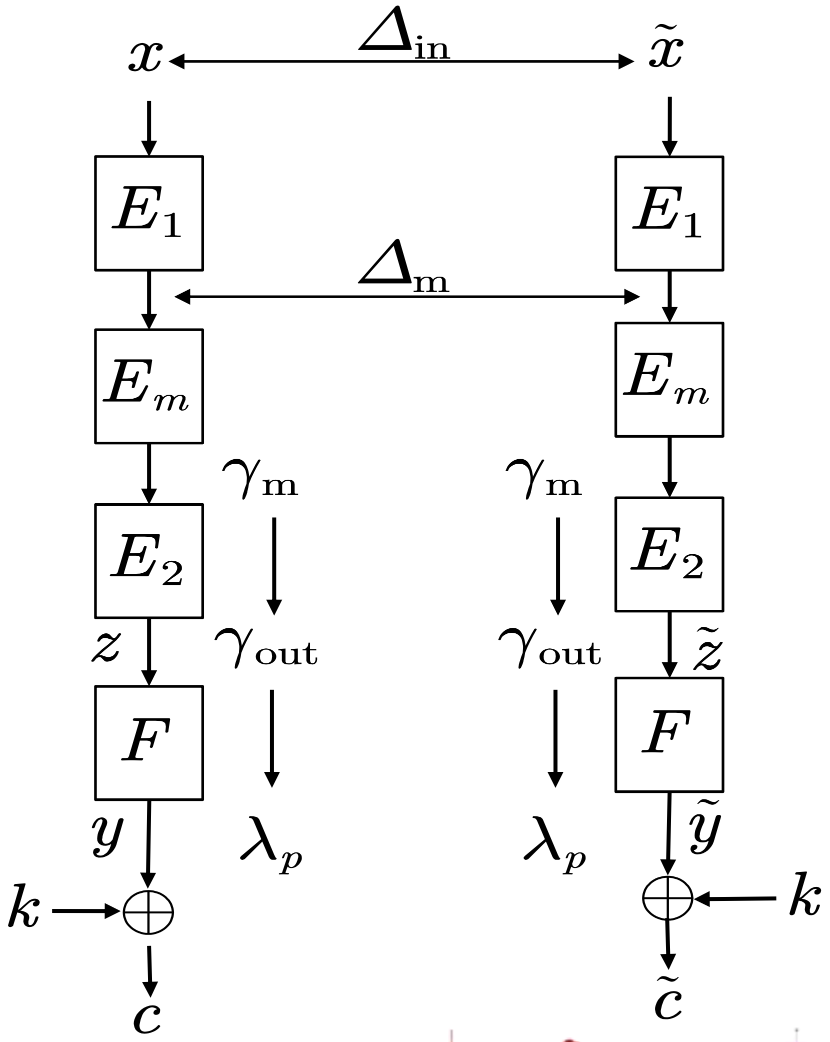
2. Conditional linear approximation:

$$\gamma_{\text{out}} \xrightarrow{F} \gamma_p \text{ for } p = b_1 \| \cdots \| b_s \in \{0, \cdots 2^s - 1\}$$

/*$\lambda_p$: linear mask in current partition*/

**Extra requirements:**

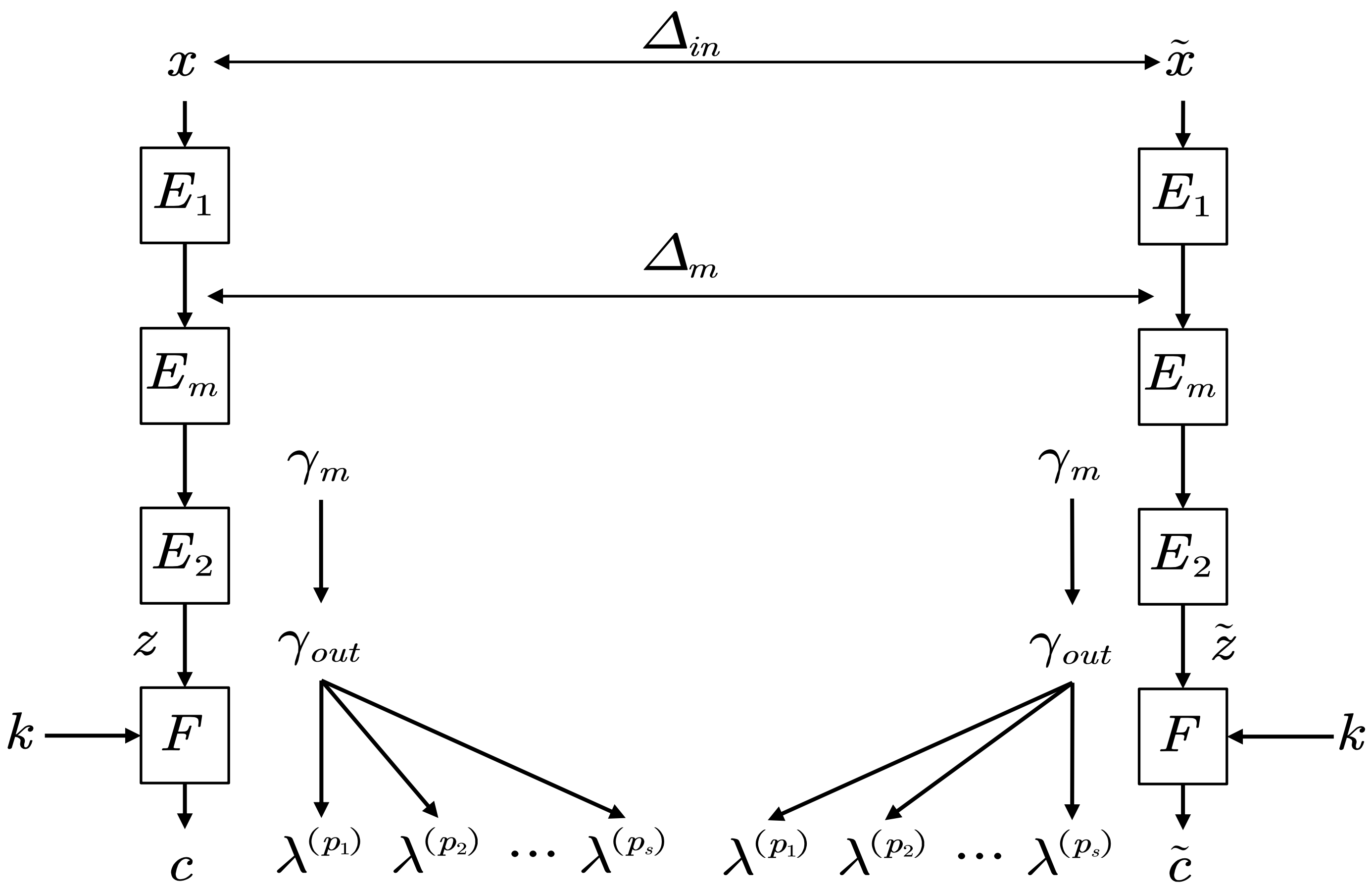
**The correlation in each partition is not zero[4].**



# Partition Tree



**Our extension for the attack proposed in [5, 4]:**




**The same task:**

$b_i$ for $i \in \{1, \cdots, s\}$

/ * partition conditions * /

$\lambda_p$ where $p = b_1 \| \cdots \| b_s$

/ * linear mask of $c \| k$ * /

# Partition Tree



## Partition Tree :

**A tree that describes the partition conditions and approximations simultaneously.**

## Leaf Node :

**Its value is known**

## Non-leaf Node :

**Its value is unknown**

## Partition Edge :

$$A \dashrightarrow B$$

***B*** **is a partition condition**

## Approximation Edge :

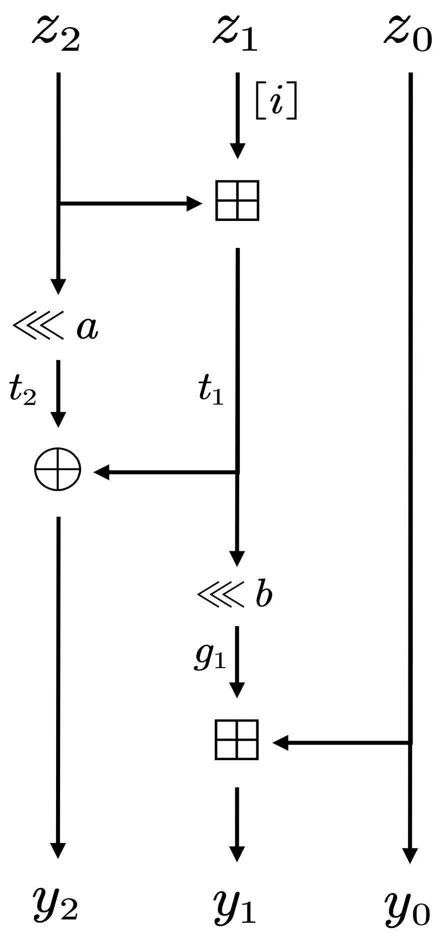
$$A \longrightarrow B \quad A = B \quad A = B$$

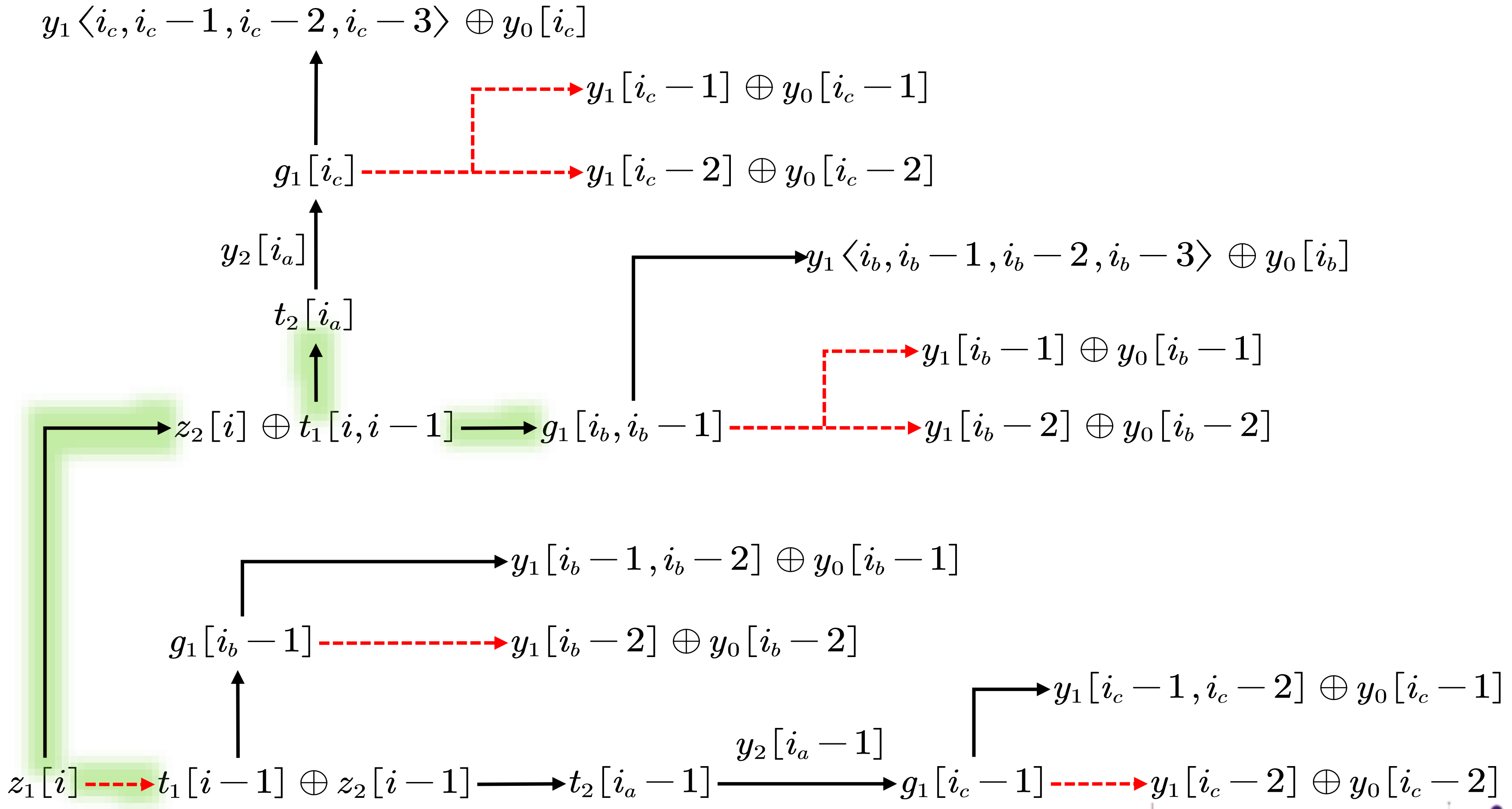
$$A \xrightarrow{X} B \quad \Longleftrightarrow \quad A = X \oplus B \text{ where } X \text{ is known.}$$

# Partition Tree



**Building process:** **Simulate the propagation of linear approximation using the tree.**

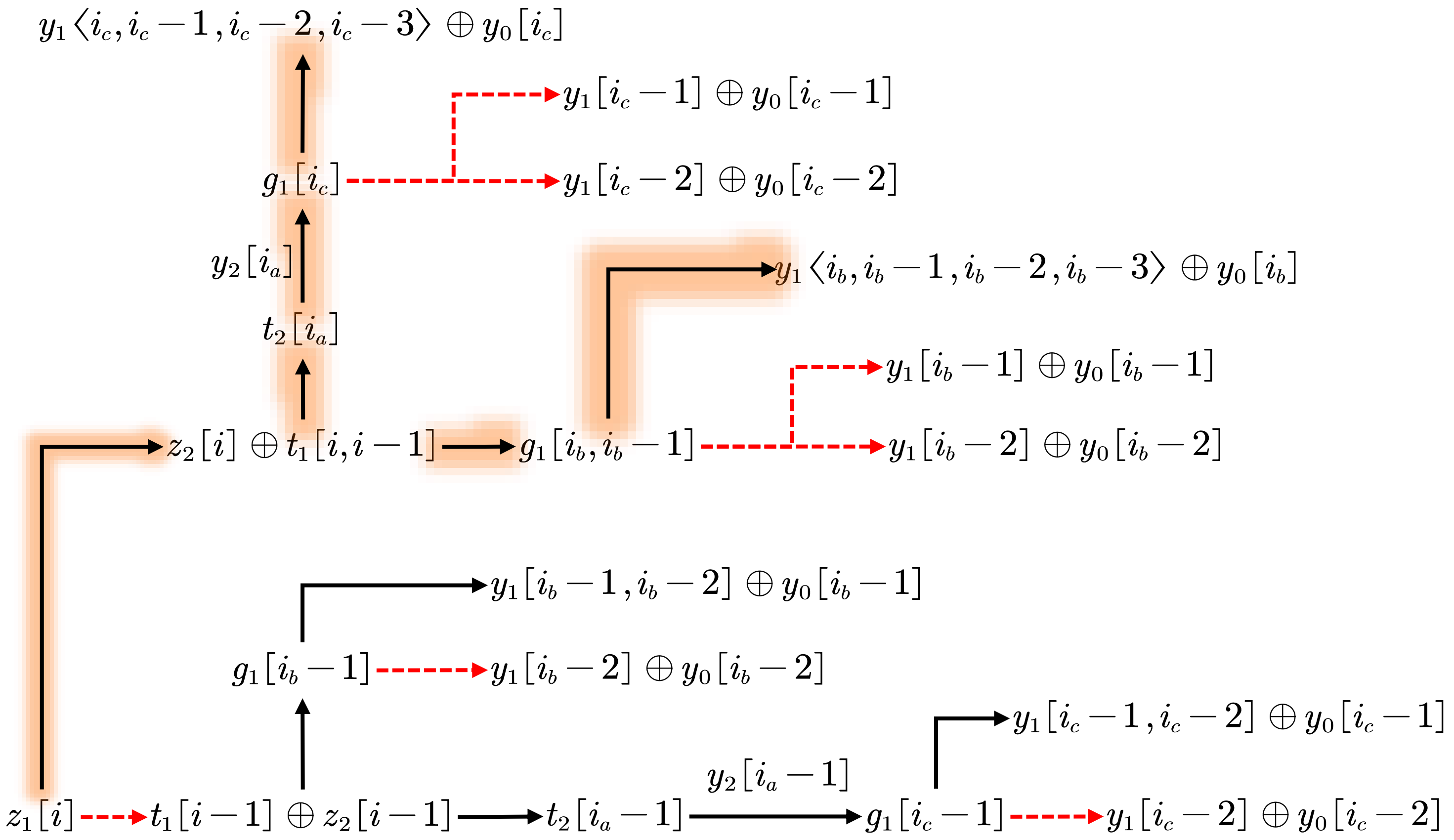


# Partition Tree



## Usage :

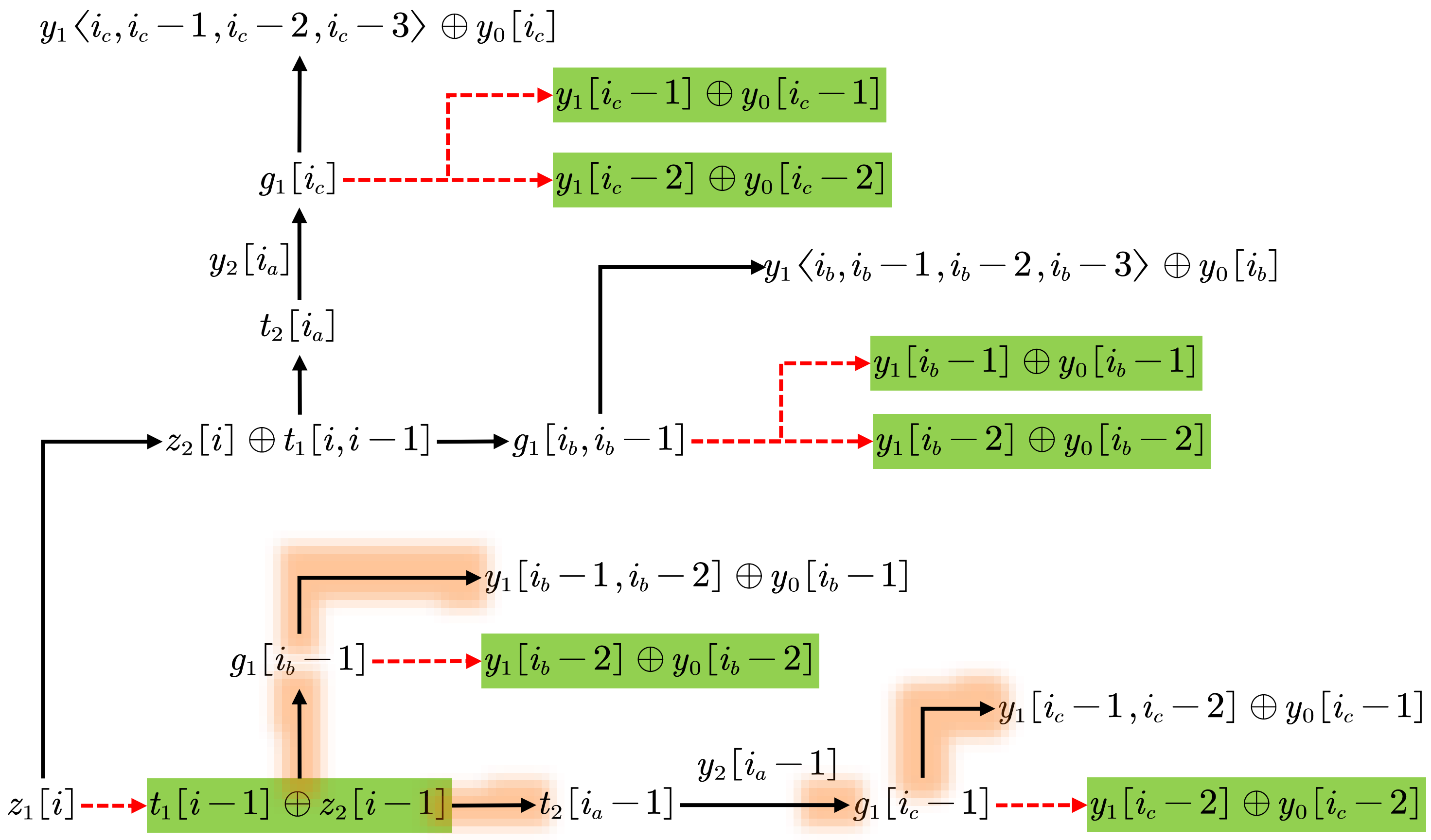



$$z_1[i] \approx \langle \gamma_p, y_2[i_a] || y_0[i_b] || y_1[i_b] || y_1[i_b-1] || y_1[i_b-2] || y_1[i_b-3] || y_0[i_c] || y_1[i_c] || y_1[i_c-1] || y_1[i_c-2] || y_1[i_c-3] \rangle$$

# Partition Tree



## Usage :




$$b_1 = y_1[i_a-1] \oplus y_1[i_b-2] \oplus y_1[i_c-2]$$

$$b_2 = y_1[i_b-1] \oplus y_0[i_b-1]$$

$$b_3 = y_1[i_b-2] \oplus y_0[i_b-2]$$

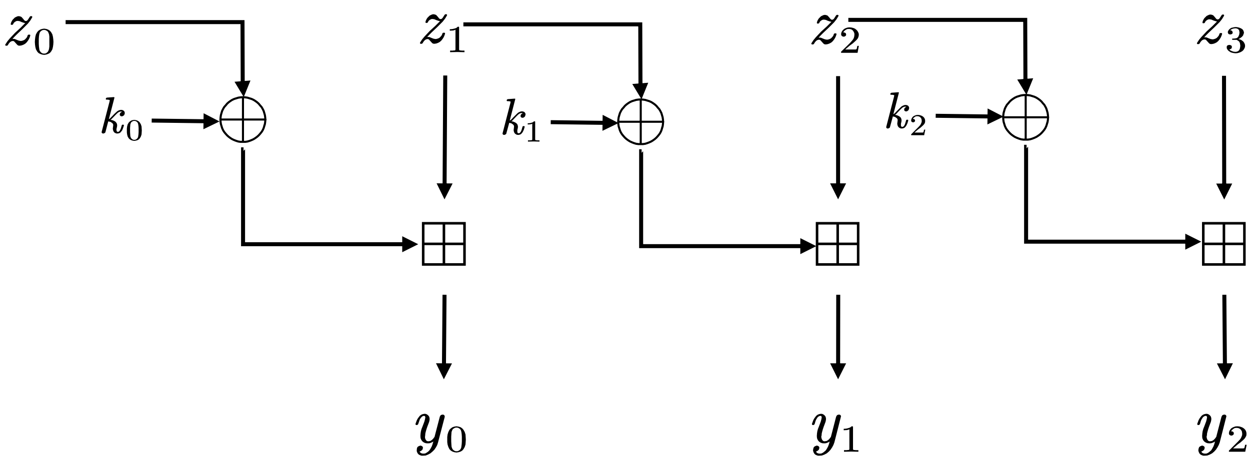
$$b_4 = y_1[i_c-1] \oplus y_0[i_c-1]$$

$$b_5 = y_1[i_c-2] \oplus y_0[i_c-2]$$

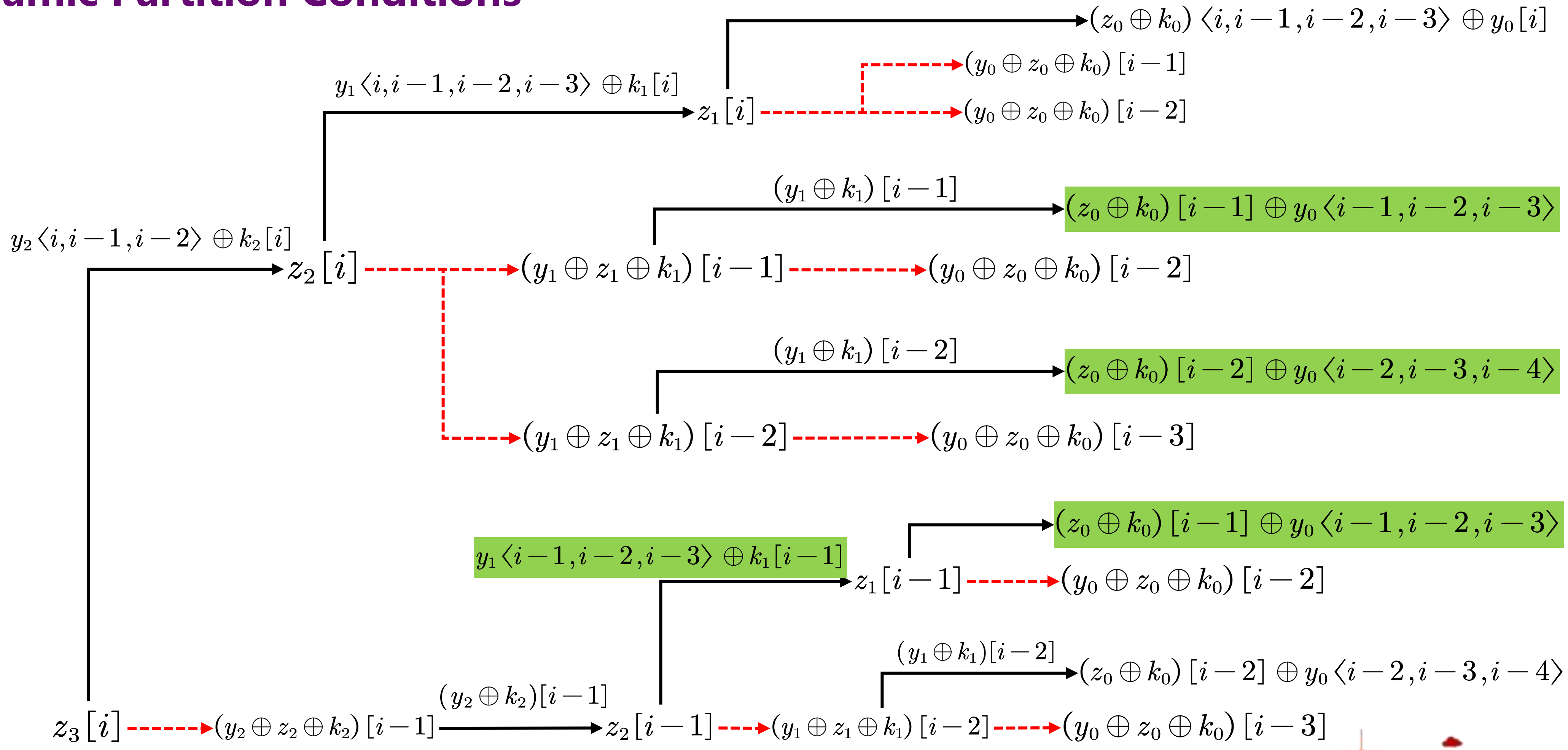
$$p = b_1||b_2||b_3||b_4||b_5$$

## Dynamic Partitioning Technique:

**When the encryption function $F$ is rather complex, in order to make the correlation in each partition be non-zero, we need to dynamically choose partition conditions for each data.**



## Dynamic Partition Conditions



# Partition Tree



## Dynamic Partition Conditions

$$
\begin{aligned}
z_3[i] = < \gamma_p,\ & y_2[i]||y_2[i-1]||y_2[i-2]||y_1[i]||y_1[i-1]||y_1[i-2]||y_1[i-3]|| \\
& y_0[i]||z_0[i]||z_0[i-1]||z_0[i-2]||z_0[i-3]||k_2[i]||k_1[i]||k_0[i] \\
& k_0[i-1]||k_0[i-2]||k_0[i-3] >
\end{aligned}
$$

$$
b_1 = (y_0 \oplus z_0 \oplus k_0)[i-1];\ b_2 = (y_0 \oplus z_0 \oplus k_0)[i-2];\ b_3 = (y_0 \oplus z_0 \oplus k_0)[i-3]
$$

$$
\boxed{
\begin{aligned}
b_4 &= \begin{cases} (y_1 \oplus k_1)[i-1] \oplus y_0[i-2], \text{ if } b_2 = 1 \\ (y_1 \oplus k_1)[i-1] \oplus y_0[i-3], \text{ if } b_2 = 0 \end{cases} \\
b_5 &= \begin{cases} (y_1 \oplus k_1)[i-2] \oplus y_0[i-3], \text{ if } b_3 = 1 \\ (y_1 \oplus k_1)[i-2] \oplus y_0[i-4], \text{ if } b_3 = 0 \end{cases} \\
b_6 &= \begin{cases} (y_2 \oplus k_2)[i-1] \oplus y_1[i-2], \text{ if } b_2 \oplus b_5 = 0 \\ (y_2 \oplus k_2)[i-1] \oplus y_1[i-3], \text{ if } b_2 \oplus b_5 = 1 \end{cases}
\end{aligned}
}
$$

# More Thoughts

- **How to further improve the DLA searching?**

   1. **Increasing the search space**

   2. **Remove the limitation on the intermediate difference $\Delta_m$**

   3. **Improve the MIQCP/MILP-based fully automated DLA searching**

- **Apply the partition tree to SPN ciphers**

   1. **Build conditional linear approximations of S-box**

Asiacrypt 2023

清華大學 Tsinghua University

# Thank you for watching

Feel free to contact us via

chenyi2023@mail.tsinghua.edu.cn ,
zzbao@mail.tsinghua.edu.cn ,
yuhongbo@mail.tsinghua.edu.cn ,

if you have any questions or ideas to discuss.