Automating Cryptanalysis:
Automated Reasoning and Structural Links Between Attacks

Hosein Hadipour, Ruhr University Bochum, Germany

March 15, 2025

SKCAM 2025 - Rome, Italy

Maria Eichlseder

Yosuke Todo

Nasour Bagheri

Patrick Derbez

Sadegh Sadeghi

Outline

Anatomy of Symmetric-Key Attacks

Theoretical Links Between Symmetric-Key Attacks

Structural Similarities Between Symmetric-Key Techniques

Research Gaps and Future Works

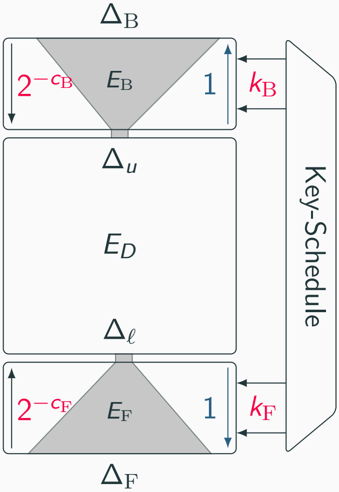
Anatomy of Symmetric-Key Attacks

Well-Known Cryptanalytic Attacks

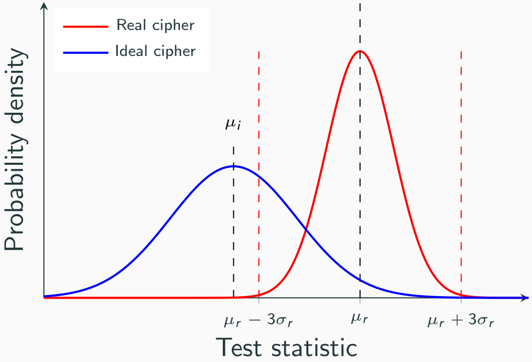
- Differential attack [BS90] (Full round DES [BS92]/AES-256 [BKN09])
- Linear attack [Mat93] (Full round DES [Mat93])
- Boomerang attack [Wag99] (Full round COCONUT98 [Wag99])
- Differential-Linear (DL) attack [LH94] (Full round COCONUT98 [BDK02])
- Impossible-Differential (ID) attack [Knu98; BBS99] (7 rounds of AES)
- Zero-Correlation attack (ZC) [BR14]
- Integral attack [Lai94a; DKR97] (Full-round MISTY1 [Tod15])
- Cube attack [DS09] (Best attack type on SHA-3 [Hua+17])
- And some others, e.g., guess-and-determine and meet-in-the-middle attacks.

Anatomy of Symmetric-Key Attacks – Overall View

- Distinguisher
- Key Recovery



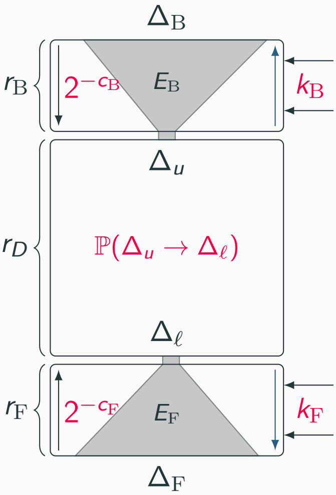
Anatomy of Symmetric-Key Attacks – Distinguisher + Key Recovery



$$r_D \left\{ \boxed{\begin{array}{c} \Delta_u \\ \\ \mathbb{P}(\Delta_u \to \Delta_\ell) \\ \\ \Delta_\ell \end{array}} \right.$$

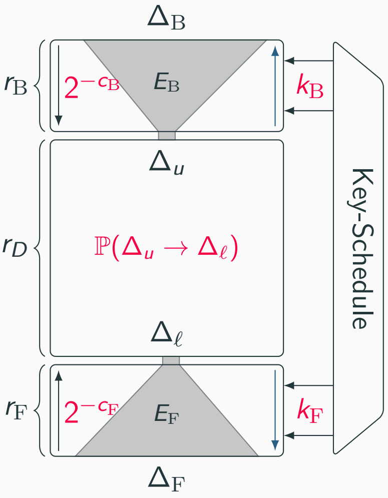
Anatomy of Symmetric-Key Attacks – Distinguisher + Key Recovery

- Common techniques in differential-based key recoveries:
  - Early abort technique [Lu+08a]
  - Probabilistic extension [Pha04; Lu+08b; Mal+10]
- Common techniques in linear-bsed and integral key recoveries:
  - FFT technique [CSQ07; FN20]
  - Partial-sum technique [Fer+00]



Anatomy of Symmetric-Key Attacks – Distinguisher + Key Recovery

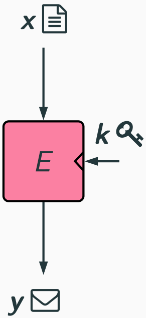
- Common techniques in differential-based key recoveries:
  - Early abort technique [Lu+08a]
  - Probabilistic extension [Pha04; Lu+08b; Mal+10]
- Common techniques in linear-bsed and integral key recoveries:
  - FFT technique [CSQ07; FN20]
  - Partial-sum technique [Fer+00]
- Generic and universal techniques (in key recovery):
  - Guess-and-Determine technique
  - Key-Bridging technique [DKS10b]



Why Do We Need to Study the Links Between Attacks?

- Avoid reinventing the wheel: save time and effort in cryptanalysis.
- Reuse discoveries in one attack to improve another.
    - Reuse more efficient automated tools from one attack to another.
    - Reuse discovered distinguishers from one attack to another.
    - Reuse discovered key-recovery techniques from one attack to another.
- Exploring the link between attacks can even lead to discovering a new attack type!

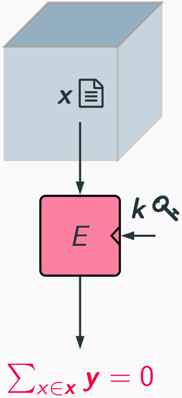
Theoretical Links Between Symmetric-Key Attacks



x
E
k
y

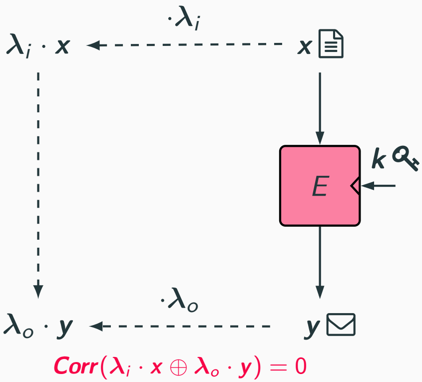
Integral and ZC Distinguishers

- Integral attack [Lai94b; DKR97]



Integral and ZC Distinguishers

- Integral attack [Lai94b; DKR97]
- Zero-correlation attack [BR14]



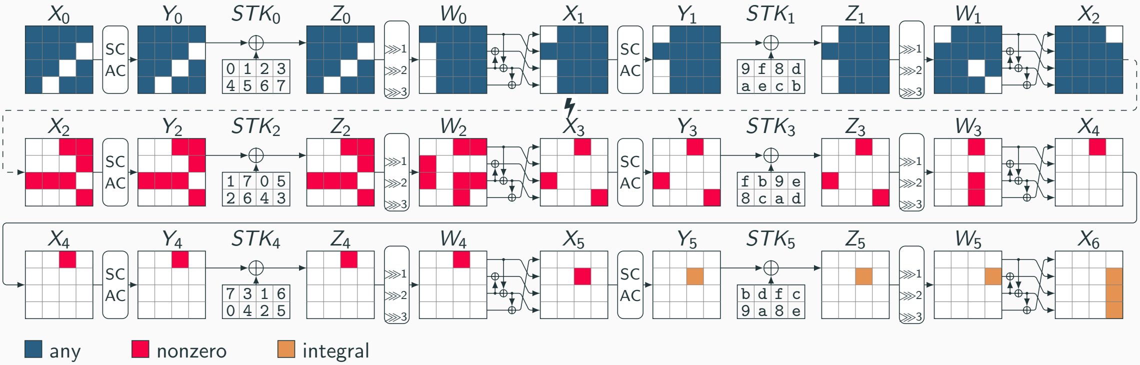
Relation Between ZC and Integral Distinguishers

Any ZC distinguisher can be converted to an integral distinguisher [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 (α, β) 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

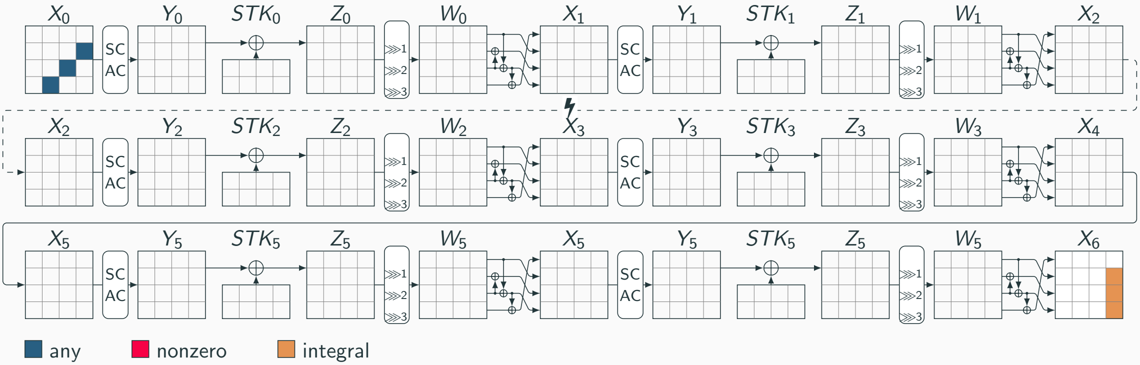
$$A^\perp = \{x \in \mathbb{F}_2^n \mid \forall\ \alpha \in A : \langle \alpha, x \rangle = 0\}.$$

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

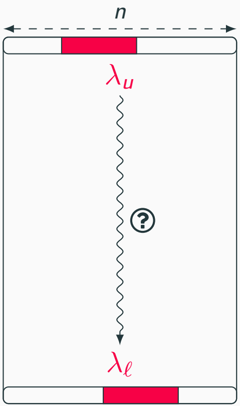
Example: Conversion of ZC Distinguisher to Integral Distinguisher



- $X_0[7, 10, 13]$ 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

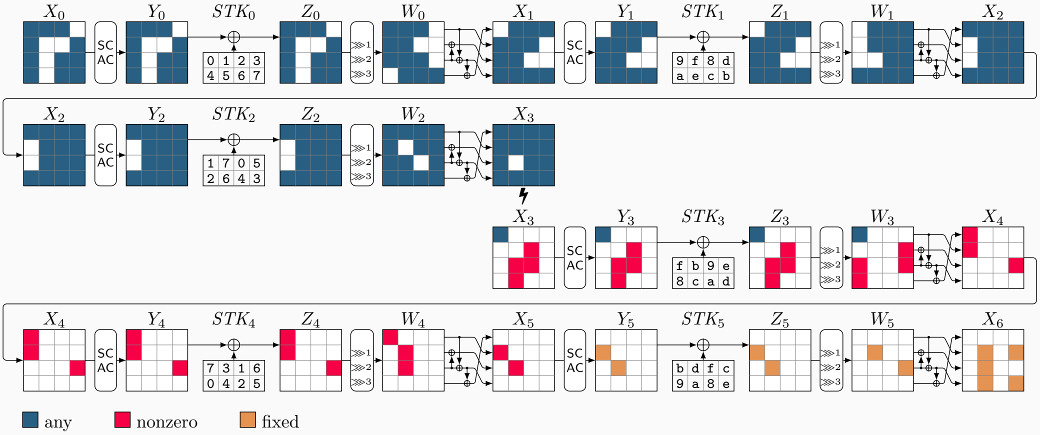
Previous Tools for ZC and Integral Attacks

- Tools based on (the negative output of the) 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]



Miss-in-the-Middle Technique [BBS99]

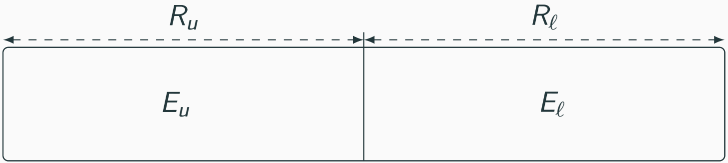
- Find two linear masks that propagate forward and backward with probability one and contradict each other somwhere in the middle.



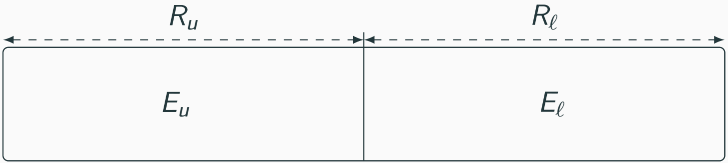
Our Positive Model to Search Distinguishers [HSE23]

E

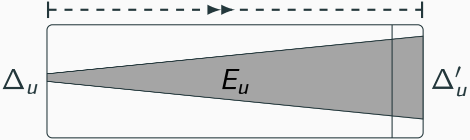
Our Positive Model to Search Distinguishers [HSE23]



Our Positive Model to Search Distinguishers [HSE23]

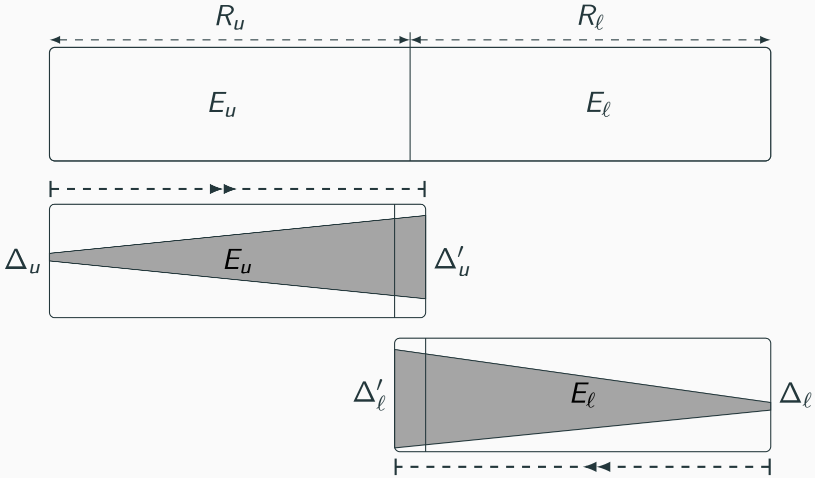


- $\mathcal{CSP}_u(\Delta_u, \Delta'_u)$



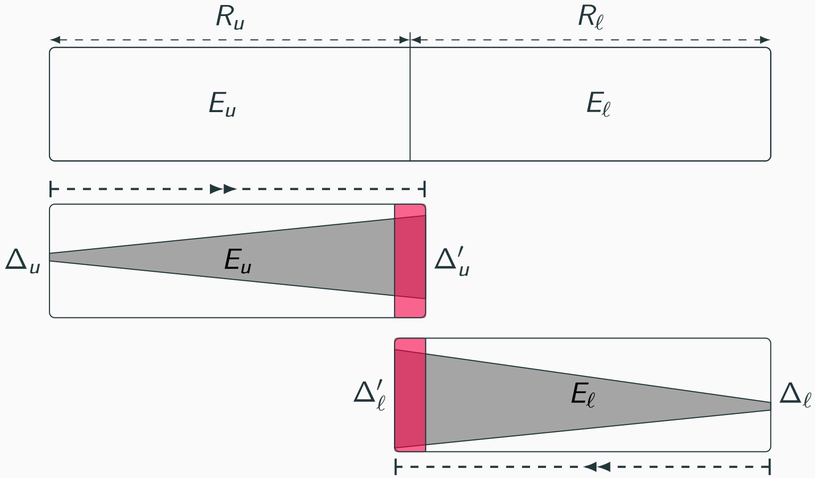
Our Positive Model to Search Distinguishers [HSE23]

- $\mathcal{CSP}_u(\Delta_u, \Delta'_u)$
- $\mathcal{CSP}_\ell(\Delta_\ell, \Delta'_\ell)$



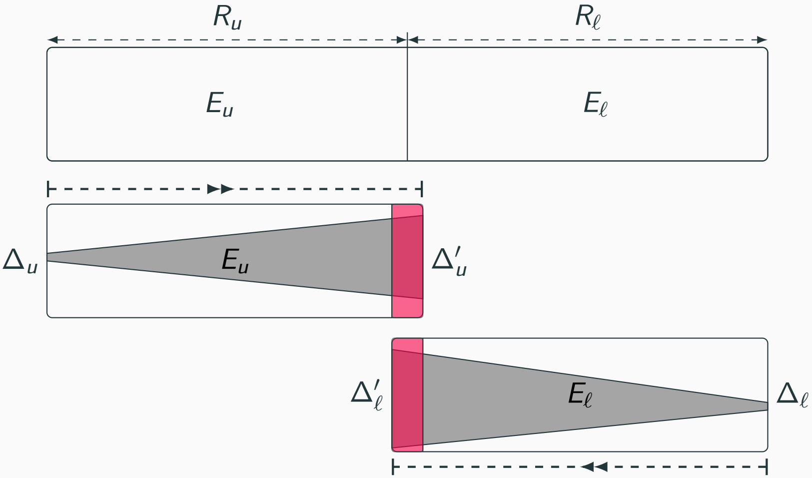
Our Positive Model to Search Distinguishers [HSE23]

- $\mathcal{CSP}_u(\Delta_u, \Delta'_u)$
- $\mathcal{CSP}_\ell(\Delta_\ell, \Delta'_\ell)$
- $\mathcal{CSP}_{\mathtt{M}}(\Delta'_u, \Delta'_\ell)$



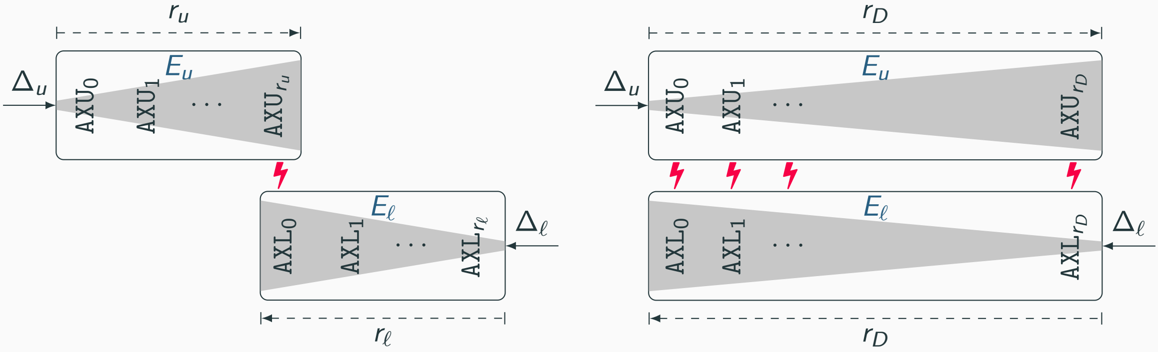
Our Positive Model to Search Distinguishers [HSE23]

- $CSP_u(\Delta_u, \Delta'_u)$
- $CSP_\ell(\Delta_\ell, \Delta'_\ell)$
- $CSP_{\mathtt{M}}(\Delta'_u, \Delta'_\ell)$



Relax the Limit of Fixing the Contradiction's Location [Hos+24]

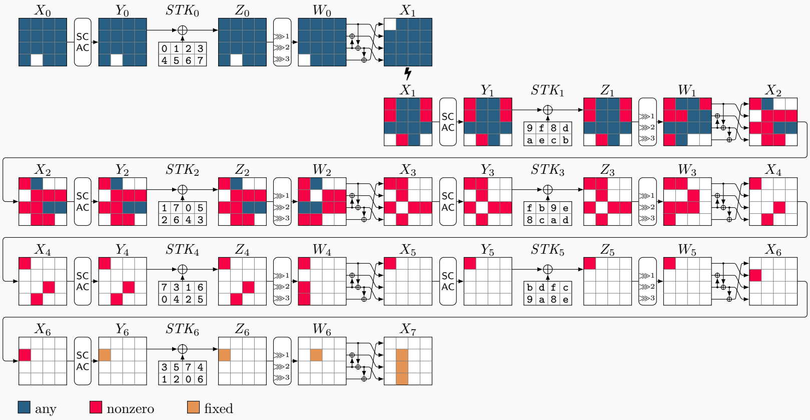
Find ZC distinguisher for $r_D (= r_u + r_\ell)$ rounds



Our first model in [HSE23].

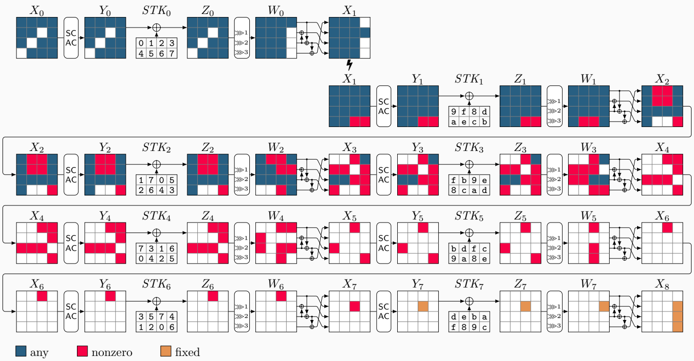
Our second model in [Hos+24]

Example: Integral Distinguisher for 7 Rounds of SKINNY



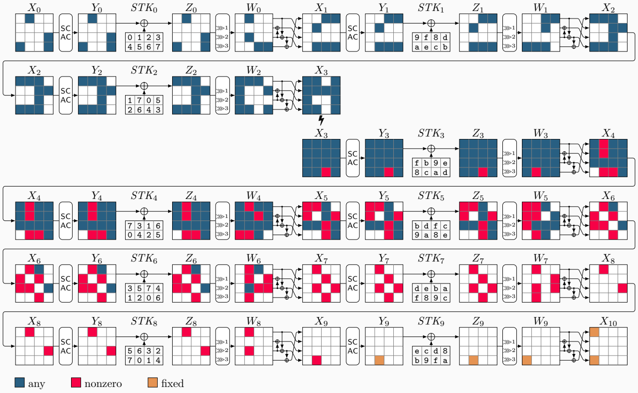
- ToSC 2019 (algebraic techniques) [Zha+20] : 7 rounds of SKINNY with data complexity $2^{2 \cdot c}$.
- Our simple model: 7-round integral distinguisher for SKINNY with data complexity 2^{c}.

Example: Integral Distinguisher for 8 Rounds of SKINNY



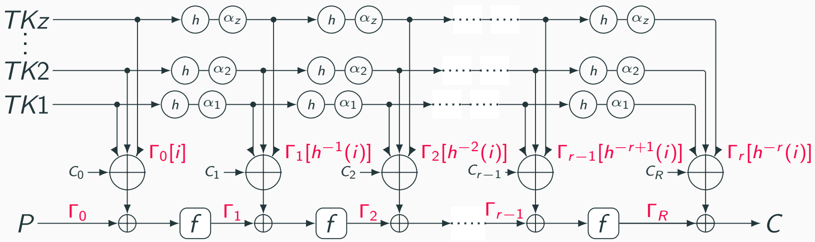
- ToSC 2020 (division property) [DF20]: 8 rounds of SKINNY-64 with data complexity 2^{15}.
- Applying division property to SKINNY-128 is computationally expensive (no results?).
- Our simple model: 8 rounds of SKINNY-64 (SKINNY-128) with data complexity 2^{12} (resp. 2^{24}).

Example: Integral Distinguisher for 10 Rounds of SKINNY



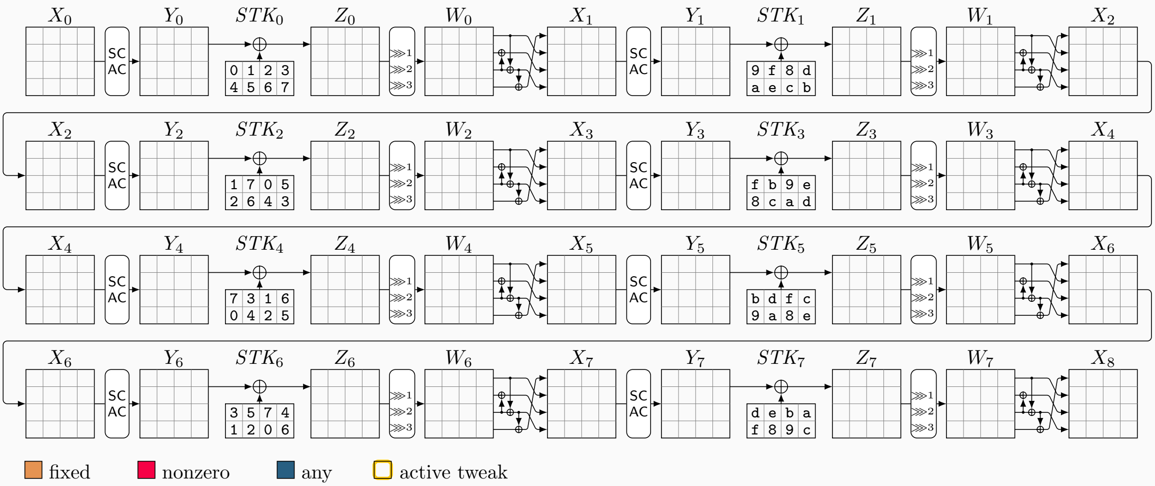
- ToSC 2019 (algebraic techniques) [Zha+20]: 10 rounds of SKINNY with data complexity $2^{15\cdot c}$.
- ToSC 2020 (division property) [DF20]: 10 rounds of SKINNY-64 with data complexity 2^{47}.
- Our simple model: 10 rounds of SKINNY with data complexity $2^{12\cdot c}$.
- Relaxing input mask constraints may even yield better results.

ZC Distinguishers for Ciphers Following the TWEAKEY Framework

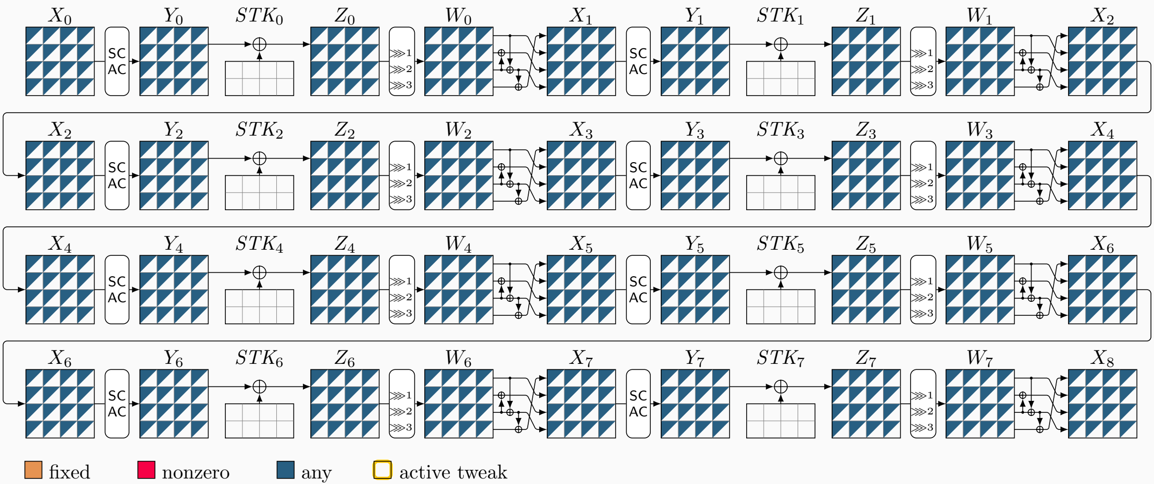


- Miss-in-the-Middle Method for ZC Distinguishers Considering Tweakey [Ank+19]:
  - Let z be the number of parallel paths in the tweakey schedule.
  - Find input/output masks activating a tweakey cell at most z times.
  - To see the formal description of the method see [Ank+19].

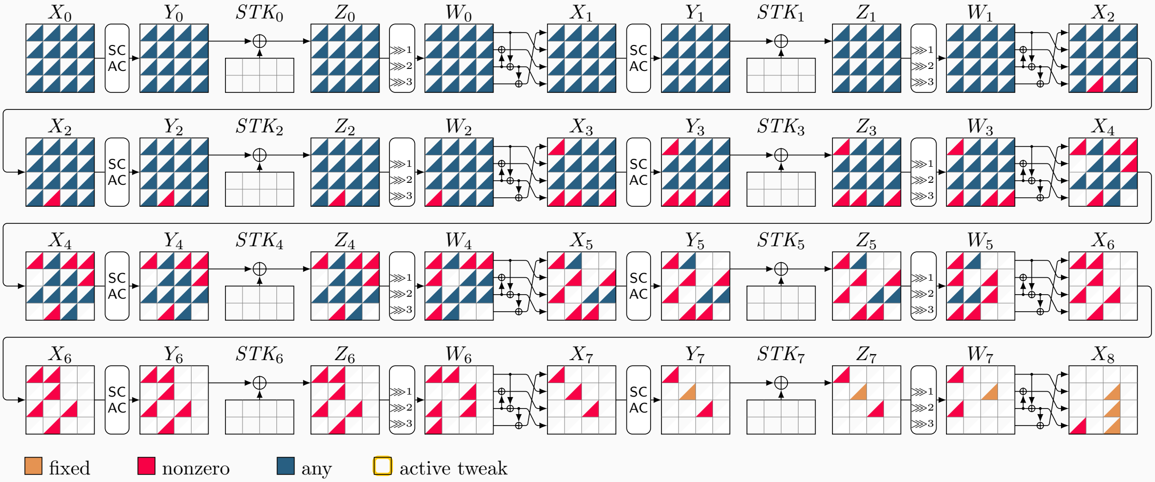
Example: ZC Distinguisher for 6 Rounds of SKINNY-TK2



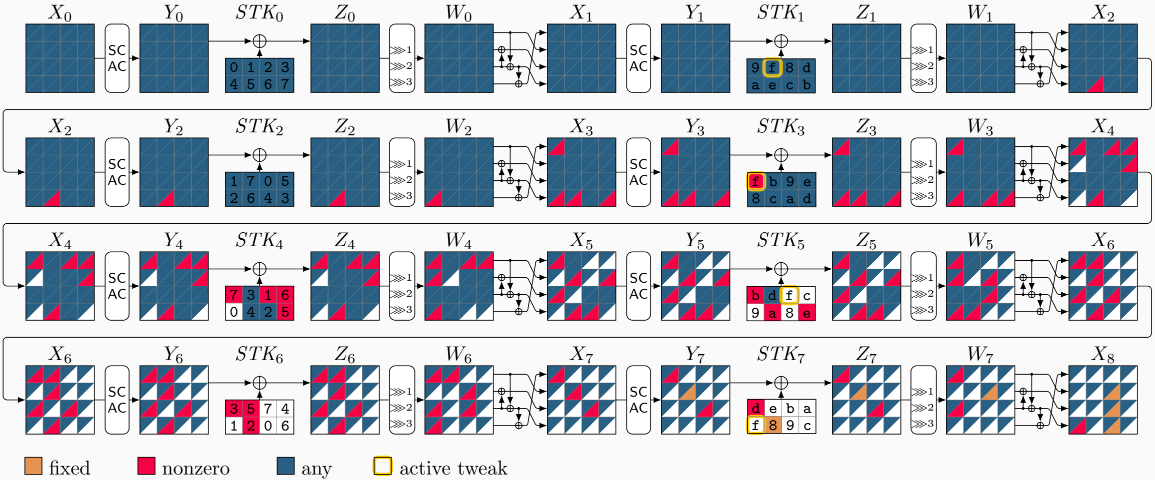
Example: ZC Distinguisher for 6 Rounds of SKINNY-TK2



Example: ZC Distinguisher for 6 Rounds of SKINNY-TK2



Example: ZC Distinguisher for 6 Rounds of SKINNY-TK2



- Look at the the tweak cell `0xf` (active at most 2 times).
- Data complexity of the corresponding integral distinguisher: $2^{2 \cdot c}$.

Chosen Tweak Integral Distinguishers for SKINNY and QARMAv2

Cipher	#Rounds	Dist.	Data complexity	Ref.
SKINNY-64-128	14	Integral	2^{60}	[HSE23]
ForkSKINNY-64-128	15	Integral	2^{60}	[Hos+24]
SKINNY-64-192	16	Integral	2^{60}	[HSE23]
ForkSKINNY-64-192	17	Integral	2^{60}	[Hos+24]
SKINNY-128-256	14	Integral	2^{112}	[HSE23]
ForkSKINNY-128-256	15	Integral	2^{112}	[Hos+24]
QARMAv2-64	5	Integral	-	[Ava+23]
QARMAv2-64 ($\mathscr{T} = 1$)	**7 / 8 / 9**	Integral	$\mathbf{2^{8}}$ / $\mathbf{2^{16}}$ / $\mathbf{2^{44}}$	[Hos+24]
QARMAv2-64 ($\mathscr{T} = 2$)	**8 / 9 / 10**	Integral	$\mathbf{2^{8}}$ / $\mathbf{2^{16}}$ / $\mathbf{2^{44}}$	[Hos+24]
QARMAv2-128($\mathscr{T} = 2$)	**10 / 11 / 12**	Integral	$\mathbf{2^{16}}$ / $\mathbf{2^{44}}$ / 2^{96}	[Hos+24]

- We also successfully applied our method to Deoxys-BC, CRAFT, MANTIS, PRESENT, Ascon, and even AndRX/ARX designs [HSE23; Hos+24; Cha+24].

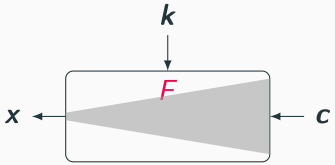
The Advantages of Our Method to Search for Distinguishers

- Based on satisfiability of the CP model (a positive 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

Naive Approach v.s. Partial-Sum Technique

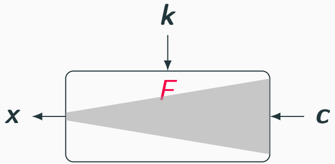
Naive approach:

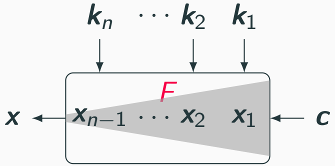
- $\boldsymbol{x} = F(\boldsymbol{k}, \boldsymbol{c})$
- $T = N \cdot 2^{|\boldsymbol{k}|}$



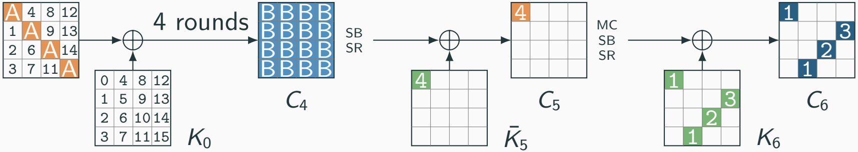
Naive Approach v.s. Partial-Sum Technique

- Naive approach:
  - $\boldsymbol{x} = F(\boldsymbol{k}, \boldsymbol{c})$
  - $T = N \cdot 2^{|\boldsymbol{k}|}$
- Partial-sum technique:
  - $\boldsymbol{x}_1 = f_1(\boldsymbol{k}_1, \boldsymbol{x}_0), \boldsymbol{x}_2 = f_2(\boldsymbol{k}_2, \boldsymbol{x}_1), \cdots, \boldsymbol{x} = f_n(\boldsymbol{k}_n, \boldsymbol{x}_{n-1})$
  - $\boldsymbol{x}_0 = \boldsymbol{c}, N_0 = N, N_i < N$
  - $T = \sum_{i=1}^{n} \frac{N_{i-1}}{n} \cdot 2^{|\boldsymbol{k}_1|+\cdots+|\boldsymbol{k}_i|} < \sum_{i=1}^{n} \frac{N}{n} \cdot 2^{|\boldsymbol{k}|}$
  - $T < N \cdot 2^{|\boldsymbol{k}|}$





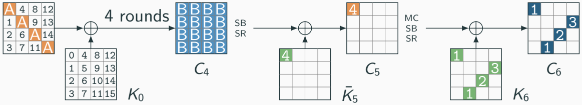
Example: Partial-Sum Integral Key Recovery for AES [Fer+00]



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

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

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



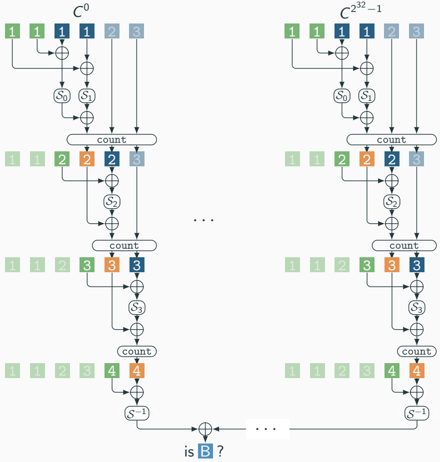
- Guess $K_6[0,7]$ and derive $\mathcal{S}_0\left(C_6[0] \oplus K_6[0]\right) \oplus \mathcal{S}_1\left(C_6[7] \oplus K_6[7]\right)$
- Guess $K_6[10]$ and derive $\mathcal{S}_2\left(C_6[10] \oplus K_6[10]\right)$
- Guess $K_6[13]$ and derive $\mathcal{S}_3\left(C_6[13] \oplus K_6[13]\right)$
- Guess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

Step 1: Key $= 2^{16}$
Data $= 2^{32}$
Time $= 2^{48}$

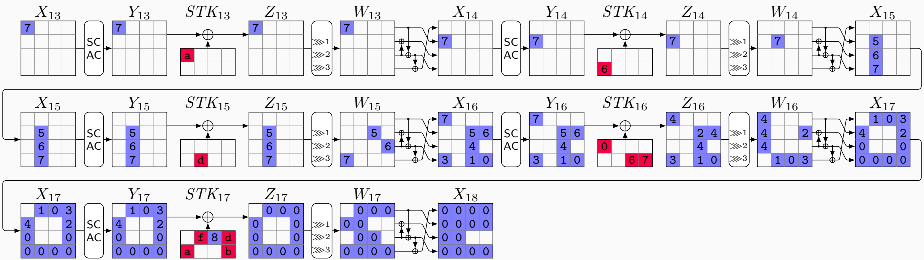
Step 2: Key $= 2^{24}$
Data $= 2^{24}$
Time $= 2^{48}$

Step 3: Key $= 2^{32}$
Data $= 2^{16}$
Time $= 2^{48}$

Step 4: Key $= 2^{40}$
Data $= 2^{8}$
Time $= 2^{48}$



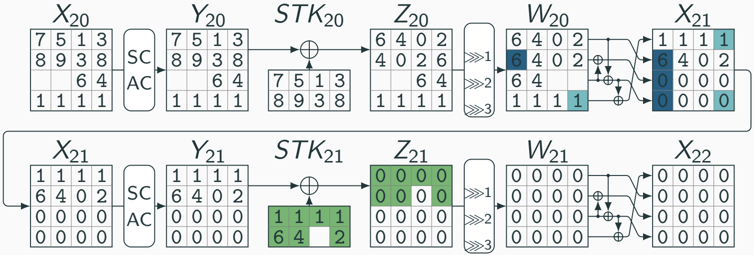
Our CP Model for Partial-Sum Technique - I



Step	Guessed	K $\times$ 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	$STK_{16}[0]$	$2^{28} \times 2^4 = 2^{32}$	$2^{44-6.2}$	$X_{13}[0]$
Σ		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.

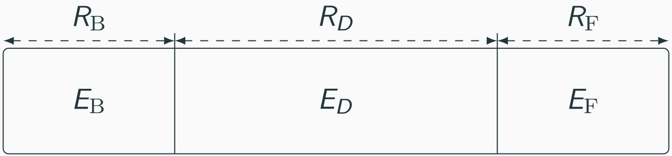


Our Unified Model for Finding Integral Attack

- 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

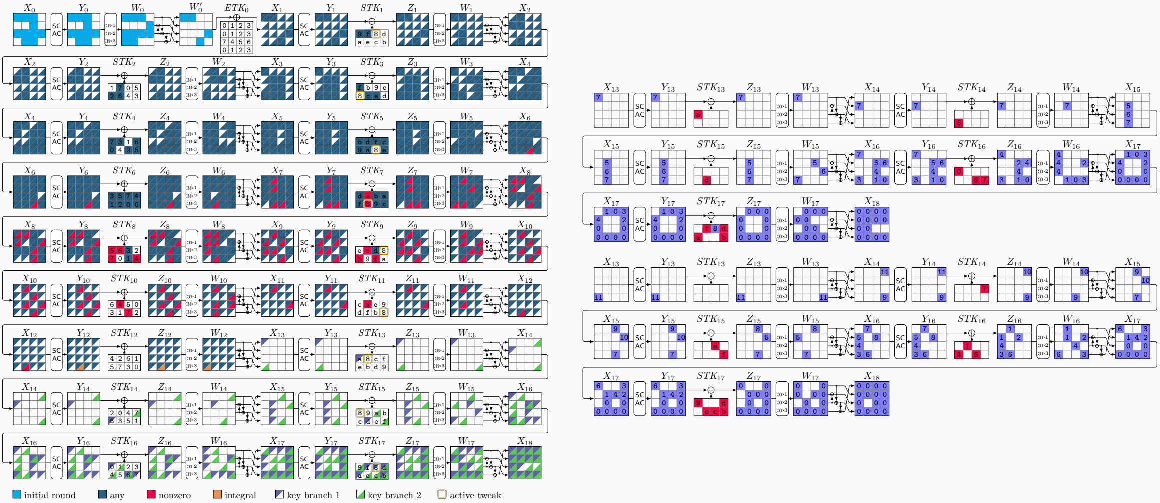
Usage of Our Tool

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



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

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



Part of Our Result

Cipher	#R	Time	Data	Mem.	Attack	Setting / Model	Ref.
SKINNY-64-64	**18**	$\mathbf{2^{53.58}}$	$2^{53.58}$	2^{48}	Int	60,SK / CP,CT	[Hos+24]
SKINNY-128-128	**18**	$\mathbf{2^{105.58}}$	$2^{105.58}$	2^{96}	Int	120,SK / CP,CT	[Hos+24]
SKINNY-64-192	23	$2^{155.60}$	$2^{73.20}$	2^{138}	Int	180,SK / CP,CT	[Ank+19]
	26	2^{172}	2^{61}	2^{172}	Int	180,SK / CP,CT	[HSE23]
SKINNY-64-128	20	$2^{97.50}$	$2^{68.40}$	2^{82}	Int	120,SK / CP,CT	[Ank+19]
	22	2^{110}	$2^{57.58}$	2^{108}	Int	120,SK / CP,CT	[HSE23]

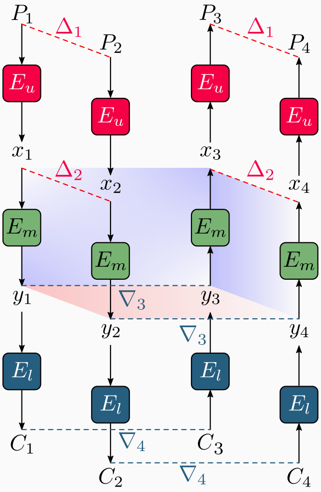
Open Question

- ZC distinguishers yield integral distinguishers but don't capture all integral properties.
- Monomial prediction (MP) captures all integral properties theoretically, but not practically.
- Automated methods based on MP or division property are negative and computationally expensive.

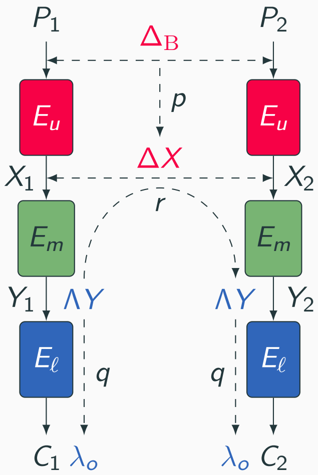
Is there a positive model based on division property or monomial prediction to automatically discover integral distinguishers?

Structural Similarities Between Symmetric-Key Techniques

Structural Similarities Between DL and Boomerang Distinguishers



$P = \mathbb{P}(P_3 \oplus P_4 = \Delta_1)$



$\mathbb{C} = \mathbb{C}(\lambda_o \cdot (C_1 \oplus C_2))$

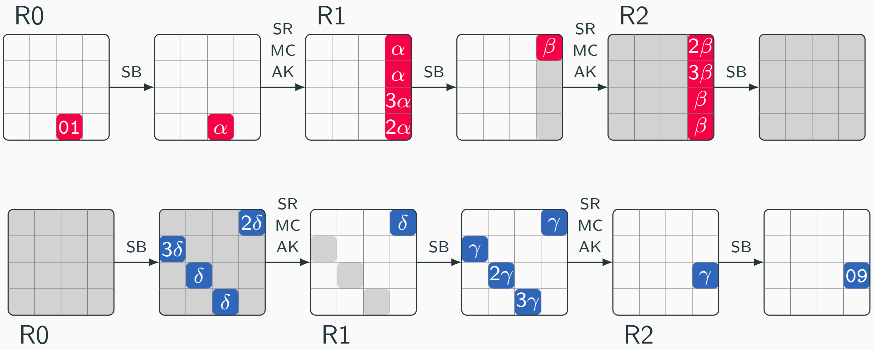
Reuse the Tools from Boomerang Aanalysis in DL Analysis [Bar+19; HDE24]

Boomerang	Differential-Linear
$\begin{tikzcd} \Delta_{\mathrm{B}} \arrow[r] \arrow[dr, "\mathtt{BCT}"] & \Delta_{\mathrm{F}} \\ \nabla_{\mathrm{B}} & \nabla_{\mathrm{F}} \arrow[l] \end{tikzcd}$	$\begin{tikzcd} \Delta_{\mathrm{B}} \arrow[r] \arrow[dr, "\mathtt{DLCT}"] & \Delta_{\mathrm{F}} \\ \lambda_{\mathrm{B}} & \lambda_{\mathrm{F}} \arrow[l] \end{tikzcd}$
$\begin{tikzcd} \Delta_{\mathrm{B}} \arrow[r] \arrow[dr, "\mathtt{UBCT}"] & \Delta_{\mathrm{F}} \\ \nabla_{\mathrm{B}} & \nabla_{\mathrm{F}} \arrow[l] \end{tikzcd}$	$\begin{tikzcd} \Delta_{\mathrm{B}} \arrow[r] \arrow[dr, "\mathtt{UDLCT}"] & \Delta_{\mathrm{F}} \\ \lambda_{\mathrm{B}} & \lambda_{\mathrm{F}} \arrow[l] \end{tikzcd}$

Boomerang	Differential-Linear
$\begin{tikzcd} \Delta_1 \arrow[r] \arrow[dr, "\mathtt{LBCT}"] & \Delta_2 \\ \nabla_1 & \nabla_2 \arrow[l] \end{tikzcd}$	$\begin{tikzcd} \Delta_{\mathrm{B}} \arrow[r] \arrow[dr, "\mathtt{LDLCT}"] & \Delta_{\mathrm{F}} \\ \lambda_{\mathrm{B}} & \lambda_{\mathrm{F}} \arrow[l] \end{tikzcd}$
$\begin{tikzcd} \Delta_1 \arrow[r] \arrow[dr, "\mathtt{EBCT}"] & \Delta_2 \\ \nabla_1 & \nabla_2 \arrow[l] \end{tikzcd}$	$\begin{tikzcd} \Delta_{\mathrm{B}} \arrow[r] \arrow[dr, "\mathtt{EDLCT}"] & \Delta_{\mathrm{F}} \\ \lambda_{\mathrm{B}} & \lambda_{\mathrm{F}} \arrow[l] \end{tikzcd}$

Boomerang	Differential-Linear
$\begin{tikzcd} \Delta_{\mathrm{B}} \arrow[r] \arrow[dr] & \Delta_m \arrow[r] \arrow[dr] \arrow[d, phantom, "\mathtt{DBCT}"] & \Delta_{\mathrm{F}} \\ \nabla_{\mathrm{B}} \arrow[r] & \nabla_m \arrow[r] & \nabla_{\mathrm{F}} \end{tikzcd}$	$\begin{tikzcd} \Delta_{\mathrm{B}} \arrow[r] \arrow[dr] & \Delta_m \arrow[r] \arrow[dr] \arrow[d, phantom, "\mathtt{DDLCT}"] & \Delta_{\mathrm{F}} \\ \lambda_{\mathrm{B}} \arrow[r] & \lambda_m \arrow[r] & \lambda_{\mathrm{F}} \end{tikzcd}$

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

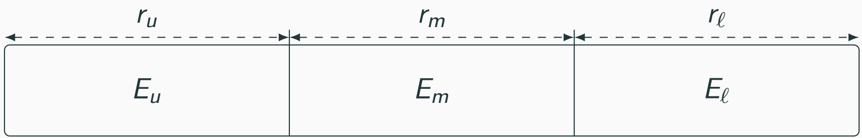


$$\sum_{\alpha,\beta,\gamma,\delta} \mathbb{C}_{\mathtt{UDLCT}}(1,\alpha,\delta)\cdot\mathbb{C}_{\mathtt{EDLCT}}(\alpha,\beta,\delta,\gamma)\cdot\mathbb{C}_{\mathtt{LDLCT}}(\beta,\gamma,9) = -2^{-7.94}$$

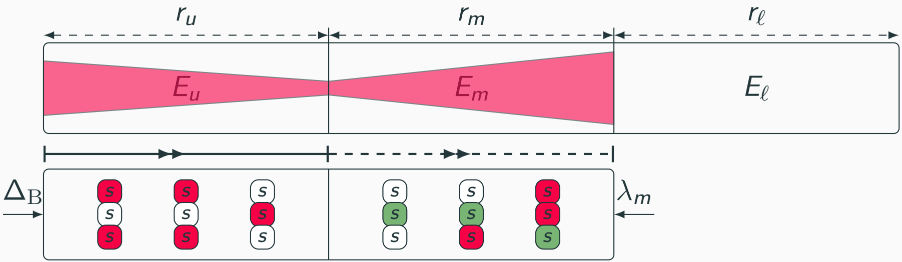
Overview of Our Method to Search for Distinguishers in Sandwich Framework

E

Overview of Our Method to Search for Distinguishers in Sandwich Framework

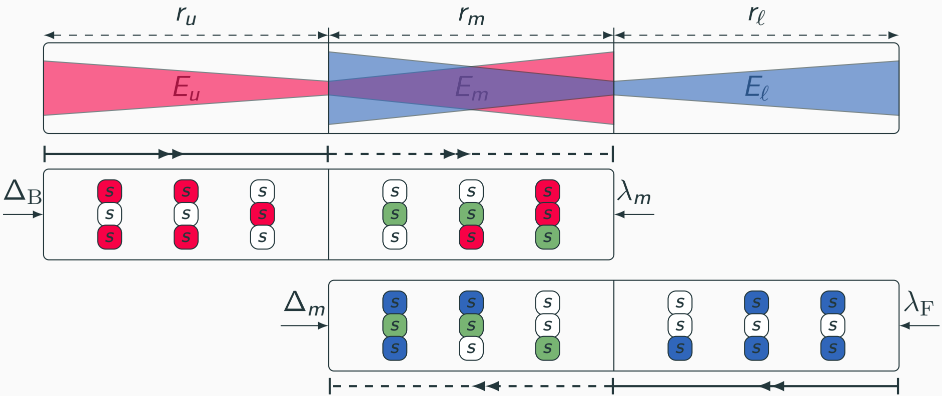


Overview of Our Method to Search for Distinguishers in Sandwich Framework



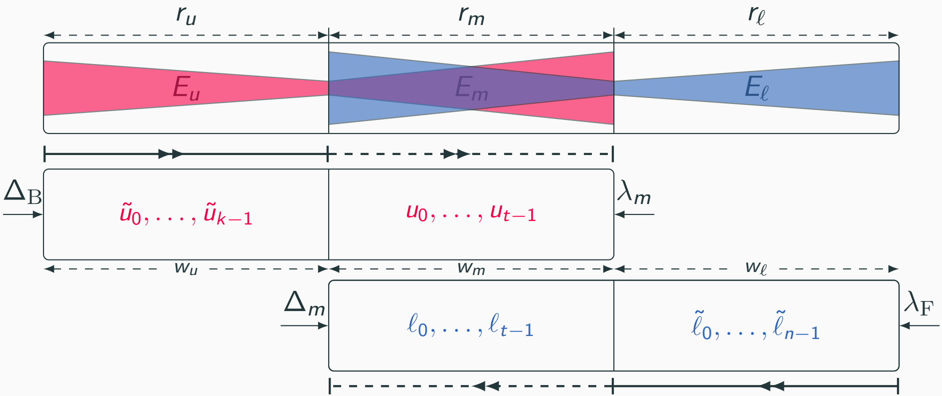
differentially active S-box   linearly active S-box   common active S-box

Overview of Our Method to Search for Distinguishers in Sandwich Framework



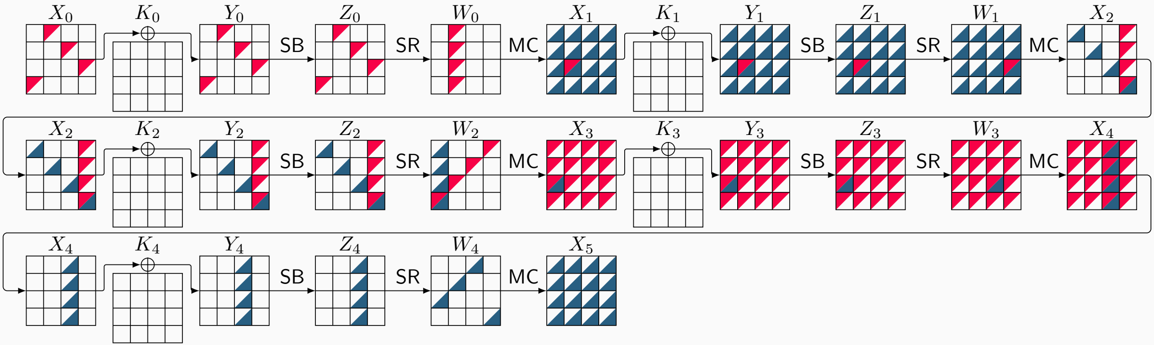
differentially active S-box linearly active S-box common active S-box

Overview of Our Method to Search for Distinguishers in Sandwich Framework



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

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`

Example: Distinguishers for up to 8 Rounds of CLEFIA [HDE24]

- Comparing the data complexity of best boomerang and DL distinguishers

# Rounds	Boomerang [HNE22]	Differential-Linear [HDE24]	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	$2^{76.03}$	$2^{66.86}$	$2^{9.17}$

Research Gaps and Future Works

Lessons and Future Works

- Lessons learned:
  - Consider the theoretical links between attacks in automated discovery.
  - Consider the structural similarities between attacks in automated discovery.
- **Future works**:
  - Connections between attacks are underutilized in automated discovery.
  - Existing methods often lack either accuracy or efficiency (hard to achieve both).
  - No unified framework exists for finding complete attacks across various types, e.g., differential, linear, boomerang.
  - Current methods are limited to strongly aligned designs, lacking approaches for weakly aligned designs.

”In this world, there is a universal law: to gain something, you must lose something else.”

– My Mom

: `https://github.com/hadipourh/talks`

Universal Bound for Data Complexity

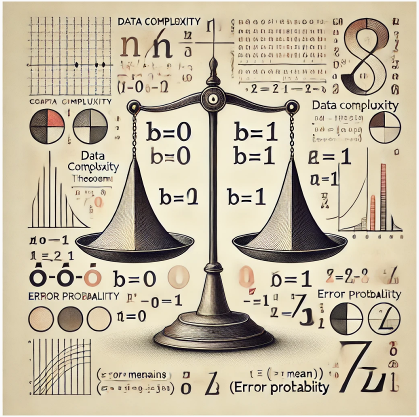
Universal Bound for Data Complexity - I

Theorem (Data Complexity)
Let X_0 and X_1 be two distributions. Given one sample from X_b, the distinguisher $\mathcal{D}$ outputs 1 with probability p if $b = 0$, and outputs 1 with probability q if $b = 1$. Assume that b is chosen uniformly at random from $\{0, 1\}$ and is fixed. Next, we run $\mathcal{D}$ on n samples, and output 1 if the sum of the outcomes is closer to $\mu_0 = np$, and 0 otherwise. If n satisfies the following inequality, then the error probability of the distinguisher is upper bounded by ε:

$$n \geq \max\left(\frac{2(3q+p)\ln\left(\frac{1}{\varepsilon}\right)}{(p-q)^2}, \frac{8p\ln\left(\frac{1}{\varepsilon}\right)}{(p-q)^2}\right).$$

Universal Bound for Data Complexity – II

- $n \geq \max\left(\frac{2(3q+p)\ln\left(\frac{1}{\varepsilon}\right)}{(p-q)^2}, \frac{8p\ln\left(\frac{1}{\varepsilon}\right)}{(p-q)^2}\right)$.
- If $p \gg q$, then $p - q \approx p$ then $n \geq \frac{8\ln\left(\frac{1}{\varepsilon}\right)}{p}$.
- If $p = \frac{1}{2} + \frac{c}{2}, q = \frac{1}{2} + \frac{c'}{2}, c \gg c'$, and $c, c' \ll \frac{1}{2}$ then $n \geq \frac{8\ln\left(\frac{1}{\varepsilon}\right)}{c^2}$.



Generated using OpenAI's DALL·E.

Differential Attack

Differential Attacks [BS90]

Input: $E_K, (\Delta_\mathrm{B}, \Delta_\mathrm{F}), N, p = \mathbb{P}(\Delta_\mathrm{B}, \Delta_\mathrm{F})$

Output: 0: **real** cipher, 1: **ideal** cipher

Initialize counter T with zero;

for $i = 0, \ldots, N-1$ **do**

- $P_1 \xleftarrow{\$} \mathbb{F}_2^n$;
- $C_1 \leftarrow E_K(P_1)$;
- $P_2 \leftarrow P_1 \oplus \Delta_\mathrm{B}$;
- $C_2 \leftarrow E_K(P_2)$;
- **if** $C_1 \oplus C_2 = \Delta_F$ **then**
  - $T \leftarrow T + 1$;

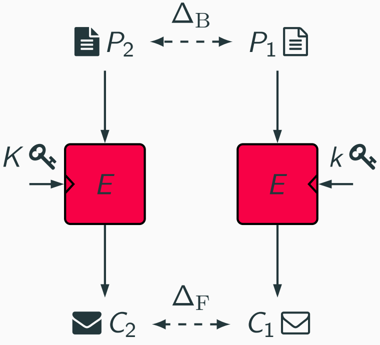
if $T \sim \mathcal{N}(\mu = Np, \sigma^2 = Np(1-p))$ **then**

- **return** 0; `// real cipher`

else

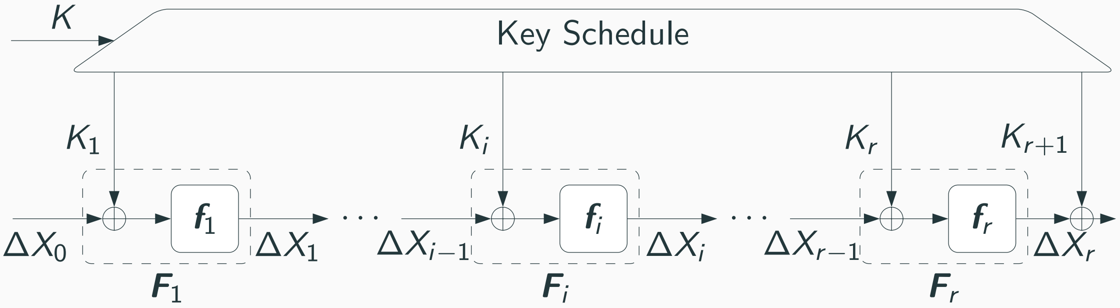
- **return** 1; `// ideal cipher`

$$N \approx \mathcal{O}(p^{-1}).$$



$p = \mathbb{P}(\Delta_\mathrm{B} \to \Delta_\mathrm{F})$

Analytical Estimation of Differential Probability



$$\mathbb{P}(\Delta X_r = \Delta_r \mid \Delta X_0 = \Delta_0) = \sum_{\Delta_1, \cdots, \Delta_{r-1}} \prod_{i=1}^{r} \mathbb{P}(\boldsymbol{f}_i(X) \oplus \boldsymbol{f}_i(X \oplus \Delta_{i-1}) = \Delta_i).$$

Difference Distribution Table ($\texttt{DDT}$) – I

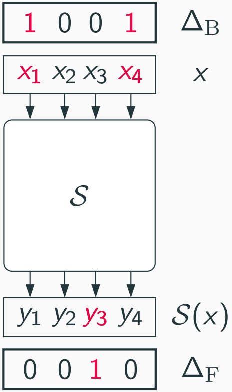
- We need a tool to handle the nonlinear operations

Differential Distribution Table ($\texttt{DDT}$)
For a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the $\texttt{DDT}$ is a $2^n \times 2^m$ table whose rows correspond to the input difference Δ_{B} to S and whose columns correspond to the output difference Δ_{F} of S. The entry at index $(\Delta_{\mathrm{B}}, \Delta_{\mathrm{F}})$ is

$$\texttt{DDT}(\Delta_{\mathrm{B}}, \Delta_{\mathrm{F}}) = |\{x \in \mathbb{F}_2^n : \; S(x) \oplus S(x \oplus \Delta_{\mathrm{B}}) = \Delta_{\mathrm{F}}\}|.$$

$\mathbb{P}\left(\Delta_{\mathrm{B}}, \Delta_{\mathrm{F}}\right) = 2^{-n} \cdot \texttt{DDT}\left(\Delta_{\mathrm{B}}, , \Delta_{\mathrm{F}}\right)$

Difference Distribution Table (DDT) – II



$$\mathbb{P}(9,2) = \frac{4}{16}$$

$\Delta_B \setminus \Delta_F$	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

Linear Attack

Linear Attacks [Mat93]

Input: E_K, Given N distinct plaintext-ciphertext pairs (P_i, C_i), $\boldsymbol{c} = \mathbb{C}(\lambda_{\mathrm{B}}, \lambda_{\mathrm{F}})$

Output: 0: **real** cipher, 1: **ideal** cipher

Initialize a counter list $V[z] \leftarrow 0$ for $z \in \{0, 1\}$;

for $t = 0, \ldots, N-1$ **do**

  $b_1 \leftarrow \lambda_{\mathrm{B}} \cdot P_t$;

  $b_2 \leftarrow \lambda_{\mathrm{F}} \cdot C_t$;

  $V[b_1 \oplus b_2] \leftarrow V[b_1 \oplus b_2] + 1$;

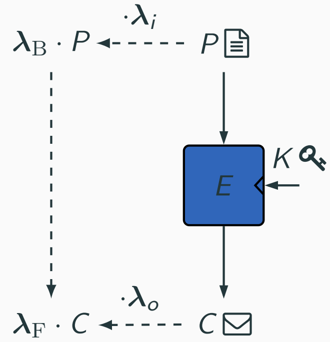
if $V[0] \sim \mathcal{N}(\mu_0 = N\frac{1+\boldsymbol{c}}{2}, \sigma_0^2 = \frac{N(1-c^2)}{4})$. **then**

  return 0; `// real cipher`

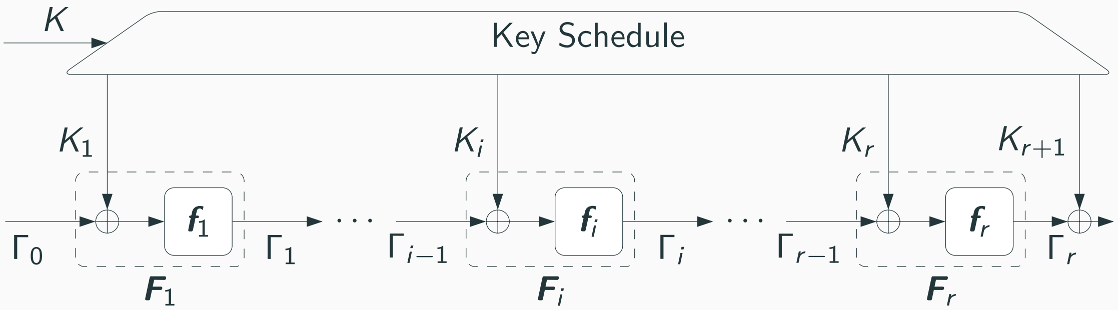
else

  return 1; `// ideal cipher`

$$N = \mathcal{O}(\boldsymbol{c}^{-2}).$$



$$\boldsymbol{c} = 2 \cdot \mathbb{P}(\boldsymbol{\lambda}_{\mathrm{B}} \cdot P \oplus \boldsymbol{\lambda}_{\mathrm{F}} \cdot C = 0) - 1$$



$$\mathbb{C}(\Gamma_0, \Gamma_{r+1}) \approx (-1)^{(\Gamma_0 \cdot K_1 \oplus \cdots \oplus \Gamma_r \cdot K_{r+1})} \prod_{i=1}^{r} \mathbb{C}_{\boldsymbol{f}_i}(\Gamma_{i-1}, \Gamma_i).$$

Linear Approximation Table ($\mathtt{LAT}$) – I

We need a metric to measure the quality of a linear approximation.

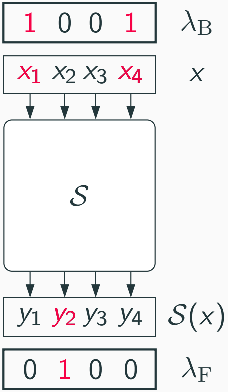
Linear Approximation Table ($\mathtt{LAT}$)
For a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the $\mathtt{LAT}$ of S is a $2^n \times 2^m$ table whose rows correspond to the input mask λ_{B} to S and whose columns correspond to the output mask λ_{F} of S. The entry at index $(\lambda_{\mathrm{B}}, \lambda_{\mathrm{F}})$ is

$$\mathtt{LAT}(\lambda_{\mathrm{B}}, \lambda_{\mathrm{F}}) = |\mathtt{LAT}_0(\lambda_{\mathrm{B}}, \lambda_{\mathrm{F}})| - |\mathtt{LAT}_1(\lambda_{\mathrm{B}}, \lambda_{\mathrm{F}})|,$$

where $\mathtt{LAT}_b(\lambda_{\mathrm{B}}, \lambda_{\mathrm{F}}) = \{x \in \mathbb{F}_2^n : \ \lambda_{\mathrm{B}} \cdot x \oplus \lambda_{\mathrm{F}} \cdot S(x) = b\}$.

$\mathbb{C}\,(\lambda_{\mathrm{B}}, \lambda_{\mathrm{F}}) = 2^{-n} \cdot \mathtt{LAT}\,(\lambda_{\mathrm{B}}, \lambda_{\mathrm{F}})$

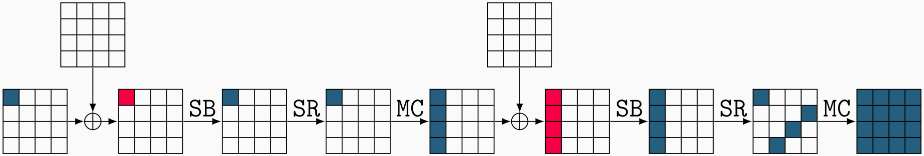
Linear Approximation Table (LAT) – II



$$\mathbb{C}(9,4) = \frac{8}{16}$$

$\lambda_B \setminus \lambda_F$	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

Minimum Number of Differentially Active S-boxes in AES



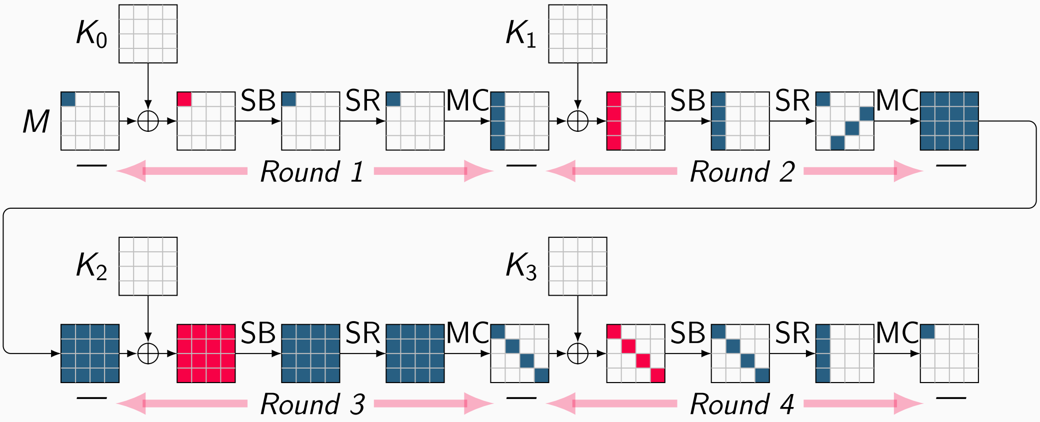
Variables:

- $s_{r,i,j} \in \{0,1\}$ is S-box in row i, column j, round r active?
- $m_{r,j} \in \{0,1\}$ is Mix-columns j in round r active?

Constraints and objective:

- $5 \cdot M_{r,j} \leq \sum_i s_{r,i,(i+j)\%4} + \sum_i s_{r+1,i,j} \leq 8 \cdot M_{r,j}; \quad \sum_{i,j} s_{0,i,j} \geq 1$
- $\min \sum_{r,i,j} s_{r,i,j}$

Security of AES Against Differential Attacks



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

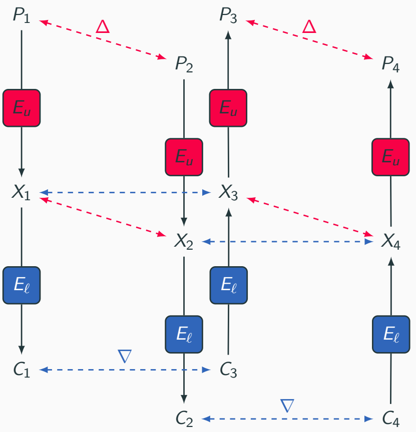
Boomerang Attack

Boomerang Distinguishers [Wag99]

Input: $E_K, (\Delta, \nabla), N, P = \mathbb{P}(P_3 \oplus P_4 = \Delta)$

Output: 0: **real** cipher, 1: **ideal** cipher

Initialize counter T with zero;
for $i = 0, \ldots, N-1$ **do**
  $P_1 \xleftarrow{\$} \mathbb{F}_2^n;\; P_2 = P_1 \oplus \Delta;$
  $C_1 \leftarrow E_K(P_1), \quad C_2 \leftarrow E_K(P_2);$
  $C_3 \leftarrow C_1 \oplus \nabla, \quad C_4 \leftarrow C_2 \oplus \nabla;$
  $P_3 \leftarrow D_K(C_3), \quad P_4 \leftarrow D_K(C_4);$
  if $P_3 \oplus P_4 = \Delta$ **then**
    $T \leftarrow T + 1;$
if $T \sim \mathcal{N}(\mu = NP, \sigma^2 = NP(1-P))$ **then**
  return 0; `// real cipher`
else
  return 1; `// ideal cipher`



Probability of Boomerang Distinguishers [Wag99]

$$\Delta \longrightarrow \boxed{E : \mathbb{F}_2^n \to \mathbb{F}_2^n} \longrightarrow \nabla$$

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

Probability of Boomerang Distinguishers [Wag99]

$$\Delta \longrightarrow \boxed{E_u \;\Big|\; E_\ell} \longrightarrow \nabla$$

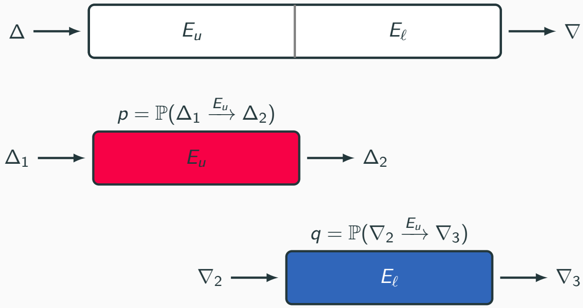
$$p = \mathbb{P}(\Delta_1 \xrightarrow{E_u} \Delta_2)$$

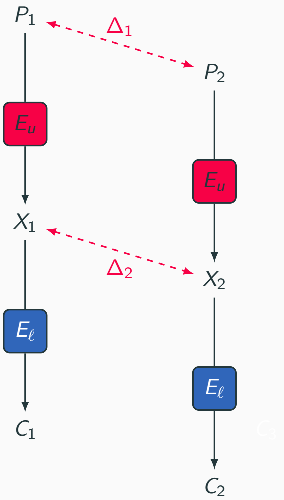
$$\Delta_1 \longrightarrow \boxed{E_u} \longrightarrow \Delta_2$$

$$q = \mathbb{P}(\nabla_2 \xrightarrow{E_u} \nabla_3)$$

$$\nabla_2 \longrightarrow \boxed{E_\ell} \longrightarrow \nabla_3$$

Probability of Boomerang Distinguishers [Wag99]

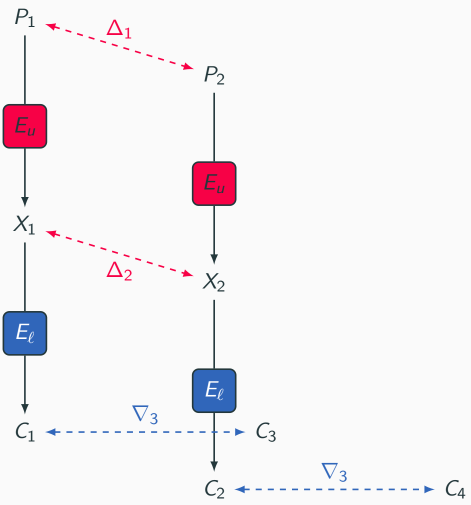




Probability of Boomerang Distinguishers [Wag99]

$p = \mathbb{P}(\Delta_1 \xrightarrow{E_u} \Delta_2)$

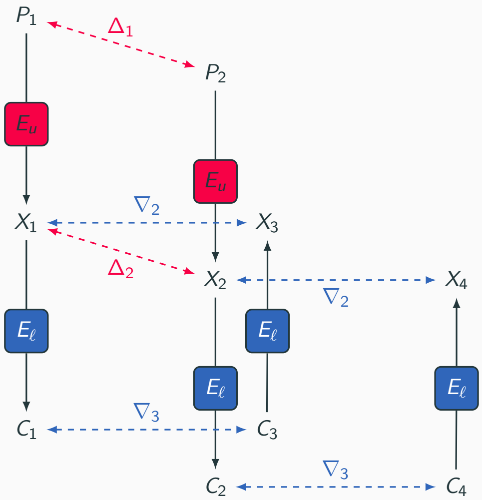
$q = \mathbb{P}(\nabla_2 \xrightarrow{E_u} \nabla_3)$



Probability of Boomerang Distinguishers [Wag99]

$p = \mathbb{P}(\Delta_1 \xrightarrow{E_u} \Delta_2)$

$q = \mathbb{P}(\nabla_2 \xrightarrow{E_u} \nabla_3)$

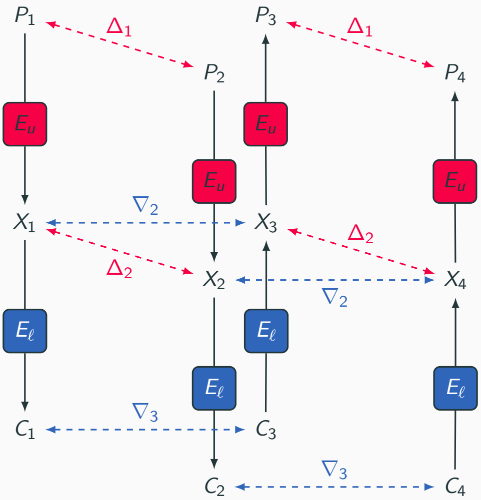


Probability of Boomerang Distinguishers [Wag99]

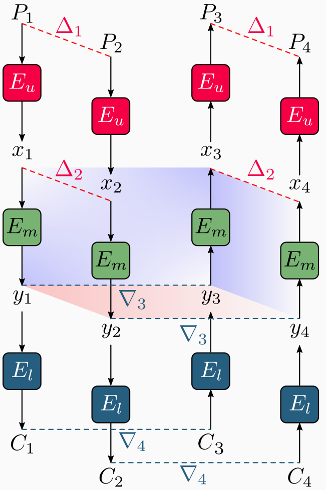
$p = \mathbb{P}(\Delta_1 \xrightarrow{E_u} \Delta_2)$

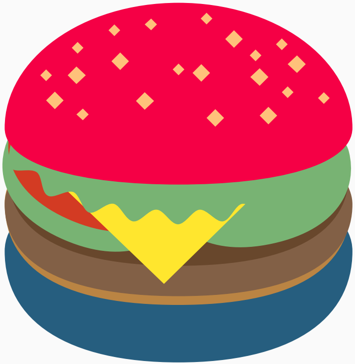
$q = \mathbb{P}(\nabla_2 \xrightarrow{E_u} \nabla_3)$

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

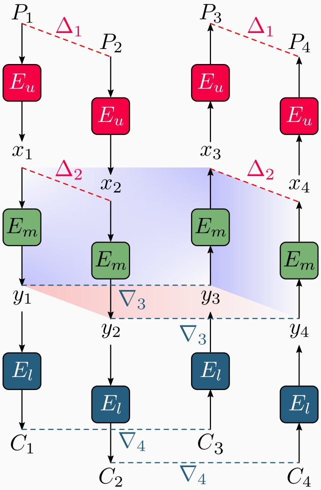


Sandwiching the Differentials! [DKS10a; DKS14]





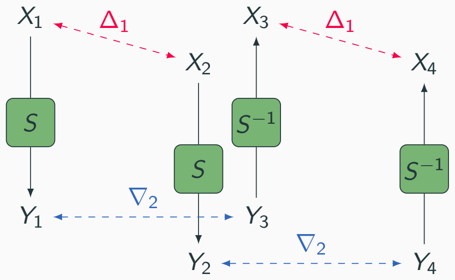
Sandwiching the Differentials! [DKS10a; DKS14]

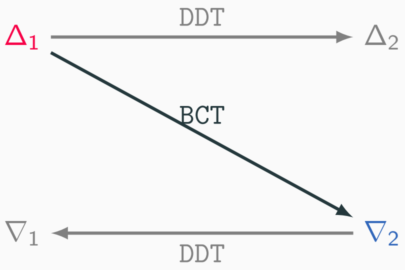


$$\mathbb{P}(P_3 \oplus P_4 = \Delta_1) \approx p^2 \times r \times q^2$$

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

Boomerang Connectivity Table (BCT) [Cid+18]

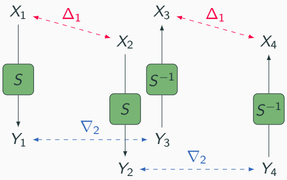




$$\mathtt{BCT}(\Delta_1, \nabla_2) := \#\{X \in \mathbb{F}_2^n \mid S^{-1}(S(X) \oplus \nabla_2) \oplus S^{-1}(S(X \oplus \Delta_1) \oplus \nabla_2) = \Delta_1\}$$

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

Generalized BCT Framework – I

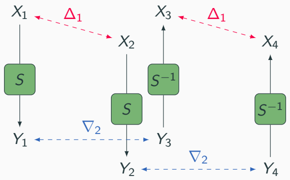


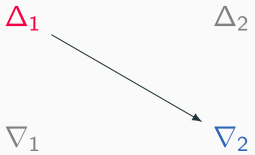
$$\Delta_1 \longrightarrow \Delta_2$$

$$\nabla_1 \longleftarrow \nabla_2$$

- $\mathcal{X}_{\texttt{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}, \quad \texttt{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\texttt{DDT}}(\Delta_1, \Delta_2)$

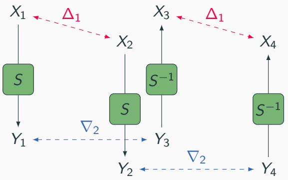
Generalized BCT Framework – I

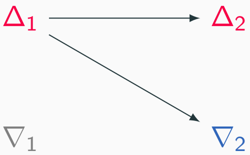




- $\mathcal{X}_{\mathtt{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}, \quad \mathtt{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\mathtt{DDT}}(\Delta_1, \Delta_2)$
- $\mathcal{X}_{\mathtt{BCT}}(\Delta_1, \nabla_2) = \{x : S^{-1}(S(x) \oplus \nabla_2) \oplus S^{-1}(S(x \oplus \Delta_1) \oplus \nabla_2) = \Delta_1\}, \; \mathtt{BCT}(\Delta_1, \nabla_2) = \#\mathcal{X}_{\mathtt{BCT}}(\Delta_1, \nabla_2)$

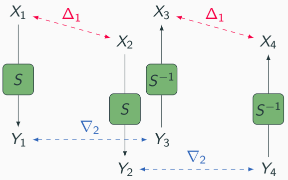
Generalized BCT Framework – I

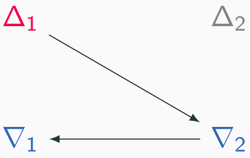




- $\mathcal{X}_{\texttt{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}, \quad \texttt{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\texttt{DDT}}(\Delta_1, \Delta_2)$
- $\mathcal{X}_{\texttt{BCT}}(\Delta_1, \nabla_2) = \{x : S^{-1}(S(x) \oplus \nabla_2) \oplus S^{-1}(S(x \oplus \Delta_1) \oplus \nabla_2) = \Delta_1\}, \ \texttt{BCT}(\Delta_1, \nabla_2) = \#\mathcal{X}_{\texttt{BCT}}(\Delta_1, \nabla_2)$
- $\texttt{UBCT}(\Delta_1, \Delta_2, \nabla_2) = \#\{x : x \in \mathcal{X}_{\texttt{BCT}}(\Delta_1, \nabla_2) \cap \mathcal{X}_{\texttt{DDT}}(\Delta_1, \Delta_2)\}$ [WP19]

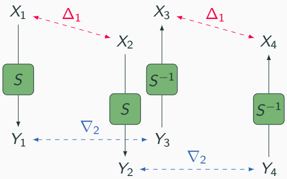
Generalized BCT Framework – I





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

Generalized BCT Framework – I



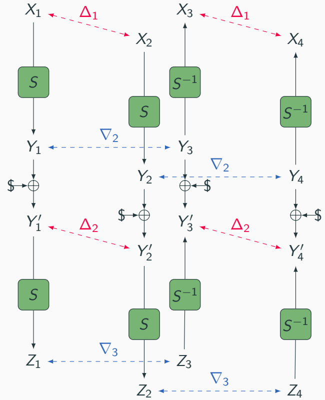
$$\begin{tikzcd} \Delta_1 \arrow[r] \arrow[dr] & \Delta_2 \\ \nabla_1 & \nabla_2 \arrow[l] \end{tikzcd}$$

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

Generalized BCT Framework (GBCT) - II

- Double Boomerang Connectivity Table ($\mathtt{DBCT}$) [HB21]

$$
\begin{tikzcd}
\Delta_1 \arrow[r] \arrow[dr] & \Delta_2 \arrow[dr] & \\
 & \nabla_2^* \arrow[r] & \nabla_3
\end{tikzcd}
$$

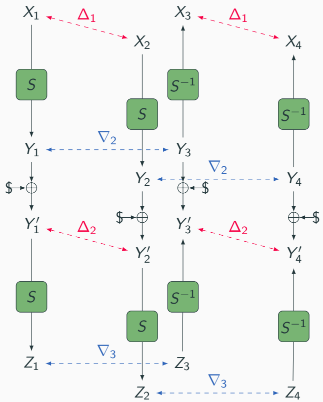


✔ $\mathtt{DBCT}^{\vdash}(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \mathtt{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \mathtt{LBCT}(\Delta_2, \nabla_2, \nabla_3)$

Generalized BCT Framework (GBCT) - II

- Double Boomerang Connectivity Table ($\mathtt{DBCT}$) [HB21]

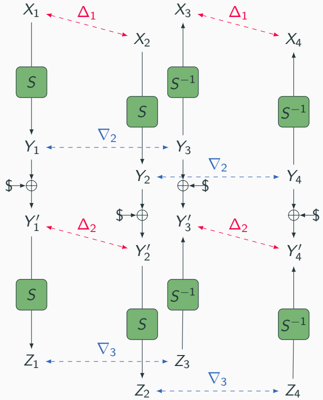
$$
\begin{tikzcd}
\Delta_1 \arrow[r] \arrow[dr] & \Delta_2^* \arrow[dr] & \\
& \nabla_2 \arrow[r] & \nabla_3
\end{tikzcd}
$$



- ✓ $\mathtt{DBCT}^{\vdash}(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \mathtt{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \mathtt{LBCT}(\Delta_2, \nabla_2, \nabla_3)$
- ✓ $\mathtt{DBCT}^{\dashv}(\Delta_1, \nabla_2, \nabla_3) = \sum_{\Delta_2} \mathtt{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \mathtt{LBCT}(\Delta_2, \nabla_2, \nabla_3)$.

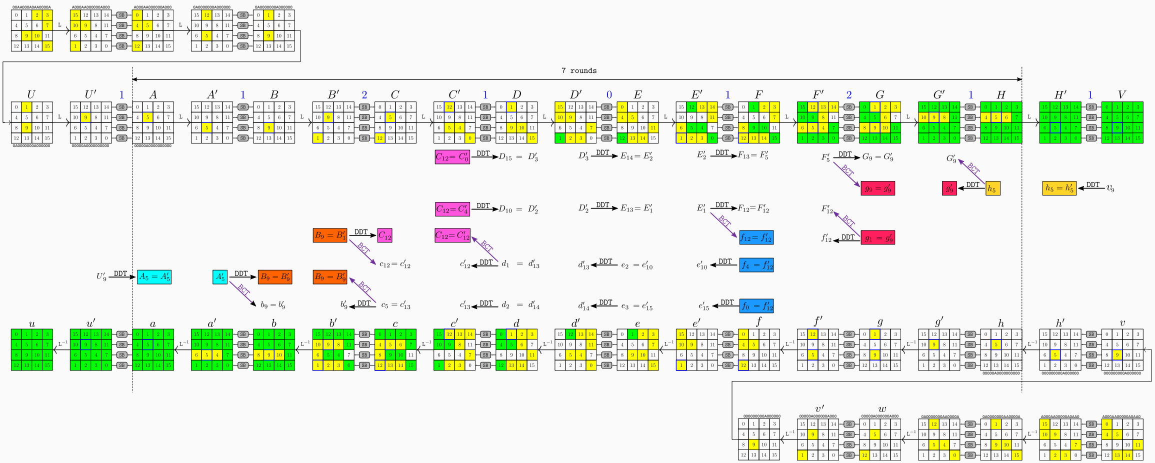
- Double Boomerang Connectivity Table (DBCT) [HB21]

$$\begin{tikzcd} \Delta_1 \arrow[r] \arrow[dr] & \Delta_2^* \arrow[dr] & \\ & \nabla_2^* \arrow[r] & \nabla_3 \end{tikzcd}$$

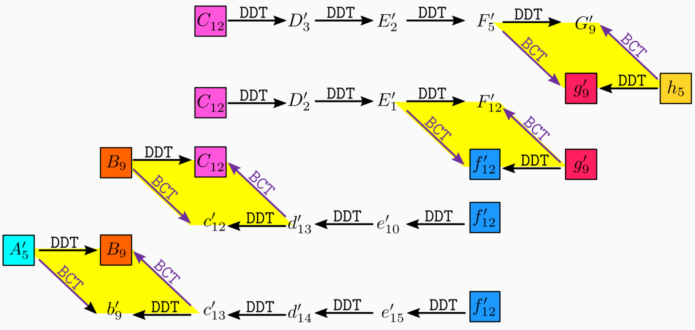


- ✓ $\mathtt{DBCT}^{\vdash}(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \mathtt{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \mathtt{LBCT}(\Delta_2, \nabla_2, \nabla_3)$
- ✓ $\mathtt{DBCT}^{\dashv}(\Delta_1, \nabla_2, \nabla_3) = \sum_{\Delta_2} \mathtt{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \mathtt{LBCT}(\Delta_2, \nabla_2, \nabla_3)$.
- ✓ $\mathtt{DBCT}(\Delta_1, \nabla_3) = \sum_{\Delta_2} \mathtt{DBCT}^{\vdash}(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \mathtt{DBCT}^{\dashv}(\Delta_1, \nabla_2, \nabla_3)$.

Application of GBCT [HB21]



Application of GBCT [HB21]



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

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

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

Differential-Linear (DL) Attack I [LH94]

Input: $E_K, (\Delta, \lambda), N, \boldsymbol{c} = \mathbb{C}(\Delta, \lambda)$

Output: 0: **real** cipher, 1: **ideal** cipher

Initialize a counter list $V[z] \leftarrow 0$ for $z \in \{0, 1\}$;

for $i = 0, \ldots, N-1$ **do**

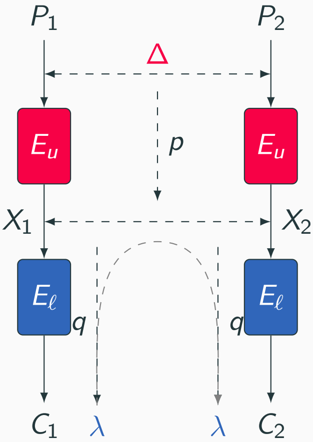
- $P_1 \xleftarrow{\$} \mathbb{F}_2^n$;
- $b_1 \leftarrow \lambda \cdot E_K(P_1)$;
- $P_2 \leftarrow P_1 \oplus \Delta$;
- $b_2 \leftarrow \lambda \cdot E_K(P_2)$;
- $V[b_1 \oplus b_2] \leftarrow V[b_1 \oplus b_2] + 1$;

if $V[0] \sim \mathcal{N}(\mu = N\frac{1+\boldsymbol{c}}{2}, \sigma^2 = N\frac{1-\boldsymbol{c}^2}{4})$ **then**

- **return** 0; `// real cipher`

else

- **return** 1; `// ideal cipher`

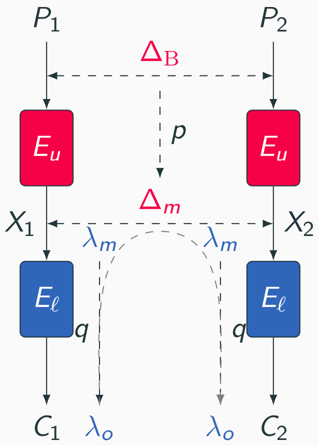


$\boldsymbol{c} = 2 \cdot \mathbb{P}(\boldsymbol{\lambda} \cdot C_1 \oplus \boldsymbol{\lambda} \cdot C_2 = 0) - 1$

Differential-Linear Attacks

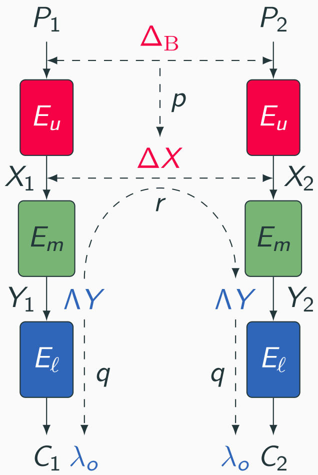
Differential-Linear (DL) Attack II [LH94]

- $p = \mathbb{P}(\Delta_{\mathrm{B}} \xrightarrow{E_u} \Delta_m)$
- $q = \mathbb{C}(\lambda_m \xrightarrow{E_\ell} \lambda_{\mathrm{F}}) = 2 \cdot \mathbb{P}(\lambda_m \cdot X \oplus \lambda_{\mathrm{F}} \cdot E_\ell(X) = 0) - 1$
- 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 $\Delta X \neq \Delta_m$
- $\mathcal{C} = \mathbb{C}\left(\lambda_{\mathrm{F}} \cdot \Delta C\right) \approx (-1)^{\lambda_m \cdot \Delta_m} \cdot pq^2 = \pm pq^2$
- Time/Data complexity: $\mathcal{O}(\mathcal{C}^{-2})$



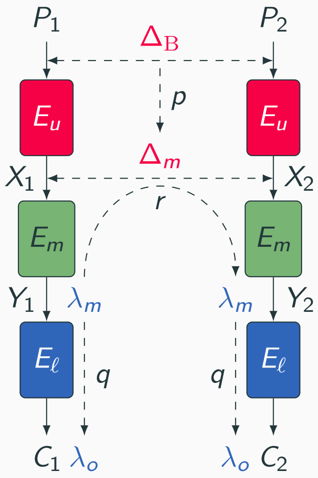
Sandwich Framework for DL Attack [BLN14; DKS14; Bar+19]

- $\mathbb{R}(\Delta X, \Lambda Y) = \mathbb{C}\left(\Lambda Y \cdot E_m(X) \oplus \Lambda Y \cdot E_m(X \oplus \Delta X)\right)$
- $\mathbb{C}(\lambda_{\mathrm{F}} \cdot \Delta C) = \sum_{\Delta X, \Lambda Y} \mathbb{P}(\Delta_{\mathrm{B}}, \Delta X) \cdot \mathbb{R}(\Delta X, \Lambda Y) \cdot \mathbb{C}^2(\Lambda Y, \lambda_{\mathrm{F}})$

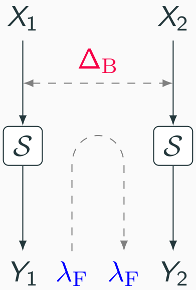


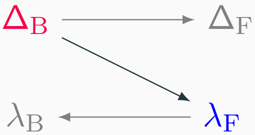
Sandwich Framework for DL Attack [BLN14; DKS14; Bar+19]

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



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



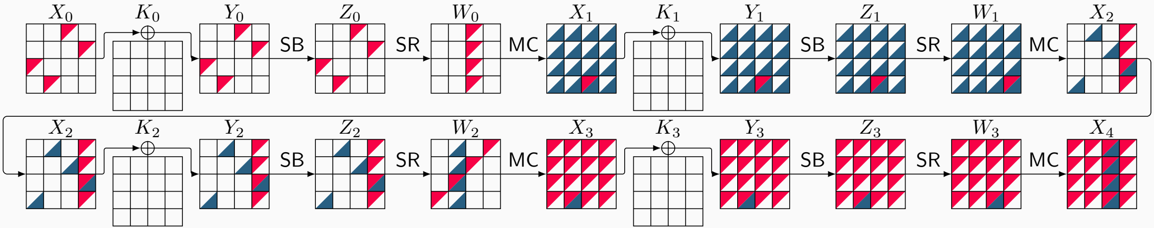


$$\mathtt{DLCT}_b(\Delta_{\mathrm{B}}, \lambda_{\mathrm{F}}) = \{x \in \mathbb{F}_2^n : \ \lambda_{\mathrm{F}} \cdot S(x) \oplus \lambda_{\mathrm{F}} \cdot S(x \oplus \Delta_{\mathrm{B}}) = b\}$$

$$\mathtt{DLCT}(\Delta_{\mathrm{B}}, \lambda_{\mathrm{F}}) = |\mathtt{DLCT}_0(\Delta_{\mathrm{B}}, \lambda_{\mathrm{F}})| - |\mathtt{DLCT}_1(\Delta_{\mathrm{B}}, \lambda_{\mathrm{F}})|$$

$$\mathbb{C}_{\mathtt{DLCT}}\left(\Delta_{\mathrm{B}}, \lambda_{\mathrm{F}}\right) = 2^{-n} \cdot \mathtt{DLCT}\left(\Delta_{\mathrm{B}}, \lambda_{\mathrm{F}}\right)$$

A 4-round DL Distinguisher for AES



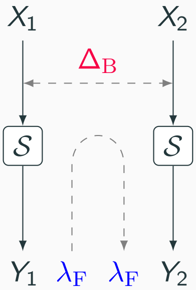
$r_u = 1, r_m = 3, r_\ell = 0,\ \ p = 2^{-24.00},\ \ r = 2^{-7.66}, q^2 = 1,\ \mathbb{C} = prq^2 = \mathbf{2^{-31.66}}$

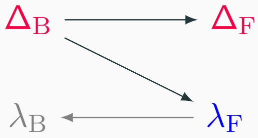
ΔX_0	`00005200000000f58f000000007b0000`	ΔX_1	`0000000000000000000000000000b400`
ΓX_4	`0032000000ab00000066000000980000`		-

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

Generalized DLCT Framework

Upper Differential-Linear Connectivity Table (UDLCT)



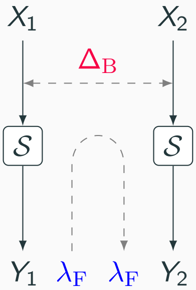


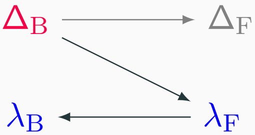
$$\mathtt{UDLCT}_b(\Delta_{\mathrm{B}}, \Delta_{\mathrm{F}}, \lambda_{\mathrm{F}}) = \{x \in \mathbb{F}_2^n : \ S(x) \oplus S(x \oplus \Delta_{\mathrm{B}}) = \Delta_{\mathrm{F}} \text{ and } \lambda_{\mathrm{F}} \cdot \Delta_{\mathrm{F}} = b\}$$

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

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

Lower Differential-Linear Connectivity Table (LDLCT)



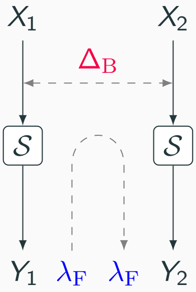


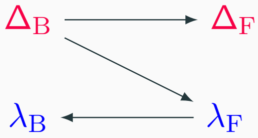
$$\mathtt{LDLCT}_b(\Delta_{\mathrm{B}}, \lambda_{\mathrm{B}}, \lambda_{\mathrm{F}}) = \{x \in \mathbb{F}_2^n : \ \lambda_{\mathrm{B}} \cdot \Delta_{\mathrm{B}} \oplus \lambda_{\mathrm{F}} \cdot S(x) \oplus \lambda_{\mathrm{F}} \cdot S(x \oplus \Delta_{\mathrm{B}}) = b\}$$

$$\mathtt{LDLCT}(\Delta_{\mathrm{B}}, \lambda_{\mathrm{B}}, \lambda_{\mathrm{F}}) = |\mathtt{LDLCT}_0(\Delta_{\mathrm{B}}, \lambda_{\mathrm{B}}, \lambda_{\mathrm{F}})| - |\mathtt{LDLCT}_1(\Delta_{\mathrm{B}}, \lambda_{\mathrm{B}}, \lambda_{\mathrm{F}})|$$

$$\mathbb{C}_{\mathtt{LDLCT}}\left(\Delta_{\mathrm{B}}, \lambda_{\mathrm{B}}, \lambda_{\mathrm{F}}\right) = 2^{-n} \cdot \mathtt{LDLCT}\left(\Delta_{\mathrm{B}}, \lambda_{\mathrm{B}}, \lambda_{\mathrm{F}}\right)$$

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





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

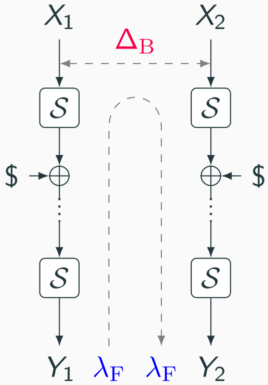
$$\mathtt{EDLCT}(\Delta_{\rm B}, \Delta_{\rm F}, \lambda_{\rm B}, \lambda_{\rm F}) = |\mathtt{EDLCT}_0(\Delta_{\rm B},\Delta_{\rm F},\lambda_{\rm B},\lambda_{\rm F})| - |\mathtt{EDLCT}_1(\Delta_{\rm B},\Delta_{\rm F},\lambda_{\rm B},\lambda_{\rm F})|$$

$$\mathbb{C}_{\mathtt{EDLCT}}\left(\Delta_{\rm B}, \Delta_{\rm F}, \lambda_{\rm B}, \lambda_{\rm F}\right) = 2^{-n} \cdot \mathtt{EDLCT}\left(\Delta_{\rm B}, \Delta_{\rm F}, \lambda_{\rm B}, \lambda_{\rm F}\right)$$

Double Differential-Linear Connectivity Table (DDLCT)

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

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



Generalized DLCT Framework (GBCT)

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

$$\begin{tikzcd} \Delta_{\mathrm{B}} \arrow[r] \arrow[dr] & \Delta_{\mathrm{F}} \\ \lambda_{\mathrm{B}} \arrow[r] & \lambda_{\mathrm{F}} \end{tikzcd}$$

$\mathtt{DLCT}\,(\Delta_{\mathrm{B}}, \lambda_{\mathrm{F}})$

$$\begin{tikzcd} \Delta_{\mathrm{B}} \arrow[r] \arrow[dr] & \Delta_{\mathrm{F}} \\ \lambda_{\mathrm{B}} \arrow[r] & \lambda_{\mathrm{F}} \end{tikzcd}$$

$\mathtt{UDLCT}\,(\Delta_{\mathrm{B}}, \Delta_{\mathrm{F}}, \lambda_{\mathrm{F}})$

$$\begin{tikzcd} \Delta_{\mathrm{B}} \arrow[r] \arrow[dr] & \Delta_{\mathrm{F}} \\ \lambda_{\mathrm{B}} \arrow[r] & \lambda_{\mathrm{F}} \end{tikzcd}$$

$\mathtt{LDLCT}\,(\Delta_{\mathrm{B}}, \lambda_{\mathrm{B}}, \lambda_{\mathrm{F}})$

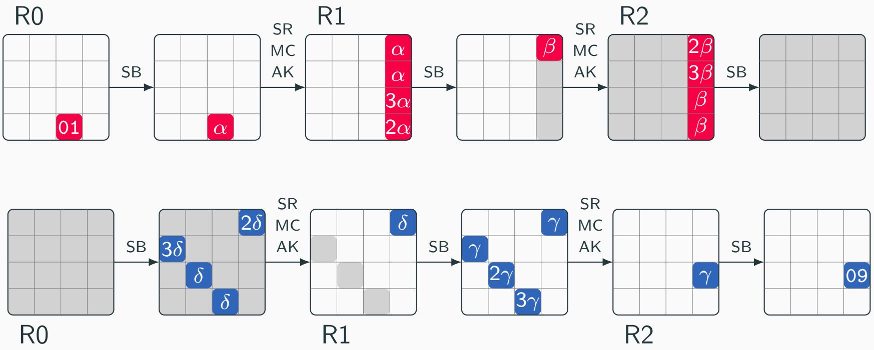
$$\begin{tikzcd} \Delta_{\mathrm{B}} \arrow[r] \arrow[dr] & \Delta_{\mathrm{F}} \\ \lambda_{\mathrm{B}} \arrow[r] & \lambda_{\mathrm{F}} \end{tikzcd}$$

$\mathtt{EDLCT}\,(\Delta_{\mathrm{B}}, \Delta_{\mathrm{F}}, \lambda_{\mathrm{B}}, \lambda_{\mathrm{F}})$

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

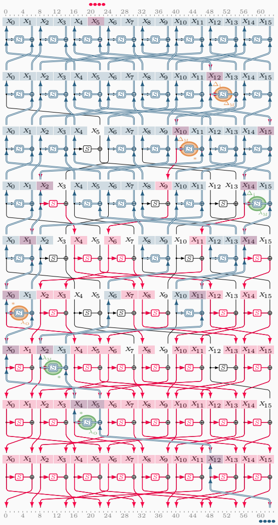
$\mathtt{DDLCT}\,(\Delta_{\mathrm{B}}, \lambda_{\mathrm{F}})$

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



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

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



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

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

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

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

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

Properties of Generalized $\mathtt{DLCT}$ Tables - I

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

Properties of Generalized DLCT Tables - II

- $\mathtt{DDLCT}(\Delta_\mathrm{B}, \lambda_\mathrm{F}) = 2^{-n} \cdot \sum_{\Delta_m} \sum_{\lambda_m} \mathtt{UDLCT}\,(\Delta_\mathrm{B}, \Delta_m, \lambda_m) \cdot \mathtt{LDLCT}\,(\Delta_m, \lambda_m, \lambda_\mathrm{F})$

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

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

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

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

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

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

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

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

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

$$\lambda_{\mathrm{B}} = (1, ?, ?, 1) \xleftarrow{S} \lambda_{\mathrm{F}} = (0, 1, 0, 0)$$

$$\lambda_{\mathrm{B}} = (1, 1, ?, ?) \xleftarrow{S} \lambda_{\mathrm{F}} = (1, 0, 0, 0)$$

$$\lambda_{\mathrm{B}} = (0, ?, ?, ?) \xleftarrow{S} \lambda_{\mathrm{F}} = (1, 1, 0, 0)$$

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

$$\Delta_{\mathrm{B}} = (0,0,0,1) \xrightarrow{S} \Delta_{\mathrm{F}} = (?,1,?,?)$$

$$\Delta_{\mathrm{B}} = (0,1,0,0) \xrightarrow{S} \Delta_{\mathrm{F}} = (1,?,?,?)$$

$$\Delta_{\mathrm{B}} = (1,0,0,0) \xrightarrow{S} \Delta_{\mathrm{F}} = (1,1,?,?)$$

$$\Delta_{\mathrm{B}} = (1,0,0,1) \xrightarrow{S} \Delta_{\mathrm{F}} = (?,0,?,?)$$

$$\Delta_{\mathrm{B}} = (1,1,0,0) \xrightarrow{S} \Delta_{\mathrm{F}} = (0,?,?,?)$$

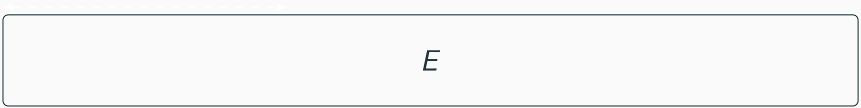
$$\lambda_{\mathrm{B}} = (1,?,?,1) \xleftarrow{S} \lambda_{\mathrm{F}} = (0,1,0,0)$$

$$\lambda_{\mathrm{B}} = (1,1,?,?) \xleftarrow{S} \lambda_{\mathrm{F}} = (1,0,0,0)$$

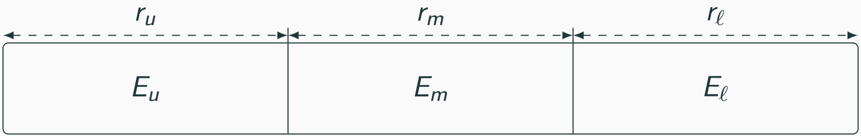
$$\lambda_{\mathrm{B}} = (0,?,?,?) \xleftarrow{S} \lambda_{\mathrm{F}} = (1,1,0,0)$$

Automatic Tools to Search for DL Distinguishers

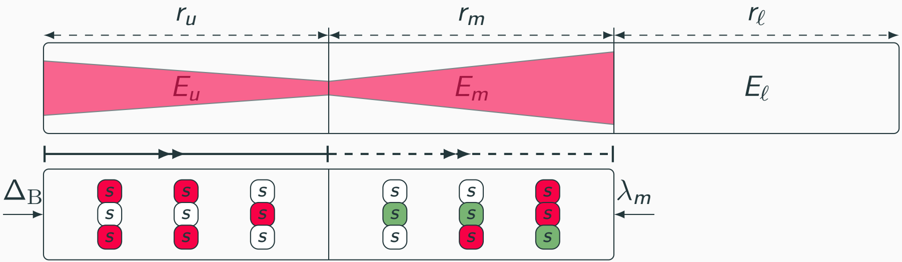
Overview of Our Method to Search for Distinguishers



Overview of Our Method to Search for Distinguishers

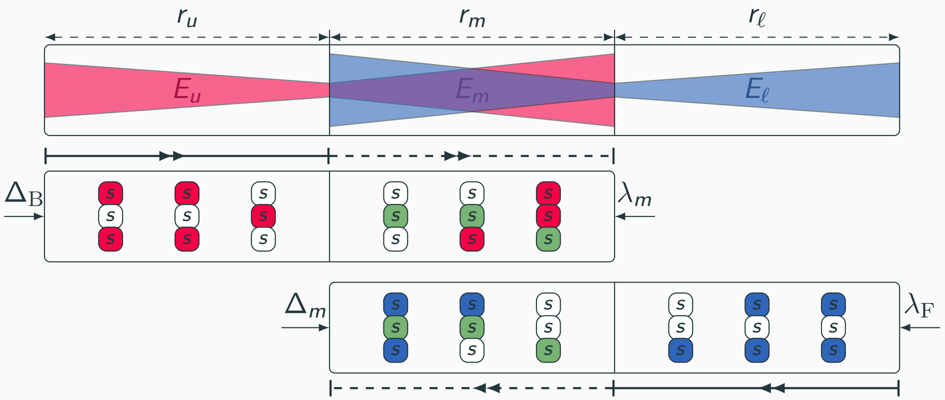


Overview of Our Method to Search for Distinguishers



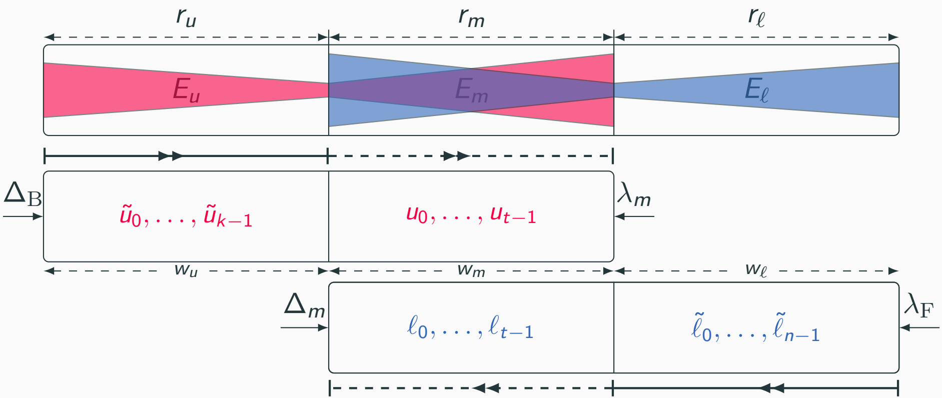
differentially active S-box linearly active S-box common active S-box

Overview of Our Method to Search for Distinguishers



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

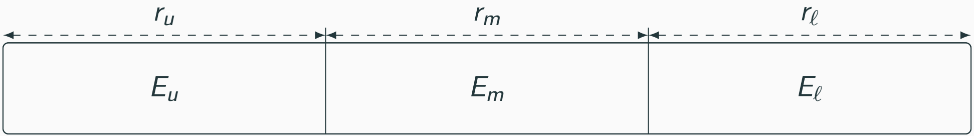
Overview of Our Method to Search for Distinguishers

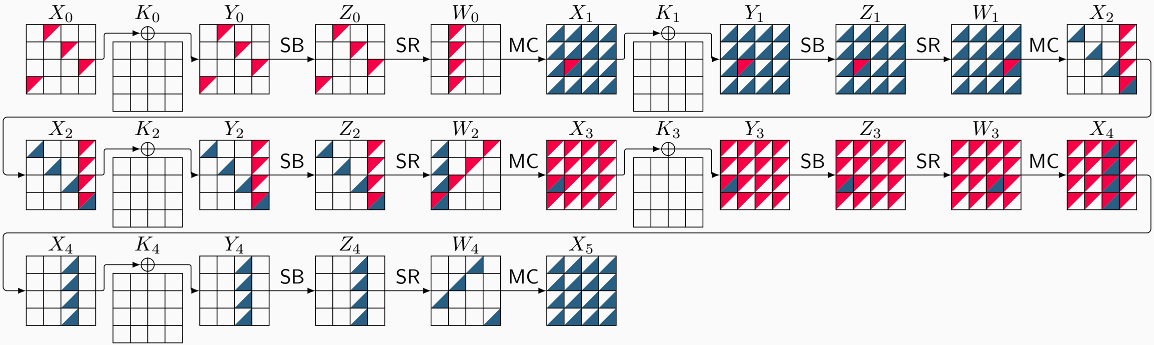


$$\min\left(\sum_{i=0}^{k-1} w_u \cdot \tilde{u}_i + \sum_{j=0}^{t-1} w_m \cdot \mathtt{bool2int}(\ell_j + u_j = 2) + \sum_{k=0}^{n-1} w_\ell \cdot \tilde{\ell}_k\right)$$

Usage of Our Tool

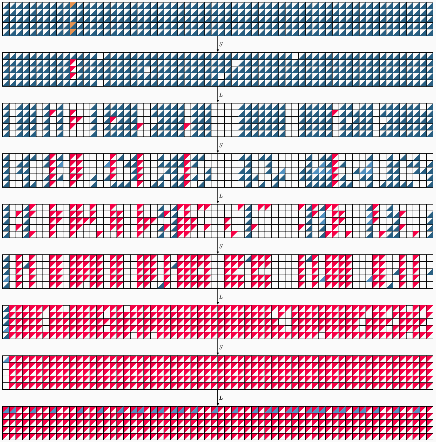
```
python3 attack.py -RU 6 -RM 10 -RL 6
```



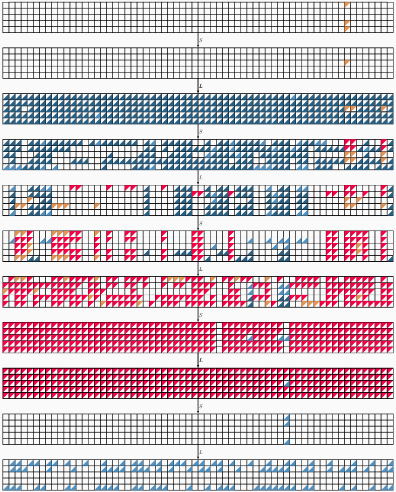


$r_0 = 1, r_m = 3, r_1 = 1,\ p = 2^{-24.00}, r = 2^{-7.66},\ q^2 = 2^{-24.00},\ prq^2 = 2^{-55.66}$			
ΔX_0	`001c00000000e200000000dfb3000000`	ΔX_1	`0000000000000000f700000000000000`
ΓX_4	`00000000000000006700000000000000`	ΓX_5	`21d3814d93b1ef228e923507f67383fd`

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



$\mathbb{C} = 1$



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

Results: 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	-	$2^{59.24}$	-

Results: 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	-	$2^{58.30}$	-

Results: 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	$2^{76.03}$	$2^{66.86}$	$2^{9.17}$

Results: 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

Results: Application to Simeck

- 💻: Experimentally verified

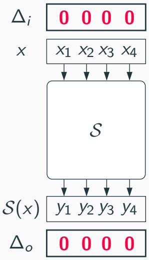
Cipher	#R	$\mathbb{C}$	💻	Ref.
Simeck-32	7	$\mathbf{1}$	✓	This work
	14	$2^{-16.63}$		[ZWH24]
	14	$\mathbf{2^{-13.92}}$	✓	This work

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

Bit-Wise Model for Finding ID/ZC/Integral 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

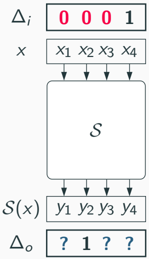
$\Delta_i \setminus \Delta_o$	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2
2	0	2	0	2	0	0	0	4	0	2	2	0	0	0	2	2
3	0	2	0	2	0	0	4	0	0	2	2	0	0	0	2	2
4	0	0	0	0	0	0	0	0	0	0	4	4	2	2	2	2
5	0	0	0	0	2	2	2	2	0	0	4	4	0	0	0	0
6	0	2	0	2	0	4	0	0	0	2	2	0	2	2	0	0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



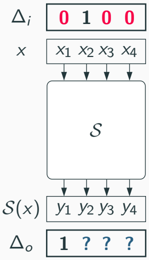
$\Delta_i \setminus \Delta_o$	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2
2	0	2	0	2	0	0	0	4	0	2	2	0	0	0	2	2
3	0	2	0	2	0	0	4	0	0	2	2	0	0	0	2	2
4	0	0	0	0	0	0	0	0	0	0	4	4	2	2	2	2
5	0	0	0	0	2	2	2	2	0	0	4	4	0	0	0	0
6	0	2	0	2	0	4	0	0	0	2	2	0	2	2	0	0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

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

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

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

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

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

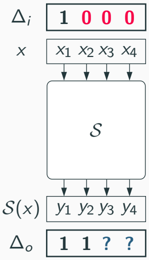
$$\Delta_i \neq (0,0,0,0) \xrightarrow{S} \Delta_o \neq (0,0,0,0)$$

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

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

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

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

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

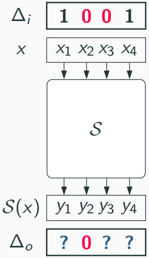
$$\Delta_i = (0,0,0,1) \xrightarrow{S} \Delta_o = (?,1,?,?)$$

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

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

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

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

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

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

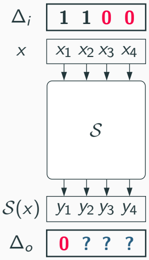
$$\Delta_i = (0,1,0,0) \xrightarrow{S} \Delta_o = (1,?,?,?)$$

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

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

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

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

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

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

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

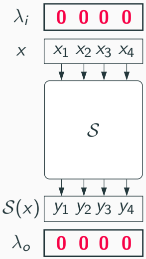
$$\Delta_i = (1,0,0,0) \xrightarrow{S} \Delta_o = (1,1,?,?)$$

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

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

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



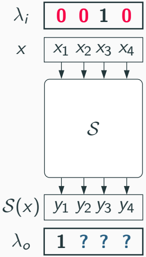
$\lambda_i \setminus \lambda_o$	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	4	-4	0	-8	-4	-4	0	0	4	-4	-8	0	4	4
2	0	0	0	0	0	0	0	0	0	8	8	0	0	8	-8	0
3	0	-8	4	4	0	0	-4	4	0	0	-4	4	-8	0	-4	-4
4	0	4	0	4	0	4	8	-4	0	4	0	4	-8	-4	0	4
5	0	4	-4	-8	0	-4	-4	0	0	4	-4	8	0	-4	-4	0
6	0	-4	8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

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

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

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

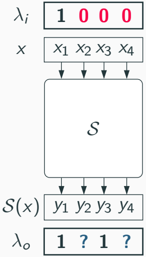
$$\lambda_i = (0,0,0,0) \xrightarrow{S} \lambda_o = (0,0,0,0)$$

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

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

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



$\lambda_i \backslash \lambda_o$	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

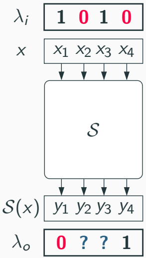
$$\lambda_i = (0,0,0,0) \xrightarrow{S} \lambda_o = (0,0,0,0)$$

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

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

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

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

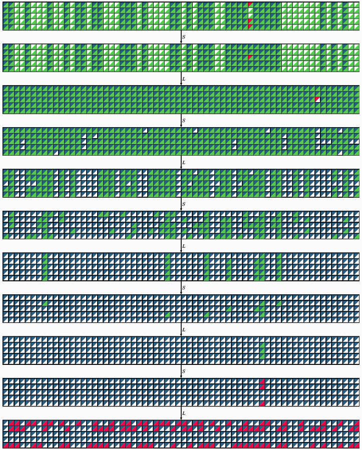
- For each bit position, we define an integer variable with domain $\{\mathbf{0}, \mathbf{1}, \mathbf{-1}\}$.
- Define CP constraints to model the propagation of deterministic bit-wise trails.

S-box

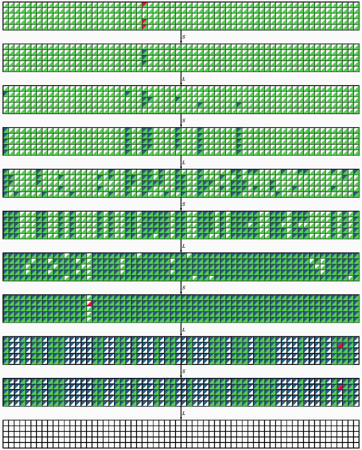
Assume that $x[i], y[i]$ are integer variables with domain $\{-1, 0, 1\}$ to encode the input and output differences at the i-th bit position, respectively. The valid deterministic differential transitions satisfy the following:

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

Example: ID/ZC Distinguishers for 5 Rounds of Ascon [Hos+24]



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



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

Generic and Common Techniques in Symmetric-Key Attacks

Guess-and-Determine: A Common Step in Most Attacks

Guess-and-Determine

Given a set of variables and a set of relations between them, find the smallest subset of variables guessing the value of which uniquely determines the value of the remaining variables.

Example

- $u, \ldots, z \in \mathbb{F}_2^{32}$
- F, G, H: bijective functions
- $c_1, \ldots, c_5$: constants

$$\begin{cases} F(u+v) \oplus G(x) \oplus y \oplus (z \lll 7) & = c_1 \\ G(u \oplus w) + (y \lll 3) + z & = c_2 \\ F(w \oplus x) + y \oplus z & = c_3 \\ F(u) \oplus G(w+z) & = c_4 \\ (F(u) \times G(w \lll 7)) + H(z \oplus v) & = c_5 \end{cases}$$

Guess-and-Determine: A Common Step in Most Attacks

Guess-and-Determine

Given a set of variables and a set of relations between them, find the smallest subset of variables guessing the value of which uniquely determines the value of the remaining variables.

Example

- Guess w, z
- Determine u (4), y (2)
- Determine x (3), v (5)

$$\begin{cases} F(u+v) \oplus G(x) \oplus y \oplus (z \lll 7) & = c_1 \\ G(u \oplus w) + (y \lll 3) + z & = c_2 \\ F(w \oplus x) + y \oplus z & = c_3 \\ F(u) \oplus G(w+z) & = c_4 \\ (F(u) \times G(w \lll 7)) + H(z \oplus v) & = c_5 \end{cases}$$

Symmetric and Implication Relations

Assumption: Relations are symmetric or implication

- **Implication relations**: $x_1, \ldots, x_n \Rightarrow y$
- **Symmetric relations**: $[x_1, \ldots, x_n]$

Example

Assume that $x, y, z, k \in \mathbb{F}_2^{32}$, and $F : \mathbb{F}_2^{32} \to \mathbb{F}_2^{32}$ is bijective:

$$z = x \times y$$

$$x, y \Rightarrow z$$

$$z = F(x + k) \oplus y$$

$$[x, y, z, k]$$

System of Equations

$$E : \begin{cases} e_1 : F(u + v) \oplus G(x) \oplus y \oplus (z \lll 7) & = c_1 \\ e_2 : G(u \oplus w) + (y \lll 3) + z & = c_2 \\ e_3 : F(w \oplus x) + y \oplus z & = c_3 \\ e_4 : F(u) \oplus G(w + z) & = c_4 \\ e_5 : (F(u) \times G(w \lll 7)) + H(z \oplus v) & = c_5 \end{cases}$$

$$X = \{u, v, w, x, y, z\}, \;\; E = \{e_1, \ldots, e_5\}$$

System of Equations $\Rightarrow$ System of Relations

$$E : \begin{cases} e_1 : F(u+v) \oplus G(x) \oplus y \oplus (z \lll 7) & = c_1 \\ e_2 : G(u \oplus w) + (y \lll 3) + z & = c_2 \\ e_3 : F(w \oplus x) + y \oplus z & = c_3 \\ e_4 : F(u) \oplus G(w+z) & = c_4 \\ e_5 : (F(u) \times G(w \lll 7)) + H(z \oplus v) & = c_5 \end{cases}$$

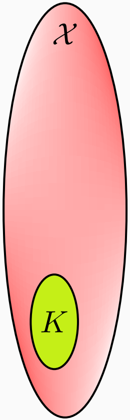
$$X = \{u, v, w, x, y, z\},\ E = \{e_1, \dots, e_5\}$$

$$\mathcal{R} : \begin{cases} r_1 : [u, v, x, y, z], & r_2 : [u, w, y, z] \\ r_3 : [w, x, y, z], & r_4 : [u, w, z] \\ r_5 : u, w \Rightarrow t, & r_6 : [t, z, v] \end{cases}$$

$$\mathcal{X} = \{u, v, w, x, y, z, t\},\ \mathcal{R} = \{r_1, \dots, r_6\}$$

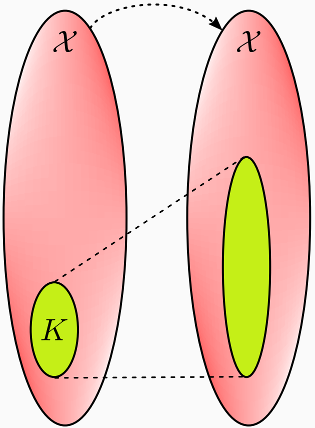
Knowledge Propagation

- $(\mathcal{X}, \mathcal{R})$
- $K \subseteq \mathcal{X}$
- K is initially known



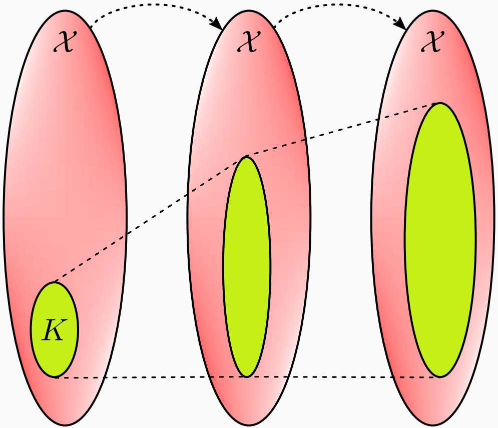
Knowledge Propagation

- $(\mathcal{X}, \mathcal{R})$
- $K \subseteq \mathcal{X}$
- K is initially known



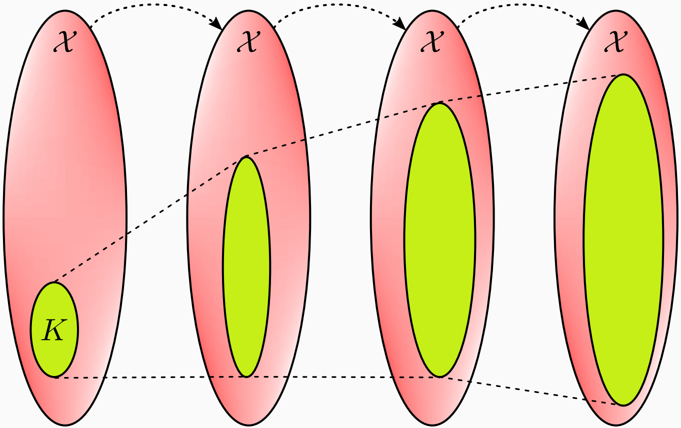
Knowledge Propagation

- $(\mathcal{X}, \mathcal{R})$
- $K \subseteq \mathcal{X}$
- K is initially known



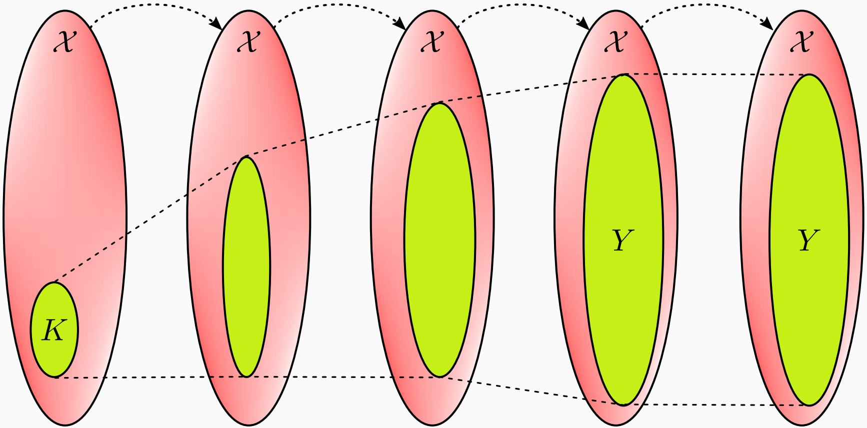
Knowledge Propagation

- $(\mathcal{X}, \mathcal{R})$
- $K \subseteq \mathcal{X}$
- K is initially known



Knowledge Propagation

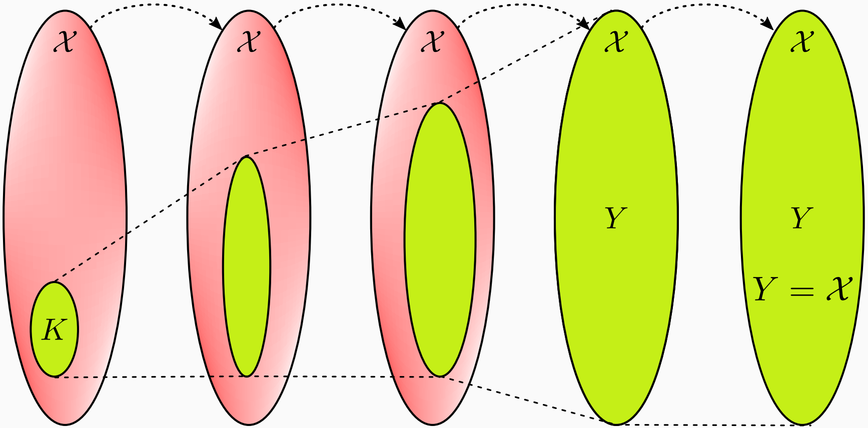
- $(\mathcal{X}, \mathcal{R})$
- $K \subseteq \mathcal{X}$
- K is initially known



$Y = \texttt{Propagate}(K)$

Knowledge Propagation

- $(\mathcal{X}, \mathcal{R})$
- $K \subseteq \mathcal{X}$
- K is initially known
- K is known



If $\mathcal{X} = \texttt{Propagate}(K)$, then we say K is a *Guess Basis*

Naive Approach for GD

Given a system of relations $(\mathcal{X}, \mathcal{R})$, where $|\mathcal{X}| = n$, is there any guess basis of size $\leq m$?

Naive Approach for GD

Given a system of relations $(\mathcal{X}, \mathcal{R})$, where $|\mathcal{X}| = n$, is there any guess basis of size $\leq m$?

Brute-force

- For $k = 1 \to m$
    - For each subset $K \subseteq \mathcal{X}$, where $|K| = k$:
        - If `Propagate(K)` $= \mathcal{X}$ then return K

Naive Approach for GD

Given a system of relations $(\mathcal{X}, \mathcal{R})$, where $|\mathcal{X}| = n$, is there any guess basis of size $\leq m$?

Brute-force

- For $k = 1 \to m$
  - For each subset $K \subseteq \mathcal{X}$, where $|K| = k$:
    - If `Propagate(K)` $= \mathcal{X}$ then return K

- Time complexity $\approx \sum_{k=1}^{m} \binom{n}{k}$
- Exponential with respect to both n and m

CP-Based Approach to Solve GD Problem

1. Convert the system of equations to a system of relations
    - We can apply a preprocessing step here (Gaussian elimination)
2. Convert the problem of finding a minimal guess basis to a CP problem
3. Employ the state-of-the-art CP solvers to solve the problem

Convert GD to a CP Problem

$r_0 : [\ x, y, z]$

$r_1 : [z, w, y]$

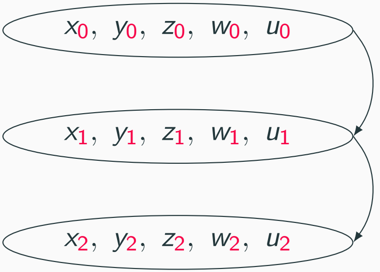
$r_2 : [w, x, u]$

Convert GD to a CP Problem

$$r_0 : [\, x, y, z]$$

$$r_1 : [z, w, y]$$

$$r_2 : [w, x, u]$$



- Fix the number of steps in knowledge propagation
- $X = \{\, x_i, y_i, z_i, w_i, u_i : 0 \leq i \leq 2\}$
- $x_i = 1$ iff x is known after the ith step of knowledge propagation, otherwise $x_i = 0$
- Initialize the set of constraints: $\mathcal{C} \leftarrow \emptyset$

Convert GD to a CP Problem

$r_0 : [x, y, z]$

$r_1 : [z, w, y]$

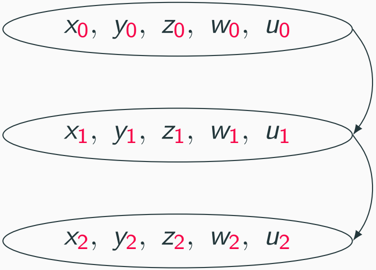
$r_2 : [w, x, u]$

$X \leftarrow X \cup \{x_{0,0}, x_{0,1}\}$

$\mathcal{C} \leftarrow \mathcal{C} \cup \{x_{0,0} = y_0 \wedge z_0\}$

$\mathcal{C} \leftarrow \mathcal{C} \cup \{x_{0,1} = w_0 \wedge u_0\}$

$\mathcal{C} \leftarrow \mathcal{C} \cup \{x_1 = x_{0,0} \vee x_{0,1}\}$



Convert GD to a CP Problem

$r_0 : [x, y, z]$

$r_1 : [z, w, y]$

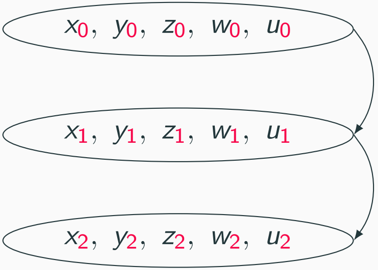
$r_2 : [w, x, u]$

$$X \leftarrow X \cup \{y_{0,0}, y_{0,1}\}$$

$$\mathcal{C} \leftarrow \mathcal{C} \cup \{y_{0,0} = x_0 \wedge z_0\}$$

$$\mathcal{C} \leftarrow \mathcal{C} \cup \{y_{0,1} = z_0 \wedge w_0\}$$

$$\mathcal{C} \leftarrow \mathcal{C} \cup \{y_1 = y_{0,0} \vee y_{0,1}\}$$

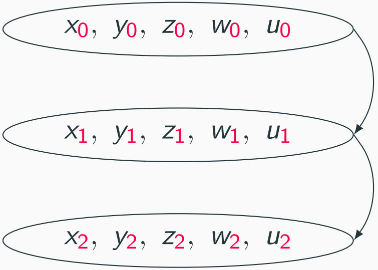


Convert GD to a CP Problem

$r_0 : [\ x, y, z]$

$r_1 : [z, w, y]$

$r_2 : [w, x, u]$



- Do it for all variables and in each step
- All variables should be known at the last step:

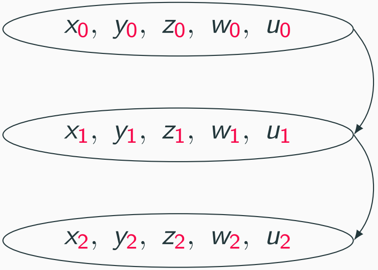
$$\mathcal{C} \leftarrow \mathcal{C} \cup \{x_2 \wedge y_2 \wedge z_2 \wedge w_2 \wedge u_2 = 1\}$$

Convert GD to a CP Problem

$r_0 : [\ x, y, z]$

$r_1 : [z, w, y]$

$r_2 : [w, x, u]$



$$\min\ x_0 + y_0 + z_0 + w_0 + u_0$$

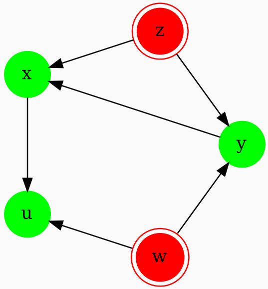
$$s.t.\ \text{all constraints in } \mathcal{C} \text{ are satisfied}$$

Convert GD to a CP Problem

$r_0 : [\ x, y, z]$

$r_1 : [z, w, y]$

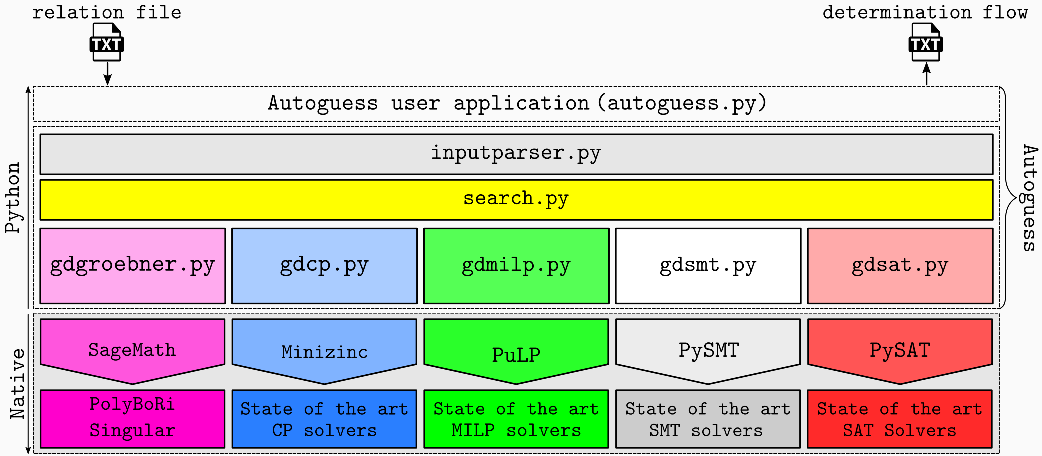
$r_2 : [w, x, u]$



$$\min\ x_0 + y_0 + z_0 + w_0 + u_0$$

$$s.t. \text{ all constraints in } \mathcal{C} \text{ are satisfied}$$

Autoguess



: https://github.com/hadipourh/autoguess

GD Attack on 1 to 3 Rounds of AES With 1 Known Plaintext