

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

Lorentz Center, 22 April, 2024 - Leiden, Netherlands

Research Gap and Our Contributions



Research Gap

- 🕒 How to formulate the correlation for more than one S-box layer?
- 🕒 How to (efficiently) find good DL distinguishers?



Contributions

- ✅ Generalizing the DLCT framework [Bar+19] to handle multiple rounds.
- ✅ 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 formulate the correlation for more than one S-box layer?
- ❑ How to (efficiently) find good DL distinguishers?



Contributions

- ❑ Generalizing the DLCT framework [Bar+19] to handle multiple rounds.
- ❑ 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

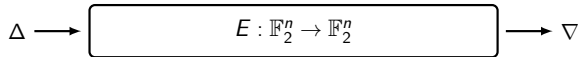
Outline

- 1 Boomerang Analysis
- 2 Differential-Linear Cryptanalysis
- 3 Generalized DLCT Framework
- 4 Differential-Linear Switches and Deterministic Trails
- 5 Automatic Tools to Search for DL Distinguishers
- 6 Contributions and Future Works

Boomerang Analysis

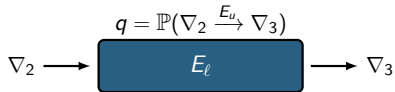
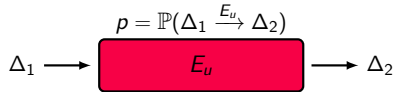


Boomerang Distinguishers [Wag99]

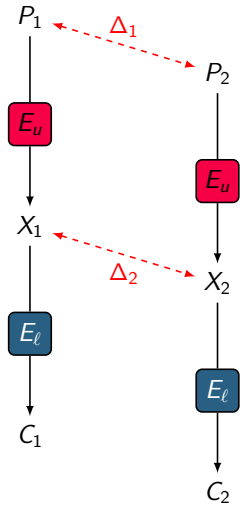
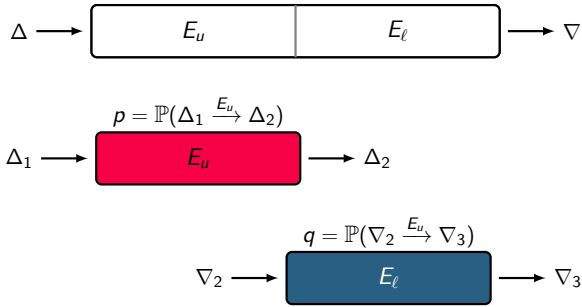


$$0 \leq \mathbb{P}(\Delta \xrightarrow{E} \nabla) \lll 2^{-n}$$

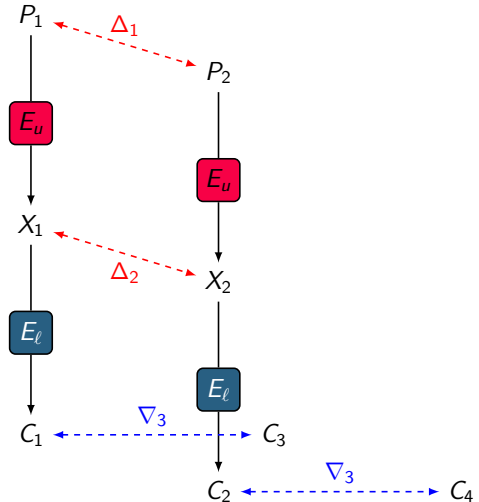
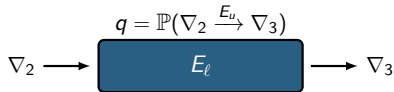
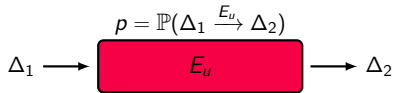
Boomerang Distinguishers [Wag99]



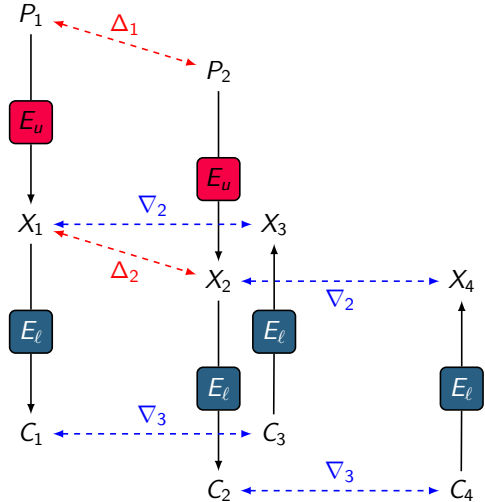
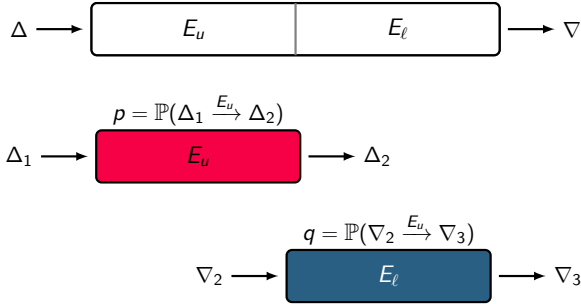
Boomerang Distinguishers [Wag99]



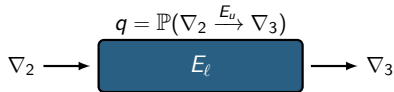
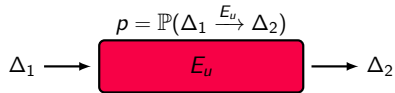
Boomerang Distinguishers [Wag99]



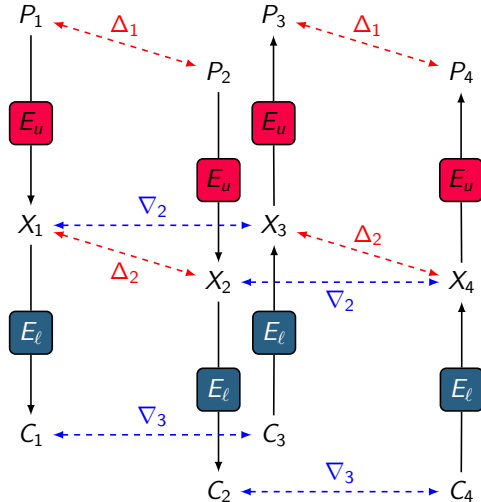
Boomerang Distinguishers [Wag99]



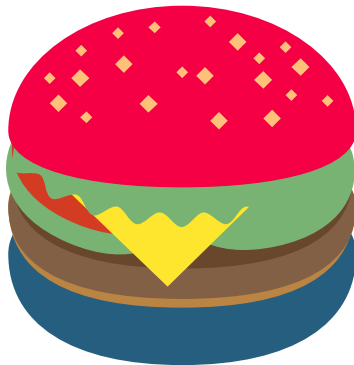
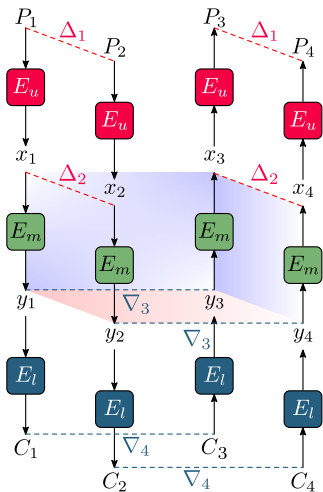
Boomerang Distinguishers [Wag99]



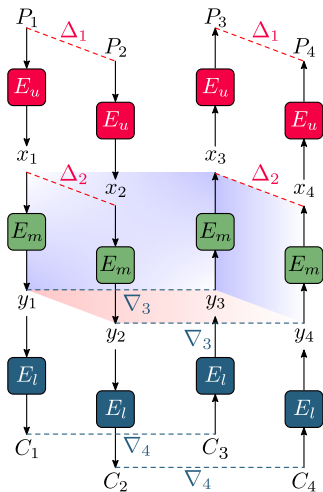
$$\mathbb{P}(P_3 \oplus P_4 = \Delta_1) = p^2 q^2$$



Sandwiching the Differentials! [DKS10; DKS14]



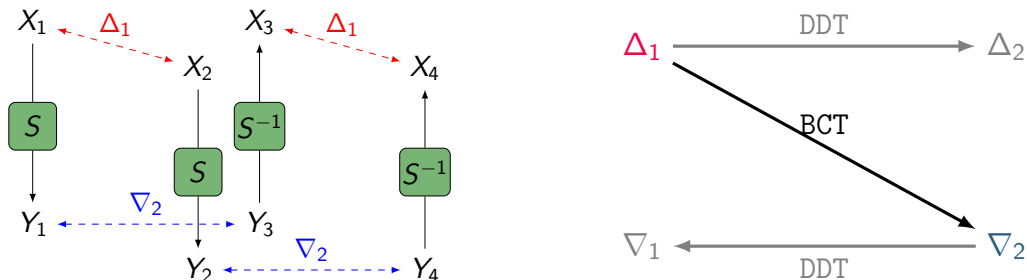
Sandwiching the Differentials! [DKS10; DKS14]



$$\mathbb{P}(P_3 \oplus P_4 = \Delta_1) \approx p^2 \times r \times q^2$$

$$r = \mathbb{P}(\Delta_2 \Leftrightarrow \nabla_3)$$

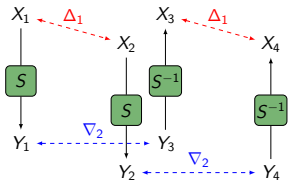
Boomerang Connectivity Table (BCT) [Cid+18]



$$\text{BCT}(\Delta_1, \nabla_2) := \#\{X \in \mathbb{F}_2^n \mid S^{-1}(S(X) \oplus \nabla_2) \oplus S^{-1}(S(X \oplus \Delta_1) \oplus \nabla_2) = \Delta_1\}$$

$$\mathbb{P}(\Delta_1 \rightleftharpoons \nabla_2) = 2^{-n} \cdot \text{BCT}(\Delta_1, \nabla_2)$$

Generalized BCT Framework (GBCT) - I

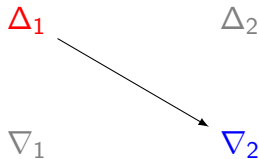
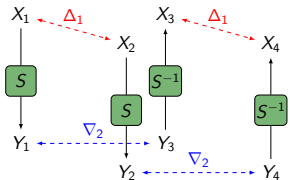


$$\Delta_1 \longrightarrow \Delta_2$$

$$\nabla_1 \longleftarrow \nabla_2$$

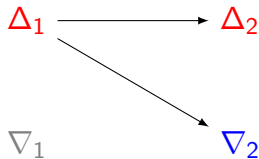
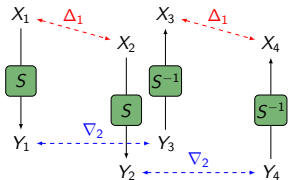
- ✓ $\mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}, \quad \text{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2)$
- ✓ $\mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) = \{x : S^{-1}(S(x) \oplus \nabla_2) \oplus S^{-1}(S(x \oplus \Delta_1) \oplus \nabla_2) = \Delta_1\}, \quad \text{BCT}(\Delta_1, \nabla_2) = \#\mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2)$
- ✓ $\text{UBCT}(\Delta_1, \Delta_2, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2)\} \quad [\text{WP19}]$
- ✓ $\text{LBCT}(\Delta_1, \nabla_1, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\nabla_1, \nabla_2)\} \quad [\text{DDV20; SQH19}]$
- ✓ $\text{EBCT}(\Delta_1, \Delta_2, \nabla_1, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2) \cap \mathcal{X}_{\text{DDT}}(\nabla_1, \nabla_2)\} \quad [\text{Bou+20; DDV20}]$

Generalized BCT Framework (GBCT) - I



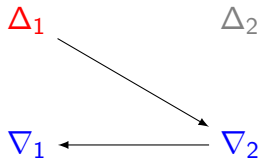
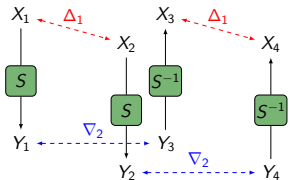
- ✓ $\mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}, \quad \text{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2)$
- ✓ $\mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) = \{x : S^{-1}(S(x) \oplus \nabla_2) \oplus S^{-1}(S(x \oplus \Delta_1) \oplus \nabla_2) = \Delta_1\}, \quad \text{BCT}(\Delta_1, \nabla_2) = \#\mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2)$
- ✓ $\text{UBCT}(\Delta_1, \Delta_2, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2)\} \quad [\text{WP19}]$
- ✓ $\text{LBCT}(\Delta_1, \nabla_1, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\nabla_1, \nabla_2)\} \quad [\text{DDV20; SQH19}]$
- ✓ $\text{EBCT}(\Delta_1, \Delta_2, \nabla_1, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2) \cap \mathcal{X}_{\text{DDT}}(\nabla_1, \nabla_2)\} \quad [\text{Bou+20; DDV20}]$

Generalized BCT Framework (GBCT) - I



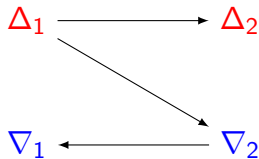
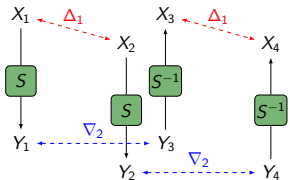
- ✓ $\mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}$, $\text{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2)$
- ✓ $\mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) = \{x : S^{-1}(S(x) \oplus \nabla_2) \oplus S^{-1}(S(x \oplus \Delta_1) \oplus \nabla_2) = \Delta_1\}$, $\text{BCT}(\Delta_1, \nabla_2) = \#\mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2)$
- ✓ $\text{UBCT}(\Delta_1, \Delta_2, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2)\}$ [WP19]
- ✓ $\text{LBCT}(\Delta_1, \nabla_1, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\nabla_1, \nabla_2)\}$ [DDV20; SQH19]
- ✓ $\text{EBCT}(\Delta_1, \Delta_2, \nabla_1, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2) \cap \mathcal{X}_{\text{DDT}}(\nabla_1, \nabla_2)\}$ [Bou+20; DDV20]

Generalized BCT Framework (GBCT) - I



- ✓ $\mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}$, $\text{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2)$
- ✓ $\mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) = \{x : S^{-1}(S(x) \oplus \nabla_2) \oplus S^{-1}(S(x \oplus \Delta_1) \oplus \nabla_2) = \Delta_1\}$, $\text{BCT}(\Delta_1, \nabla_2) = \#\mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2)$
- ✓ $\text{UBCT}(\Delta_1, \Delta_2, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2)\}$ [WP19]
- ✓ $\text{LBCT}(\Delta_1, \nabla_1, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\nabla_1, \nabla_2)\}$ [DDV20; SQH19]
- ✓ $\text{EBCT}(\Delta_1, \Delta_2, \nabla_1, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2) \cap \mathcal{X}_{\text{DDT}}(\nabla_1, \nabla_2)\}$ [Bou+20; DDV20]

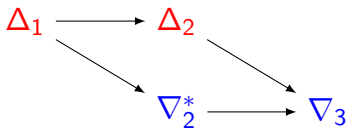
Generalized BCT Framework (GBCT) - I



- ✓ $\mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}$, $\text{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2)$
- ✓ $\mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) = \{x : S^{-1}(S(x) \oplus \nabla_2) \oplus S^{-1}(S(x \oplus \Delta_1) \oplus \nabla_2) = \Delta_1\}$, $\text{BCT}(\Delta_1, \nabla_2) = \#\mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2)$
- ✓ $\text{UBCT}(\Delta_1, \Delta_2, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2)\}$ [WP19]
- ✓ $\text{LBCT}(\Delta_1, \nabla_1, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\nabla_1, \nabla_2)\}$ [DDV20; SQH19]
- ✓ $\text{EBCT}(\Delta_1, \Delta_2, \nabla_1, \nabla_2) = \#\{x : x \in \mathcal{X}_{\text{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\text{DDT}}(\Delta_1, \Delta_2) \cap \mathcal{X}_{\text{DDT}}(\nabla_1, \nabla_2)\}$ [Bou+20; DDV20]

Generalized BCT Framework (GBCT) - II

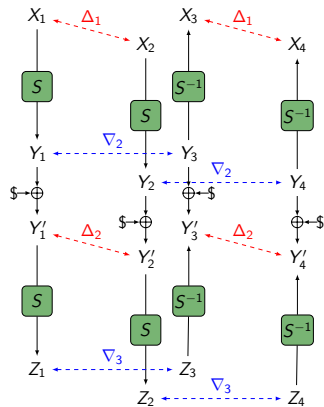
■ Double Boomerang Connectivity Table (DBCT) [HB21]



✔ $\text{DBCT}^+(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \text{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \text{LBCT}(\Delta_2, \nabla_2, \nabla_3)$

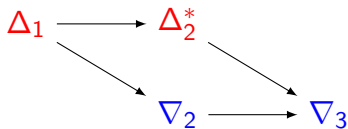
✔ $\text{DBCT}^-(\Delta_1, \nabla_2, \nabla_3) = \sum_{\Delta_2} \text{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \text{LBCT}(\Delta_2, \nabla_2, \nabla_3)$.

✔ $\text{DBCT}(\Delta_1, \nabla_3) = \sum_{\Delta_2} \text{DBCT}^+(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \text{DBCT}^-(\Delta_1, \nabla_2, \nabla_3)$.



Generalized BCT Framework (GBCT) - II

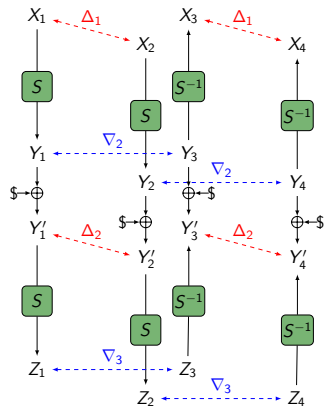
Double Boomerang Connectivity Table (DBCT) [HB21]



$$\text{DBCT}^{\perp}(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \text{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \text{LBCT}(\Delta_2, \nabla_2, \nabla_3)$$

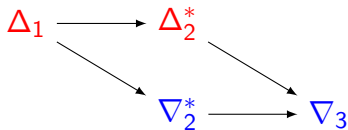
$$\text{DBCT}^{-1}(\Delta_1, \nabla_2, \nabla_3) = \sum_{\Delta_2} \text{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \text{LBCT}(\Delta_2, \nabla_2, \nabla_3).$$

$$\text{DBCT}(\Delta_1, \nabla_3) = \sum_{\Delta_2} \text{DBCT}^{\perp}(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \text{DBCT}^{-1}(\Delta_1, \nabla_2, \nabla_3).$$



Generalized BCT Framework (GBCT) - II

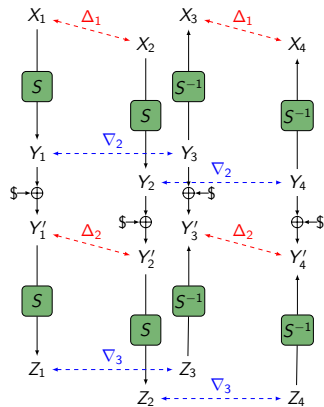
■ Double Boomerang Connectivity Table (DBCT) [HB21]



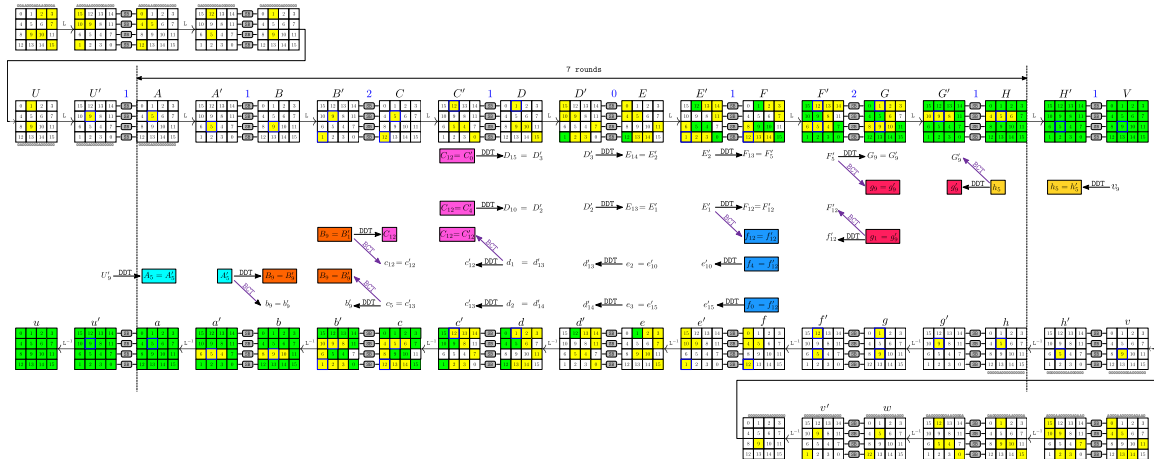
$$\textcircled{\checkmark} \text{DBCT}^\perp(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \text{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \text{LBCT}(\Delta_2, \nabla_2, \nabla_3)$$

$$\textcircled{\checkmark} \text{DBCT}^\perp(\Delta_1, \nabla_2, \nabla_3) = \sum_{\Delta_2} \text{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \text{LBCT}(\Delta_2, \nabla_2, \nabla_3).$$

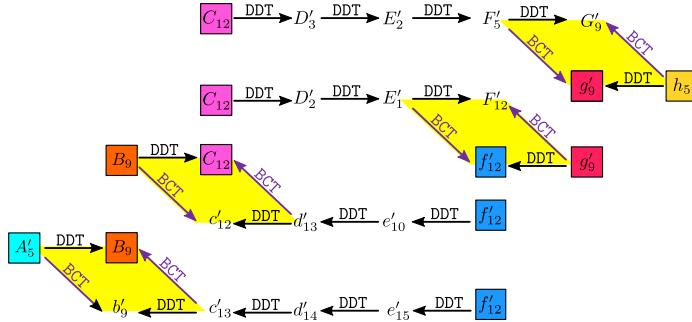
$$\textcircled{\checkmark} \text{DBCT}(\Delta_1, \nabla_3) = \sum_{\Delta_2} \text{DBCT}^\perp(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \text{DBCT}^\perp(\Delta_1, \nabla_2, \nabla_3).$$



Application of GBCT [HB21]



Application of GBCT [HB21]



$$\text{DBCT}_{\text{total}} = \text{DBCT}^{\perp}(A_5, B_9, c_5) \cdot \text{DBCT}^{\perp}(B_9, C_{12}, d_1) \cdot \text{DBCT}^{\perp}(E'_1, f'_{12}, g'_9) \cdot \text{DBCT}^{\perp}(F'_5, g'_9, h_5)$$

$$\text{Pr}_{\text{total}} = \Pr(d_1 \xleftarrow{2 \text{ DDT}} f'_{12}) \cdot \Pr(c_5 \xleftarrow{3 \text{ DDT}} f'_{12}) \cdot \Pr(C_{12} \xrightarrow{2 \text{ DDT}} E'_1) \cdot \Pr(C_{12} \xrightarrow{3 \text{ DDT}} F'_5)$$

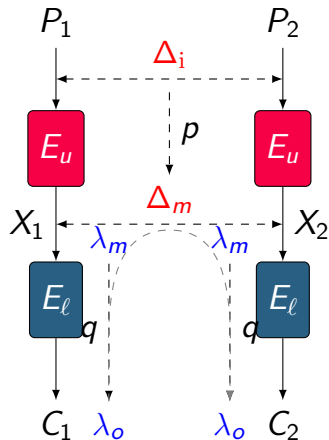
$$r = 2^{-8 \cdot n} \cdot \sum_{B_9} \sum_{C_{12}} \sum_{g'_9} \sum_{f'_{12}} \sum_{c_5} \sum_{d_1} \sum_{E'_1} \sum_{F'_5} \text{DBCT}_{\text{total}} \cdot \text{Pr}_{\text{total}}.$$

Differential-Linear Cryptanalysis



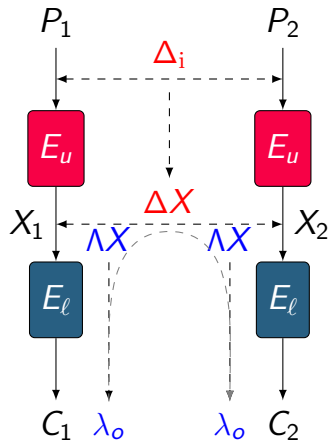
Differential-Linear (DL) Attack [LH94]

- $\mathbb{P}(\Delta_i \xrightarrow{E_u} \Delta_m) = p$
- $\mathbb{C}(\lambda_m \xrightarrow{E_\ell} \lambda_o) = q$
- Assumptions ($\Delta X = X_1 \oplus X_2$):
 1. E_u , and E_ℓ are statistically independent
 2. $\mathbb{P}(\lambda_m \cdot \Delta X = 0) = 1/2$ when $\Delta X \neq \Delta_m$
- $\mathbb{C}(\lambda_o \cdot C_1 \oplus \lambda_o \cdot C_2) = (-1)^{\lambda_m \cdot \Delta_m} \cdot pq^2 = \pm pq^2$



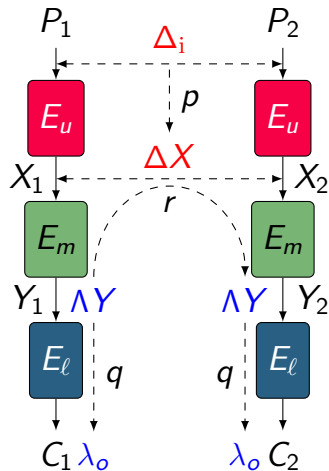
Differential-Linear Attack Revisited [BLN14; BLN17]

- $\mathbb{C}(\Lambda X \xrightarrow{E_\ell} \lambda_o) = \mathbb{C}(\Lambda X, \lambda_o)$
- Assumptions:
 1. E_u , and E_ℓ are statistically independent
- $\mathbb{C}(\lambda_o \cdot C_1 \oplus \lambda_o \cdot C_2) = \sum_{\Delta X, \Lambda X} \mathbb{C}(\Lambda X \cdot \Delta X) \cdot \mathbb{C}^2(\Lambda X, \lambda_o)$



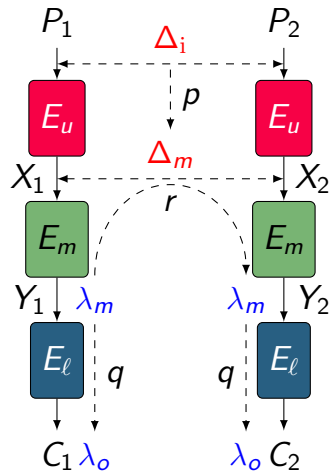
Sandwich Framework for DL Attack [DKS14; Bar+19]

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



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

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



Differential-Linear Connectivity Table (DLCT)

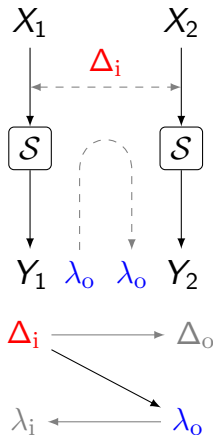
Differential-Linear Connectivity Table (DLCT) [Bar+19]

For a vectorial Boolean function $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$, the DLCT of S is a $2^n \times 2^m$ table whose rows correspond to the input difference Δ_i to S and whose columns correspond to the output mask λ_o of S . The entry at index (Δ_i, λ_o) is

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

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

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



Generalized DLCT Framework



Upper Differential-Linear Connectivity Table (UDLCT)

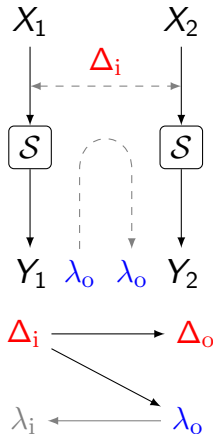
Upper Differential-Linear Connectivity Table (UDLCT)

For a vectorial Boolean function $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$, the UDLCT of S is a $2^n \times 2^n \times 2^m$ table. The entry at index $(\Delta_i, \Delta_o, \lambda_o)$ is

$$\text{UDLCT}(\Delta_i, \Delta_o, \lambda_o) = |\text{UDLCT}_0(\Delta_i, \Delta_o, \lambda_o)| - |\text{UDLCT}_1(\Delta_i, \Delta_o, \lambda_o)|,$$

where $\text{UDLCT}_b(\Delta_i, \Delta_o, \lambda_o) = \{x \in \mathbb{F}_2^n : S(x) \oplus S(x \oplus \Delta_i) = \Delta_o \text{ and } \lambda_o \cdot \Delta_o = b\}$.

$$\mathbb{C}_{\text{UDLCT}}(\Delta_i, \Delta_o, \lambda_o) = 2^{-n} \cdot \text{UDLCT}(\Delta_i, \Delta_o, \lambda_o)$$



Lower Differential-Linear Connectivity Table (LDLCT)

Lower Differential-Linear Connectivity Table (LDLCT)

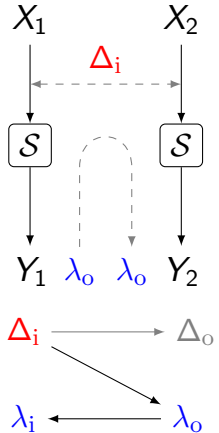
For a vectorial Boolean function $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$, the LDLCT of S is a $2^n \times 2^m \times 2^m$ table. The entry at index $(\Delta_i, \lambda_i, \lambda_o)$ is

$$\text{LDLCT}(\Delta_i, \lambda_i, \lambda_o) = |\text{LDLCT}_0(\Delta_i, \lambda_i, \lambda_o)| - |\text{LDLCT}_1(\Delta_i, \lambda_i, \lambda_o)|,$$

where

$$\text{LDLCT}_b(\Delta_i, \lambda_i, \lambda_o) = \{x \in \mathbb{F}_2^n : \lambda_i \cdot \Delta_i \oplus \lambda_o \cdot S(x) \oplus \lambda_o \cdot S(x \oplus \Delta_i) = b\}.$$

$$\mathbb{C}_{\text{LDLCT}}(\Delta_i, \lambda_i, \lambda_o) = 2^{-n} \cdot \text{LDLCT}(\Delta_i, \lambda_i, \lambda_o)$$



Extended Differential-Linear Connectivity Table (EDLCT)

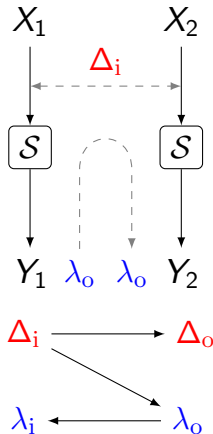
Extended Differential-Linear Connectivity Table (EDLCT)

For a vectorial Boolean function $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$, the EDLCT of S is a $2^n \times 2^n \times 2^m \times 2^m$ table. The entry at index $(\Delta_i, \Delta_o, \lambda_i, \lambda_o)$ is

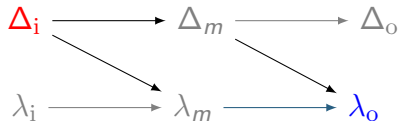
$$\text{EDLCT}(\Delta_i, \Delta_o, \lambda_i, \lambda_o) = |\text{EDLCT}_0(\Delta_i, \Delta_o, \lambda_i, \lambda_o)| - |\text{EDLCT}_1(\Delta_i, \Delta_o, \lambda_i, \lambda_o)|,$$

where $\text{EDLCT}_b(\Delta_i, \Delta_o, \lambda_i, \lambda_o) = \{x \in \mathbb{F}_2^n : S(x) \oplus S(x \oplus \Delta_i) = \Delta_o \text{ and } \lambda_i \cdot \Delta_i \oplus \lambda_o \cdot \Delta_o = b\}$.

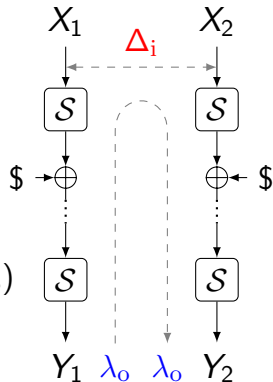
$$\mathbb{C}_{\text{EDLCT}}(\Delta_i, \Delta_o, \lambda_i, \lambda_o) = 2^{-n} \cdot \text{EDLCT}(\Delta_i, \Delta_o, \lambda_i, \lambda_o)$$



Double Differential-Linear Connectivity Table (DDLCT)

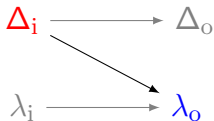


$$\text{DDLCT}(\Delta_i, \lambda_o) = 2^{-n} \sum_{\Delta_m} \sum_{\lambda_m} \text{UDLCT}(\Delta_i, \Delta_m, \lambda_m) \cdot \text{LDLCT}(\Delta_m, \lambda_m, \lambda_o)$$

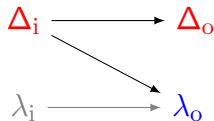


Generalized DLCT Framework (GBCT)

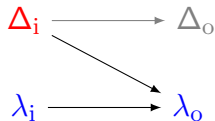
- How to formulate the correlation for more than 1 round?



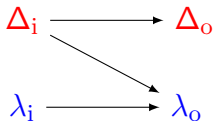
DLCT (Δ_i, λ_o)



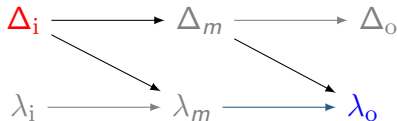
UDLCT ($\Delta_i, \Delta_o, \lambda_o$)



LDLCT ($\Delta_i, \lambda_i, \lambda_o$)

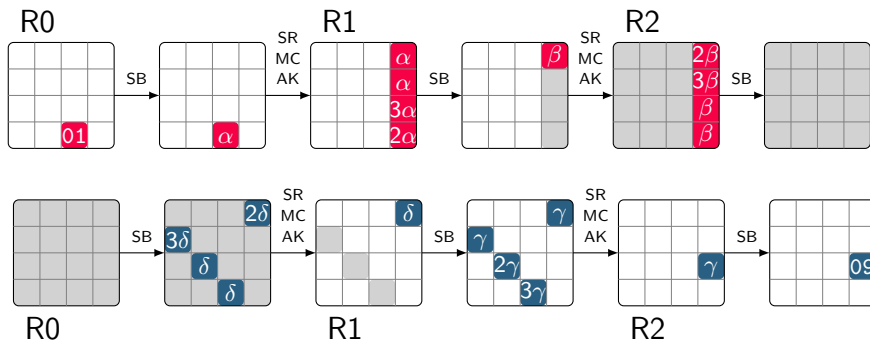


EDLCT ($\Delta_i, \Delta_o, \lambda_i, \lambda_o$)



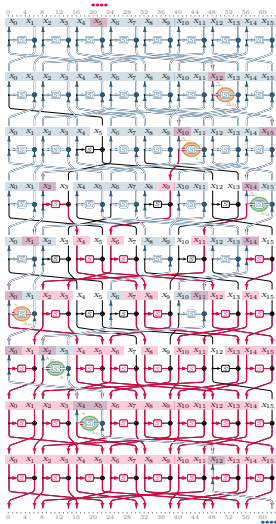
DDLCT (Δ_i, λ_o)

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



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

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



$$\begin{aligned}\mathbb{C}(\Delta_i, \lambda_o) &= \sum_{\Delta_m} \mathbb{P}_{\text{DDT}}(\Delta_i, \Delta_m) \cdot \mathbb{C}_{\text{DDLCT}}(\Delta_m, \lambda_o) \\ &= \sum_{\lambda_m} \mathbb{C}_{\text{DDLCT}}(\Delta_i, \lambda_m) \cdot \mathbb{C}_{\text{LAT}}^2(\lambda_m, \lambda_o).\end{aligned}$$

$$\mathbb{C}_{\text{tot}}(\Delta_i, \lambda_o) = \mathbb{C}^2(\Delta_i, \lambda_o).$$

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

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
$S(x)$	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:
 $DLCT(\Delta_i, 0) = DLCT(0, \lambda_o) = 2^n$ for all Δ_i, λ_o

Bit-wise switches:
 $DLCT(\Delta_i, \lambda_o) = \pm 2^n$ for $\Delta_i, \lambda_o \neq 0$

Example: $C(9, 4) = \frac{16}{16}$

Cell-Wise and Bit-Wise Switches

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	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:
 $DLCT(\Delta_i, 0) = DLCT(0, \lambda_o) = 2^n$ for all Δ_i, λ_o

- Bit-wise switches:
 $DLCT(\Delta_i, \lambda_o) = \pm 2^n$ for $\Delta_i, \lambda_o \neq 0$

Example: $C(9, 4) = \frac{16}{16}$

Cell-Wise and Bit-Wise Switches

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	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:
 $\text{DLCT}(\Delta_i, 0) = \text{DLCT}(0, \lambda_o) = 2^n$ for all Δ_i, λ_o
- Bit-wise switches:
 $\text{DLCT}(\Delta_i, \lambda_o) = \pm 2^n$ for $\Delta_i, \lambda_o \neq 0$
 - Example: $\mathbb{C}(9, 4) = \frac{16}{16}$

Deterministic Bit-Wise Differential Trails (Forward)

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
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

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

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

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

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

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

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

Deterministic Bit-Wise Linear Trails (Backward)

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
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

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

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

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

Bit-Wise Switches and Deterministic Trails

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	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

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

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

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

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

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

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

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

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

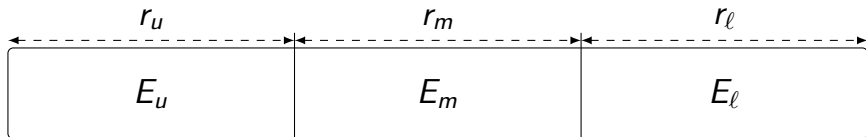
Automatic Tools to Search for DL Distinguishers



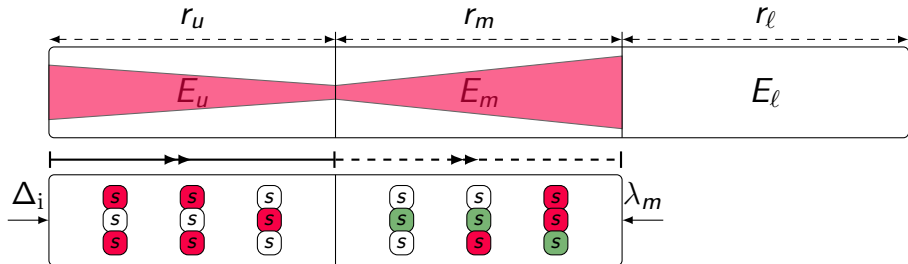
Overview of Our Method to Search for Distinguishers

E

Overview of Our Method to Search for Distinguishers

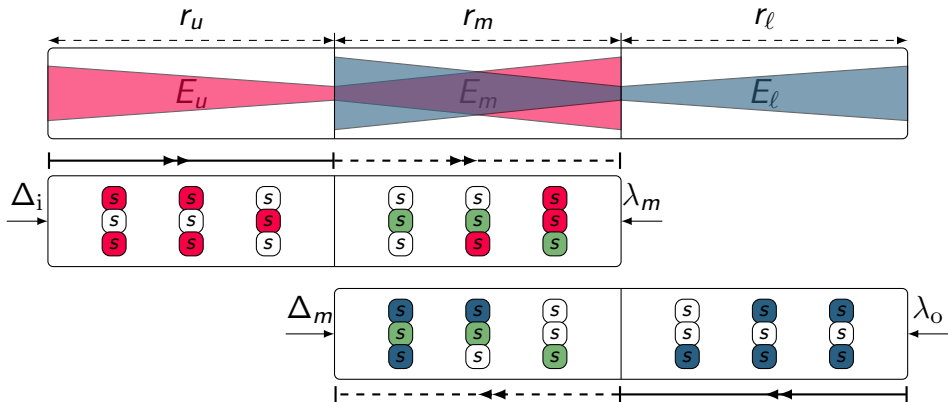


Overview of Our Method to Search for Distinguishers



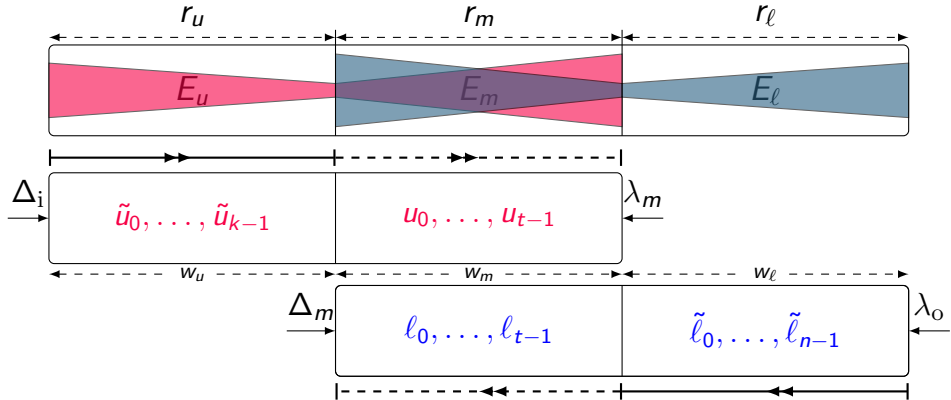
■ differentially active S-box ■ linearly active S-box ■ common active S-box

Overview of Our Method to Search for Distinguishers



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

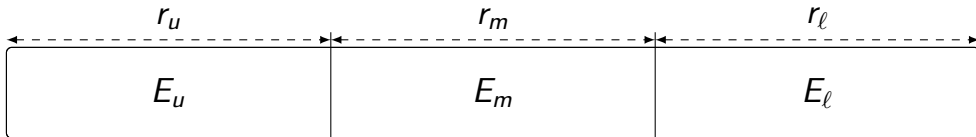
Overview of Our Method to Search for Distinguishers



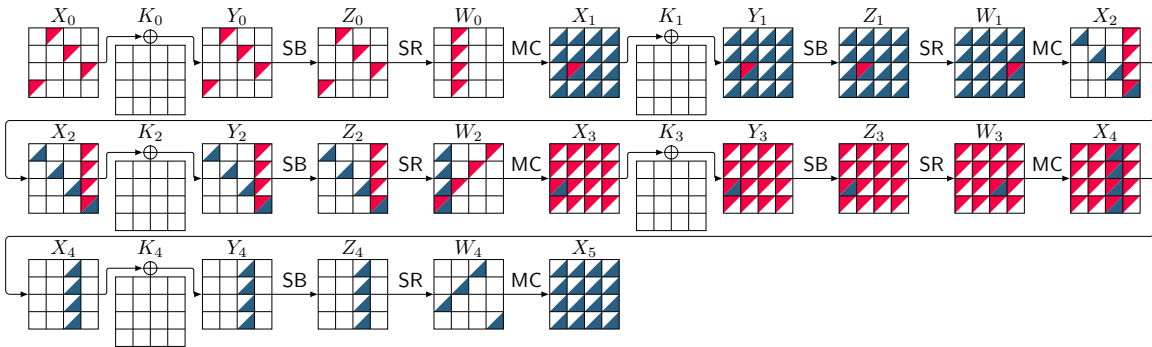
$$\min \left(\sum_{i=0}^{k-1} w_u \cdot \tilde{u}_i + \sum_{j=0}^{t-1} w_m \cdot \text{bool2int}(\ell_j + u_j = 2) + \sum_{k=0}^{n-1} w_\ell \cdot \tilde{\ell}_k \right)$$

Usage of Our Tool

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



Example: A 5-round DL Distinguisher for AES



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

$$\Delta X_0 \text{ 001c00000000e200000000dfb3000000} \quad \Delta X_1 \text{ 000000000000000000f7000000000000}$$

ΓX_4 00000000000000000670000000000000 ΓX_5 21d3814d93b1ef228e923507f67383fd

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

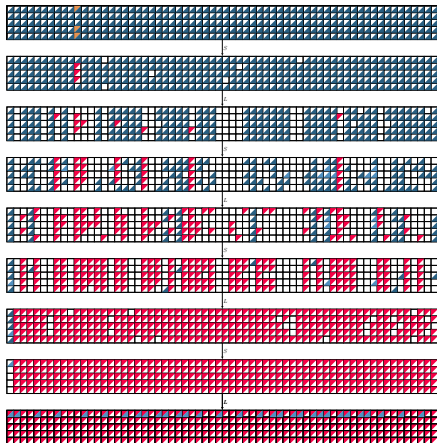
# Rounds	Boomerang [HNE22]	Differential-Linear	Gain
5	1	1	1
7	$2^{2.97}$	1	$2^{2.97}$
13	$2^{30.28}$	$2^{23.78}$	$2^{6.50}$
14	$2^{38.86}$	$2^{30.34}$	$2^{8.52}$
15	$2^{46.90}$	$2^{38.26}$	$2^{8.64}$
16	$2^{57.16}$	$2^{46.26}$	$2^{10.90}$
17	-	$2^{58.30}$	-

Example: Distinguishers for up to 8 Rounds of CLEFIA

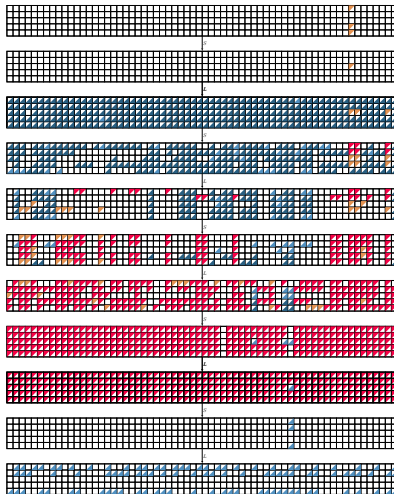
- Comparing the data complexity of best boomerang and DL distinguishers

# 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 Ascon-p(active difference unknown difference active mask unknown mask)




$C = 1$




$C = 2^{-4.33}$


Application to SERPENT


- : Experimentally verified


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

Application to Simeck

- : Experimentally verified

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

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

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

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://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](https://doi.org/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](https://doi.org/10.1007/978-3-662-46706-0_21).
- [BLN17] Céline Blondeau, Gregor Leander, and Kaisa Nyberg. **Differential-Linear Cryptanalysis Revisited**. *J. Cryptol.* 30.3 (2017), pp. 859–888. DOI: [10.1007/s00145-016-9237-5](https://doi.org/10.1007/s00145-016-9237-5).

Bibliography II

- [Bou+20] Hamid Boukerrou et al. **On the Feistel Counterpart of the Boomerang Connectivity Table Introduction and Analysis of the FBCT**. *IACR Trans. Symmetric Cryptol.* 2020.1 (2020), pp. 331–362. DOI: [10.13154/TOSC.V2020.I1.331-362](https://doi.org/10.13154/TOSC.V2020.I1.331-362).
- [Cid+18] Carlos Cid et al. **Boomerang Connectivity Table: A New Cryptanalysis Tool**. EUROCRYPT 2018. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10821. LNCS. Springer, 2018, pp. 683–714. DOI: [10.1007/978-3-319-78375-8_22](https://doi.org/10.1007/978-3-319-78375-8_22).
- [DDV20] Stéphanie Delaune, Patrick Derbez, and Mathieu Vavrille. **Catching the Fastest Boomerangs Application to SKINNY**. *IACR Trans. Symmetric Cryptol.* 2020.4 (2020), pp. 104–129. DOI: [10.46586/TOSC.V2020.I4.104-129](https://doi.org/10.46586/TOSC.V2020.I4.104-129).

Bibliography III

- [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](https://doi.org/10.1007/978-3-540-89754-5_24).
- [DKS10] Orr Dunkelman, Nathan Keller, and Adi Shamir. **A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony**. CRYPTO. Vol. 6223. LNCS. Springer, 2010, pp. 393–410. DOI: [10.1007/978-3-642-14623-7_21](https://doi.org/10.1007/978-3-642-14623-7_21).
- [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](https://doi.org/10.1007/s00145-013-9154-9).

Bibliography IV

- [HB21] Hosein Hadipour and Nasour Bagheri. **Improved Rectangle Attacks on SKINNY and CRAFT**. *IACR Trans. Symmetric Cryptol.* 2021.2 (2021), pp. 140–198. DOI: [10.46586/TOSC.V2021.I2.140-198](https://doi.org/10.46586/TOSC.V2021.I2.140-198).
- [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](https://doi.org/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](https://doi.org/10.1007/3-540-48658-5_3).

Bibliography V

- [SQH19] Ling Song, Xianrui Qin, and Lei Hu. **Boomerang Connectivity Table Revisited. Application to SKINNY and AES.** *IACR Trans. Symmetric Cryptol.* 2019.1 (2019), pp. 118–141. DOI: [10.13154/TOSC.V2019.I1.118-141](https://doi.org/10.13154/TOSC.V2019.I1.118-141). URL: <https://doi.org/10.13154/tosc.v2019.i1.118-141>.
- [Wag99] David A. Wagner. **The Boomerang Attack.** FSE. Vol. 1636. LNCS. Springer, 1999, pp. 156–170. DOI: [10.1007/3-540-48519-8_12](https://doi.org/10.1007/3-540-48519-8_12).
- [WP19] Haoyang Wang and Thomas Peyrin. **Boomerang Switch in Multiple Rounds. Application to AES Variants and Deoxys.** *IACR Trans. Symmetric Cryptol.* 2019.1 (2019), pp. 142–169. DOI: [10.13154/TOSC.V2019.I1.142-169](https://doi.org/10.13154/TOSC.V2019.I1.142-169).

Bibliography VI

- [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](https://doi.org/10.1049/2024/8315115).

Properties of Generalized DLCT Tables - I

- $\text{DLCT}(\Delta_i, \lambda_o) = \sum_{\Delta_o} \text{UDLCT}(\Delta_i, \Delta_o, \lambda_o)$
- $\text{UDLCT}(\Delta_i, \Delta_o, \lambda_o) = (-1)^{\Delta_o \cdot \lambda_o} \text{DDT}(\Delta_i, \Delta_o)$
- $\text{LDLCT}(\Delta_i, \lambda_i, \lambda_o) = (-1)^{\Delta_i \cdot \lambda_i} \text{DLCT}(\Delta_i, \lambda_o)$
- $\text{EDLCT}(\Delta_i, \Delta_o, \lambda_i, \lambda_o) = (-1)^{\lambda_i \cdot \Delta_i \oplus \lambda_o \cdot \Delta_o} \text{DDT}(\Delta_i, \Delta_o)$
- $\text{LDLCT}(\Delta_i, \lambda_i, \lambda_o) = \sum_{\Delta_o} \text{EDLCT}(\Delta_i, \Delta_o, \lambda_i, \lambda_o)$
- $\sum_{\Delta_i} \text{LDLCT}(\Delta_i, \lambda_i, \lambda_o) = \text{LAT}^2(\lambda_i, \lambda_o)$

Properties of Generalized DLCT Tables - II

- $\text{DDLCT}(\Delta_i, \lambda_o) = \sum_{\Delta_m} \sum_{\lambda_m} \text{UDLCT}(\Delta_i, \Delta_m, \lambda_m) \cdot \text{LDLCT}(\Delta_m, \lambda_m, \lambda_o)$

$$\begin{aligned}\text{DDLCT}(\Delta_i, \lambda_o) &= \sum_{\Delta_m} \text{DDT}(\Delta_i, \Delta_m) \cdot \text{DLCT}(\Delta_m, \lambda_o) \\ &= 2^{-n} \sum_{\lambda_m} \text{DLCT}(\Delta_i, \lambda_m) \cdot \text{LAT}^2(\lambda_m, \lambda_o).\end{aligned}$$