# Improved Search for Integral, Impossible Differential and Zero-Correlation Attacks

**Hosein Hadipour** Simon Gerhalter Sadegh Sadeghi Maria Eichlseder

FSE 2024 - Leuven, Belgium

› hsn.hadipour@gmail.com

SCIENCE
PASSION
TECHNOLOGY

# Motivation and Our Contributions

## Motivation

- Providing a tool to find **complete** integral, and ID/ZC attacks

## Contributions

- Improving the CP-based methods to find ID/ZC, and integral distinguishers
- Introducing a CP model for the partial-sum technique for the first time
- Improving distinguishers of Ascon, QARMAv2, and ForkSKINNY (**25** Dists.)
- Improving key recovery attacks of SKINNY, and ForkSKINNY (**24** Attacks)

# Motivation and Our Contributions

## Motivation

- Providing a tool to find **complete** integral, and ID/ZC attacks

## Contributions

- Improving the CP-based methods to find ID/ZC, and integral distinguishers
- Introducing a CP model for the partial-sum technique for the first time
- Improving distinguishers of Ascon, QARMAv2, and ForkSKINNY (**25** Dists.)
- Improving key recovery attacks of SKINNY, and ForkSKINNY (**24** Attacks)

# Part of Our Results Regarding Distinguishing Attacks

| Cipher | #Rounds | Dist. | Data complexity | Ref. |
|---|---|---|---|---|
| QARMAv2-64 | 5 | Integral | - | [Ava+23] |
| QARMAv2-64 ($\mathscr{T} = 1$) | **7 / 8 / 9** | Integral | $\mathbf{2^{8}}$ / $\mathbf{2^{16}}$ / $\mathbf{2^{44}}$ | This work |
| QARMAv2-64 ($\mathscr{T} = 2$) | **8 / 9 / 10** | Integral | $\mathbf{2^{8}}$ / $\mathbf{2^{16}}$ / $\mathbf{2^{44}}$ | This work |
| QARMAv2-128($\mathscr{T} = 2$) | **10 / 11 / 12** | Integral | $\mathbf{2^{16}}$ / $\mathbf{2^{44}}$ / $2^{96}$ | This work |
| ForkSKINNY-64-192 | 16 | Integral | $2^{72}$ | [Niu+21] |
| ForkSKINNY-64-192 | **17** | Integral | $\mathbf{2^{60}}$ | This work |
| ForkSKINNY-64-192 | 16 | ID | - | [HSE23] |
| ForkSKINNY-64-192 | **21** | ID | - | This work |
| ForkSKINNY-128-256 | 14 | Integral | $2^{56}$ | [HSE23] |
| ForkSKINNY-128-256 | **15** | Integral | $\mathbf{2^{56}}$ | This work |

# Part of Our Results Regarding Key Recovery Attacks

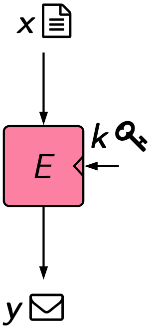
| Cipher | #R | Time | Data | Mem. | Attack | Setting / Model | Ref. |
|---|---|---|---|---|---|---|---|
| SKINNY-64-64 | 17 | $2^{59}$ | $2^{58.79}$ | $2^{40}$ | ID | STK / CP | [HSE23] |
| | **18** | $\mathbf{2^{53.58}}$ | $2^{53.58}$ | $2^{48}$ | Int | 60,SK / CP,CT | This work |
| SKINNY-128-128 | 17 | $2^{116.51}$ | $2^{116.37}$ | $2^{80}$ | ID | STK / CP | [HSE23] |
| | **18** | $\mathbf{2^{105.58}}$ | $2^{105.58}$ | $2^{96}$ | Int | 120,SK / CP,CT | This work |
| SKINNY-128-384 | 26 | $2^{344}$ | $2^{121}$ | $2^{340}$ | Int | 360,SK / CP,CT | [HSE23] |
| | 26 | $\mathbf{2^{331}}$ | $2^{122}$ | $2^{328}$ | Int | 360,SK / CP,CT | This work |
| ForkSKINNY-128-256 | 26 | $2^{250.30}$ | $2^{127}$ | $2^{160}$ | ID | 256,RTK / CP | [BDL20] |
| | 26 | $\mathbf{2^{238.50}}$ | $2^{128.60}$ | $2^{175.60}$ | ID | 256,RTK / CP | This paper |

# Outline

# Background and the Research Gap

# Integral, ID, and ZC Distinguishers

- Integral attack [Lai94; DKR97]
- Impossible-differential attack [BBS99; Knu98]
- Zero-correlation attack [BR14]

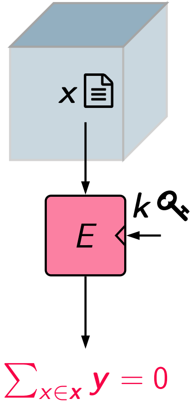




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Integral, ID, and ZC Distinguishers

- Integral attack [Lai94; DKR97]
- Impossible-differential attack [BBS99; Knu98]
- Zero-correlation attack [BR14]

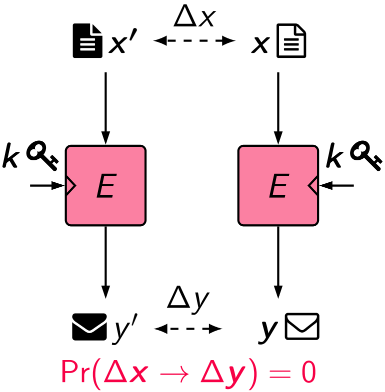




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

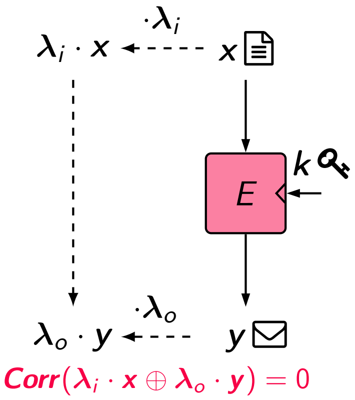
# Integral, ID, and ZC Distinguishers

- Integral attack [Lai94; DKR97]
- Impossible-differential attack [BBS99; Knu98]
- Zero-correlation attack [BR14]



# Integral, ID, and ZC Distinguishers

- Integral attack [Lai94; DKR97]
- Impossible-differential attack [BBS99; Knu98]
- Zero-correlation attack [BR14]

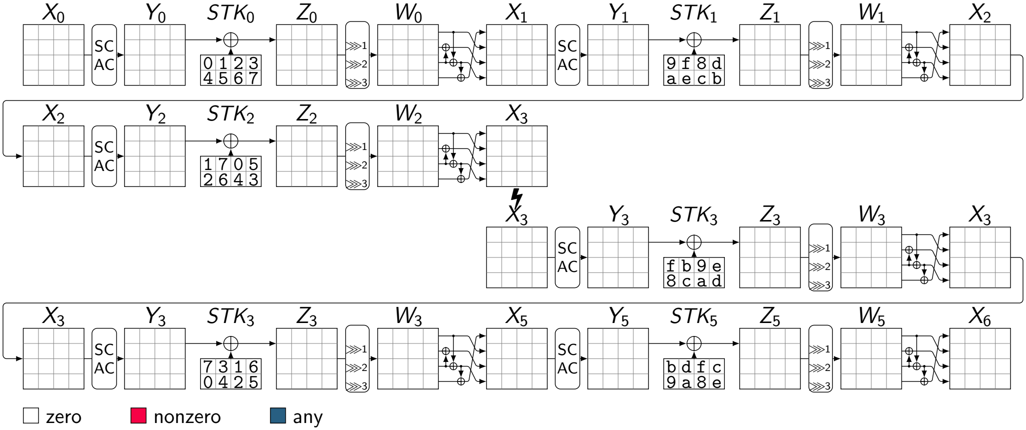




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

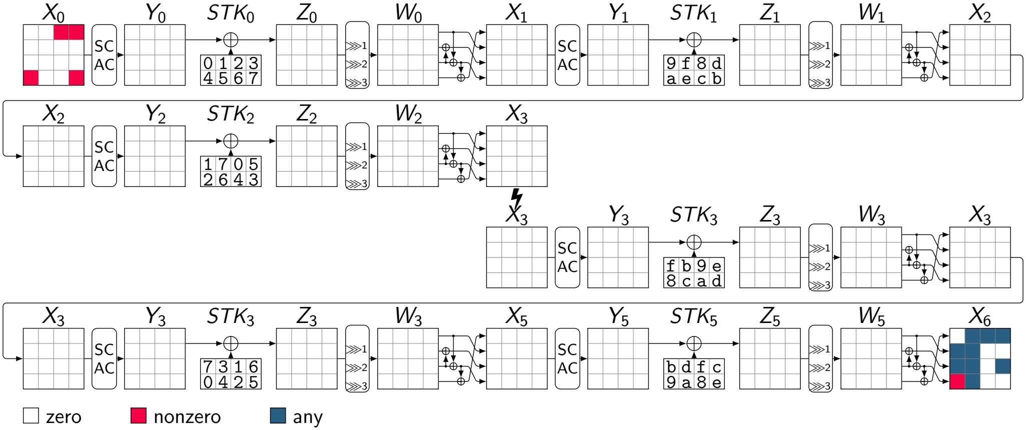
# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle



# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle

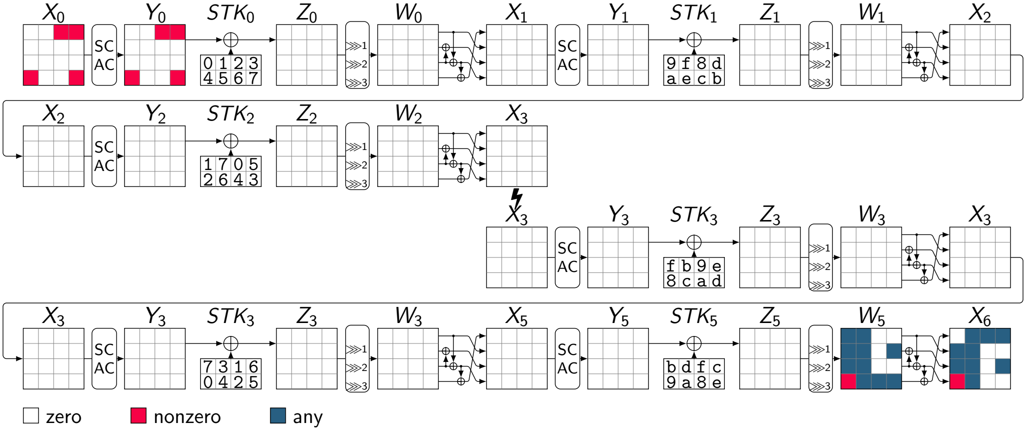




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

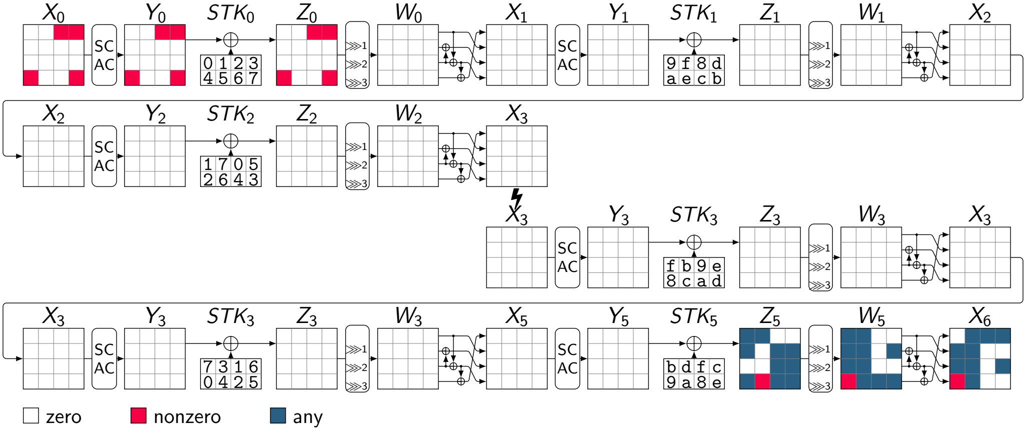
# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle



# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle

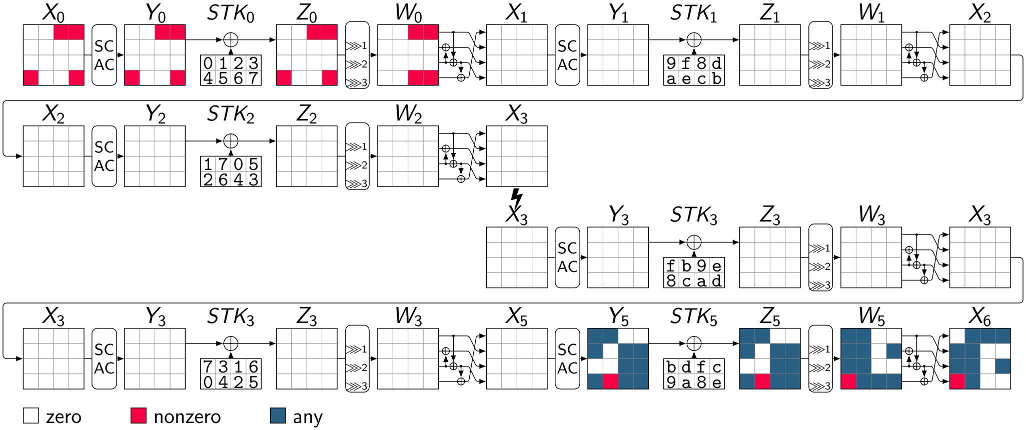




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle

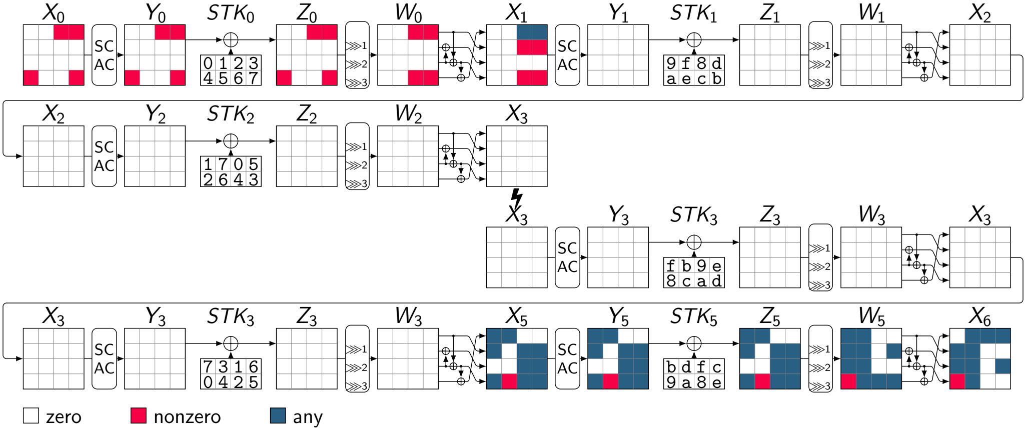




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle

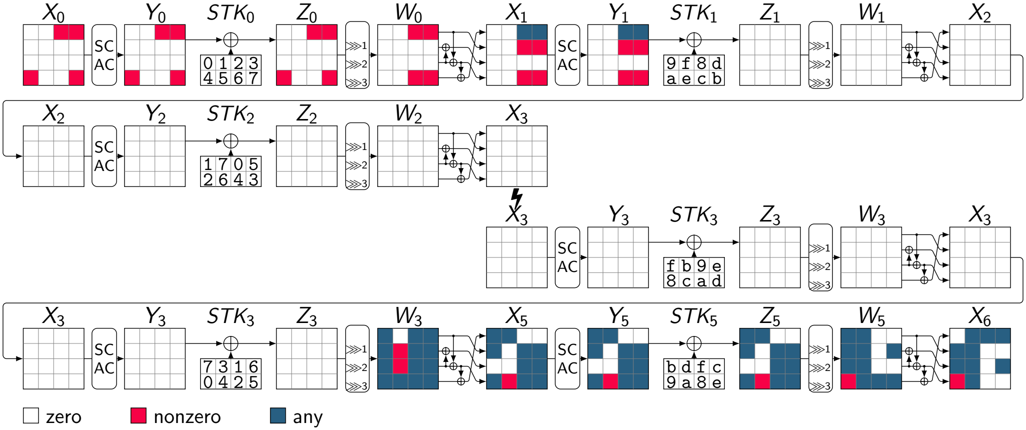




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

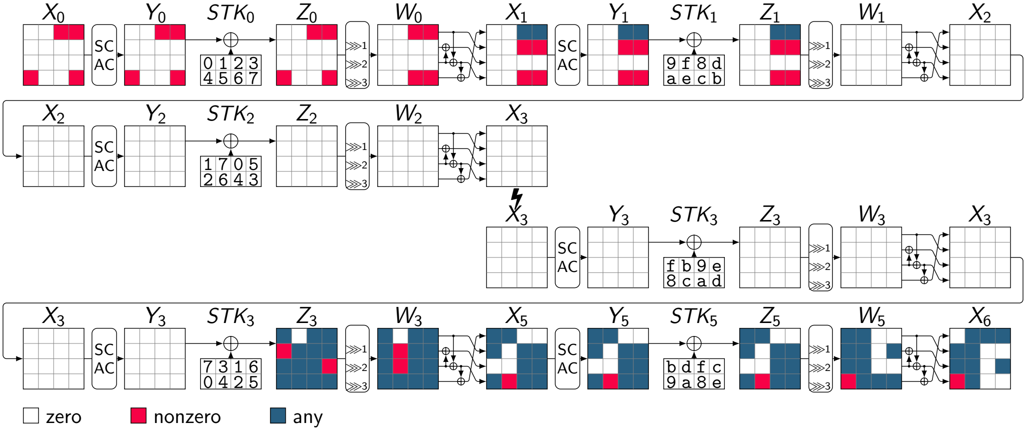
# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle



# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle

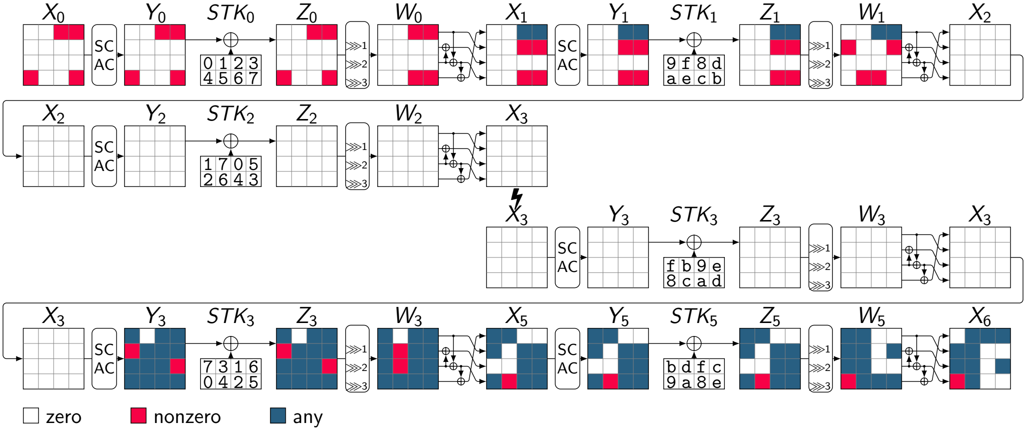




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle

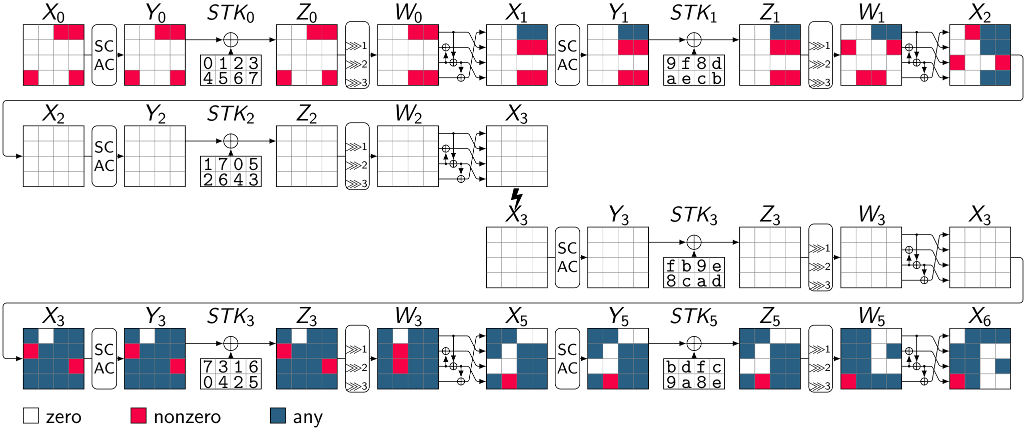




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder
FSE 2024 - Leuven, Belgium

# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle

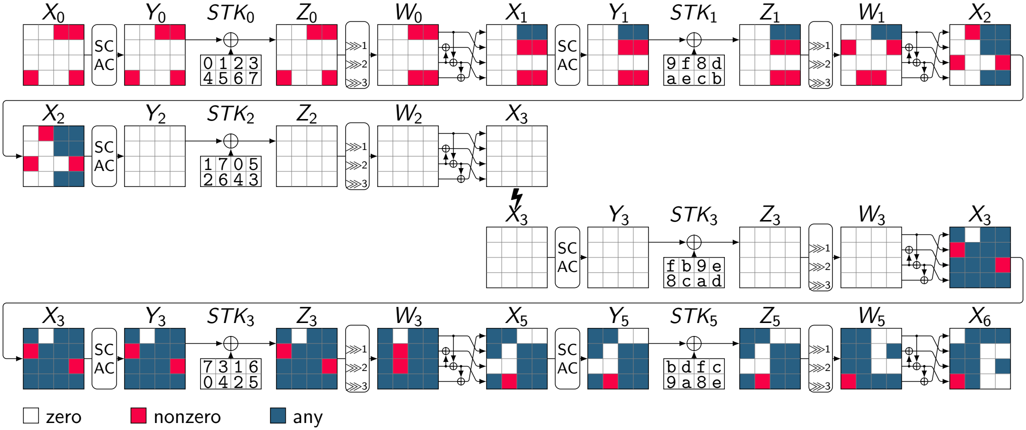




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder
FSE 2024 - Leuven, Belgium

# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle

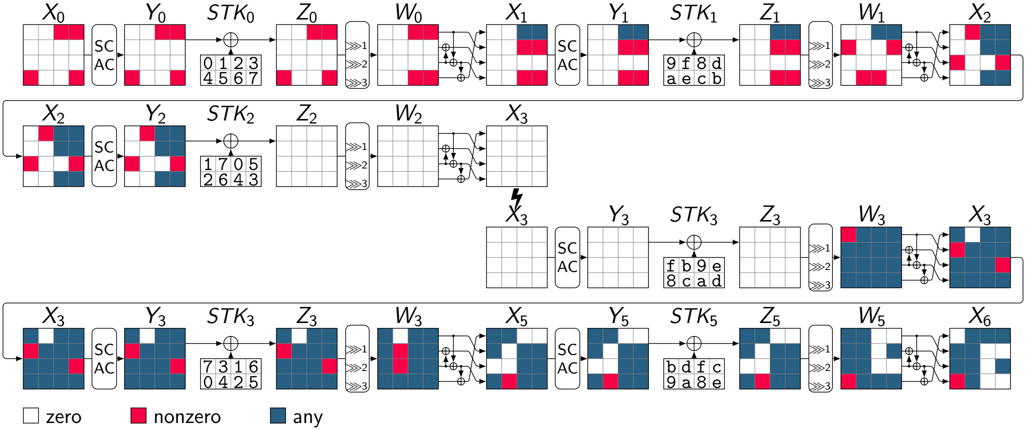




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle

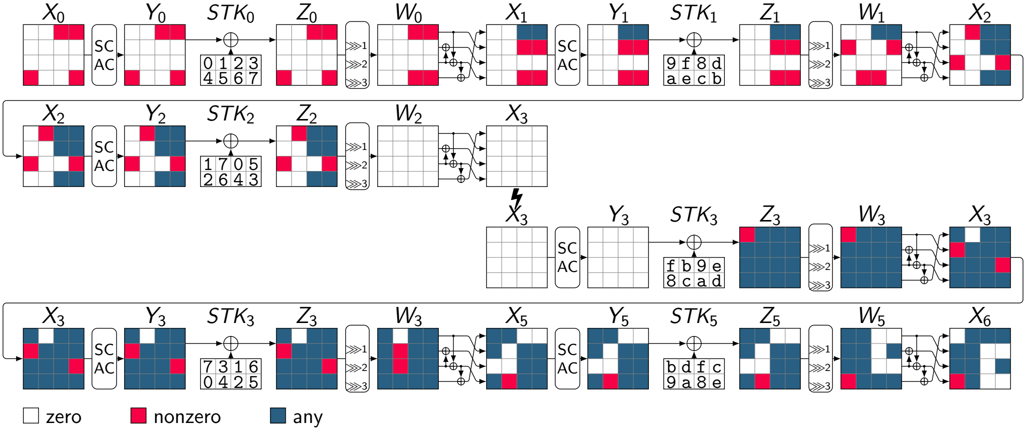




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder
FSE 2024 - Leuven, Belgium

# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle

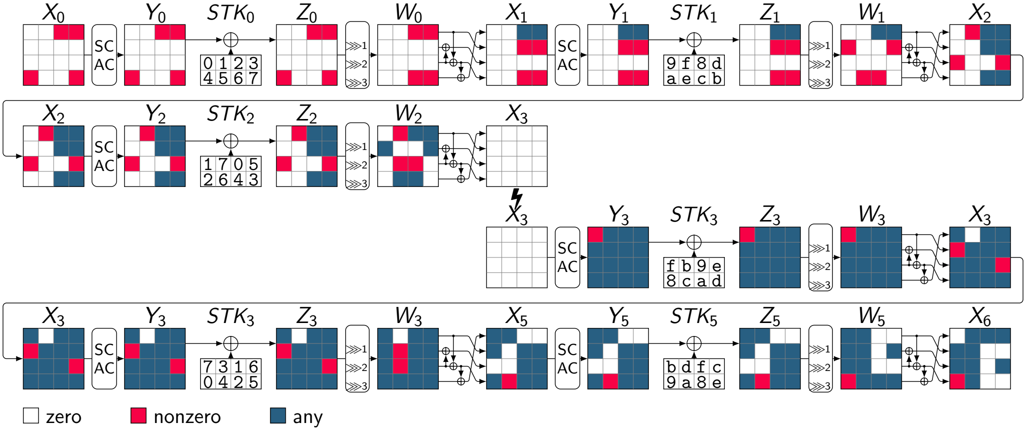




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

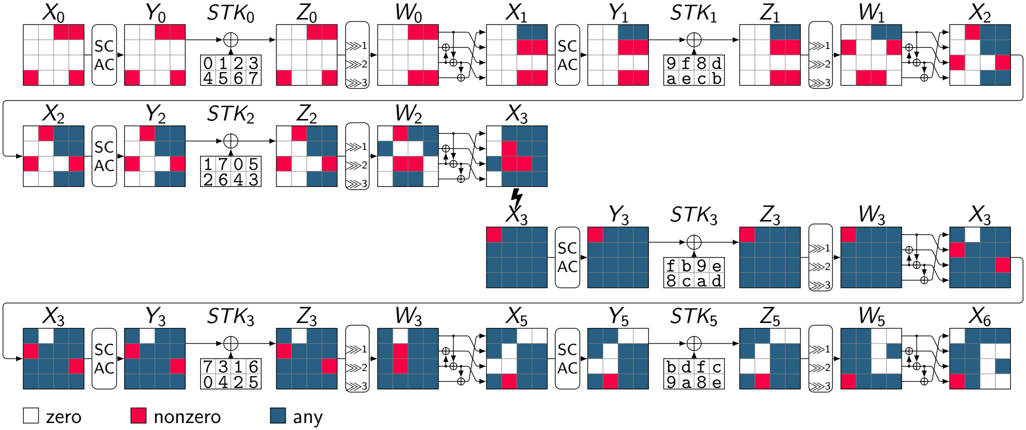
# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle



# Miss-in-the-Middle Technique [BBS99]

- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle





**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Relation Between ZC and Integral Distinguishers

- Any ZC distinguisher can be converted to an integral distinguisher [Sun+15].

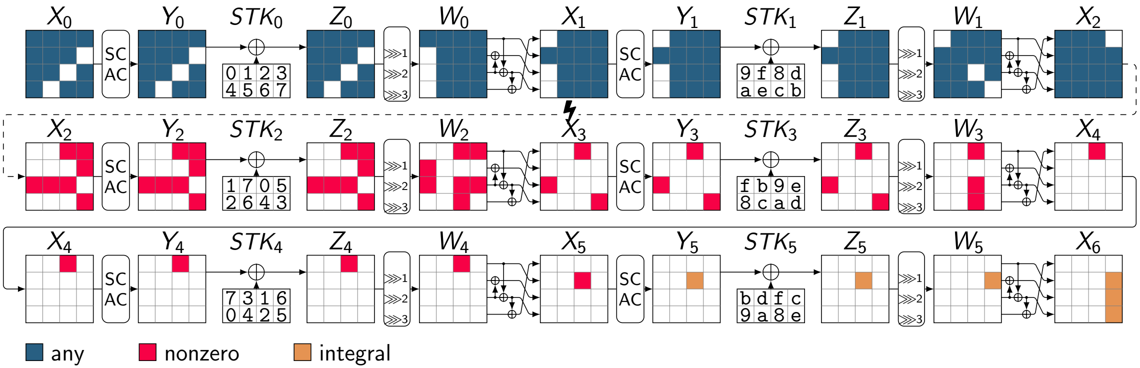
**Link Between ZC and Integral Distinguishers [Sun+15]**

Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a vectorial Boolean function. Assume $A$ is a subspace of $\mathbb{F}_2^n$ and $\beta \in \mathbb{F}_2^n \setminus \{0\}$ such that $(\alpha, \beta)$ is a ZC approximation for any $\alpha \in A$. Then, for any $\lambda \in \mathbb{F}_2^n$, $\langle \beta, F(x + \lambda) \rangle$ is balanced over the set

$$A^\perp = \{x \in \mathbb{F}_2^n \mid \forall\, \alpha \in A : \langle \alpha, x \rangle = 0\}.$$


**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder
FSE 2024 - Leuven, Belgium

# Example: Conversion of ZC Distinguisher to Integral Distinguisher




- $X_0[7, 10, 13]$ takes all possible values and the remaining cells take a fixed value
- $X_6[7] \oplus X_6[11] \oplus X_6[15]$ is balanced

# ID, ZC, and Integral Key Recovery

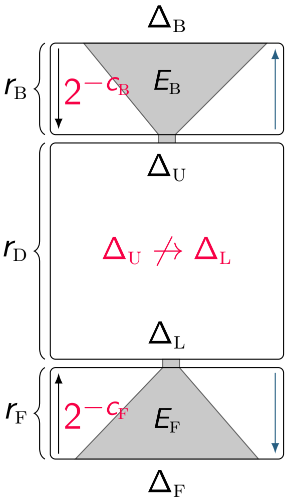
- Common technique for ID key recovery:
  - Early abort technique [Lu+08]
- Common technique for ZC/Integral key recovery:
  - Partial-sum technique [Fer+00]

$$r_{\mathrm{D}}\left\{\begin{array}{|c|}\hline \Delta_{\mathrm{U}} \\ \\ \Delta_{\mathrm{U}} \not\rightarrow \Delta_{\mathrm{L}} \\ \\ \Delta_{\mathrm{L}} \\ \hline\end{array}\right.$$


**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

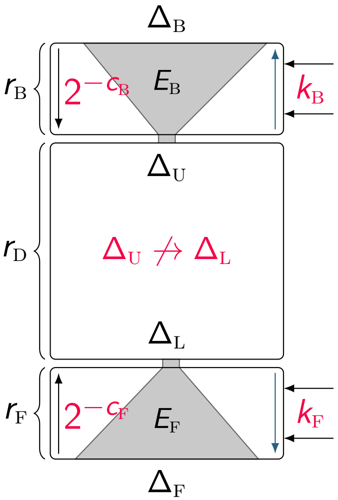
# ID, ZC, and Integral Key Recovery

- Common technique for ID key recovery:
  - Early abort technique [Lu+08]
- Common technique for ZC/Integral key recovery:
  - Partial-sum technique [Fer+00]



# ID, ZC, and Integral Key Recovery

- Common technique for ID key recovery:
  - Early abort technique [Lu+08]
- Common technique for ZC/Integral key recovery:
  - Partial-sum technique [Fer+00]

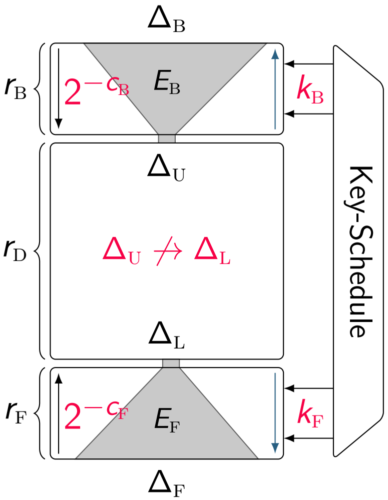




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# ID, ZC, and Integral Key Recovery

- Common technique for ID key recovery:
  - Early abort technique [Lu+08]
- Common technique for ZC/Integral key recovery:
  - Partial-sum technique [Fer+00]

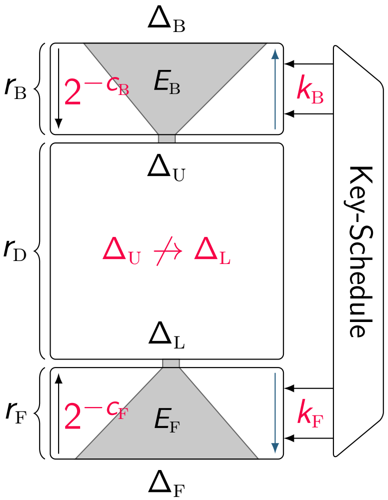




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

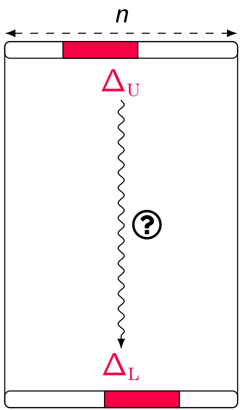
# ID, ZC, and Integral Key Recovery

- Common technique for ID key recovery:
  - Early abort technique [Lu+08]
- Common technique for ZC/Integral key recovery:
  - Partial-sum technique [Fer+00]



# Previous Tools for ID/ZC, and Integral Attacks

- Tools based on dedicated algorithms:
  - CRYPTO 2016 ($\mathcal{DC}$-MITM, ID) [DF16]
- Tools based on general purpose solvers:
  - Eprint 2016 (ID) [Cui+16]
  - ASIACRYPT 2016 (Integral) [Xia+16]
  - EUROCRYPT 2017 (ID, ZC) [ST17]
  - ToSC 2017 (ID, ZC) [Sun+17]
  - ToSC 2020 (ID, ZC) [Sun+20]



# Search for Distinguishers

# Our Previous Method to Search Distinguishers [HSE23]

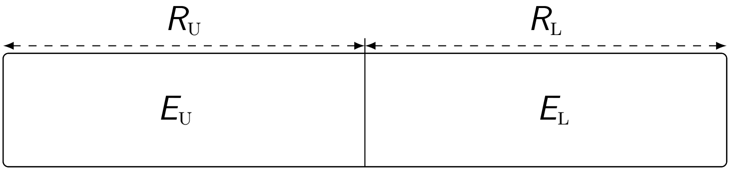
$E$

$CSP_{\mathrm{U}}(\Delta_{\mathrm{U}}, \Delta'_{\mathrm{U}})$

$CSP_{\mathrm{L}}(\Delta_{\mathrm{L}}, \Delta'_{\mathrm{L}})$

$CSP_{\mathrm{M}}(\Delta'_{\mathrm{U}}, \Delta'_{\mathrm{L}})$

# Our Previous Method to Search Distinguishers [HSE23]

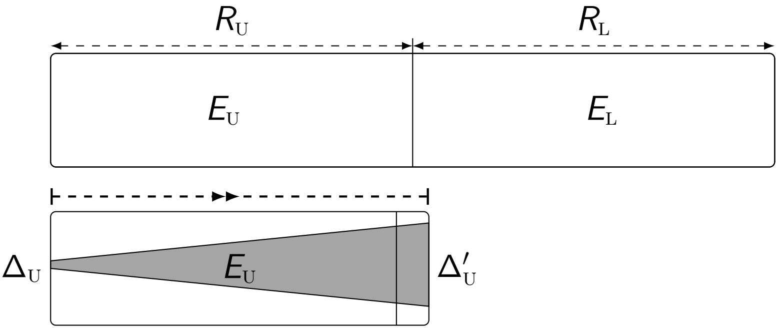



- $CSP_{\mathrm{U}}(\Delta_{\mathrm{U}}, \Delta'_{\mathrm{U}})$
- $CSP_{\mathrm{L}}(\Delta_{\mathrm{L}}, \Delta'_{\mathrm{L}})$
- $CSP_{\mathrm{M}}(\Delta'_{\mathrm{U}}, \Delta'_{\mathrm{L}})$


**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Our Previous Method to Search Distinguishers [HSE23]

- $CSP_{\mathrm{U}}(\Delta_{\mathrm{U}}, \Delta'_{\mathrm{U}})$
- $CSP_{\mathrm{L}}(\Delta_{\mathrm{L}}, \Delta'_{\mathrm{L}})$
- $CSP_{\mathrm{M}}(\Delta'_{\mathrm{U}}, \Delta'_{\mathrm{L}})$

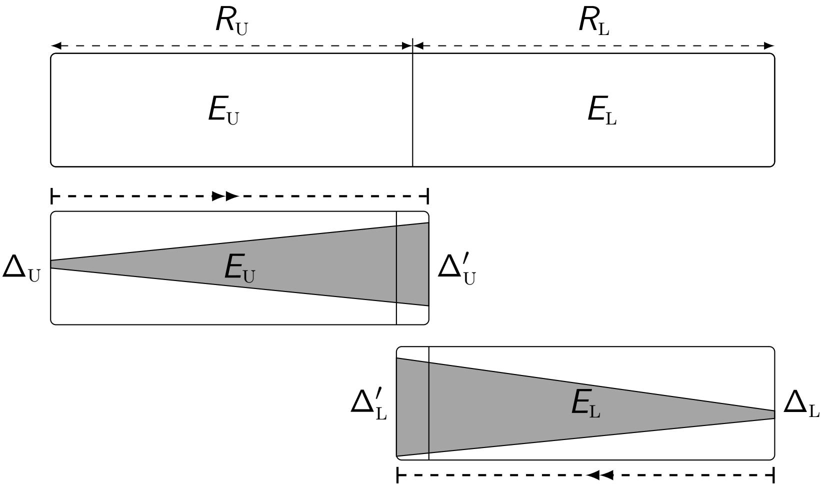




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Our Previous Method to Search Distinguishers [HSE23]

- $CSP_{\mathrm{U}}(\Delta_{\mathrm{U}}, \Delta'_{\mathrm{U}})$
- $CSP_{\mathrm{L}}(\Delta_{\mathrm{L}}, \Delta'_{\mathrm{L}})$
- $CSP_{\mathrm{M}}(\Delta'_{\mathrm{U}}, \Delta'_{\mathrm{L}})$

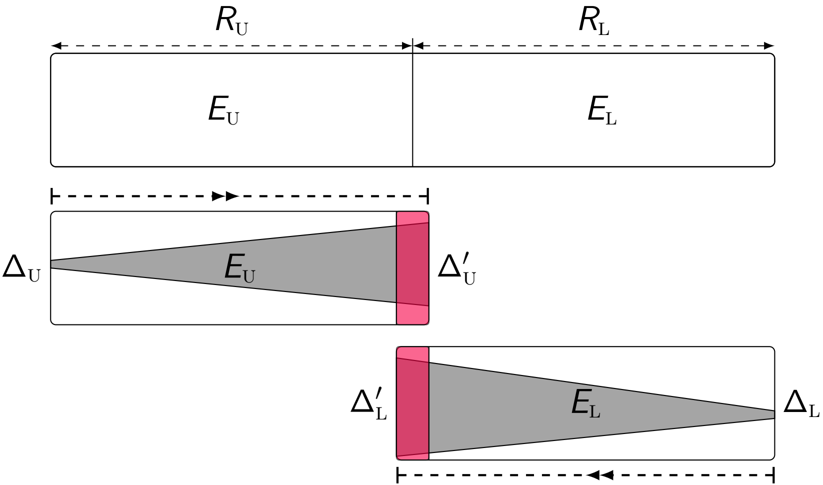




**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Our Previous Method to Search Distinguishers [HSE23]

- $CSP_{\mathrm{U}}(\Delta_{\mathrm{U}}, \Delta'_{\mathrm{U}})$
- $CSP_{\mathrm{L}}(\Delta_{\mathrm{L}}, \Delta'_{\mathrm{L}})$
- $CSP_{\mathtt{M}}(\Delta'_{\mathrm{U}}, \Delta'_{\mathrm{L}})$

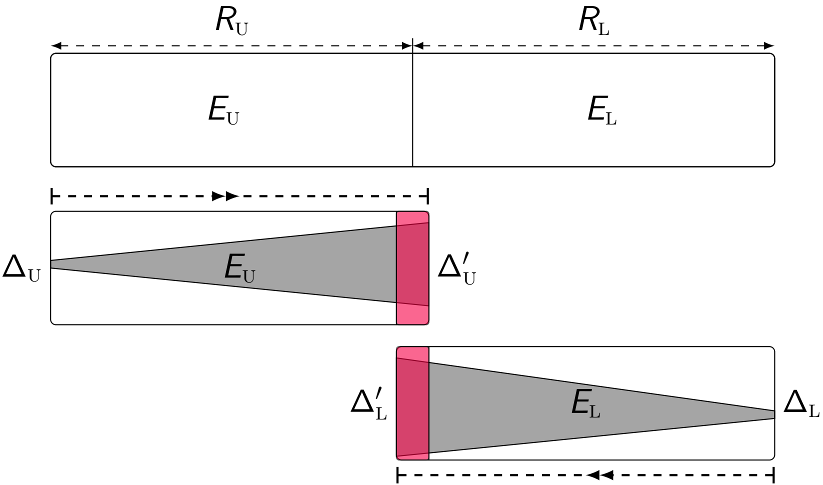


# Our Previous Method to Search Distinguishers [HSE23]

$CSP_{\mathrm{U}}(\Delta_{\mathrm{U}}, \Delta'_{\mathrm{U}})$

$CSP_{\mathrm{L}}(\Delta_{\mathrm{L}}, \Delta'_{\mathrm{L}})$

$CSP_{\mathtt{M}}(\Delta'_{\mathrm{U}}, \Delta'_{\mathrm{L}})$



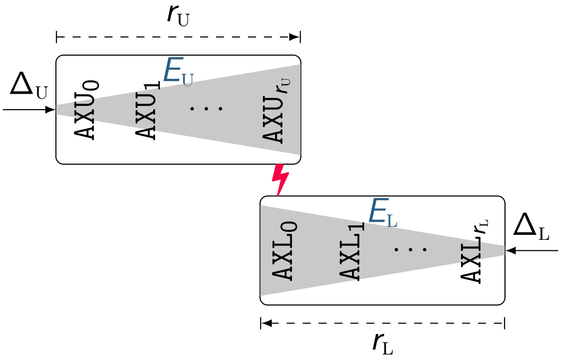


**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

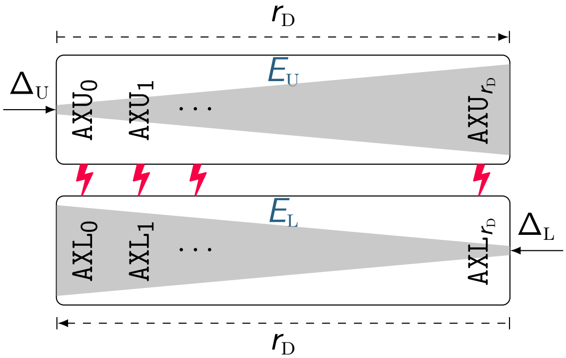
# Our New Word-Wise Method for Finding Distinguishers

# Relax the Limit of Fixing the Contradiction's Location

Find ID distinguisher for $r_{\mathrm{D}}(= r_{\mathrm{U}} + r_{\mathrm{L}})$ rounds




Modeling the distinguishers in [HSE23].

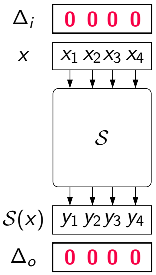



Our modeling of the distinguishers.

# Our New Bit-Wise Method for Finding Distinguishers

# Deterministic Bit-Wise Differential Trails (a.k.a. Undisturbed Bits [Tez14])

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{S}(x)$ | 4 | 0 | a | 7 | b | e | 1 | d | 9 | f | 6 | 8 | 5 | 2 | c | 3 |



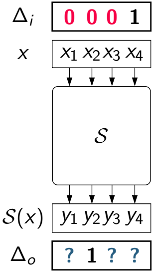

| $\Delta_i \setminus \Delta_o$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 |
| 2 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 4 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 2 |
| 3 | 0 | 2 | 0 | 2 | 0 | 0 | 4 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 2 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 2 | 2 | 2 | 2 |
| 5 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 4 | 4 | 0 | 0 | 0 | 0 |
| 6 | 0 | 2 | 0 | 2 | 0 | 4 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 |
| 7 | 0 | 2 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 |
| 9 | 0 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| a | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| b | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |
| c | 0 | 4 | 4 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| d | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| e | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| f | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |

$$\Delta_i = (0,0,0,0) \xrightarrow{S} \Delta_o = (0,0,0,0)$$

$$\Delta_i \neq (0,0,0,0) \xrightarrow{S} \Delta_o \neq (0,0,0,0)$$

# Deterministic Bit-Wise Differential Trails (a.k.a. Undisturbed Bits [Tez14])

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{S}(x)$ | 4 | 0 | a | 7 | b | e | 1 | d | 9 | f | 6 | 8 | 5 | 2 | c | 3 |




| $\Delta_i \setminus \Delta_o$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 |
| 2 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 4 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 2 |
| 3 | 0 | 2 | 0 | 2 | 0 | 0 | 4 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 2 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 2 | 2 | 2 | 2 |
| 5 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 4 | 4 | 0 | 0 | 0 | 0 |
| 6 | 0 | 2 | 0 | 2 | 0 | 4 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 |
| 7 | 0 | 2 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 |
| 9 | 0 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| a | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| b | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |
| c | 0 | 4 | 4 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| d | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| e | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| f | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |

$$\Delta_i = (0,0,0,0) \xrightarrow{S} \Delta_o = (0,0,0,0)$$

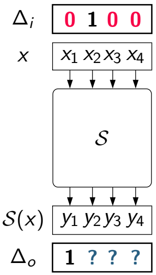
$$\Delta_i \neq (0,0,0,0) \xrightarrow{S} \Delta_o \neq (0,0,0,0)$$

$$\Delta_i = (0,0,0,1) \xrightarrow{S} \Delta_o = (?,1,?,?)$$


**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Deterministic Bit-Wise Differential Trails (a.k.a. Undisturbed Bits [Tez14])

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{S}(x)$ | 4 | 0 | a | 7 | b | e | 1 | d | 9 | f | 6 | 8 | 5 | 2 | c | 3 |




| $\Delta_i \setminus \Delta_o$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 |
| 2 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 4 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 2 |
| 3 | 0 | 2 | 0 | 2 | 0 | 0 | 4 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 2 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 2 | 2 | 2 | 2 |
| 5 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 4 | 4 | 0 | 0 | 0 | 0 |
| 6 | 0 | 2 | 0 | 2 | 0 | 4 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 |
| 7 | 0 | 2 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 |
| 9 | 0 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| a | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| b | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |
| c | 0 | 4 | 4 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| d | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| e | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| f | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |

$$\Delta_i = (0,0,0,0) \xrightarrow{S} \Delta_o = (0,0,0,0)$$

$$\Delta_i \neq (0,0,0,0) \xrightarrow{S} \Delta_o \neq (0,0,0,0)$$

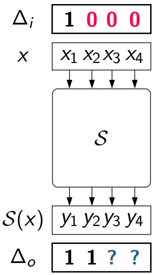
$$\Delta_i = (0,0,0,1) \xrightarrow{S} \Delta_o = (?,1,?,?)$$

$$\Delta_i = (0,1,0,0) \xrightarrow{S} \Delta_o = (1,?,?,?)$$


**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Deterministic Bit-Wise Differential Trails (a.k.a. Undisturbed Bits [Tez14])

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{S}(x)$ | 4 | 0 | a | 7 | b | e | 1 | d | 9 | f | 6 | 8 | 5 | 2 | c | 3 |




| $\Delta_i \setminus \Delta_o$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 |
| 2 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 4 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 2 |
| 3 | 0 | 2 | 0 | 2 | 0 | 0 | 4 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 2 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 2 | 2 | 2 | 2 |
| 5 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 4 | 4 | 0 | 0 | 0 | 0 |
| 6 | 0 | 2 | 0 | 2 | 0 | 4 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 |
| 7 | 0 | 2 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 |
| 9 | 0 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| a | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| b | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |
| c | 0 | 4 | 4 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| d | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| e | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| f | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |

$$\Delta_i = (0,0,0,0) \xrightarrow{S} \Delta_o = (0,0,0,0)$$

$$\Delta_i \neq (0,0,0,0) \xrightarrow{S} \Delta_o \neq (0,0,0,0)$$

$$\Delta_i = (0,0,0,1) \xrightarrow{S} \Delta_o = (?,1,?,?)$$

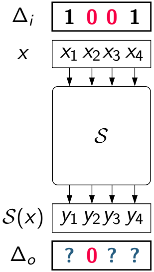
$$\Delta_i = (0,1,0,0) \xrightarrow{S} \Delta_o = (1,?,?,?)$$

$$\Delta_i = (1,0,0,0) \xrightarrow{S} \Delta_o = (1,1,?,?)$$


**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Deterministic Bit-Wise Differential Trails (a.k.a. Undisturbed Bits [Tez14])

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{S}(x)$ | 4 | 0 | a | 7 | b | e | 1 | d | 9 | f | 6 | 8 | 5 | 2 | c | 3 |




| $\Delta_i \setminus \Delta_o$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 |
| 2 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 4 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 2 |
| 3 | 0 | 2 | 0 | 2 | 0 | 0 | 4 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 2 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 2 | 2 | 2 | 2 |
| 5 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 4 | 4 | 0 | 0 | 0 | 0 |
| 6 | 0 | 2 | 0 | 2 | 0 | 4 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 |
| 7 | 0 | 2 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 |
| 9 | 0 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| a | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| b | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |
| c | 0 | 4 | 4 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| d | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| e | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| f | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |

$$\Delta_i = (0,0,0,0) \xrightarrow{S} \Delta_o = (0,0,0,0)$$

$$\Delta_i \neq (0,0,0,0) \xrightarrow{S} \Delta_o \neq (0,0,0,0)$$

$$\Delta_i = (0,0,0,1) \xrightarrow{S} \Delta_o = (?,1,?,?)$$

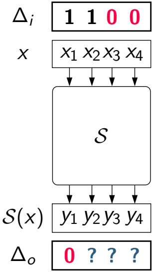
$$\Delta_i = (0,1,0,0) \xrightarrow{S} \Delta_o = (1,?,?,?)$$

$$\Delta_i = (1,0,0,0) \xrightarrow{S} \Delta_o = (1,1,?,?)$$

$$\Delta_i = (1,0,0,1) \xrightarrow{S} \Delta_o = (?,0,?,?)$$

# Deterministic Bit-Wise Differential Trails (a.k.a. Undisturbed Bits [Tez14])

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{S}(x)$ | 4 | 0 | a | 7 | b | e | 1 | d | 9 | f | 6 | 8 | 5 | 2 | c | 3 |




| $\Delta_i \setminus \Delta_o$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 |
| 2 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 4 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 2 |
| 3 | 0 | 2 | 0 | 2 | 0 | 0 | 4 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 2 | 2 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 2 | 2 | 2 | 2 |
| 5 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 4 | 4 | 0 | 0 | 0 | 0 |
| 6 | 0 | 2 | 0 | 2 | 0 | 4 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 |
| 7 | 0 | 2 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 |
| 9 | 0 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| a | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| b | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |
| c | 0 | 4 | 4 | 0 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| d | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 |
| e | 0 | 0 | 2 | 2 | 0 | 2 | 2 | 0 | 4 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| f | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 4 | 0 | 0 | 0 | 2 | 0 | 2 | 0 |

$$\Delta_i = (0,0,0,0) \xrightarrow{S} \Delta_o = (0,0,0,0)$$

$$\Delta_i \neq (0,0,0,0) \xrightarrow{S} \Delta_o \neq (0,0,0,0)$$

$$\Delta_i = (0,0,0,1) \xrightarrow{S} \Delta_o = (?,1,?,?)$$

$$\Delta_i = (0,1,0,0) \xrightarrow{S} \Delta_o = (1,?,?,?)$$

$$\Delta_i = (1,0,0,0) \xrightarrow{S} \Delta_o = (1,1,?,?)$$

$$\Delta_i = (1,0,0,1) \xrightarrow{S} \Delta_o = (?,0,?,?)$$

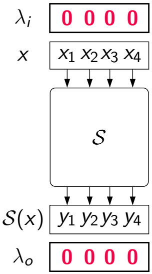
$$\Delta_i = (1,1,0,0) \xrightarrow{S} \Delta_o = (0,?,?,?)$$



**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Deterministic Bit-Wise Linear Trails

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{S}(x)$ | 4 | 0 | a | 7 | b | e | 1 | d | 9 | f | 6 | 8 | 5 | 2 | c | 3 |




| $\lambda_i \setminus \lambda_o$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 4 | -4 | 0 | -8 | -4 | -4 | 0 | 0 | 4 | -4 | -8 | 0 | 4 | 4 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 8 | 0 | 0 | 8 | -8 | 0 |
| 3 | 0 | -8 | 4 | 4 | 0 | 0 | -4 | 4 | 0 | 0 | -4 | 4 | -8 | 0 | -4 | -4 |
| 4 | 0 | 4 | 0 | 4 | 0 | 4 | 8 | -4 | 0 | 4 | 0 | 4 | -8 | -4 | 0 | 4 |
| 5 | 0 | 4 | -4 | -8 | 0 | -4 | -4 | 0 | 0 | 4 | -4 | 8 | 0 | -4 | -4 | 0 |
| 6 | 0 | -4 | 8 | 4 | 0 | -4 | 0 | -4 | 0 | 4 | 0 | 4 | 8 | -4 | 0 | 4 |
| 7 | 0 | 4 | 4 | 0 | 0 | -4 | 4 | -8 | 0 | -4 | -4 | 0 | 0 | 4 | -4 | -8 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 8 | 0 | 0 | 8 | -8 |
| 9 | 0 | 0 | -4 | 4 | 8 | 0 | -4 | -4 | 0 | 0 | 4 | -4 | 0 | -8 | -4 | -4 |
| a | 0 | 8 | 0 | 8 | 0 | -8 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b | 0 | 0 | -4 | 4 | -8 | 0 | -4 | -4 | 0 | 8 | -4 | -4 | 0 | 0 | 4 | -4 |
| c | 0 | 4 | 0 | 4 | 0 | 4 | -8 | -4 | 8 | -4 | 0 | 4 | 0 | 4 | 0 | 4 |
| d | 0 | 4 | 4 | 0 | -8 | 4 | -4 | 0 | -8 | -4 | 4 | 0 | 0 | -4 | -4 | 0 |
| e | 0 | 4 | 8 | -4 | 0 | 4 | 0 | 4 | 8 | 4 | 0 | -4 | 0 | -4 | 0 | -4 |
| f | 0 | -4 | -4 | 0 | -8 | -4 | 4 | 0 | 8 | -4 | 4 | 0 | 0 | -4 | -4 | 0 |

$$\lambda_i = (0,0,0,0) \xrightarrow{S} \lambda_o = (0,0,0,0)$$

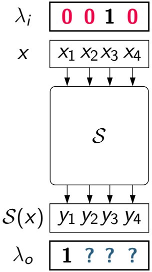
$$\lambda_i \neq (0,0,0,0) \xrightarrow{S} \lambda_o \neq (0,0,0,0)$$



**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Deterministic Bit-Wise Linear Trails

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{S}(x)$ | 4 | 0 | a | 7 | b | e | 1 | d | 9 | f | 6 | 8 | 5 | 2 | c | 3 |




| $\lambda_i \setminus \lambda_o$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 4 | -4 | 0 | -8 | -4 | -4 | 0 | 0 | 4 | -4 | -8 | 0 | 4 | 4 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 8 | 0 | 0 | 8 | -8 | 0 |
| 3 | 0 | -8 | 4 | 4 | 0 | 0 | -4 | 4 | 0 | 0 | -4 | 4 | -8 | 0 | -4 | -4 |
| 4 | 0 | 4 | 0 | 4 | 0 | 4 | 8 | -4 | 0 | 4 | 0 | 4 | -8 | -4 | 0 | 4 |
| 5 | 0 | 4 | -4 | -8 | 0 | -4 | -4 | 0 | 0 | 4 | -4 | 8 | 0 | -4 | -4 | 0 |
| 6 | 0 | -4 | 8 | 4 | 0 | -4 | 0 | -4 | 0 | 4 | 0 | 4 | 8 | -4 | 0 | 4 |
| 7 | 0 | 4 | 4 | 0 | 0 | -4 | 4 | -8 | 0 | -4 | -4 | 0 | 0 | 4 | -4 | -8 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 8 | 0 | 0 | 8 | -8 |
| 9 | 0 | 0 | -4 | 4 | 8 | 0 | -4 | -4 | 0 | 0 | 4 | -4 | 0 | -8 | -4 | -4 |
| a | 0 | 8 | 0 | 8 | 0 | -8 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b | 0 | 0 | -4 | 4 | -8 | 0 | -4 | -4 | 0 | 8 | -4 | -4 | 0 | 0 | 4 | -4 |
| c | 0 | 4 | 0 | 4 | 0 | 4 | -8 | -4 | 8 | -4 | 0 | 4 | 0 | 4 | 0 | 4 |
| d | 0 | 4 | 4 | 0 | -8 | 4 | -4 | 0 | -8 | -4 | 4 | 0 | 0 | -4 | -4 | 0 |
| e | 0 | 4 | 8 | -4 | 0 | 4 | 0 | 4 | 8 | 4 | 0 | -4 | 0 | -4 | 0 | -4 |
| f | 0 | -4 | -4 | 0 | -8 | -4 | 4 | 0 | 8 | -4 | 4 | 0 | 0 | -4 | -4 | 0 |

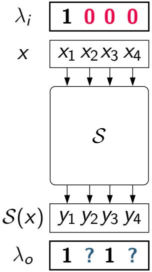
$$\lambda_i = (0,0,0,0) \xrightarrow{S} \lambda_o = (0,0,0,0)$$

$$\lambda_i \neq (0,0,0,0) \xrightarrow{S} \lambda_o \neq (0,0,0,0)$$

$$\lambda_i = (0,0,1,0) \xrightarrow{S} \lambda_o = (1,?,?,?)$$

# Deterministic Bit-Wise Linear Trails

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{S}(x)$ | 4 | 0 | a | 7 | b | e | 1 | d | 9 | f | 6 | 8 | 5 | 2 | c | 3 |




| $\lambda_i \setminus \lambda_o$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 4 | -4 | 0 | -8 | -4 | -4 | 0 | 0 | 4 | -4 | -8 | 0 | 4 | 4 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 8 | 0 | 0 | 8 | -8 | 0 |
| 3 | 0 | -8 | 4 | 4 | 0 | 0 | -4 | 4 | 0 | 0 | -4 | 4 | -8 | 0 | -4 | -4 |
| 4 | 0 | 4 | 0 | 4 | 0 | 4 | 8 | -4 | 0 | 4 | 0 | 4 | -8 | -4 | 0 | 4 |
| 5 | 0 | 4 | -4 | -8 | 0 | -4 | -4 | 0 | 0 | 4 | -4 | 8 | 0 | -4 | -4 | 0 |
| 6 | 0 | -4 | 8 | 4 | 0 | -4 | 0 | -4 | 0 | 4 | 0 | 4 | 8 | -4 | 0 | 4 |
| 7 | 0 | 4 | 4 | 0 | 0 | -4 | 4 | -8 | 0 | -4 | -4 | 0 | 0 | 4 | -4 | -8 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 8 | 0 | 0 | 8 | -8 |
| 9 | 0 | 0 | -4 | 4 | 8 | 0 | -4 | -4 | 0 | 0 | 4 | -4 | 0 | -8 | -4 | -4 |
| a | 0 | 8 | 0 | 8 | 0 | -8 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b | 0 | 0 | -4 | 4 | -8 | 0 | -4 | -4 | 0 | 8 | -4 | -4 | 0 | 0 | 4 | -4 |
| c | 0 | 4 | 0 | 4 | 0 | 4 | -8 | -4 | 8 | -4 | 0 | 4 | 0 | 4 | 0 | 4 |
| d | 0 | 4 | 4 | 0 | -8 | 4 | -4 | 0 | -8 | -4 | 4 | 0 | 0 | -4 | -4 | 0 |
| e | 0 | 4 | 8 | -4 | 0 | 4 | 0 | 4 | 8 | 4 | 0 | -4 | 0 | -4 | 0 | -4 |
| f | 0 | -4 | -4 | 0 | -8 | -4 | 4 | 0 | 8 | -4 | 4 | 0 | 0 | -4 | -4 | 0 |

$$\lambda_i = (0,0,0,0) \xrightarrow{S} \lambda_o = (0,0,0,0)$$

$$\lambda_i \neq (0,0,0,0) \xrightarrow{S} \lambda_o \neq (0,0,0,0)$$

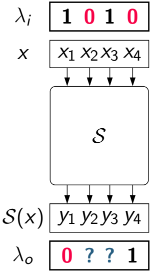
$$\lambda_i = (0,0,1,0) \xrightarrow{S} \lambda_o = (1,?,?,?)$$

$$\lambda_i = (1,0,0,0) \xrightarrow{S} \lambda_o = (1,?,1,?)$$


**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder
FSE 2024 - Leuven, Belgium

# Deterministic Bit-Wise Linear Trails

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{S}(x)$ | 4 | 0 | a | 7 | b | e | 1 | d | 9 | f | 6 | 8 | 5 | 2 | c | 3 |




| $\lambda_i \setminus \lambda_o$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 4 | -4 | 0 | -8 | -4 | -4 | 0 | 0 | 4 | -4 | -8 | 0 | 4 | 4 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 8 | 0 | 0 | 8 | -8 | 0 |
| 3 | 0 | -8 | 4 | 4 | 0 | 0 | -4 | 4 | 0 | 0 | -4 | 4 | -8 | 0 | -4 | -4 |
| 4 | 0 | 4 | 0 | 4 | 0 | 4 | 8 | -4 | 0 | 4 | 0 | 4 | -8 | -4 | 0 | 4 |
| 5 | 0 | 4 | -4 | -8 | 0 | -4 | -4 | 0 | 0 | 4 | -4 | 8 | 0 | -4 | -4 | 0 |
| 6 | 0 | -4 | 8 | 4 | 0 | -4 | 0 | -4 | 0 | 4 | 0 | 4 | 8 | -4 | 0 | 4 |
| 7 | 0 | 4 | 4 | 0 | 0 | -4 | 4 | -8 | 0 | -4 | -4 | 0 | 0 | 4 | -4 | -8 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 8 | 0 | 0 | 8 | -8 |
| 9 | 0 | 0 | -4 | 4 | 8 | 0 | -4 | -4 | 0 | 0 | 4 | -4 | 0 | -8 | -4 | -4 |
| a | 0 | 8 | 0 | 8 | 0 | -8 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b | 0 | 0 | -4 | 4 | -8 | 0 | -4 | -4 | 0 | 8 | -4 | -4 | 0 | 0 | 4 | -4 |
| c | 0 | 4 | 0 | 4 | 0 | 4 | -8 | -4 | 8 | -4 | 0 | 4 | 0 | 4 | 0 | 4 |
| d | 0 | 4 | 4 | 0 | -8 | 4 | -4 | 0 | -8 | -4 | 4 | 0 | 0 | -4 | -4 | 0 |
| e | 0 | 4 | 8 | -4 | 0 | 4 | 0 | 4 | 8 | 4 | 0 | -4 | 0 | -4 | 0 | -4 |
| f | 0 | -4 | -4 | 0 | -8 | -4 | 4 | 0 | 8 | -4 | 4 | 0 | 0 | -4 | -4 | 0 |

$$\lambda_i = (0,0,0,0) \xrightarrow{S} \lambda_o = (0,0,0,0)$$

$$\lambda_i \neq (0,0,0,0) \xrightarrow{S} \lambda_o \neq (0,0,0,0)$$

$$\lambda_i = (0,0,1,0) \xrightarrow{S} \lambda_o = (1,?,?,?)$$

$$\lambda_i = (1,0,0,0) \xrightarrow{S} \lambda_o = (1,?,1,?)$$

$$\lambda_i = (1,0,1,0) \xrightarrow{S} \lambda_o = (0,?,?,1)$$

# CP Model for Deterministic Bit-Wise Trails - I

- For each bit position, we define an integer variable with domain $\{\mathbf{0}, \mathbf{1}, \mathbf{-1}\}$.
- Define CP constraints to model the propagation of deterministic bit-wise trails.

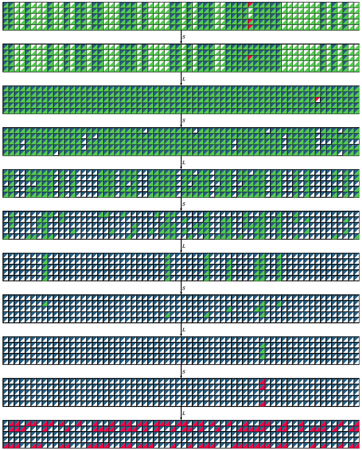
## S-box

Assume that $x[i], y[i]$ are integer variables with domain $\{-1, 0, 1\}$ to encode the input and output differences at the $i$-th bit position, respectively. The valid deterministic differential transitions satisfy the following:

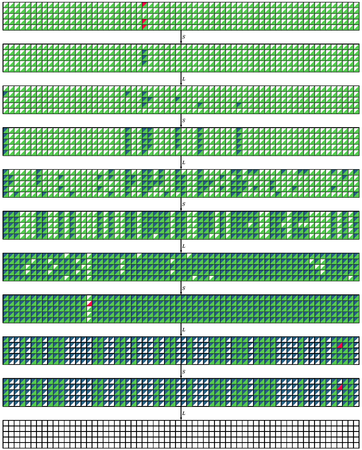
$$\begin{cases} \mathtt{if}(x[0] = 0 \land x[1] = 0 \land x[2] = 0 \land x[3] = 0)\ \mathtt{then}\ (y[0] = 0 \land y[1] = 0 \land y[2] = 0 \land y[3] = 0) \\ \mathtt{elseif}(x[0] = 0 \land x[1] = 0 \land x[2] = 0 \land x[3] = 1)\ \mathtt{then}\ (y[0] = -1 \land y[1] = 1 \land y[2] = -1 \land y[3] = -1) \\ \mathtt{elseif}(x[0] = 0 \land x[1] = 1 \land x[2] = 0 \land x[3] = 0)\ \mathtt{then}\ (y[0] = 1 \land y[1] = -1 \land y[2] = -1 \land y[3] = -1) \\ \mathtt{elseif}(x[0] = 1 \land x[1] = 0 \land x[2] = 0 \land x[3] = 0)\ \mathtt{then}\ (y[0] = 1 \land y[1] = 1 \land y[2] = -1 \land y[3] = -1) \\ \mathtt{elseif}(x[0] = 1 \land x[1] = 0 \land x[2] = 0 \land x[3] = 1)\ \mathtt{then}\ (y[0] = -1 \land y[1] = 0 \land y[2] = -1 \land y[3] = -1) \\ \mathtt{elseif}(x[0] = 1 \land x[1] = 1 \land x[2] = 0 \land x[3] = 0)\ \mathtt{then}\ (y[0] = 0 \land y[1] = -1 \land y[2] = -1 \land y[3] = -1) \\ \mathtt{else}(y[0] = -1 \land y[1] = -1 \land y[2] = -1 \land y[3] = -1)\ \mathtt{endif}; \end{cases}$$


**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Example: ID/ZC Distinguishers for 5 Rounds of Ascon



$2^{155}$ ZC Distinguishers (upper/lower nonzero: /)



$2^{155}$ ID Distinguishers (upper/lower unknown: /)

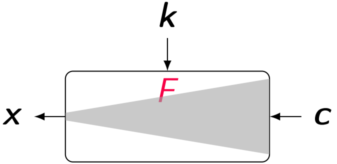
# The Advantages of Our Method to Search for Distinguishers

- Based on satisfiability of the CP model
- Any feasible solutions of our CP model is a distinguisher
- We do not fix the input/output of distinguisher
- Extendable to a unified model for key-recovery
  - Enables us to find a distinguisher optimized for key-recovery
  - Enables us to consider key-recovery techniques:
    - MitM
    - Key bridging
    - *Partial-sum technique*

# Our Unified CP Model for Partial-Sum Key-Recovery

# Naive Approach v.s. Partial-Sum Technique

- Naive approach:
  - $x = F(\boldsymbol{k}, \boldsymbol{c})$
  - $T = N \cdot 2^{|\boldsymbol{k}|}$

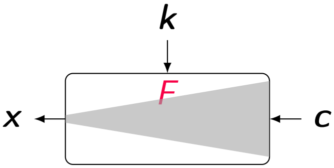



- Partial-sum technique:
  - $x_1 = f_1(\boldsymbol{k}_1, x_0), x_2 = f_2(\boldsymbol{k}_2, x_1), \cdots, x = f_n(\boldsymbol{k}_n, x_{n-1})$
  - $x_0 = \boldsymbol{c}, N_0 = N, N_i < N$
  - $T = \sum_{i=1}^{n} \frac{N_{i-1}}{n} \cdot 2^{|\boldsymbol{k}_1| + \cdots + |\boldsymbol{k}_i|} < \sum_{i=1}^{n} \frac{N}{n} \cdot 2^{|\boldsymbol{k}|}$
  - $T < N \cdot 2^{|\boldsymbol{k}|}$

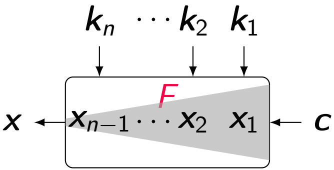

**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder
FSE 2024 - Leuven, Belgium

# Naive Approach v.s. Partial-Sum Technique

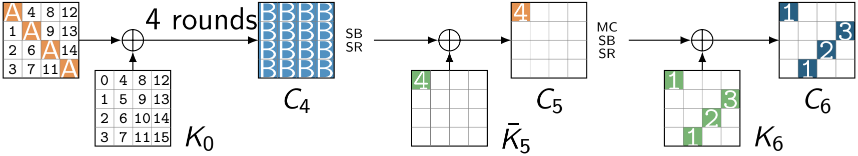
- Naive approach:
  - $x = F(k, c)$
  - $T = N \cdot 2^{|k|}$




- Partial-sum technique:
  - $x_1 = f_1(k_1, x_0), x_2 = f_2(k_2, x_1), \cdots, x = f_n(k_n, x_{n-1})$
  - $x_0 = c, N_0 = N, N_i < N$
  - $T = \sum_{i=1}^{n} \frac{N_{i-1}}{n} \cdot 2^{|k_1|+\cdots+|k_i|} < \sum_{i=1}^{n} \frac{N}{n} \cdot 2^{|k|}$
  - $T < N \cdot 2^{|k|}$



# Example: Partial-Sum Integral Key Recovery for AES [Fer+00]

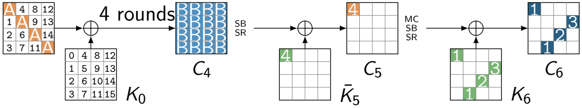


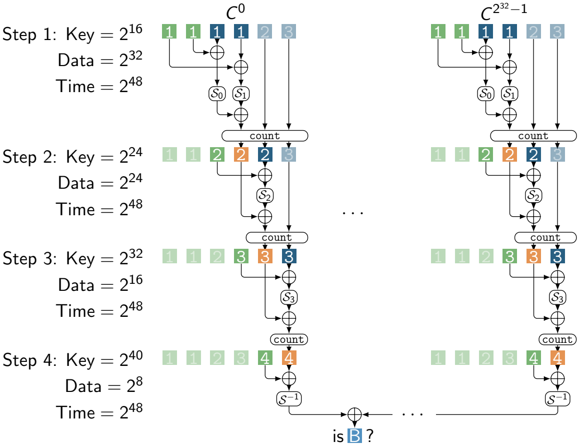

$$C_4[0] = \mathcal{S}^{-1}\Big(\bar{K}_5[0] \oplus \texttt{0E} \cdot \mathcal{S}^{-1}(C_6[0] \oplus K_6[0]) \oplus \texttt{09} \cdot \mathcal{S}^{-1}(C_6[7] \oplus K_6[7])$$
$$\oplus \texttt{0D} \cdot \mathcal{S}^{-1}(C_6[10] \oplus K_6[10]) \oplus \texttt{0B} \cdot \mathcal{S}^{-1}(C_6[13] \oplus K_6[13])\Big)$$

- Time complexity of naive key recovery: $6 \times 2^{32} \times 2^{40} \approx 2^{74.58}$


**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Partial-sum Technique for Integral Key Recovery [Fer+00]



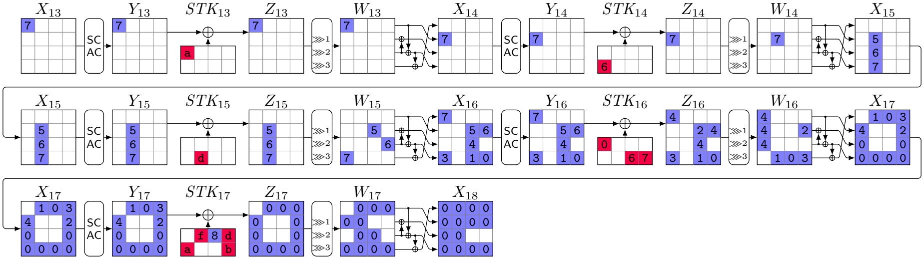




- Guess $K_6[0, 7]$ and derive $\mathcal{S}_0\left(C_6[0] \oplus K_6[0]\right) \oplus \mathcal{S}_1\left(C_6[7] \oplus K_6[7]\right)$
- Guess $K_6[10]$ and derive $\mathcal{S}_2\left(C_6[10] \oplus K_6[10]\right)$
- Guess $K_6[13]$ and derive $\mathcal{S}_3\left(C_6[13] \oplus K_6[13]\right)$
- Guess $\bar{K}_5[0]$ and derive $C_4[0]$
- Time complexity: $6 \times 4 \times 2^{48} \approx 2^{52}$ S-box lookups



**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

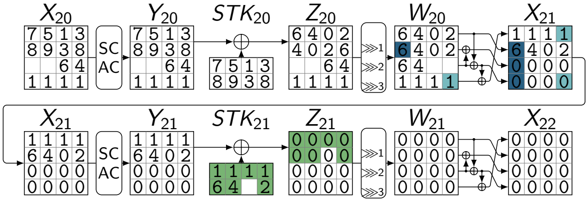
# Our CP Model for Partial-Sum Technique - I




| Step | Guessed | K × D =Mem | Time | Stored Texts |
|---|---|---|---|---|
| 0 | – | $2^{0} \times 2^{40} = 2^{40}$ | $2^{40-5.2}$ | $Z_{17}[1,3,4,7]$; $X_{17}[8,11,12,13,15]$; $X_{16}[15]$ |
| 1 | $STK_{17}[1]$ | $2^{4} \times 2^{36} = 2^{40}$ | $2^{44-7.2}$ | $Z_{17}[3,4,7]$; $X_{17}[8,11,12,15]$; $X_{16}[14,15]$ |
| 2 | $STK_{17}[7]$ | $2^{8} \times 2^{32} = 2^{40}$ | $2^{44-8.2}$ | $Z_{17}[3,4]$; $X_{17}[8,12,15]$; $Z_{16}[6]$; $X_{16}[14,15]$ |
| 3 | $STK_{17}[3]$ | $2^{12} \times 2^{28} = 2^{40}$ | $2^{44-7.2}$ | $Z_{17}[4]$; $X_{17}[8,12]$; $Z_{16}[6]$; $X_{16}[12,14,15]$ |
| 4 | $STK_{17}[4]$ | $2^{16} \times 2^{28} = 2^{44}$ | $2^{44-7.2}$ | $Z_{16}[0,6,7]$; $X_{16}[10,12,14,15]$ |
| 5 | $STK_{16}[6]$ | $2^{20} \times 2^{20} = 2^{40}$ | $2^{48-7.2}$ | $Z_{16}[0,7]$; $X_{16}[12,15]$; $X_{15}[5]$ |
| 6 | $STK_{16}[7]$ | $2^{24} \times 2^{16} = 2^{40}$ | $2^{44-7.2}$ | $Z_{16}[0]$; $X_{16}[12]$; $X_{15}[5,9]$ |
| 7 | $STK_{16}[0]$ | $2^{28} \times 2^{4} = 2^{32}$ | $2^{44-6.2}$ | $X_{13}[0]$ |
| $\Sigma$ | | $2^{44}$ | $2^{41.32}$ | |

# Our CP Model for Partial-Sum Technique - II

- Assume that in each step we guess at least one cell of the involved keys.
- We define the number of steps $s$ which is less than the number of involved key cells.
- For each cell we define an integer variable with domain $\{0, \cdots, s\}$.
- We define some constraints to compute the step number of deriving each cell.





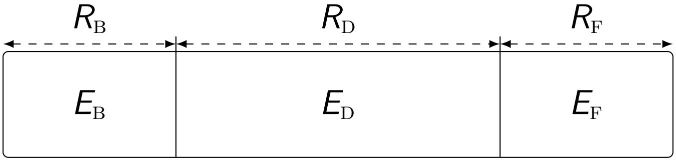

**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Our Unified Model for Finding Integral Attack

- Our CP model for finding complete integral attack includes the following modules:
  - Model the distinguisher part
  - Model the meet-in-the-middle technique
  - Model the involved cells in key recovery
  - Model the step assignment
  - Model the tweakey schedule (key-bridging)
  - Model the time/memory complexity evaluation
- Objective function: minimize the total time complexity

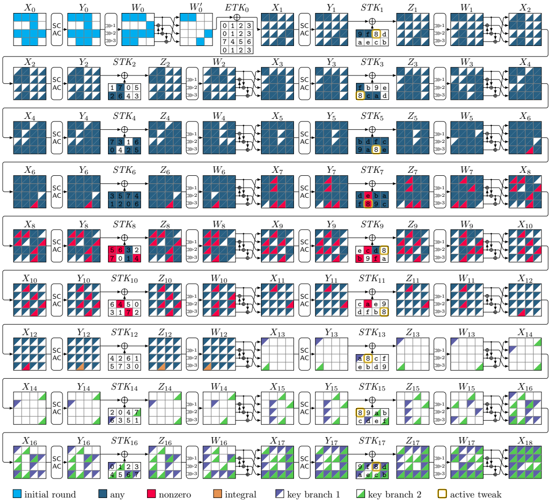
# Usage of Our Tool

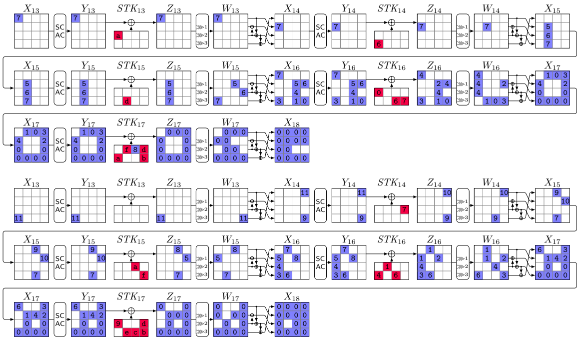
```
python3 attack.py -RB 1 -RD 12 -RF 5
```




- We use MiniZinc [Net+07] to create our CP models
- We use Gurobi [Gur22] and OrTools [PF] as the CP solvers
- Our tool can find the results in a few seconds running on a regular laptop

# Example: 18-round Integral Attack on SKINNY-$n$-$n$



# Contributions and Future Works

# Contributions and Future Works

- Contributions
  - Improving unified models for finding complete ID/ZC/integral attacks
  - Introducing a CP model for the partial-sum technique for the first time
  - Found improved attacks for SKINNY, and ForskSKINNY, and QARMAv2
- Future works
  - Extending our distinguisher models for ID/ZC to find indirect contradictions
  - Extending our tools to AndRX and ARX ciphers, e.g., Simeck, and SPECK.
  - Extending our approach to division property or monomial prediction techniques
  - Improving the key-recovery part of our CP models for ZC attacks

: https://github.com/hadipourh/zeroplus

: https://ia.cr/2023/1701


**Hosein Hadipour**, Simon Gerhalter, Sadegh Sadeghi, Maria Eichlseder

# Bibliography I


[Ava+23] Roberto Avanzi et al. **The QARMAv2 Family of Tweakable Block Ciphers**. *IACR Trans. Symmetric Cryptol.* 2023.3 (2023), pp. 25–73. DOI: 10.46586/TOSC.V2023.I3.25-73.

[BBS99] Eli Biham, Alex Biryukov, and Adi Shamir. **Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials**. EUROCRYPT 1999. Vol. 1592. LNCS. Springer, 1999, pp. 12–23. DOI: 10.1007/3-540-48910-X_2.

[BDL20] Augustin Bariant, Nicolas David, and Gaëtan Leurent. **Cryptanalysis of Forkciphers**. *IACR Trans. Symmetric Cryptol.* 2020.1 (2020), pp. 233–265. DOI: 10.13154/tosc.v2020.i1.233-265.

[BR14] Andrey Bogdanov and Vincent Rijmen. **Linear hulls with correlation zero and linear cryptanalysis of block ciphers**. *Des. Codes Cryptogr.* 70.3 (2014), pp. 369–383. DOI: 10.1007/s10623-012-9697-z.

# Bibliography II


[Cui+16] Tingting Cui et al. **New Automatic Search Tool for Impossible Differentials and Zero-Correlation Linear Approximations**. IACR Cryptology ePrint Archive, Report 2016/689. 2016. URL: https://eprint.iacr.org/2016/689.

[DF16] Patrick Derbez and Pierre-Alain Fouque. **Automatic Search of Meet-in-the-Middle and Impossible Differential Attacks**. CRYPTO 2016. Vol. 9815. LNCS. Springer, 2016, pp. 157–184.

[DKR97] Joan Daemen, Lars R. Knudsen, and Vincent Rijmen. **The Block Cipher Square**. FSE 1997. Vol. 1267. LNCS. Springer, 1997, pp. 149–165. DOI: 10.1007/BFb0052343.

[Fer+00] Niels Ferguson et al. **Improved Cryptanalysis of Rijndael**. FSE 2000. Vol. 1978. LNCS. Springer, 2000, pp. 213–230. DOI: 10.1007/3-540-44706-7_15.

# Bibliography III


[Gur22] Gurobi Optimization, LLC. **Gurobi Optimizer Reference Manual**. 2022. URL: https://www.gurobi.com.

[HSE23] Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder. **Finding the Impossible: Automated Search for Full Impossible Differential, Zero-Correlation, and Integral Attacks**. EUROCRYPT 2023. Vol. 14007. LNCS. Springer, 2023, pp. 128–157. DOI: 10.1007/978-3-031-30634-1_5.

[Knu98] Lars Knudsen. **DEAL-a 128-bit block cipher**. *complexity* 258.2 (1998), p. 216.

[Lai94] Xuejia Lai. **Higher order derivatives and differential cryptanalysis**. *Communications and cryptography*. Springer, 1994, pp. 227–233.

# Bibliography IV


[Lu+08] Jiqiang Lu et al. **Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1**. CT-RSA 2008. Vol. 4964. LNCS. Springer, 2008, pp. 370–386. DOI: 10.1007/978-3-540-79263-5_24.

[Net+07] Nicholas Nethercote et al. **MiniZinc: Towards a Standard CP Modelling Language**. CP 2007. Vol. 4741. LNCS. Springer, 2007, pp. 529–543.

[Niu+21] Chao Niu et al. **Zero-Correlation Linear Cryptanalysis with Equal Treatment for Plaintexts and Tweakeys**. CT-RSA 2021. Vol. 12704. LNCS. Springer, 2021, pp. 126–147. DOI: 10.1007/978-3-030-75539-3_6.

[PF] Laurent Perron and Vincent Furnon. **OR-Tools**. Version 9.3. Google. URL: https://developers.google.com/optimization/.

# Bibliography V


[ST17] Yu Sasaki and Yosuke Todo. **New Impossible Differential Search Tool from Design and Cryptanalysis Aspects**. EUROCRYPT 2017. Cham: Springer International Publishing, 2017, pp. 185–215. DOI: 10.1007/978-3-319-56617-7_7.

[Sun+15] Bing Sun et al. **Links Among Impossible Differential, Integral and Zero Correlation Linear Cryptanalysis**. CRYPTO 2015. Vol. 9215. LNCS. Springer, 2015, pp. 95–115. DOI: 10.1007/978-3-662-47989-6_5.

[Sun+17] Siwei Sun et al. **Analysis of AES, SKINNY, and Others with Constraint Programming**. *IACR Transactions on Symmetric Cryptology* 2017.1 (Mar. 2017), pp. 281–306. DOI: 10.13154/tosc.v2017.i1.281-306.

# Bibliography VI

[Sun+20] Ling Sun et al. **On the Usage of Deterministic (Related-Key) Truncated Differentials and Multidimensional Linear Approximations for SPN Ciphers**. *IACR Transactions on Symmetric Cryptology* 2020.3 (Sept. 2020), pp. 262–287. DOI: 10.13154/tosc.v2020.i3.262-287.

[Tez14] Cihangir Tezcan. **Improbable differential attacks on Present using undisturbed bits**. *J. Comput. Appl. Math.* 259 (2014), pp. 503–511. DOI: 10.1016/j.cam.2013.06.023.

[Xia+16] Zejun Xiang et al. **Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers**. ASIACRYPT 2016. Vol. 10031. LNCS. 2016, pp. 648–678. DOI: 10.1007/978-3-662-53887-6_24.