Revisiting Differential-Linear Attacks via a Boomerang Perspective

Applications to AES, Ascon, CLEFIA, SKINNY, PRESENT, KNOT, TWINE, WARP, LBlock, Simeck, and SERPENT

Hosein Hadipour Patrick Derbez Maria Eichlseder

CRYPTO 2024 - California, USA

› hsn.hadipour@gmail.com

TU Graz
SCIENCE PASSION TECHNOLOGY

Research Gap and Our Contributions

Research Gap

- How to analytically estimate the correlation of DL distinguishers?
- How to (efficiently) find good DL distinguishers?

Contributions

- Generalizing the DLCT framework [Bar+19] for analytical correlation estimation.
- Introducing an efficient method to search for DL distinguishers applicable to:
  - Strongly aligned SPN primitives: AES, SKINNY
  - Weakly aligned SPN primitives: Ascon, SERPENT, KNOT, PRESENT
  - Feistel structures: CLEFIA, TWINE, LBlock, LBlock-s, WARP
  - AndRX designs: Simeck

Research Gap and Our Contributions

Research Gap

- How to analytically estimate the correlation of DL distinguishers?
- How to (efficiently) find good DL distinguishers?

Contributions

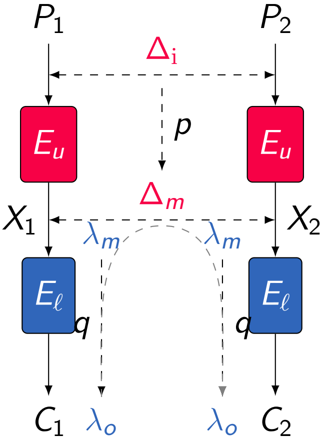
- Generalizing the `DLCT` framework [Bar+19] for analytical correlation estimation.
- Introducing an efficient method to search for DL distinguishers applicable to:
  - Strongly aligned SPN primitives: AES, SKINNY
  - Weakly aligned SPN primitives: Ascon, SERPENT, KNOT, PRESENT
  - Feistel structures: CLEFIA, TWINE, LBlock, LBlock-s, WARP
  - AndRX designs: Simeck

Outline

Background

Differential-Linear (DL) Attack [LH94]

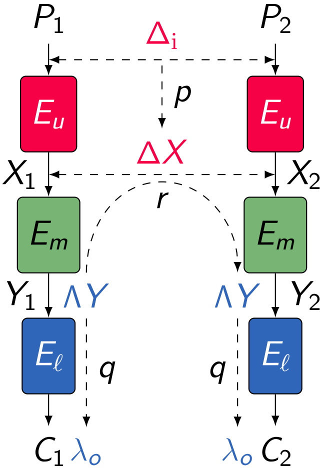
- $\mathbb{P}(\Delta_{\mathrm{i}} \xrightarrow{E_u} \Delta_m) = p$
- $\mathbb{C}(\lambda_m \xrightarrow{E_\ell} \lambda_{\mathrm{o}}) = q$
- Assumptions ($\Delta X = X_1 \oplus X_2$):
  1. E_u, and E_ℓ are statistically independent
  2. $\mathbb{P}\left(\lambda_m \cdot \Delta X = 0\right) = 1/2$ when $\Delta X \neq \Delta_m$
- $\mathbb{C}\left(\lambda_{\mathrm{o}} \cdot \Delta C\right) = (-1)^{\lambda_m \cdot \Delta_m} \cdot pq^2 = \pm pq^2$



Hosein Hadipour, Patrick Derbez, Maria Eichlseder

Sandwich Framework for DL Attack [BLN14; DKS14; Bar+19]

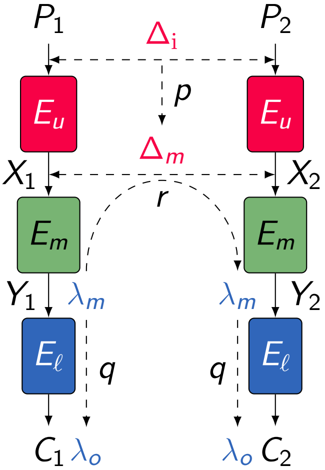
- $\mathbb{R}(\Delta X, \Lambda Y) = \mathbb{C}\left(\Lambda Y \cdot E_m(X) \oplus \Lambda Y \cdot E_m(X \oplus \Delta X)\right)$
- $\mathbb{C}(\lambda_\mathrm{o} \cdot \Delta C) = \sum_{\Delta X, \Lambda Y} \mathbb{P}(\Delta_\mathrm{i}, \Delta X) \cdot \mathbb{R}(\Delta X, \Lambda Y) \cdot \mathbb{C}^2(\Lambda Y, \lambda_\mathrm{o})$
- $\mathbb{P}(\Delta_\mathrm{i} \xrightarrow{E_u} \Delta_m) = p$
- $\mathbb{R}(\Delta_m, \lambda_m) = r$
- $\mathbb{C}(\lambda_m \xrightarrow{E_\ell} \lambda_\mathrm{o}) = q$
- $\mathbb{C}(\lambda_\mathrm{o} \cdot \Delta C) \approx prq^2$



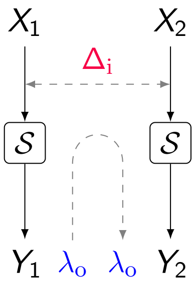
Hosein Hadipour, Patrick Derbez, Maria Eichlseder

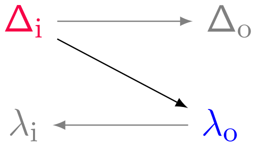
Sandwich Framework for DL Attack [BLN14; DKS14; Bar+19]

- $\mathbb{R}(\Delta X, \Lambda Y) = \mathbb{C}\left(\Lambda Y \cdot E_m(X) \oplus \Lambda Y \cdot E_m(X \oplus \Delta X)\right)$
- $\mathbb{C}(\lambda_o \cdot \Delta C) = \sum_{\Delta X, \Lambda Y} \mathbb{P}(\Delta_i, \Delta X) \cdot \mathbb{R}(\Delta X, \Lambda Y) \cdot \mathbb{C}^2(\Lambda Y, \lambda_o)$
- $\mathbb{P}(\Delta_i \xrightarrow{E_u} \Delta_m) = p$
- $\mathbb{R}(\Delta_m, \lambda_m) = r$
- $\mathbb{C}(\lambda_m \xrightarrow{E_\ell} \lambda_o) = q$
- $\mathbb{C}(\lambda_o \cdot \Delta C) \approx prq^2$



Differential-Linear Connectivity Table ($\mathtt{DLCT}$) [Bar+19]



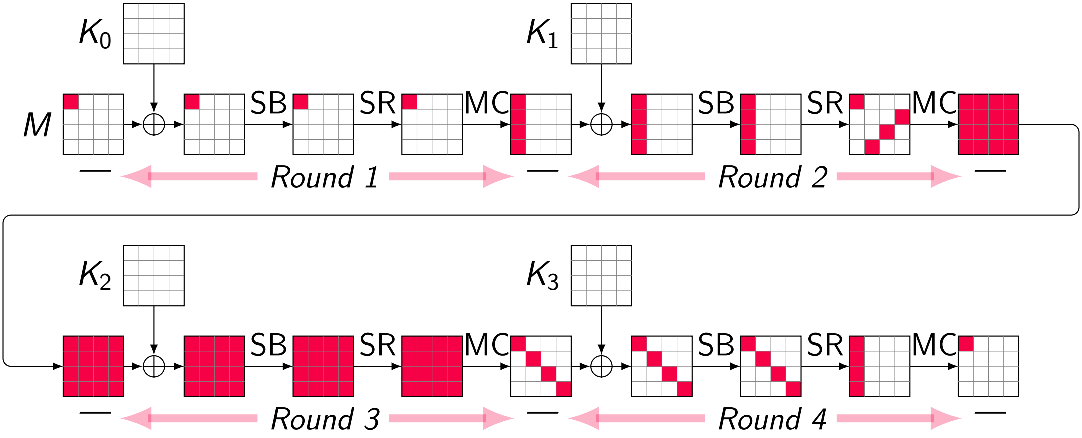


$$\mathtt{DLCT}_b(\Delta_i, \lambda_o) = \{x \in \mathbb{F}_2^n : \ \lambda_o \cdot S(x) \oplus \lambda_o \cdot S(x \oplus \Delta_i) = b\}$$

$$\mathtt{DLCT}(\Delta_i, \lambda_o) = |\mathtt{DLCT}_0(\Delta_i, \lambda_o)| - |\mathtt{DLCT}_1(\Delta_i, \lambda_o)|$$

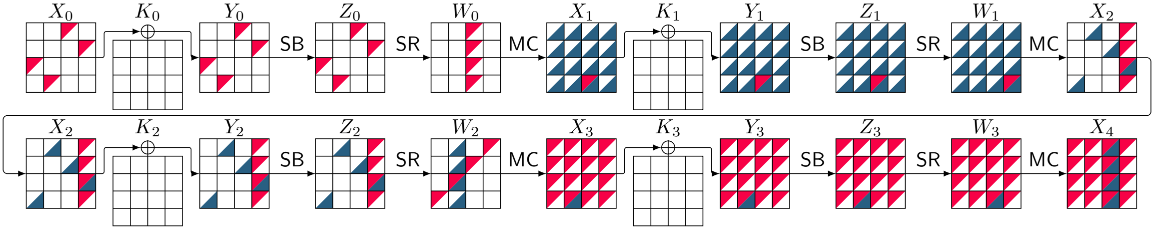
$$\mathbb{C}_{\mathtt{DLCT}}(\Delta_i, \lambda_o) = 2^{-n} \cdot \mathtt{DLCT}(\Delta_i, \lambda_o)$$

Security of AES Against Differential/Linear Attacks



$$\mathbb{P}_{4 \text{ rounds}} \leq 2^{-150}, \; \mathbb{C}^2_{4 \text{ rounds}} \leq 2^{-150}$$

A 4-round DL Distinguisher for AES



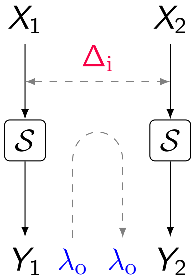
$r_u = 1, r_m = 3, r_\ell = 0,\ p = 2^{-24.00},\ r = 2^{-7.66}, q^2 = 1,\ \mathbb{C} = prq^2 = \mathbf{2^{-31.66}}$	
ΔX_0 `00005200000000f58f000000007b0000`	ΔX_1 `0000000000000000000000000000b400`
ΓX_4 `0032000000ab00000066000000980000`	-

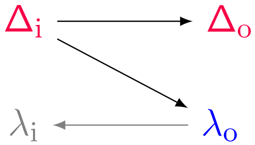
$\mathbf{2^{63.32}}$ v.s. 2^{150}

Hosein Hadipour, Patrick Derbez, Maria Eichlseder

Generalized DLCT Framework

Upper Differential-Linear Connectivity Table ($\mathtt{UDLCT}$)



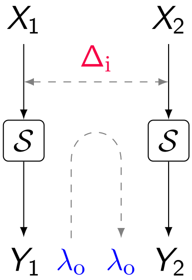


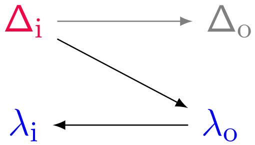
$$\mathtt{UDLCT}_b(\Delta_{\mathrm{i}}, \Delta_{\mathrm{o}}, \lambda_{\mathrm{o}}) = \{x \in \mathbb{F}_2^n : \ S(x) \oplus S(x \oplus \Delta_{\mathrm{i}}) = \Delta_{\mathrm{o}} \text{ and } \lambda_{\mathrm{o}} \cdot \Delta_{\mathrm{o}} = b\}$$

$$\mathtt{UDLCT}(\Delta_{\mathrm{i}}, \Delta_{\mathrm{o}}, \lambda_{\mathrm{o}}) = |\mathtt{UDLCT}_0(\Delta_{\mathrm{i}}, \Delta_{\mathrm{o}}, \lambda_{\mathrm{o}})| - |\mathtt{UDLCT}_1(\Delta_{\mathrm{i}}, \Delta_{\mathrm{o}}, \lambda_{\mathrm{o}})|$$

$$\mathbb{C}_{\mathtt{UDLCT}}\left(\Delta_{\mathrm{i}}, \Delta_{\mathrm{o}}, \lambda_{\mathrm{o}}\right) = 2^{-n} \cdot \mathtt{UDLCT}\left(\Delta_{\mathrm{i}}, \Delta_{\mathrm{o}}, \lambda_{\mathrm{o}}\right)$$

Lower Differential-Linear Connectivity Table ($\texttt{LDLCT}$)



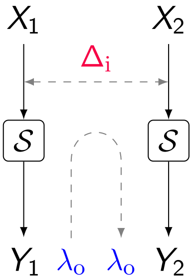


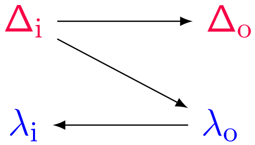
$$\texttt{LDLCT}_b(\Delta_\mathrm{i}, \lambda_\mathrm{i}, \lambda_\mathrm{o}) = \{x \in \mathbb{F}_2^n : \ \lambda_\mathrm{i} \cdot \Delta_\mathrm{i} \oplus \lambda_\mathrm{o} \cdot S(x) \oplus \lambda_\mathrm{o} \cdot S(x \oplus \Delta_\mathrm{i}) = b\}$$

$$\texttt{LDLCT}(\Delta_\mathrm{i}, \lambda_\mathrm{i}, \lambda_\mathrm{o}) = |\texttt{LDLCT}_0(\Delta_\mathrm{i}, \lambda_\mathrm{i}, \lambda_\mathrm{o})| - |\texttt{LDLCT}_1(\Delta_\mathrm{i}, \lambda_\mathrm{i}, \lambda_\mathrm{o})|$$

$$\mathbb{C}_{\texttt{LDLCT}}\left(\Delta_\mathrm{i}, \lambda_\mathrm{i}, \lambda_\mathrm{o}\right) = 2^{-n} \cdot \texttt{LDLCT}\left(\Delta_\mathrm{i}, \lambda_\mathrm{i}, \lambda_\mathrm{o}\right)$$

Extended Differential-Linear Connectivity Table ($\mathtt{EDLCT}$)





$$\mathtt{EDLCT}_b(\Delta_{\mathrm{i}}, \Delta_{\mathrm{o}}, \lambda_{\mathrm{i}}, \lambda_{\mathrm{o}}) = \{x \in \mathbb{F}_2^n : \; S(x) \oplus S(x \oplus \Delta_{\mathrm{i}}) = \Delta_{\mathrm{o}} \text{ and } \lambda_{\mathrm{i}} \cdot \Delta_{\mathrm{i}} \oplus \lambda_{\mathrm{o}} \cdot \Delta_{\mathrm{o}} = b\}$$

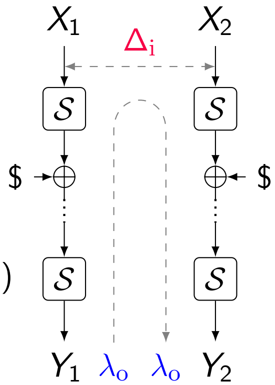
$$\mathtt{EDLCT}(\Delta_{\mathrm{i}}, \Delta_{\mathrm{o}}, \lambda_{\mathrm{i}}, \lambda_{\mathrm{o}}) = |\mathtt{EDLCT}_0(\Delta_{\mathrm{i}}, \Delta_{\mathrm{o}}, \lambda_{\mathrm{i}}, \lambda_{\mathrm{o}})| - |\mathtt{EDLCT}_1(\Delta_{\mathrm{i}}, \Delta_{\mathrm{o}}, \lambda_{\mathrm{i}}, \lambda_{\mathrm{o}})|$$

$$\mathbb{C}_{\mathtt{EDLCT}}(\Delta_{\mathrm{i}}, \Delta_{\mathrm{o}}, \lambda_{\mathrm{i}}, \lambda_{\mathrm{o}}) = 2^{-n} \cdot \mathtt{EDLCT}(\Delta_{\mathrm{i}}, \Delta_{\mathrm{o}}, \lambda_{\mathrm{i}}, \lambda_{\mathrm{o}})$$

Double Differential-Linear Connectivity Table (DDLCT)

$$
\begin{tikzcd}
\Delta_{\mathrm{i}} \arrow[r] \arrow[dr] & \Delta_m \arrow[r] \arrow[dr] & \Delta_{\mathrm{o}} \\
\lambda_{\mathrm{i}} \arrow[r] & \lambda_m \arrow[r] & \lambda_{\mathrm{o}}
\end{tikzcd}
$$

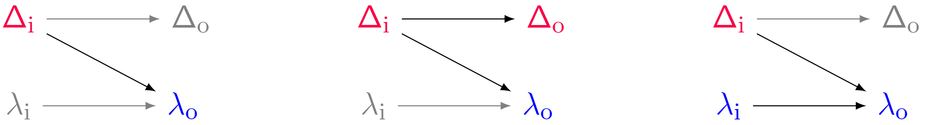
$$\mathtt{DDLCT}(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = 2^{-n} \cdot \sum_{\Delta_m} \sum_{\lambda_m} \mathtt{UDLCT}\left(\Delta_{\mathrm{i}}, \Delta_m, \lambda_m\right) \cdot \mathtt{LDLCT}\left(\Delta_m, \lambda_m, \lambda_{\mathrm{o}}\right)$$



Hosein Hadipour, Patrick Derbez, Maria Eichlseder

Generalized DLCT Framework (GBCT)

- How to formulate the correaltion for more than 1 round?

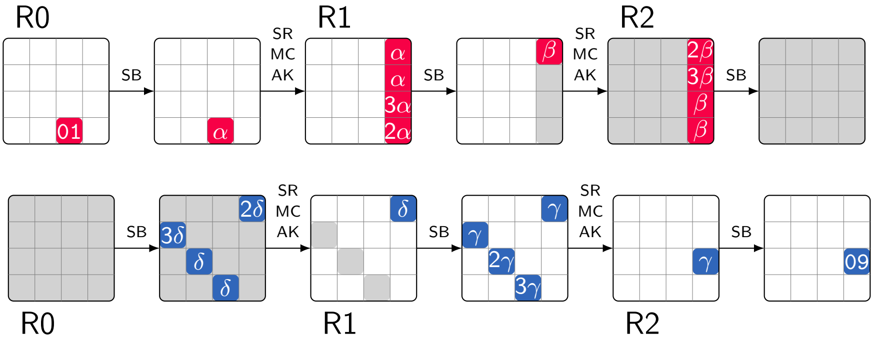


$\mathtt{DLCT}(\Delta_i, \lambda_o)$ $\mathtt{UDLCT}(\Delta_i, \Delta_o, \lambda_o)$ $\mathtt{LDLCT}(\Delta_i, \lambda_i, \lambda_o)$

$\mathtt{EDLCT}(\Delta_i, \Delta_o, \lambda_i, \lambda_o)$ $\mathtt{DDLCT}(\Delta_i, \lambda_o)$

Hosein Hadipour, Patrick Derbez, Maria Eichlseder
CRYPTO 2024 - California, USA

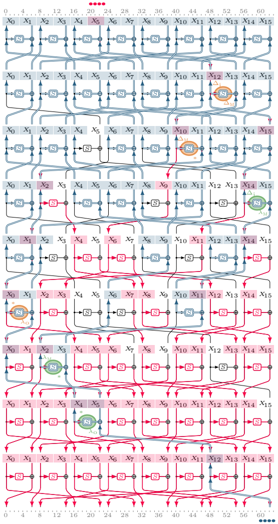
Application of the Generalized DLCT Tables - AES (– differential – linear)



$$\sum_{\alpha,\beta,\gamma,\delta} \mathbb{C}_{\mathtt{UDLCT}}(1,\alpha,\delta)\cdot\mathbb{C}_{\mathtt{EDLCT}}(\alpha,\beta,\delta,\gamma)\cdot\mathbb{C}_{\mathtt{LDLCT}}(\beta,\gamma,9) = -2^{-7.94}$$

Hosein Hadipour, Patrick Derbez, Maria Eichlseder

Application of the Generalized DLCT Tables - TWINE (- differential - linear)



$$\mathbb{C}(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = \sum_{\Delta_m} \mathbb{P}_{\mathtt{DDT}}(\Delta_{\mathrm{i}}, \Delta_m) \cdot \mathbb{C}_{\mathtt{DDLCT}}(\Delta_m, \lambda_{\mathrm{o}})$$

$$= \sum_{\lambda_m} \mathbb{C}_{\mathtt{DDLCT}}(\Delta_{\mathrm{i}}, \lambda_m) \cdot \mathbb{C}^2_{\mathtt{LAT}}(\lambda_m, \lambda_{\mathrm{o}}).$$

$$\mathbb{C}_{tot}(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = \mathbb{C}^2(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}).$$

Input/Output Differences/Linear-mask	Formula	Exp. Correlation
$(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = (\mathtt{0xb4}, \mathtt{0x67})$	$-2^{-7.66}$	$-2^{-7.64}$
$(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = (\mathtt{0x02}, \mathtt{0x02})$	$-2^{-7.92}$	$-2^{-7.93}$
$(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = (\mathtt{0x55}, \mathtt{0x55})$	$-2^{-7.99}$	$-2^{-7.98}$
$(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = (\mathtt{0xbf}, \mathtt{0xef})$	$-2^{-8.05}$	$-2^{-8.06}$
$(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = (\mathtt{0xfe}, \mathtt{0x06})$	$-2^{-8.26}$	$-2^{-8.25}$
$(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = (\mathtt{0x4b}, \mathtt{0x1a})$	$-2^{-8.43}$	$-2^{-8.44}$

Differential-Linear Switches and Deterministic Trails

Cell-Wise and Bit-Wise Switches

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

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

- Cell-wise switches: $\mathtt{DLCT}(\Delta_{\mathrm{i}}, 0) = \mathtt{DLCT}(0, \lambda_{\mathrm{o}}) = 2^n$ for all $\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}$
- Bit-wise switches: $\mathtt{DLCT}(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = \pm 2^n$ for $\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}} \neq 0$
  - Example: $\mathbb{C}(9, 4) = \frac{16}{16}$

Hosein Hadipour, Patrick Derbez, Maria Eichlseder

Cell-Wise and Bit-Wise Switches

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

$\Delta \backslash \lambda$	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
1	16	0	0	0	-16	0	0	0	0	0	0	0	0	0	0	0
2	16	-8	-8	0	0	0	8	-8	0	-8	0	8	0	0	0	0
3	16	0	-8	-8	0	-8	8	0	0	0	0	0	0	-8	0	8
4	16	0	-8	0	0	0	-8	0	-16	0	8	0	0	0	8	0
5	16	0	-8	0	0	0	-8	0	0	0	8	0	-16	0	8	0
6	16	-8	8	-8	0	0	-8	0	0	-8	0	0	0	0	0	8
7	16	0	8	0	0	-8	-8	-8	0	0	0	8	0	-8	0	0
8	16	0	0	0	-16	0	0	0	-16	0	0	0	16	0	0	0
9	16	-8	0	-8	16	-8	0	-8	0	8	0	-8	0	8	0	-8
a	16	0	0	8	0	8	0	0	0	0	-8	0	0	-8	-8	-8
b	16	8	0	0	0	0	0	8	0	-8	-8	-8	0	0	-8	0
c	16	0	0	-8	0	0	0	-8	16	0	0	-8	0	0	0	-8
d	16	-8	0	0	0	-8	0	0	0	8	0	0	-16	8	0	0
e	16	0	0	0	0	8	0	8	0	0	-8	-8	0	-8	-8	0
f	16	8	0	8	0	0	0	0	0	-8	-8	0	0	0	-8	-8

- Cell-wise switches: $\mathtt{DLCT}(\Delta_{\mathrm{i}}, 0) = \mathtt{DLCT}(0, \lambda_{\mathrm{o}}) = 2^n$ for all $\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}$
- Bit-wise switches: $\mathtt{DLCT}(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = \pm 2^n$ for $\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}} \neq 0$
  - Example: $\mathbb{C}(9, 4) = \frac{16}{16}$

Hosein Hadipour, Patrick Derbez, Maria Eichlseder

Cell-Wise and Bit-Wise Switches

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

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

- Cell-wise switches: $\mathtt{DLCT}(\Delta_{\mathrm{i}}, 0) = \mathtt{DLCT}(0, \lambda_{\mathrm{o}}) = 2^n$ for all $\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}$
- Bit-wise switches: $\mathtt{DLCT}(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = \pm 2^n$ for $\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}} \neq 0$
  - Example: $\mathbb{C}\,(9, 4) = \frac{16}{16}$

Hosein Hadipour, Patrick Derbez, Maria Eichlseder

Deterministic Bit-Wise Differential Trails (Forward)

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

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

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

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

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

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

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

Hosein Hadipour, Patrick Derbez, Maria Eichlseder

Deterministic Bit-Wise Linear Trails (Backward)

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

$$\lambda_{\mathrm{i}} = (1, ?, ?, 1) \xleftarrow{S} \lambda_{\mathrm{o}} = (0, 1, 0, 0)$$

$$\lambda_{\mathrm{i}} = (1, 1, ?, ?) \xleftarrow{S} \lambda_{\mathrm{o}} = (1, 0, 0, 0)$$

$$\lambda_{\mathrm{i}} = (0, ?, ?, ?) \xleftarrow{S} \lambda_{\mathrm{o}} = (1, 1, 0, 0)$$

Bit-Wise Switches and Deterministic 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

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

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

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

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

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

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

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

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

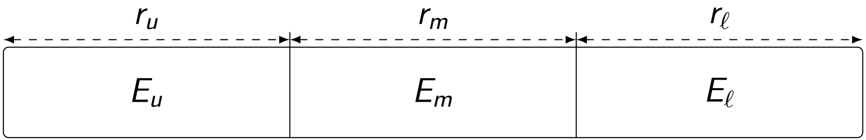
$$\lambda_i = (0,?,?,?) \xleftarrow{S} \lambda_o = (1,1,0,0)$$

Automatic Tools to Search for DL Distinguishers

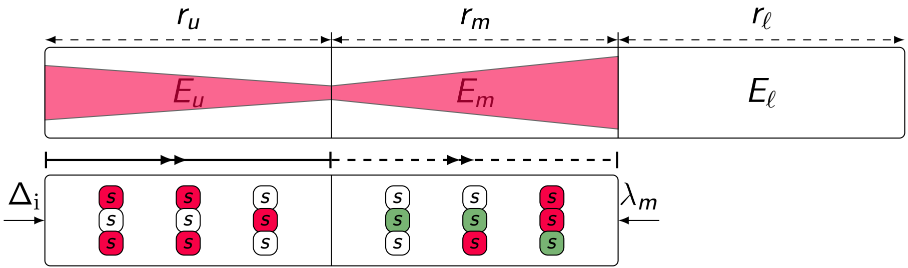
Overview of Our Method to Search for Distinguishers

$$\boxed{E}$$

Overview of Our Method to Search for Distinguishers

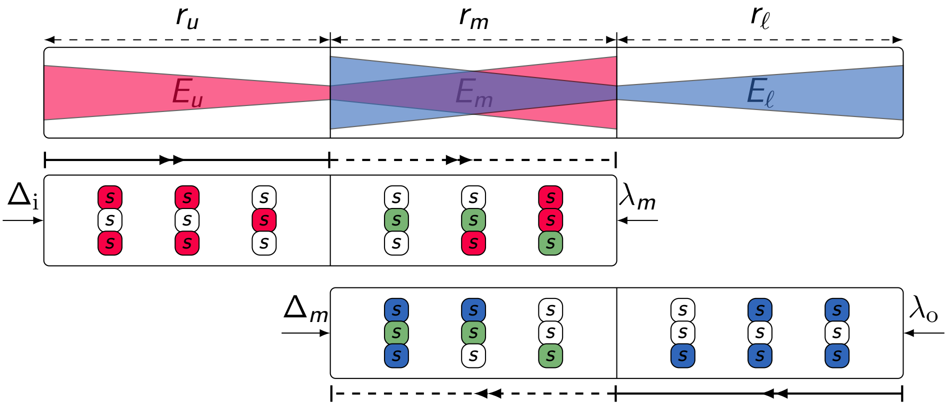


Overview of Our Method to Search for Distinguishers



differentially active S-box   linearly active S-box   common active S-box

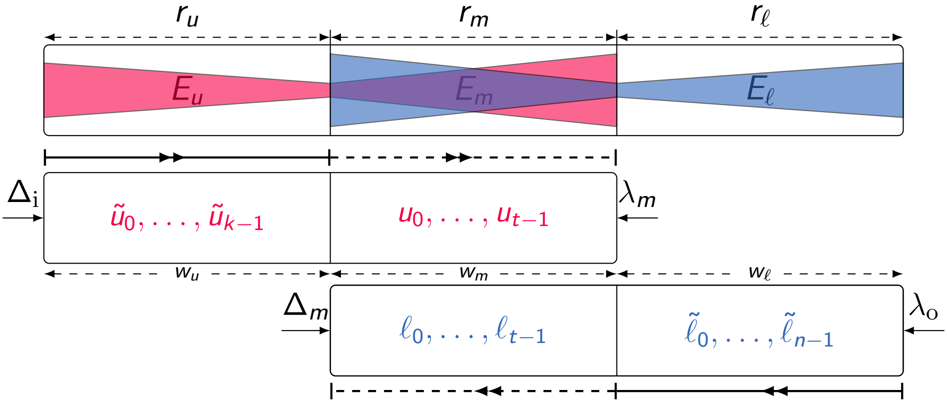
Overview of Our Method to Search for Distinguishers



differentially active S-box · linearly active S-box · common active S-box

Hosein Hadipour, Patrick Derbez, Maria Eichlseder

Overview of Our Method to Search for Distinguishers

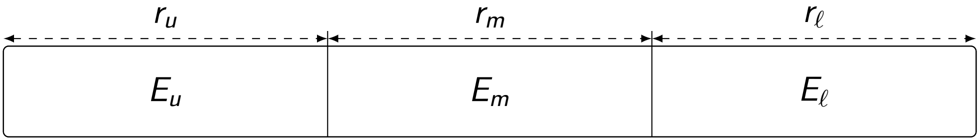


$$\min\left(\sum_{i=0}^{k-1} w_u \cdot \tilde{u}_i + \sum_{j=0}^{t-1} w_m \cdot \mathtt{bool2int}\left(\ell_j + u_j = 2\right) + \sum_{k=0}^{n-1} w_\ell \cdot \tilde{\ell}_k\right)$$

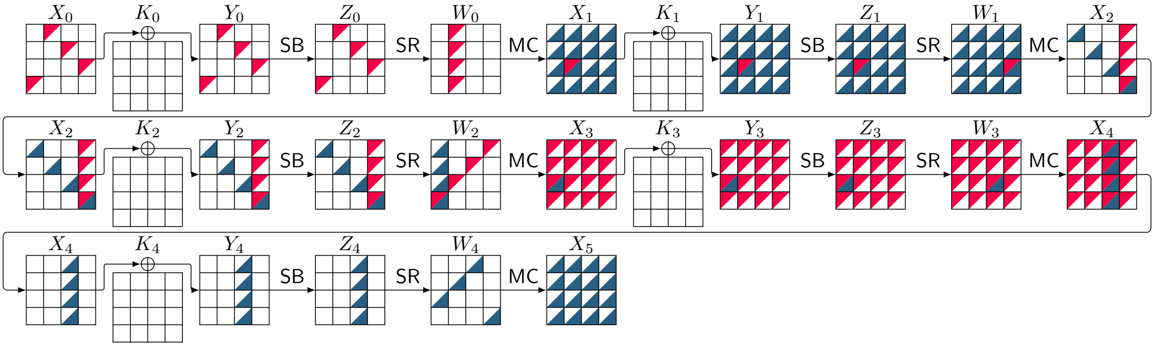
Hosein Hadipour, Patrick Derbez, Maria Eichlseder

Usage of Our Tool

```
python3 attack.py -RU 6 -RM 10 -RL 6
```



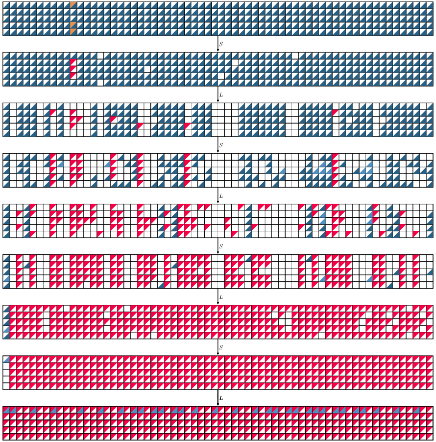
Example: A 5-round DL Distinguisher for AES



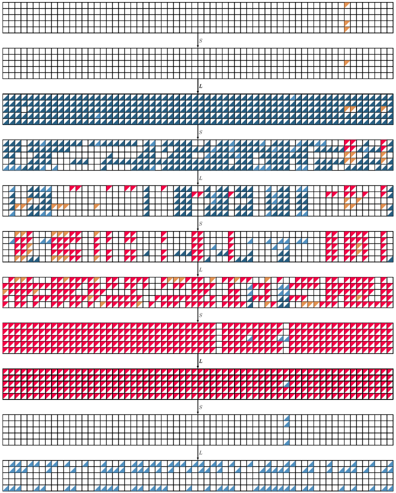
$$r_0 = 1, r_m = 3, r_1 = 1,\ \ p = 2^{-24.00}, r = 2^{-7.66},\ \ q^2 = 2^{-24.00},\ \ prq^2 = 2^{-55.66}$$

ΔX_0	`001c00000000e200000000dfb3000000`	ΔX_1	`000000000000000000f7000000000000`
ΓX_4	`00000000000000006700000000000000`	ΓX_5	`21d3814d93b1ef228e923507f67383fd`

Application to Ascon-p(active difference unknown difference active mask unknown mask)



$\mathbb{C} = 1$



$\mathbb{C} = 2^{-4.33}$

Contributions and Future Works

Contributions and Future Works

- Contributions
  - We generalized the DLCT framework from one S-box layer to multiple rounds
  - We proposed an automatic tool for finding optimum DL distinguishers
  - We applied our tool to almost any design paradigm
- Future works
  - Extending the application of our tool to other primitives, e.g., ARX
  - Extending our tool to a unified model for finding complete attack (key recovery)

: https://github.com/hadipourh/DL

: https://ia.cr/2024/255

Bibliography I

[Bar+19] Achiya Bar-On et al. **DLCT: A New Tool for Differential-Linear Cryptanalysis**. EUROCRYPT 2019. Vol. 11476. LNCS. Springer, 2019, pp. 313–342. DOI: 10.1007/978-3-030-17653-2_11.

[BLN14] Céline Blondeau, Gregor Leander, and Kaisa Nyberg. **Differential-Linear Cryptanalysis Revisited**. FSE 2014. Ed. by Carlos Cid and Christian Rechberger. Vol. 8540. LNCS. Springer, 2014, pp. 411–430. DOI: 10.1007/978-3-662-46706-0_21.

[DIK08] Orr Dunkelman, Sebastiaan Indesteege, and Nathan Keller. **A Differential-Linear Attack on 12-Round Serpent**. INDOCRYPT 2008. Ed. by Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das. Vol. 5365. LNCS. Springer, 2008, pp. 308–321. DOI: 10.1007/978-3-540-89754-5_24.

Bibliography II

[DKS14] Orr Dunkelman, Nathan Keller, and Adi Shamir. **A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony**. *J. Cryptol.* 27.4 (2014), pp. 824–849. DOI: 10.1007/s00145-013-9154-9.

[HNE22] Hosein Hadipour, Marcel Nageler, and Maria Eichlseder. **Throwing Boomerangs into Feistel Structures Application to CLEFIA, WARP, LBlock, LBlock-s and TWINE**. *IACR Trans. Symmetric Cryptol.* 2022.3 (2022), pp. 271–302. DOI: 10.46586/TOSC.V2022.I3.271-302.

[LH94] Susan K. Langford and Martin E. Hellman. **Differential-Linear Cryptanalysis**. CRYPTO '94. Vol. 839. Springer, 1994, pp. 17–25. DOI: 10.1007/3-540-48658-5_3.

Bibliography III

[ZWH24] Yanyan Zhou, Senpeng Wang, and Bin Hu. **MILP/MIQCP-Based Fully Automatic Method of Searching for Differential-Linear Distinguishers for SIMON-Like Ciphers**. *IET Information Security* 2024 (2024). DOI: 10.1049/2024/8315115.

Properties of Generalized DLCT Tables - I

- $\mathtt{DLCT}(\Delta_\mathrm{i}, \lambda_\mathrm{o}) = \sum_{\Delta_\mathrm{o}} \mathtt{UDLCT}(\Delta_\mathrm{i}, \Delta_\mathrm{o}, \lambda_\mathrm{o})$
- $\mathtt{UDLCT}(\Delta_\mathrm{i}, \Delta_\mathrm{o}, \lambda_\mathrm{o}) = (-1)^{\Delta_\mathrm{o} \cdot \lambda_\mathrm{o}} \mathtt{DDT}(\Delta_\mathrm{i}, \Delta_\mathrm{o})$
- $\mathtt{LDLCT}(\Delta_\mathrm{i}, \lambda_\mathrm{i}, \lambda_\mathrm{o}) = (-1)^{\Delta_\mathrm{i} \cdot \lambda_\mathrm{i}} \mathtt{DLCT}(\Delta_\mathrm{i}, \lambda_\mathrm{o})$
- $\mathtt{EDLCT}(\Delta_\mathrm{i}, \Delta_\mathrm{o}, \lambda_\mathrm{i}, \lambda_\mathrm{o}) = (-1)^{\lambda_\mathrm{i} \cdot \Delta_\mathrm{i} \oplus \lambda_\mathrm{o} \cdot \Delta_\mathrm{o}} \mathtt{DDT}(\Delta_\mathrm{i}, \Delta_\mathrm{o})$
- $\mathtt{LDLCT}(\Delta_\mathrm{i}, \lambda_\mathrm{i}, \lambda_\mathrm{o}) = \sum_{\Delta_\mathrm{o}} \mathtt{EDLCT}(\Delta_\mathrm{i}, \Delta_\mathrm{o}, \lambda_\mathrm{i}, \lambda_\mathrm{o})$
- $\sum_{\Delta_\mathrm{i}} \mathtt{LDLCT}(\Delta_\mathrm{i}, \lambda_\mathrm{i}, \lambda_\mathrm{o}) = \mathtt{LAT}^2(\lambda_\mathrm{i}, \lambda_\mathrm{o})$

Properties of Generalized $\texttt{DLCT}$ Tables - II

- $\texttt{DDLCT}(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) = 2^{-n} \cdot \sum_{\Delta_m} \sum_{\lambda_m} \texttt{UDLCT}\left(\Delta_{\mathrm{i}}, \Delta_m, \lambda_m\right) \cdot \texttt{LDLCT}\left(\Delta_m, \lambda_m, \lambda_{\mathrm{o}}\right)$

$$
\begin{aligned}
\texttt{DDLCT}(\Delta_{\mathrm{i}}, \lambda_{\mathrm{o}}) &= \sum_{\Delta_m} \texttt{DDT}(\Delta_{\mathrm{i}}, \Delta_m) \cdot \texttt{DLCT}(\Delta_m, \lambda_{\mathrm{o}}) \\
&= 2^{-n} \sum_{\lambda_m} \texttt{DLCT}(\Delta_{\mathrm{i}}, \lambda_m) \cdot \texttt{LAT}^2(\lambda_m, \lambda_{\mathrm{o}}).
\end{aligned}
$$

Example: Distinguishers for up to 17 Rounds of TWINE

- Comparing the data complexity of best boomerang and DL distinguishers

# Rounds	Boomerang [HNE22]	Differential-Linear	Gain
5	1	1	1
7	$2^{3.20}$	1	$2^{3.20}$
13	$2^{34.32}$	$2^{27.16}$	$2^{7.16}$
14	$2^{42.25}$	$2^{31.28}$	$2^{10.97}$
15	$2^{51.03}$	$2^{38.98}$	$2^{12.05}$
16	$2^{58.04}$	$2^{47.28}$	$2^{10.76}$
17	-	$2^{59.24}$	-

Example: Distinguishers for up to 17 Rounds of LBlock

- Comparing the data complexity of best boomerang and DL distinguishers

# Rounds	Boomerang [HNE22]	Differential-Linear	Gain
5	1	1	1
7	$2^{2.97}$	1	$2^{2.97}$
13	$2^{30.28}$	$2^{23.78}$	$2^{6.50}$
14	$2^{38.86}$	$2^{30.34}$	$2^{8.52}$
15	$2^{46.90}$	$2^{38.26}$	$2^{8.64}$
16	$2^{57.16}$	$2^{46.26}$	$2^{10.90}$
17	-	$2^{58.30}$	-

Example: Distinguishers for up to 8 Rounds of CLEFIA

- Comparing the data complexity of best boomerang and DL distinguishers

# Rounds	Boomerang [HNE22]	Differential-Linear	Gain
3	1	1	1
4	$2^{6.32}$	1	$2^{6.32}$
5	$2^{12.26}$	$2^{5.36}$	$2^{6.90}$
6	$2^{22.45}$	$2^{14.14}$	$2^{8.31}$
7	$2^{32.67}$	$2^{23.50}$	$2^{9.17}$
8	$2^{76.03}$	$2^{66.86}$	$2^{9.17}$

Application to SERPENT

- 💻: Experimentally verified

Cipher	#R	$\mathbb{C}$	💻	Ref.
SERPENT	3	$\mathbf{2^{-0.68}}$	✓	This work
	4	$2^{-12.75}$		[DIK08]
	4	$\mathbf{2^{-5.54}}$	✓	This work
	5	$2^{-16.75}$		[DIK08]
	5	$\mathbf{2^{-11.10}}$	✓	This work
	8	$2^{-39.18}$		This work
	9	$2^{-56.50}$		[DIK08]
	9	$\mathbf{2^{-50.95}}$		This work

Application to Simeck

- 💻: Experimentally verified

Cipher	#R	$\mathbb{C}$	💻	Ref.
Simeck-32	7	$\mathbf{1}$	✓	This work
	14	$2^{-16.63}$		[ZWH24]
	14	$\mathbf{2^{-13.92}}$	✓	This work

Cipher	#R	$\mathbb{C}$	💻	Ref.
Simeck-48	8	$\mathbf{1}$	✓	This work
	17	$2^{-22.37}$		[ZWH24]
	17	$\mathbf{2^{-13.89}}$	✓	This work
	18	$2^{-24.75}$		[ZWH24]
	18	$\mathbf{2^{-15.89}}$		This work
	19	$\mathbf{2^{-17.89}}$		This work
	20	$\mathbf{2^{-21.89}}$		This work

Cipher	#R	$\mathbb{C}$	💻	Ref.
Simeck-64	10	$\mathbf{1}$	✓	This work
	24	$2^{-38.13}$		[ZWH24]
	24	$\mathbf{2^{-25.14}}$		This work
	25	$2^{-41.04}$		[ZWH24]
	25	$\mathbf{2^{-27.14}}$		This work
	26	$\mathbf{2^{-30.35}}$		This work