SCIENCE
PASSION
TECHNOLOGY

Automated Methods in Cryptanalysis and Design of Symmetric-Key Cryptographic Primitives

Hosein Hadipour

Ph.D. Defense, Graz University of Technology

› hsn.hadipour@gmail.com

- Special thanks to my advisor, Maria Eichlseder.
- I am grateful to my examiners, Gregor Leander and María Naya-Plasencia.
- Thanks to the chair of the deefense meeting, Bernhard Aichernig.

Maria Eichlseder

María Naya-Plasencia

Gregor Leander

Outline

Research Gaps and Thesis Contributions

Challenges and Gaps

Automated cryptanalysis has advanced significantly in the past decade, **yet**:

- Accuracy and efficiency of current methods remain suboptimal.
- Many cryptanalytic techniques were not (fully) automated before this work.
- New cryptanalytic techniques demand novel automation tools.
- Emerging primitives necessitate new methodologies for automated cryptanalysis.
- The connections between attacks are not well used in automated discovery.

Our Contributions/Publications – I

- **Hosein Hadipour** and Maria Eichlseder. **Autoguess: A Tool for Finding Guess-and-Determine Attacks and Key Bridges**. **ACNS** 2022. Ed. by Giuseppe Ateniese and Daniele Venturi. Vol. 13269. LNCS. Springer, 2022, pp. 230–250. DOI: 10.1007/978-3-031-09234-3_12

- **Hosein Hadipour** and Maria Eichlseder. **Integral Cryptanalysis of WARP based on Monomial Prediction**. *IACR Trans. Symmetric Cryptol.* 2022.2 (2022), pp. 92–112. DOI: 10.46586/tosc.v2022.i2.92-112

- **Hosein Hadipour**, Marcel Nageler, and Maria Eichlseder. **Throwing Boomerangs into Feistel Structures: Application to CLEFIA, WARP, LBlock, LBlock-s and TWINE**. *IACR Trans. Symmetric Cryptol.* 2022.3 (2022), pp. 271–302. DOI: 10.46586/tosc.v2022.i3.271-302

Our Contributions/Publications – II

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

- **Hosein Hadipour** et al. **Improved Search for Integral, Impossible-Differential and Zero-Correlation Attacks: Application to Ascon, ForkSKINNY, SKINNY, MANTIS, PRESENT and QARMAv2**. *IACR Trans. Symmetric Cryptol.* 2024.1 (2024), pp. 234–325. DOI: 10.46586/tosc.v2024.i1.234-325

- **Hosein Hadipour** and Yosuke Todo. **Cryptanalysis of QARMAv2**. *IACR Trans. Symmetric Cryptol.* 2024.1 (2024), pp. 188–213. DOI: 10.46586/tosc.v2024.i1.188-213

Our Contributions/Publications – III

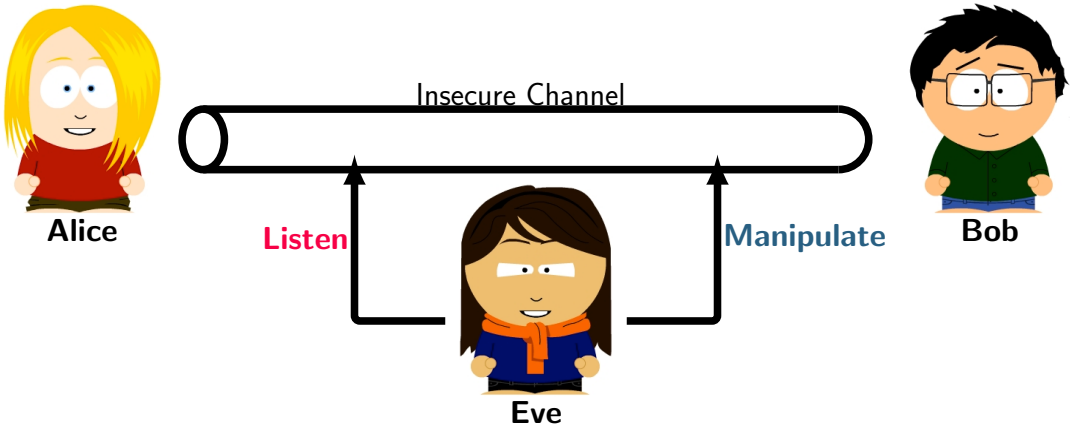
- **Hosein Hadipour**, Patrick Derbez, and Maria Eichlseder. **Revisiting Differential-Linear Attacks via a Boomerang Perspective With Application to AES, Ascon, CLEFIA, SKINNY, PRESENT, KNOT, TWINE, WARP, LBlock, Simeck, and SERPENT**. **CRYPTO** 2024. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14922. LNCS. Springer, 2024, pp. 290–305. DOI: 10.1007/978-3-031-68385-5_2

- Debasmita Chakraborty et al. **Finding Complete Impossible Differential Attacks on AndRX Ciphers and Efficient Distinguishers for ARX Designs**. *IACR Trans. Symmetric Cryptol.* 2024.3 (2024), pp. 84–176. DOI: 10.46586/tosc.v2024.i3.84-176

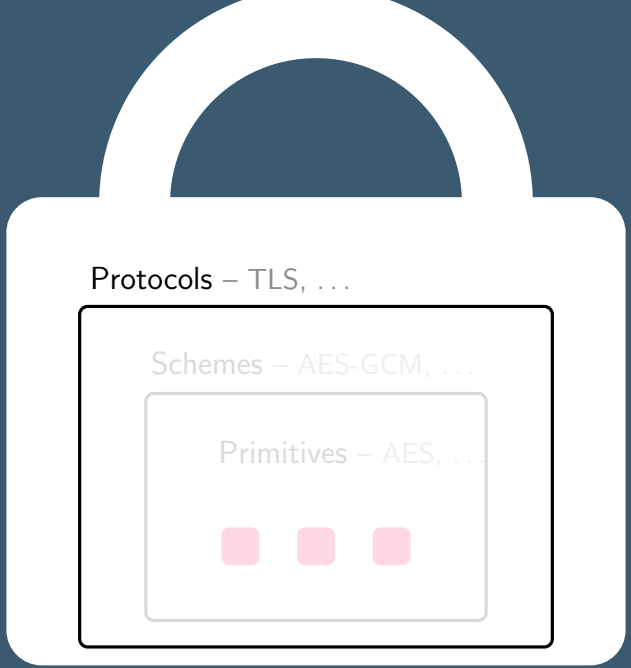
- Hadi Soleimany et al. **Practical Multiple Persistent Faults Analysis**. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2022.1 (2022), pp. 367–390. DOI: 10.46586/TCHES.V2022.I1.367-390

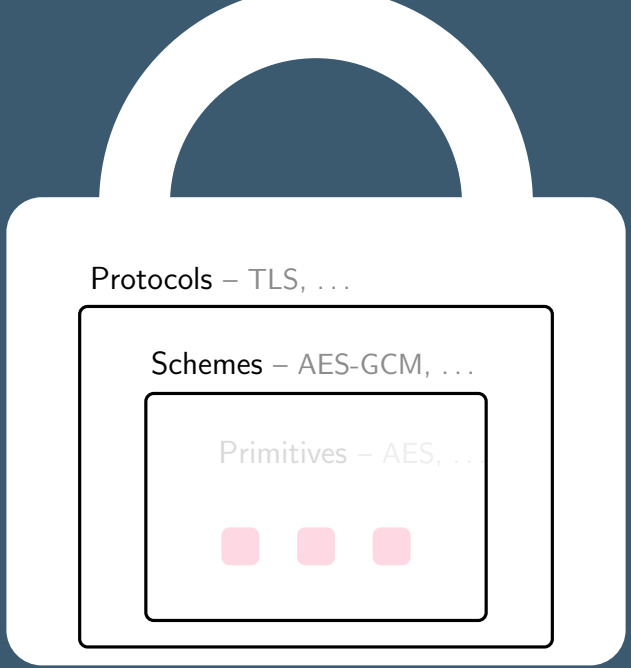
Background

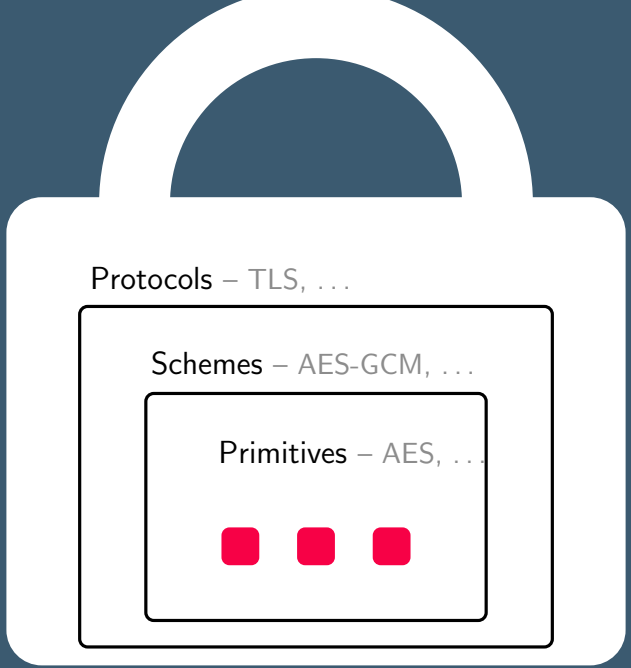
Cryptography – Primary Goals

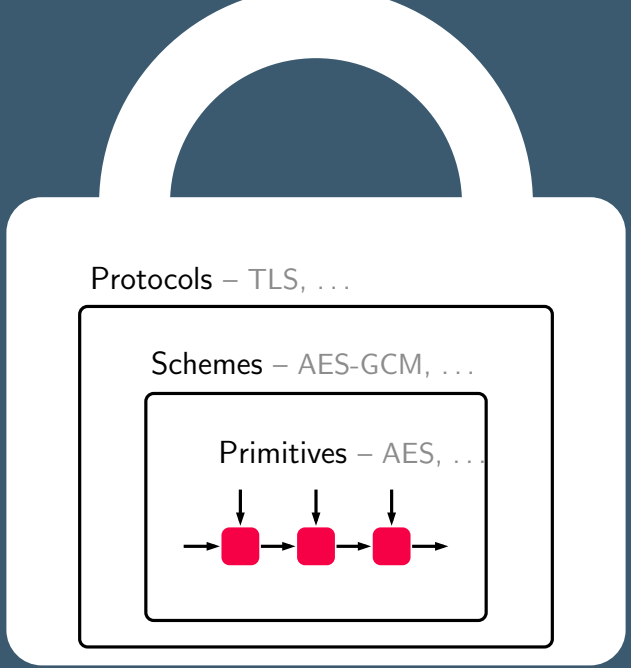


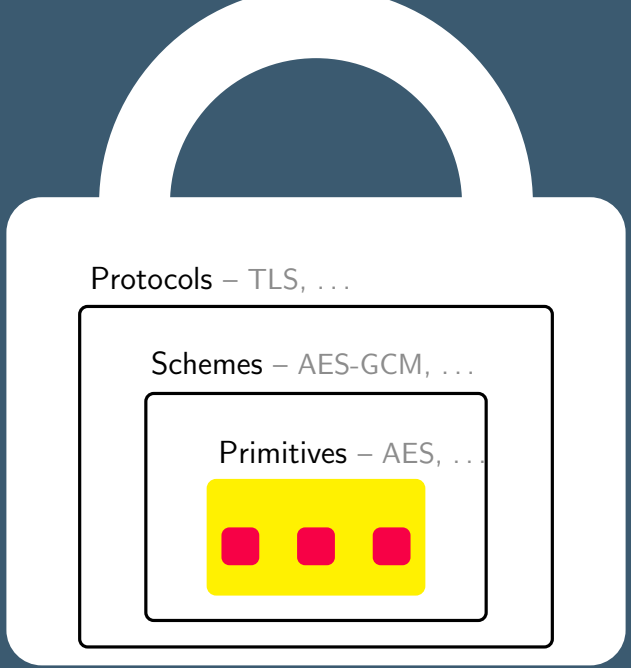
1- **Confidentiality** 2- **Integrity** 3- **Authentication**


Protocols – TLS, . . .
Schemes – AES-GCM, . . .
Primitives – AES, . . .


Protocols – TLS, . . .
Schemes – AES-GCM, . . .
Primitives – AES, . . .


Protocols – TLS, . . .
Schemes – AES-GCM, . . .
Primitives – AES, . . .

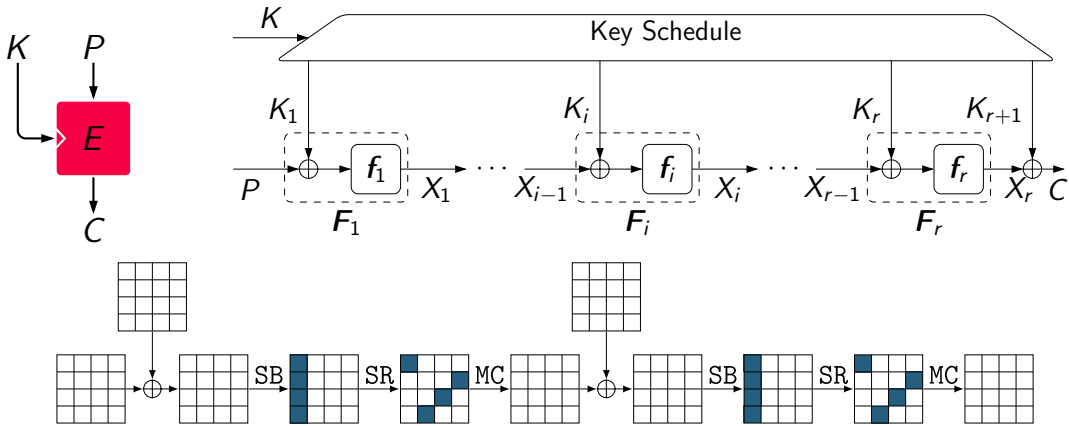

Protocols – TLS, . . .
Schemes – AES-GCM, . . .
Primitives – AES, . . .


Protocols – TLS, . . .
Schemes – AES-GCM, . . .
Primitives – AES, . . .

Cryptographic Primitives

- Symmetric-Key Primitives
  - Block ciphers (AES [DR99], CLEFIA [Shi+07])
  - Tweakable block ciphers (SKINNY [Bei+16], QARMAv2 [Ava+23])
  - Unkeyed primitives (Ascon [Dob+21b; Dob+21a], Keccak [Ber+13])
  - Stream ciphers (Trivium [CP08], ZUC [ETS11], Enocoro-128v2 [WOK10])
- Public-Key Primitives
  - Public-key encryption algorithms (RSA, ECC)
  - Digital signature algorithms (RSA, ECC)
  - Key-exchange algorithms (DH)
- **Hybrid Approach**

Overview of Block Cipher Designs



Cryptanalysis

- **Information-Theoretic Approach**
  - Provides generic bounds for security against unconditional adversaries
- **Complexity-Theoretic Approach**
  - Based on reduction techniques
  - More common in public-key cryptography
- **Cryptanalytic Approach**
  - Evaluates security against concrete known attacks
  - More common in symmetric-key cryptography

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)
- Integral attack [Lai94a; DKR97] (Full-round MISTY1 [Tod15])
- Cube attack [DS09] (Best attack type on SHA-3 [Hua+17])
- And some others, e.g., zero-correlation (ZC), meet-in-the-middle attacks.

Automated Methods in Cryptanalysis

Mounting cryptanalytic attacks against symmetric-key primitives:

- Requires tracing the propagation of a certain property at the bit-level
- Implies solving a hard combinatorial optimization problem
- Is very time-consuming
- Is potentially an error-prone process

Automated Methods in Cryptanalysis

Leveraging machines to **find**, **build**, or **optimize** cryptanalytic attacks.

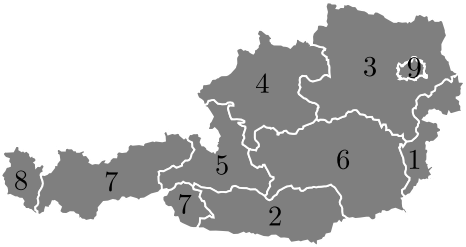
Different Approaches for Automatic Cryptanalysis

- Dedicated algorithms
- **Constraint Programming (CP)**
  - Satistifiability (SAT)
  - Satistifiability Modulo Theories (SMT)
  - Mixed Integer Linear Programming (MILP)
  - Constraint Satisfaction/Optimization Problem (CSP/COP)
- Artificial Intelligence (AI)

Constraint Programming (CP)

- Constraint Satisfaction/Optimization Problem (CSP/COP):
  - We define a set of variables: $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$
  - We specify the domain of each variable: $\mathbb{F}_2, \mathbb{Z}, \mathbb{R}, \ldots$
  - We define a set of constraints: $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_2\}$
  - We define an objective function (if it is required)
- Constraint Programming (CP): Searching for a solution for a CSP/COP
- MILP and SMT/SAT are special cases of CP

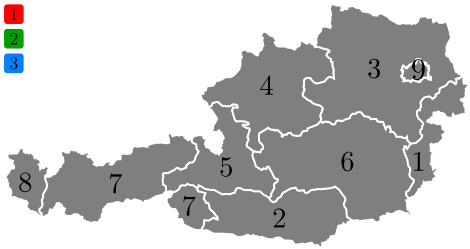
Constraint Programming – Example I




```
int: NC = 3;
array[1..9] of var 1..NC: R;
constraint R[1] != R[3]; constraint R[1] != R[6];
constraint R[2] != R[5]; constraint R[2] != R[6];
constraint R[2] != R[7]; constraint R[3] != R[9];
constraint R[3] != R[6]; constraint R[3] != R[4];
constraint R[4] != R[6]; constraint R[4] != R[5];
constraint R[5] != R[6]; constraint R[5] != R[7];
constraint R[7] != R[8];
solve satisfy;


R = [3, 3, 2, 3, 2, 1, 1, 2, 1];
```

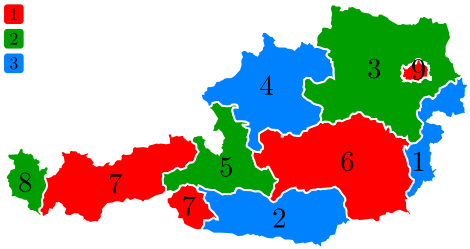
Constraint Programming – Example I




```
int: NC = 3;
array[1..9] of var 1..NC: R;
constraint R[1] != R[3]; constraint R[1] != R[6];
constraint R[2] != R[5]; constraint R[2] != R[6];
constraint R[2] != R[7]; constraint R[3] != R[9];
constraint R[3] != R[6]; constraint R[3] != R[4];
constraint R[4] != R[6]; constraint R[4] != R[5];
constraint R[5] != R[6]; constraint R[5] != R[7];
constraint R[7] != R[8];
solve satisfy;
```

```
R = [3, 3, 2, 3, 2, 1, 1, 2, 1];
```

Constraint Programming – Example I



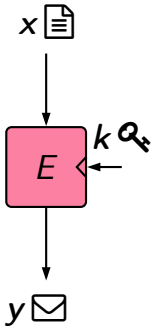

```
int: NC = 3;
array[1..9] of var 1..NC: R;
constraint R[1] != R[3]; constraint R[1] != R[6];
constraint R[2] != R[5]; constraint R[2] != R[6];
constraint R[2] != R[7]; constraint R[3] != R[9];
constraint R[3] != R[6]; constraint R[3] != R[4];
constraint R[4] != R[6]; constraint R[4] != R[5];
constraint R[5] != R[6]; constraint R[5] != R[7];
constraint R[7] != R[8];
solve satisfy;
```

```
R = [3, 3, 2, 3, 2, 1, 1, 2, 1];
```

Automated Tools for Impossible-Differential, Zero-Correlation and Integral Attacks

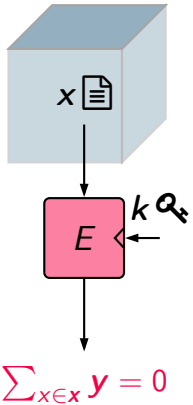
Integral, ID, and ZC Distinguishers

- Integral attack [Lai94b; DKR97]
- Impossible-differential attack [BBS99; Knu98]
- Zero-correlation attack [BR14]



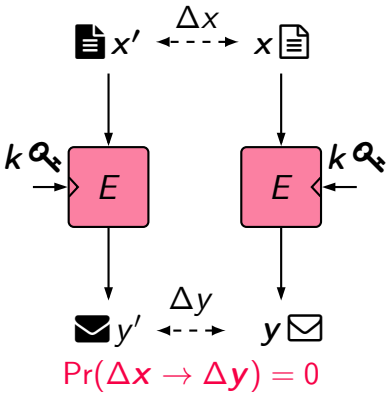
Integral, ID, and ZC Distinguishers

- Integral attack [Lai94b; DKR97]
- Impossible-differential attack [BBS99; Knu98]
- Zero-correlation attack [BR14]



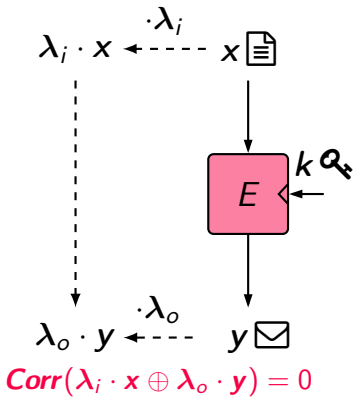
Integral, ID, and ZC Distinguishers

- Integral attack [Lai94b; DKR97]
- Impossible-differential attack [BBS99; Knu98]
- Zero-correlation attack [BR14]



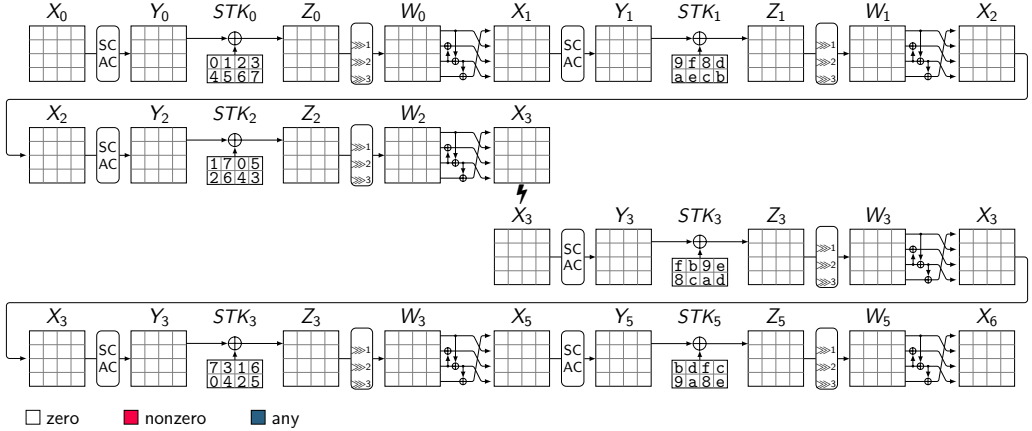
Integral, ID, and ZC Distinguishers

- Integral attack [Lai94b; DKR97]
- Impossible-differential attack [BBS99; Knu98]
- Zero-correlation attack [BR14]



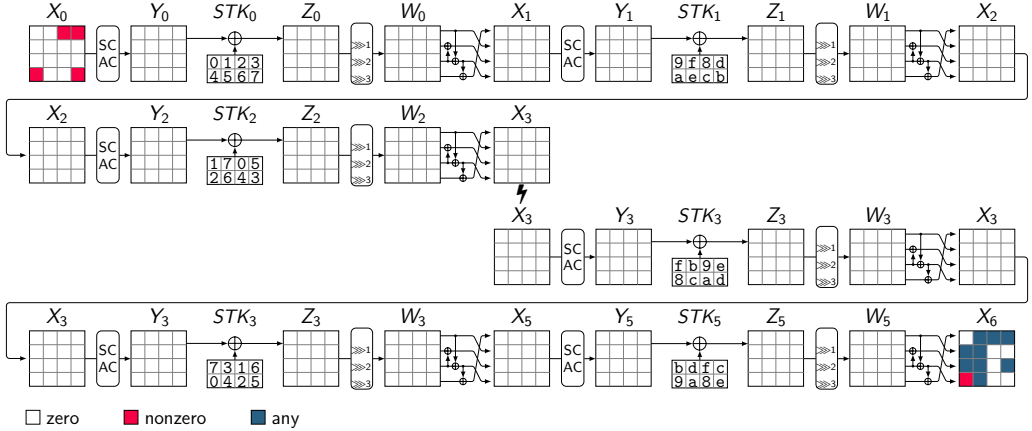
Miss-in-the-Middle Technique [BBS99]

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



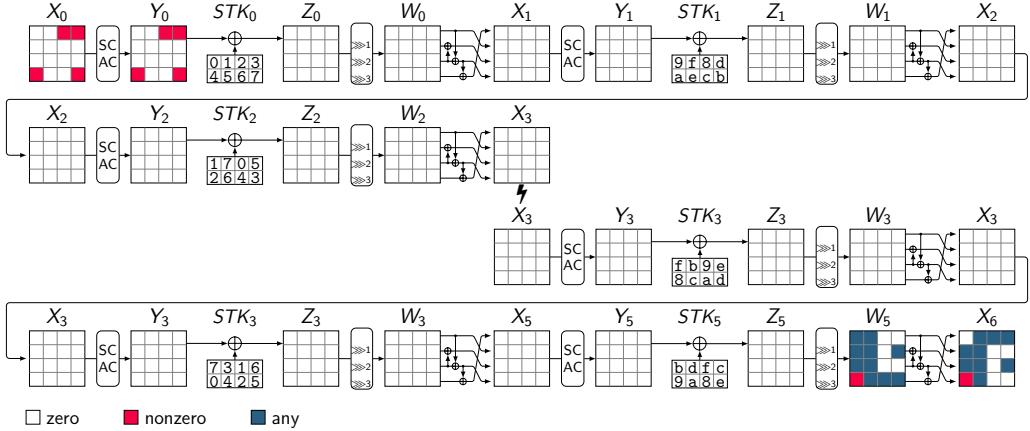
Miss-in-the-Middle Technique [BBS99]

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



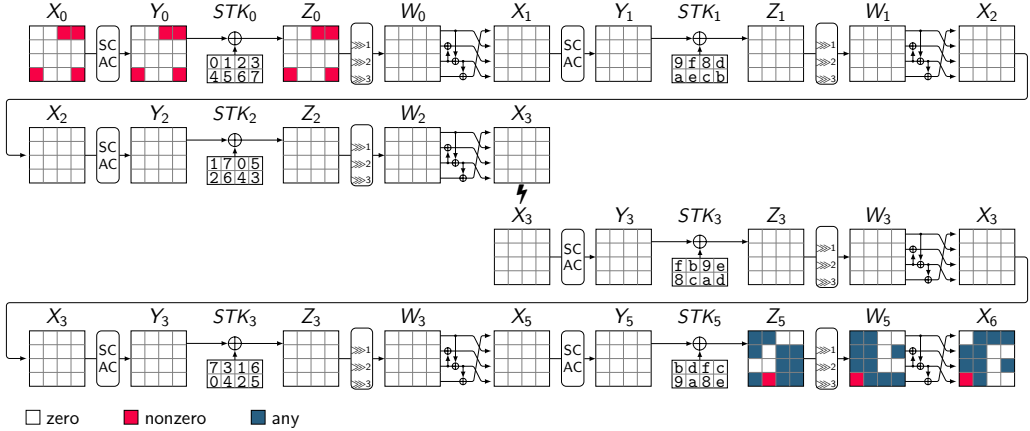
Miss-in-the-Middle Technique [BBS99]

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



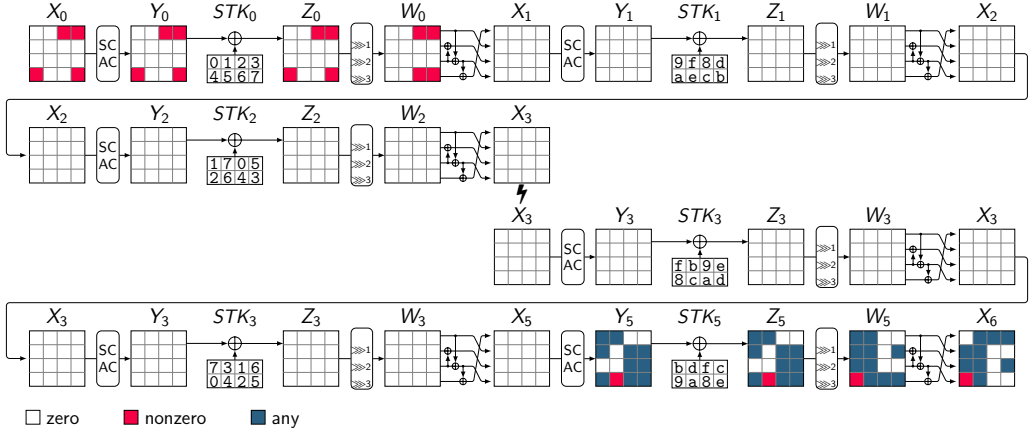
Miss-in-the-Middle Technique [BBS99]

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



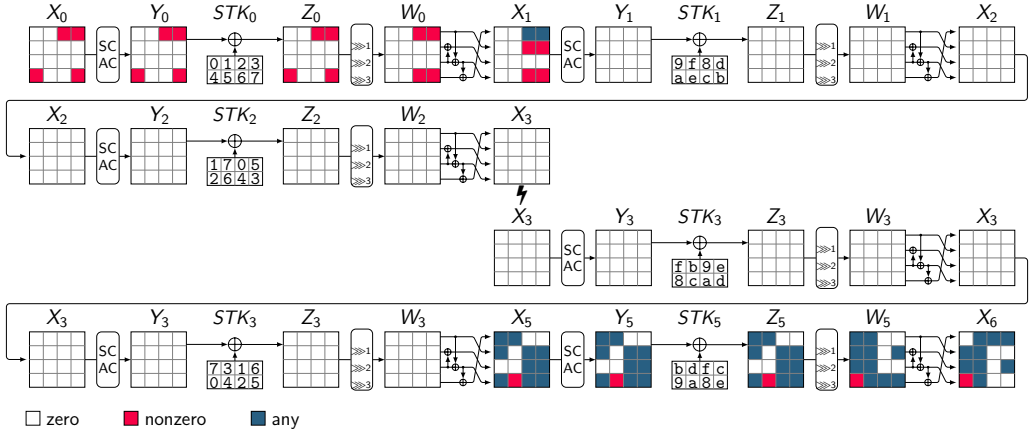
Miss-in-the-Middle Technique [BBS99]

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



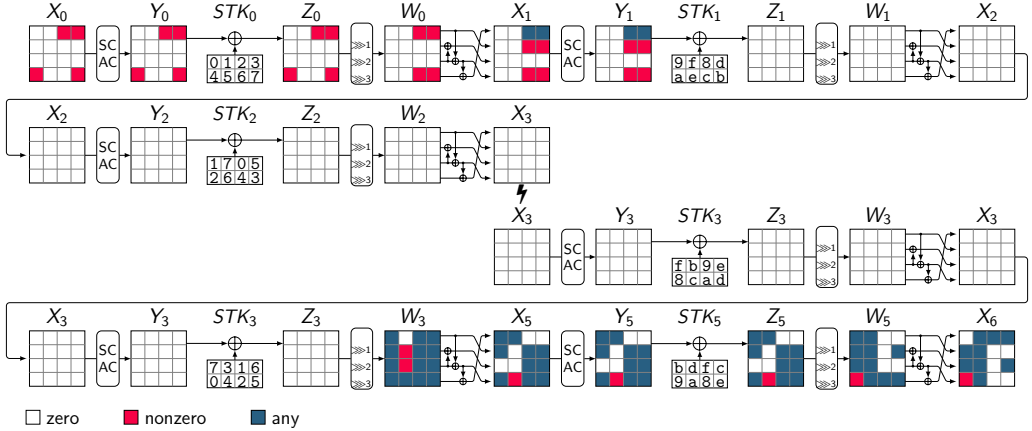
Miss-in-the-Middle Technique [BBS99]

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



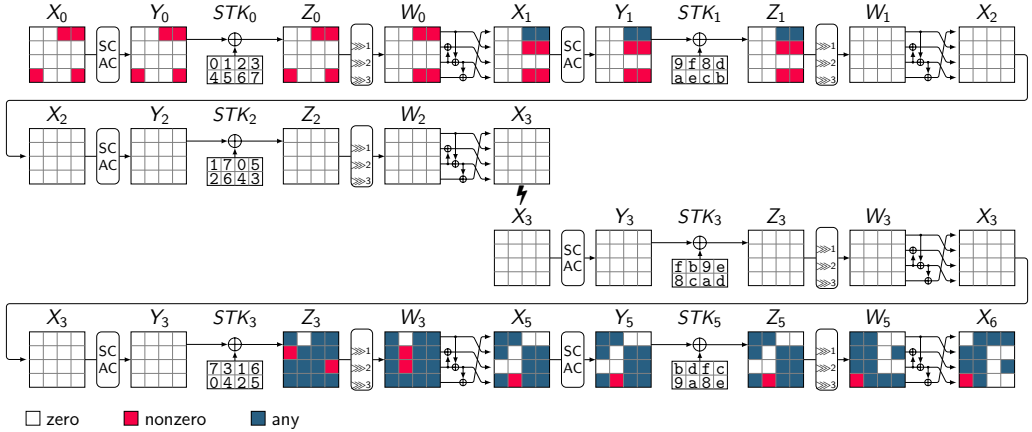
Miss-in-the-Middle Technique [BBS99]

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



Miss-in-the-Middle Technique [BBS99]

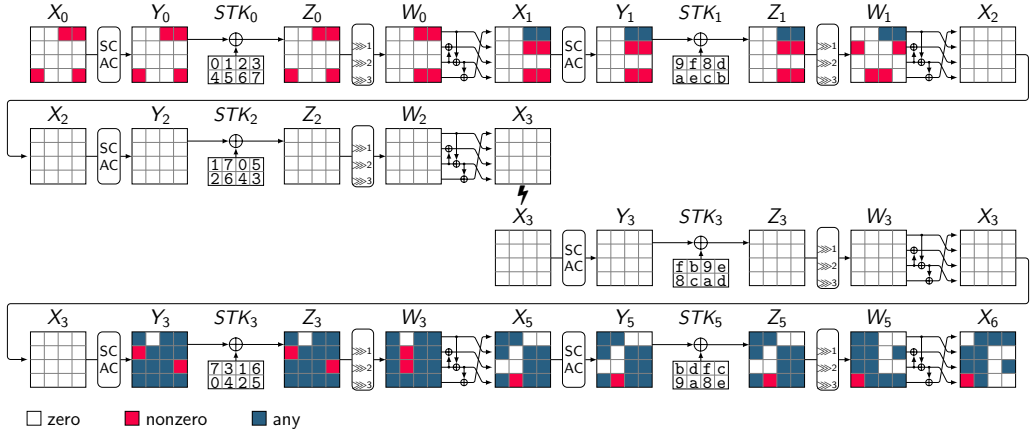
- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle



Hosein Hadipour
Ph.D. Defense, Graz University of Technology

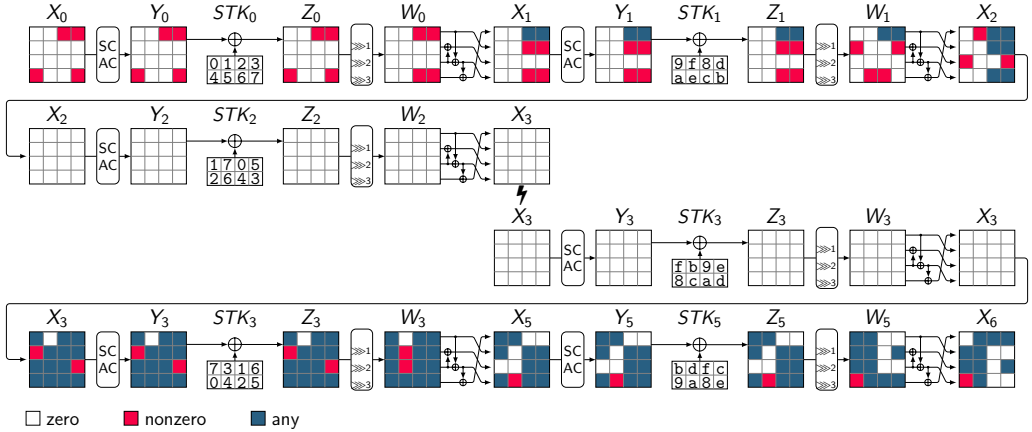
Miss-in-the-Middle Technique [BBS99]

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



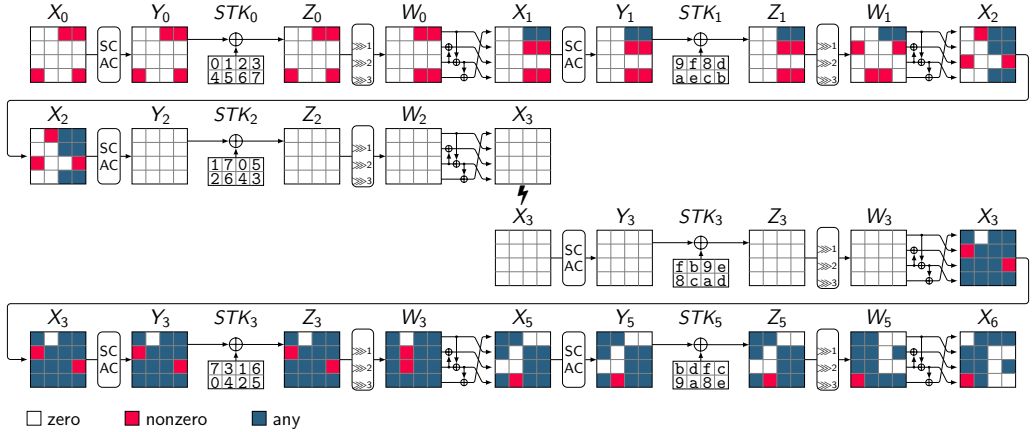
Miss-in-the-Middle Technique [BBS99]

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



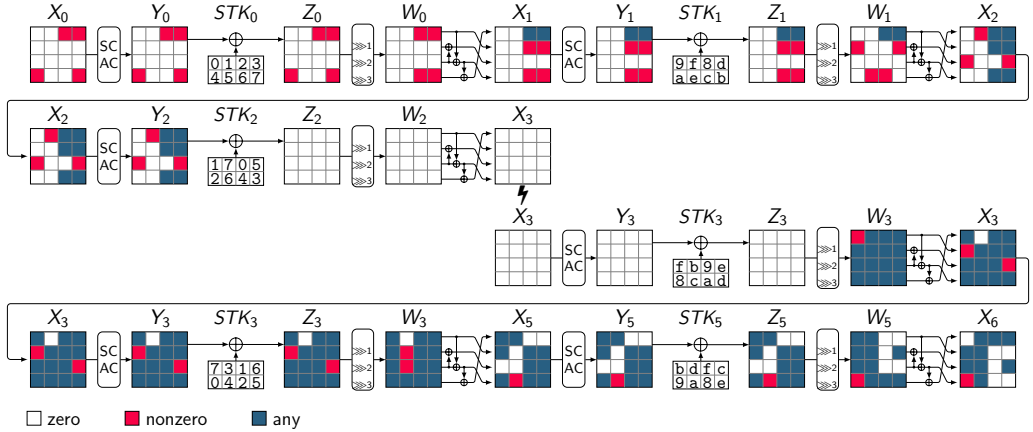
Miss-in-the-Middle Technique [BBS99]

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



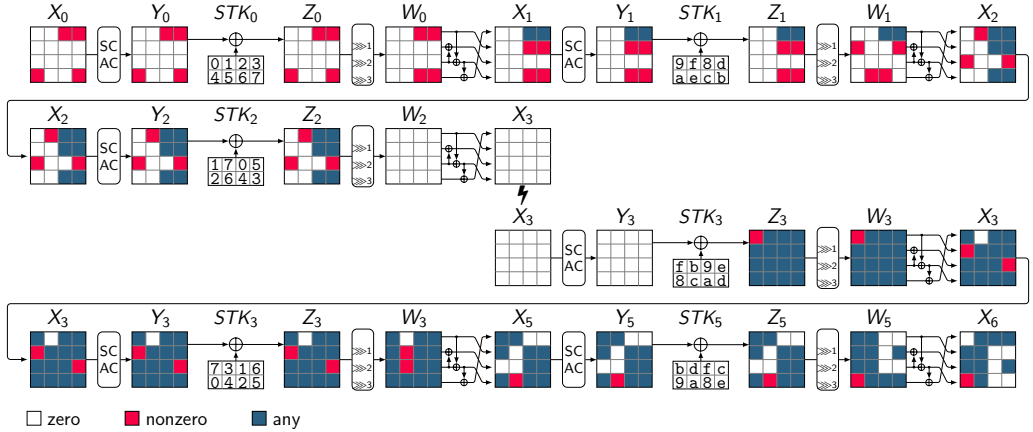
Miss-in-the-Middle Technique [BBS99]

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



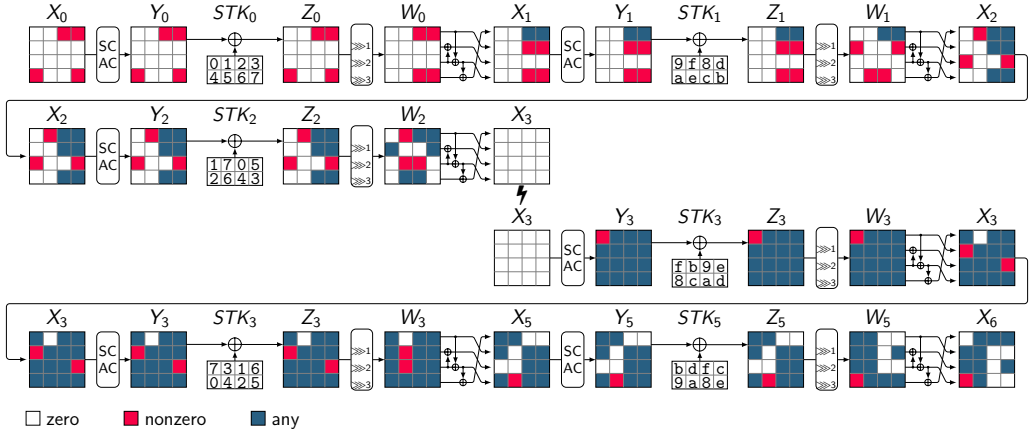
Miss-in-the-Middle Technique [BBS99]

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



Miss-in-the-Middle Technique [BBS99]

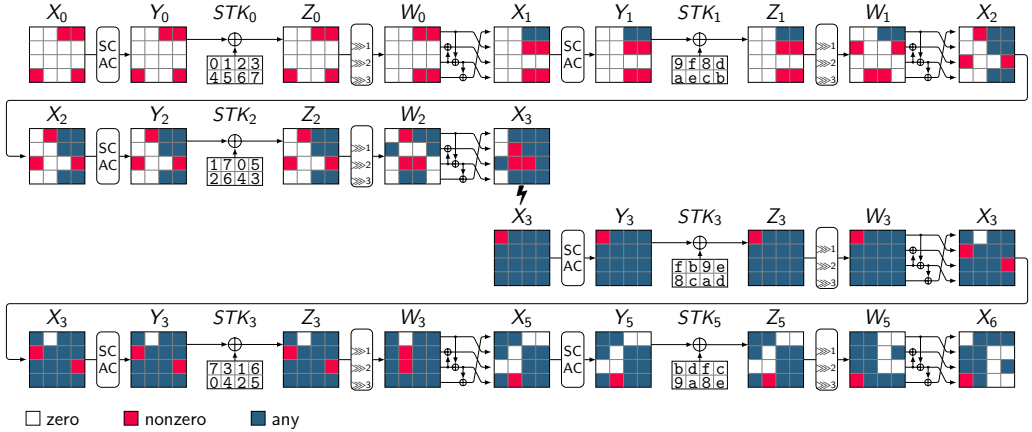
- Find two differences (linear masks) that propagate forward and backward with probability one and contradict each other in the middle



Hosein Hadipour
Ph.D. Defense, Graz University of Technology

Miss-in-the-Middle Technique [BBS99]

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



Relation Between ZC and Integral Distinguishers

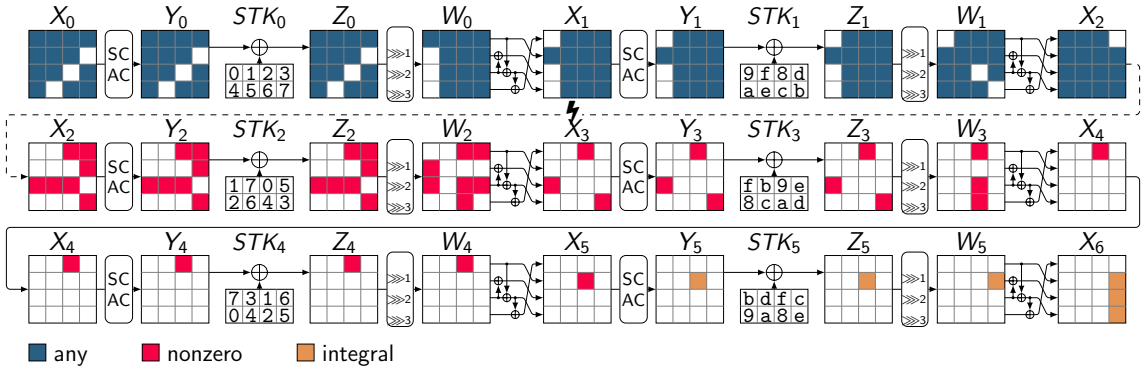
- Any ZC distinguisher can be converted to an integral distinguisher [Sun+15].

Link Between ZC and Integral Distinguishers [Sun+15]

Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a vectorial Boolean function. Assume A is a subspace of $\mathbb{F}_2^n$ and $\beta \in \mathbb{F}_2^n \setminus \{0\}$ such that (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

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

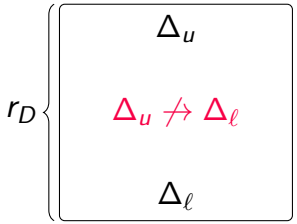
Example: Conversion of ZC Distinguisher to Integral Distinguisher



- $X_0[7, 10, 13]$ 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

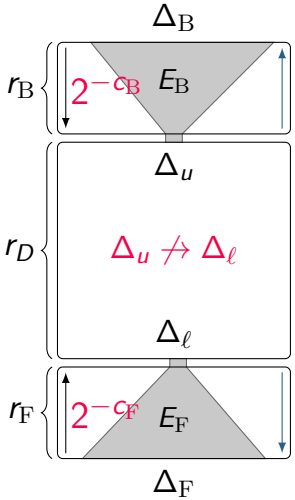
ID, ZC, and Integral Key Recovery

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



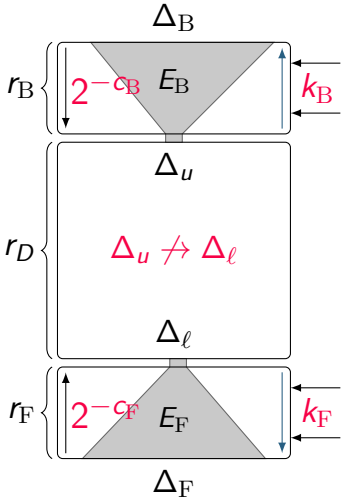
ID, ZC, and Integral Key Recovery

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



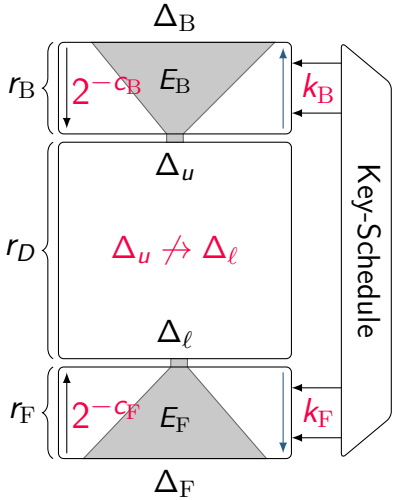
ID, ZC, and Integral Key Recovery

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



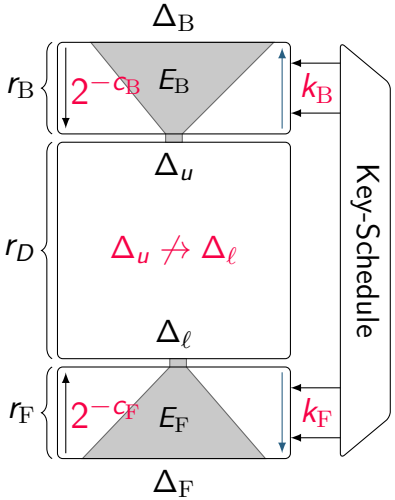
ID, ZC, and Integral Key Recovery

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



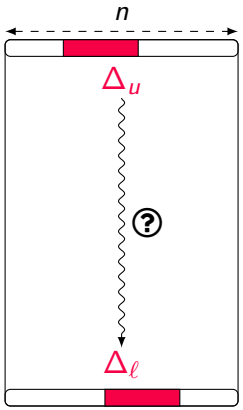
ID, ZC, and Integral Key Recovery

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

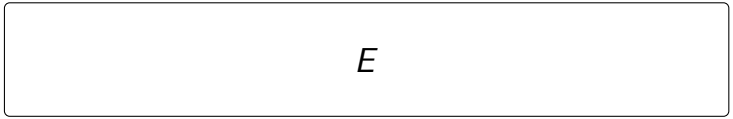


Previous Tools for ID/ZC, and Integral Attacks

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

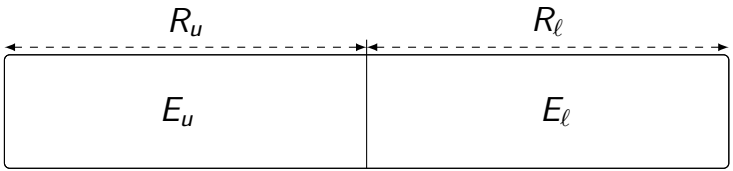


Our First Method to Search Distinguishers [HSE23]



- $CSP_u(\Delta_u, \Delta'_u)$
- $CSP_\ell(\Delta_\ell, \Delta'_\ell)$
- $CSP_M(\Delta'_u, \Delta'_\ell)$

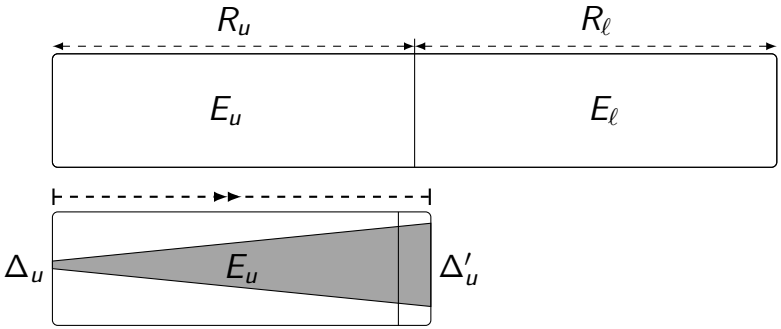
Our First Method to Search Distinguishers [HSE23]



- $CSP_u(\Delta_u, \Delta'_u)$
- $CSP_\ell(\Delta_\ell, \Delta'_\ell)$
- $CSP_M(\Delta'_u, \Delta'_\ell)$

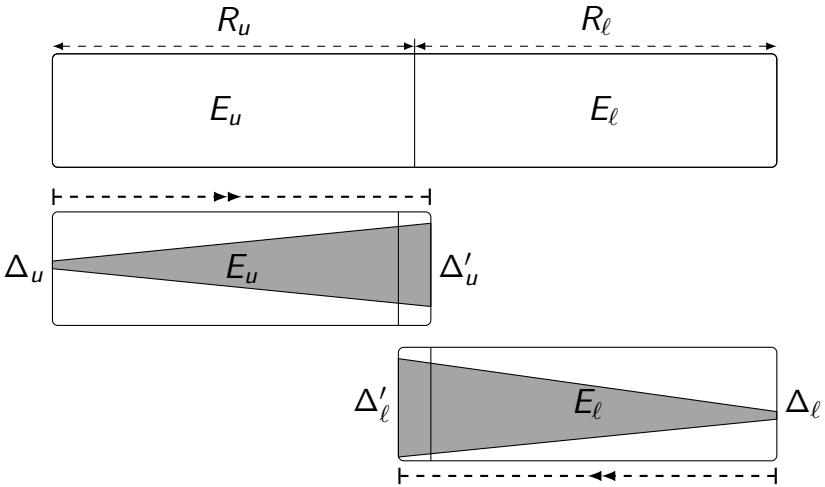
Our First Method to Search Distinguishers [HSE23]

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



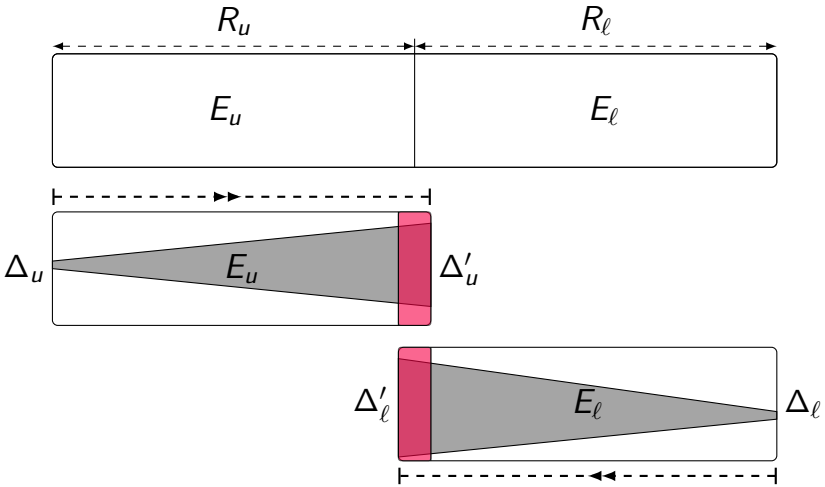
Our First Method to Search Distinguishers [HSE23]

- $CSP_u(\Delta_u, \Delta'_u)$
- $CSP_\ell(\Delta_\ell, \Delta'_\ell)$
- $CSP_M(\Delta'_u, \Delta'_\ell)$



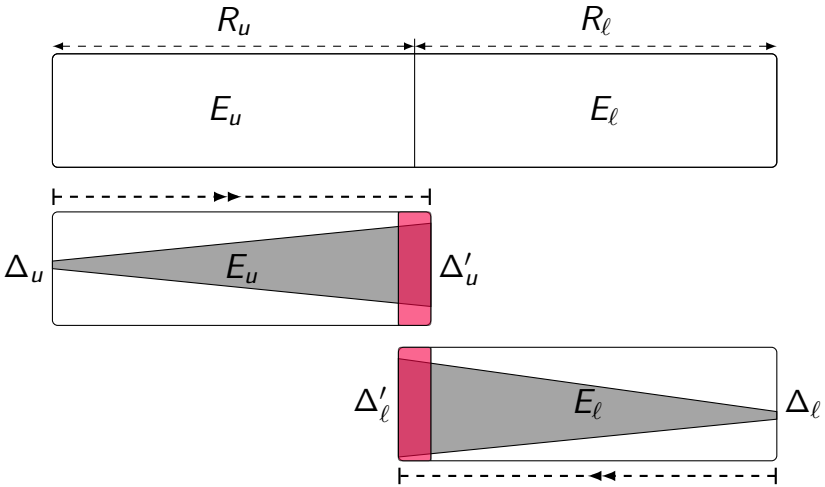
Our First Method to Search Distinguishers [HSE23]

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



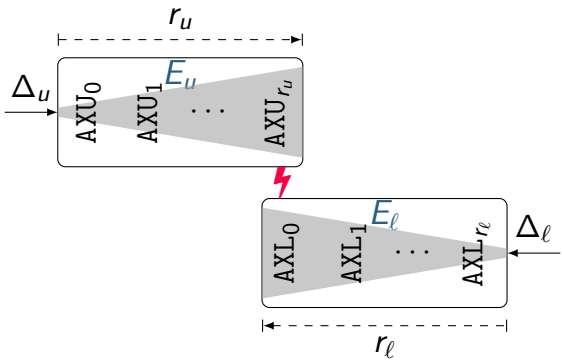
Our First Method to Search Distinguishers [HSE23]

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

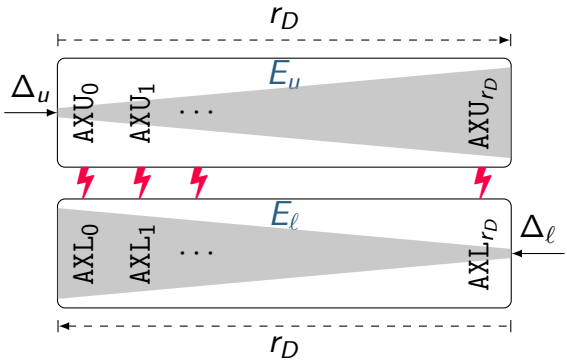


Relax the Limit of Fixing the Contradiction's Location [Hos+24]

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



Our first model in [HSE23].



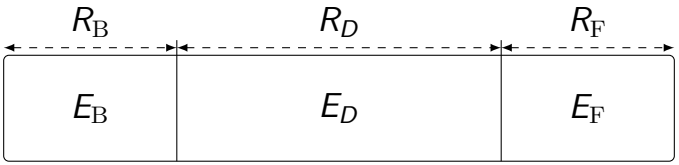
Our second model in [Hos+24]

The Advantages of Our Method to Search for Distinguishers

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

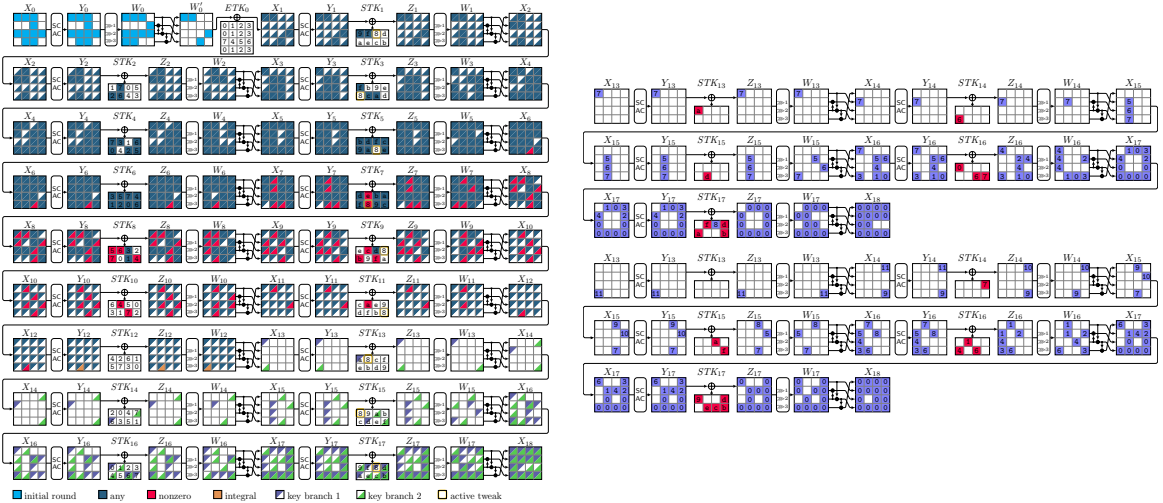
Usage of Our Tool

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



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

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



Hosein Hadipour
Ph.D. Defense, Graz University of Technology

Part of Our Results Regarding Distinguishing Attacks

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

Part of Our Result Regarding Key-Recovery Attacks

Cipher	#R	Time	Data	Mem.	Attack	Setting / Model	Ref.
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	This work
SKINNY-64-128	18	2^{126}	$2^{62.68}$	2^{64}	ZC	STK / KP	[SMB18]
	19	$2^{119.12}$	$2^{62.89}$	2^{49}	ZC	STK / KP	This work
	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	This work
SKINNY-64-64	14	2^{62}	$2^{62.58}$	2^{64}	ZC	STK / KP	[SMB18]
	16	$2^{62.71}$	$2^{61.35}$	$2^{37.80}$	ZC	STK / KP	This work
CRAFT	**20**	$2^{120.43}$	$2^{62.89}$	2^{49}	ZC	STK / KP	This work
	21	$2^{106.53}$	$2^{60.99}$	2^{100}	ID	STK / CP	This work

Autoguess: Automated Guess-and-Determine Attacks

Guess-and-Determine (GD)

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 (GD)

Guess-and-Determine

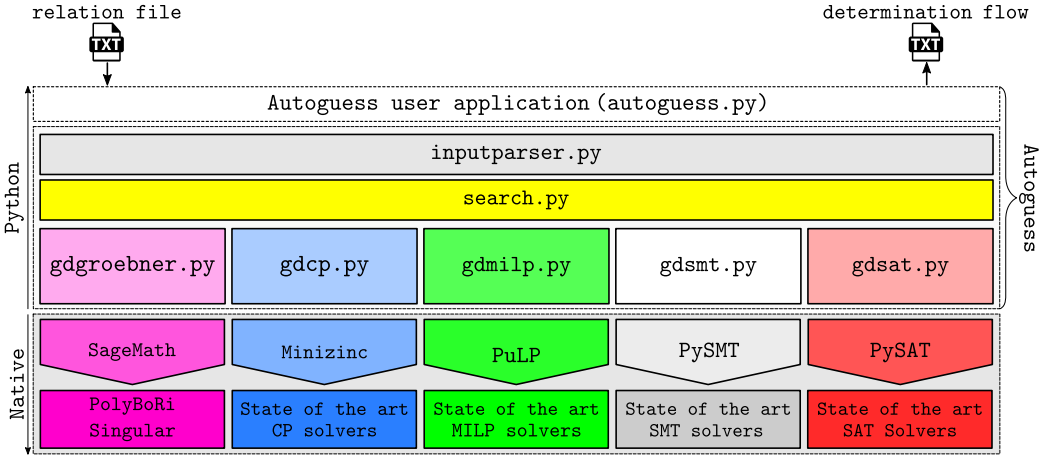
Given a set of variables and a set of relations between them, find the smallest subset of variables guessing the value of which uniquely determines the value of the remaining variables.

Example

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

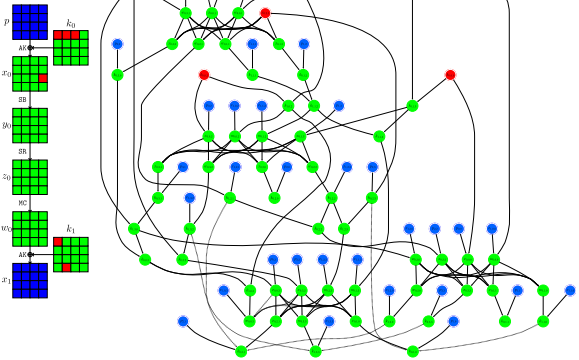
$$\begin{cases} F(u+v) \oplus G(x) \oplus y \oplus (z \lll 7) & = c_1 \\ G(u \oplus w) + (y \lll 3) + z & = c_2 \\ F(w \oplus x) + y \oplus z & = c_3 \\ F(u) \oplus G(w+z) & = c_4 \\ (F(u) \times G(w \lll 7)) + H(z \oplus v) & = c_5 \end{cases}$$

Autoguess

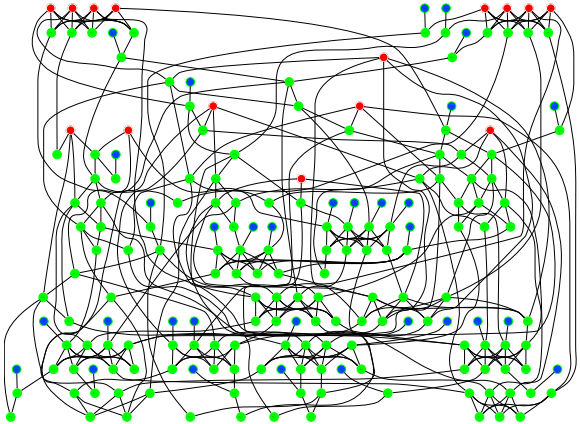


: https://github.com/hadipourh/autoguess

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

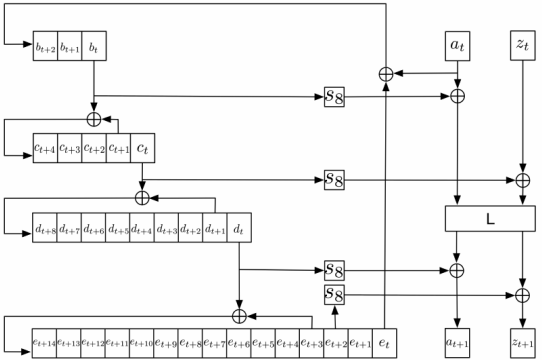


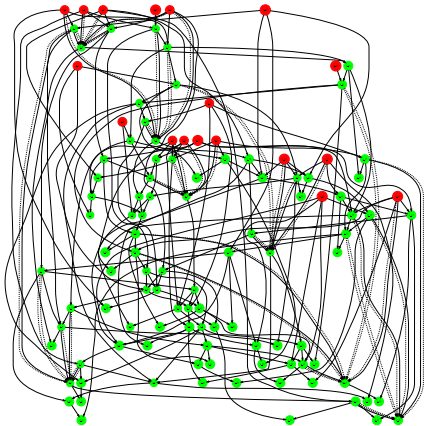
Found in 0.02 seconds on a standard laptop



Found in 34.51 seconds on a standard laptop

GD (State Recovery) Attack on Enocoro128-v2 (16 clock cycles)





Found in less than a second running on a single core of a regular laptop (`i7-1165G7 @ 2.80GHz`)

Conclusion and Acknowledgments

Conclusion – I

- Developed a generic, automatic tool for guess-and-determine attacks and the key-bridging technique [HE22a].
- Proposed an efficient method to identify effective boomerang/rectangle distinguishers [HNE22].
- Designed a SAT-based method to search for integral distinguishers based on monomial prediction for block ciphers [HE22b].
- Created a graph-based automatic tool for constructing key recovery of integral attacks, leveraging the FFT technique [HE22b].
- Presented the first CP/MILP-based tool for finding complete impossible-differential, zero-correlation, and integral attacks [HSE23; Hos+24].

Conclusion – II

- Proposed the first CP/MILP model for the partial-sum technique in key recovery for integral attacks [Hos+24].
- Designed an automatic tool to exploit the Meet-in-the-Middle (MiTM) technique in key recovery for integral attacks, achieving the best-known attacks on QARMAv2 to date [HT24].
- Solved an open problem from [Bar+19] and extended the `DLCT` framework for differential-linear (DL) attacks to multiple rounds [HDE24].
- Linked boomerang and differential-linear attacks, and developed a CP/MILP-based tool for identifying DL distinguishers for nearly any design paradigm [HDE24].

Source Code of Our Tools

- **Guess-and-Determine Attacks**: https://github.com/hadipourh/autoguess
- **Boomerang Attacks**: https://github.com/hadipourh/comeback
- **Inegral Attacks**:
  - **Based on Monomial Prediction**: https://github.com/hadipourh/mpt
  - **Based on ZC Distinguishers**: https://github.com/hadipourh/QARMAnalysis
- **Impossible Differential, Zero-Correlation and Integral Attacks:**
  - **Zero**: https://github.com/hadipourh/zero
  - **Zeroplus**: https://github.com/hadipourh/zeroplus
- **Differential-Linear Attacks**: https://github.com/hadipourh/DL
- **S-box Analyzer**: https://github.com/hadipourh/sboxanalyzer

- Special thanks to my advisor, Maria Eichlseder.
- I am grateful to my examiners, Gregor Leander and María Naya-Plasencia.
- Thanks to my co-authors for their collaboration.
- Heartfelt thanks to my family and friends.

Maria Eichlseder

María Naya-Plasencia

Gregor Leander

Takanori Isobe

Patrick Derbez

Virginie Lallemand

Yosuke Todo

Sadegh Sadeghi

Nasour Bagheri

Bibliography I

[Ank+19] Ralph Ankele et al. **Zero-Correlation Attacks on Tweakable Block Ciphers with Linear Tweakey Expansion**. *IACR Transactions on Symmetric Cryptology* 2019.1 (Mar. 2019), pp. 192–235. DOI: 10.13154/tosc.v2019.i1.192-235.

[Ava+23] Roberto Avanzi et al. **The QARMAv2 Family of Tweakable Block Ciphers**. *IACR Transactions on Symmetric Cryptology* 2023.3 (Sept. 2023), pp. 25–73. DOI: 10.46586/tosc.v2023.i3.25-73.

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

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

Bibliography II

[BDK02] Eli Biham, Orr Dunkelman, and Nathan Keller. **Enhancing Differential-Linear Cryptanalysis**. ASIACRYPT 2002. Vol. 2501. LNCS. Springer, 2002, pp. 254–266. DOI: 10.1007/3-540-36178-2_16.

[Bei+16] Christof Beierle et al. **The SKINNY family of block ciphers and its low-latency variant MANTIS**. CRYPTO 2016. Springer. 2016, pp. 123–153. DOI: 10.1007/978-3-662-53008-5_5.

[Ber+13] Guido Bertoni et al. **Keccak**. EUROCRYPT. Vol. 7881. Lecture Notes in Computer Science. Springer, 2013, pp. 313–314. DOI: 10.1007/978-3-642-38348-9_19.

Bibliography III

[BKN09] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. **Distinguisher and Related-Key Attack on the Full AES-256**. CRYPTO 2009. Ed. by Shai Halevi. Vol. 5677. LNCS. Springer, 2009, pp. 231–249. DOI: 10.1007/978-3-642-03356-8_14.

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

[Bou+20] Hamid Boukerrou et al. **On the Feistel Counterpart of the Boomerang Connectivity Table Introduction and Analysis of the FBCT**. *IACR Trans. Symmetric Cryptol.* 2020.1 (2020), pp. 331–362. DOI: 10.13154/TOSC.V2020.I1.331-362.

Bibliography IV

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

[BS90] Eli Biham and Adi Shamir. **Differential Cryptanalysis of DES-like Cryptosystems**. CRYPTO '90. Ed. by Alfred Menezes and Scott A. Vanstone. Vol. 537. LNCS. Springer, 1990, pp. 2–21. DOI: 10.1007/3-540-38424-3_1.

[BS92] Eli Biham and Adi Shamir. **Differential Cryptanalysis of the Full 16-Round DES**. CRYPTO '92. Ed. by Ernest F. Brickell. Vol. 740. LNCS. Springer, 1992, pp. 487–496. DOI: 10.1007/3-540-48071-4_34.

Bibliography V

[Cha+24] Debasmita Chakraborty et al. **Finding Complete Impossible Differential Attacks on AndRX Ciphers and Efficient Distinguishers for ARX Designs**. *IACR Trans. Symmetric Cryptol.* 2024.3 (2024), pp. 84–176. DOI: 10.46586/tosc.v2024.i3.84-176.

[Cid+18] Carlos Cid et al. **Boomerang Connectivity Table: A New Cryptanalysis Tool**. EUROCRYPT 2018. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10821. LNCS. Springer, 2018, pp. 683–714. DOI: 10.1007/978-3-319-78375-8_22.

[CP08] Christophe De Cannière and Bart Preneel. **Trivium**. *New stream cipher designs*. Springer, 2008, pp. 244–266.

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

Bibliography VI

[DDV20] Stéphanie Delaune, Patrick Derbez, and Mathieu Vavrille. **Catching the Fastest Boomerangs Application to SKINNY**. *IACR Trans. Symmetric Cryptol.* 2020.4 (2020), pp. 104–129. DOI: 10.46586/TOSC.V2020.I4.104-129.

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

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

Bibliography VII

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

[DKS10] Orr Dunkelman, Nathan Keller, and Adi Shamir. **A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony**. CRYPTO. Vol. 6223. LNCS. Springer, 2010, pp. 393–410. DOI: 10.1007/978-3-642-14623-7_21.

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

Bibliography VIII

[Dob+21a] Christoph Dobraunig et al. **Ascon v1.2 (Submission to NIST)**. Finalist submission to the NIST lightweight cryptography standardization process. 2021. URL: https://csrc.nist.gov/Projects/Lightweight-Cryptography.

[Dob+21b] Christoph Dobraunig et al. **Ascon v1.2: Lightweight Authenticated Encryption and Hashing**. *Journal of Cryptology* 34.3 (2021), p. 33. DOI: 10.1007/s00145-021-09398-9.

[DR99] Joan Daemen and Vincent Rijmen. **AES proposal: Rijndael**. (1999).

[DS09] Itai Dinur and Adi Shamir. **Cube Attacks on Tweakable Black Box Polynomials**. EUROCRYPT 2009. Ed. by Antoine Joux. Vol. 5479. LNCS. Springer, 2009, pp. 278–299. DOI: 10.1007/978-3-642-01001-9_16.

Bibliography IX

[ETS11] ETSI/SAGE. **Specification of the 3GPP confidentiality and integrity algorithms 128-EEA3 and 128-EIA3: ZUC specification**. *ETSI/SAGE, Document 2, Version 1.6* (2011).

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

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

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

Bibliography X

[HDE24] **Hosein Hadipour**, Patrick Derbez, and Maria Eichlseder. **Revisiting Differential-Linear Attacks via a Boomerang Perspective With Application to AES, Ascon, CLEFIA, SKINNY, PRESENT, KNOT, TWINE, WARP, LBlock, Simeck, and SERPENT**. **CRYPTO** 2024. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14922. LNCS. Springer, 2024, pp. 290–305. DOI: 10.1007/978-3-031-68385-5_2.

[HE22a] **Hosein Hadipour** and Maria Eichlseder. **Autoguess: A Tool for Finding Guess-and-Determine Attacks and Key Bridges**. **ACNS** 2022. Ed. by Giuseppe Ateniese and Daniele Venturi. Vol. 13269. LNCS. Springer, 2022, pp. 230–250. DOI: 10.1007/978-3-031-09234-3_12.

Bibliography XI

[HE22b] **Hosein Hadipour** and Maria Eichlseder. **Integral Cryptanalysis of WARP based on Monomial Prediction**. *IACR Trans. Symmetric Cryptol.* 2022.2 (2022), pp. 92–112. DOI: 10.46586/tosc.v2022.i2.92-112.

[HNE22] **Hosein Hadipour**, Marcel Nageler, and Maria Eichlseder. **Throwing Boomerangs into Feistel Structures: Application to CLEFIA, WARP, LBlock, LBlock-s and TWINE**. *IACR Trans. Symmetric Cryptol.* 2022.3 (2022), pp. 271–302. DOI: 10.46586/tosc.v2022.i3.271-302.

[Hos+24] **Hosein Hadipour** et al. **Improved Search for Integral, Impossible-Differential and Zero-Correlation Attacks: Application to Ascon, ForkSKINNY, SKINNY, MANTIS, PRESENT and QARMAv2**. *IACR Trans. Symmetric Cryptol.* 2024.1 (2024), pp. 234–325. DOI: 10.46586/tosc.v2024.i1.234-325.

Bibliography XII

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

[HT24] **Hosein Hadipour** and Yosuke Todo. **Cryptanalysis of QARMAv2**. *IACR Trans. Symmetric Cryptol.* 2024.1 (2024), pp. 188–213. DOI: 10.46586/tosc.v2024.i1.188-213.

[Hua+17] Senyang Huang et al. **Conditional Cube Attack on Reduced-Round Keccak Sponge Function**. EUROCRYPT 2017. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10211. LNCS. 2017, pp. 259–288. DOI: 10.1007/978-3-319-56614-6_9.

Bibliography XIII

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

[Lai94a] Xuejia Lai. **Higher Order Derivatives and Differential Cryptanalysis**. (1994), pp. 227–233. DOI: 10.1007/978-1-4615-2694-0_23.

[Lai94b] Xuejia Lai. **Higher order derivatives and differential cryptanalysis**. *Communications and cryptography*. Springer, 1994, pp. 227–233.

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

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

Bibliography XIV

[Mat93] Mitsuru Matsui. **Linear Cryptanalysis Method for DES Cipher**. EUROCRYPT '93. Ed. by Tor Helleseth. Vol. 765. LNCS. Springer, 1993, pp. 386–397. DOI: 10.1007/3-540-48285-7_33.

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

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

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

Bibliography XV

[Shi+07] Taizo Shirai et al. **The 128-Bit Blockcipher CLEFIA (Extended Abstract)**. FSE 2007. Vol. 4593. LNCS. Springer, 2007, pp. 181–195.

[SMB18] Sadegh Sadeghi, Tahereh Mohammadi, and Nasour Bagheri. **Cryptanalysis of Reduced round SKINNY Block Cipher**. *IACR Trans. Symmetric Cryptol.* 2018.3 (2018), pp. 124–162. DOI: 10.13154/tosc.v2018.i3.124-162.

[Sol+22] Hadi Soleimany et al. **Practical Multiple Persistent Faults Analysis**. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2022.1 (2022), pp. 367–390. DOI: 10.46586/TCHES.V2022.I1.367-390.

[SQH19] Ling Song, Xianrui Qin, and Lei Hu. **Boomerang Connectivity Table Revisited. Application to SKINNY and AES**. *IACR Trans. Symmetric Cryptol.* 2019.1 (2019), pp. 118–141. DOI: 10.13154/TOSC.V2019.I1.118-141. URL: https://doi.org/10.13154/tosc.v2019.i1.118-141.

Bibliography XVI

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

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

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

Bibliography XVII

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

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

[Tod15] Yosuke Todo. **Integral Cryptanalysis on Full MISTY1**. CRYPTO 2015. Ed. by Rosario Gennaro and Matthew Robshaw. Vol. 9215. LNCS. Springer, 2015, pp. 413–432. DOI: 10.1007/978-3-662-47989-6_20.

[Wag99] David A. Wagner. **The Boomerang Attack**. FSE. Vol. 1636. LNCS. Springer, 1999, pp. 156–170. DOI: 10.1007/3-540-48519-8_12.

Bibliography XVIII

[WOK10] D Watanabe, K Okamoto, and T Kaneko. **A Hardware-Oriented Light Weight Pseudo-Random Number Generator Enocoro-128v2. The 2010 Symposium on Cryptography and Information Security, SCIS 2010, 3D1-3**. (2010).

[WP19] Haoyang Wang and Thomas Peyrin. **Boomerang Switch in Multiple Rounds. Application to AES Variants and Deoxys**. *IACR Trans. Symmetric Cryptol.* 2019.1 (2019), pp. 142–169. DOI: 10.13154/TOSC.V2019.I1.142-169.

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

Bibliography XIX

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

Universal Bound for Data Complexity

Universal Bound for Data Complexity - I

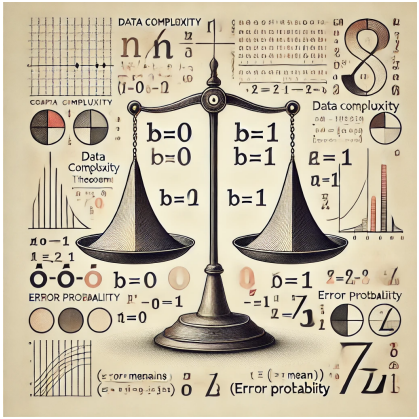
Theorem (Data Complexity)

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.

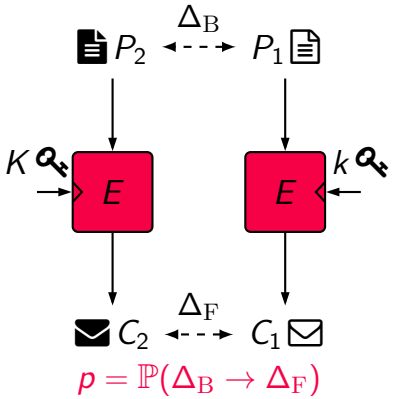
Differential Attack

Differential Attacks [BS90]

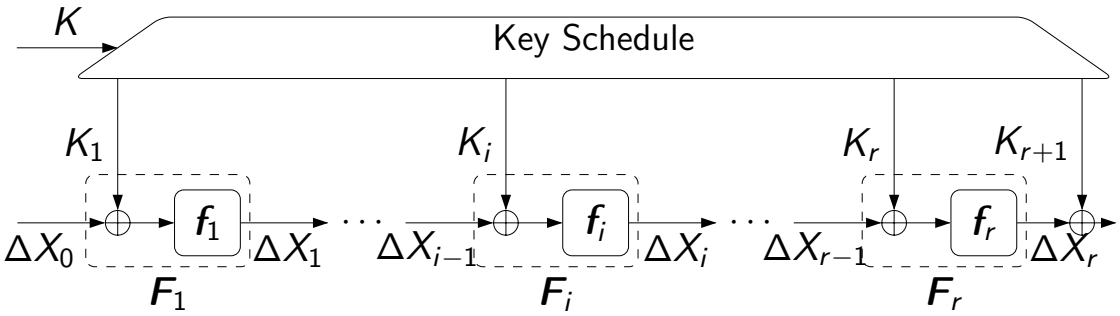
Input: $E_K, (\Delta_B, \Delta_F), N, p = \mathbb{P}(\Delta_B, \Delta_F)$
Output: 0: **real** cipher, 1: **ideal** cipher

Initialize counter T with zero;
for $i = 0, \ldots, N-1$ **do**
  $P_1 \xleftarrow{\$} \mathbb{F}_2^n$;
  $C_1 \leftarrow E_K(P_1)$;
  $P_2 \leftarrow P_1 \oplus \Delta_B$;
  $C_2 \leftarrow E_K(P_2)$;
  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

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



Analytical Estimation of Differential Probability



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

Difference Distribution Table (DDT) – I

We need a tool to handle the nonlinear operations

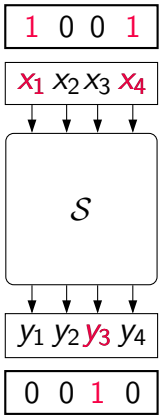
Differential Distribution Table (DDT)

For a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the 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

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

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

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

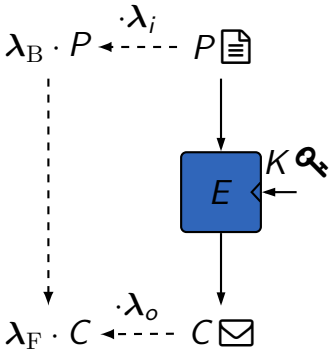
Linear Attack

Linear Attacks [Mat93]

Input: E_K, Given N distinct plaintext-ciphertext pairs (P_i, C_i), $\boldsymbol{c} = \mathbb{C}(\lambda_B, \lambda_F)$
Output: 0: **real** cipher, 1: **ideal** cipher

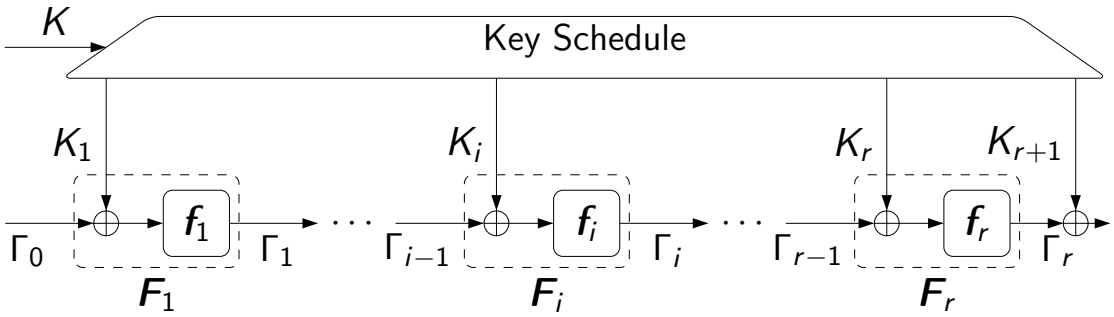
Initialize a counter list $V[z] \leftarrow 0$ for $z \in \{0,1\}$;
for $t = 0, \ldots, N-1$ **do**
  $b_1 \leftarrow \lambda_B \cdot P_t$;
  $b_2 \leftarrow \lambda_F \cdot C_t$;
  $V[b_1 \oplus b_2] \leftarrow V[b_1 \oplus b_2] + 1$;
if $V[0] \sim \mathcal{N}(\mu_0 = N\frac{1+\boldsymbol{c}}{2}, \sigma_0^2 = \frac{N(1-\boldsymbol{c}^2)}{4})$. **then**
  return 0; // real cipher
else
  return 1; // ideal cipher

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



$\boldsymbol{c} = 2 \cdot \mathbb{P}(\lambda_B \cdot P \oplus \lambda_F \cdot C = 0) - 1$

Analytical Estimation of Correlation



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

Linear Approximation Table (LAT) – I

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

Linear Approximation Table (LAT)

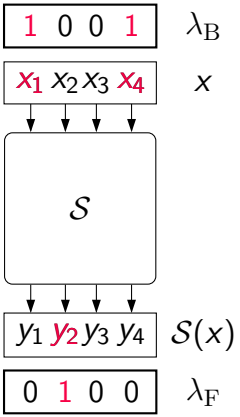
For a vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$, the LAT of S is a $2^n \times 2^m$ table whose rows correspond to the input mask λ_B to S and whose columns correspond to the output mask λ_F of S. The entry at index (λ_B, λ_F) is

$$\mathtt{LAT}(\lambda_B, \lambda_F) = |\mathtt{LAT}_0(\lambda_B, \lambda_F)| - |\mathtt{LAT}_1(\lambda_B, \lambda_F)|,$$

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

$$\mathbb{C}\,(\lambda_B, \lambda_F) = 2^{-n} \cdot \mathtt{LAT}\,(\lambda_B, \lambda_F)$$

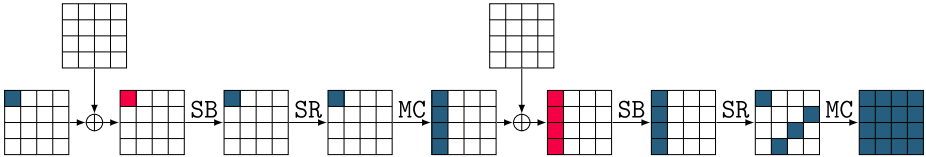
Linear Approximation Table (LAT) – II



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

$\lambda_B \setminus \lambda_F$	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



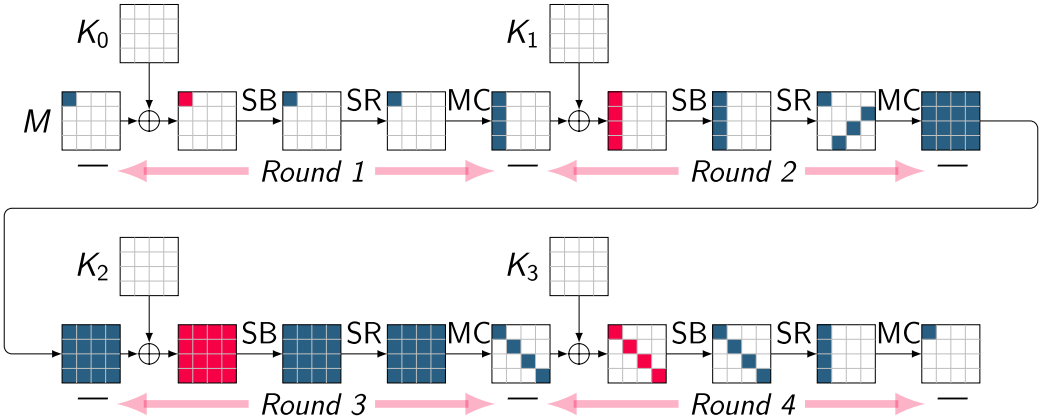
Variables:

- $s_{r,i,j} \in \{0,1\}$ 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



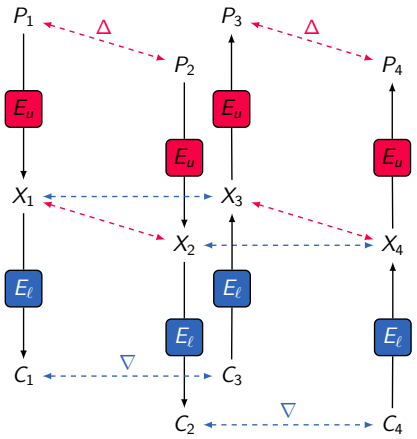
$$\mathbb{P}_{\text{4 rounds}} \leq 2^{-150}$$

Boomerang Attack

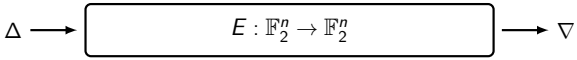
Boomerang Distinguishers [Wag99]

Input: $E_K, (\Delta, \nabla), N, P = \mathbb{P}(P_3 \oplus P_4 = \Delta)$
Output: 0: **real** cipher, 1: **ideal** cipher

1 Initialize counter T with zero;
2 **for** $i = 0, \ldots, N-1$ **do**
3 $\quad P_1 \xleftarrow{\$} \mathbb{F}_2^n$; $P_2 = P_1 \oplus \Delta$;
4 $\quad C_1 \leftarrow E_K(P_1)$, $\quad C_2 \leftarrow E_K(P_2)$;
5 $\quad C_3 \leftarrow C_1 \oplus \nabla$, $\quad C_4 \leftarrow C_2 \oplus \nabla$;
6 $\quad P_3 \leftarrow D_K(C_3)$, $\quad P_4 \leftarrow D_K(C_4)$;
7 $\quad$ **if** $P_3 \oplus P_4 = \Delta$ **then**
8 $\quad\quad T \leftarrow T + 1$;
9 **if** $T \sim \mathcal{N}(\mu = NP, \sigma^2 = NP(1-P))$ **then**
10 $\quad$ **return** 0; `// real cipher`
11 **else**
12 $\quad$ **return** 1; `// ideal cipher`

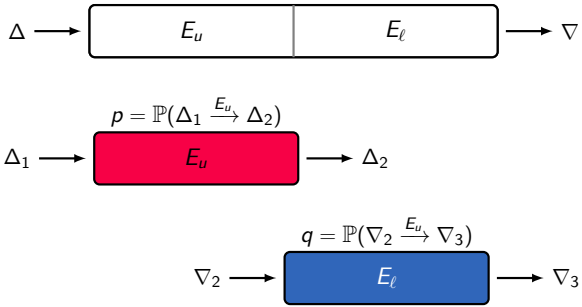


Probability of Boomerang Distinguishers [Wag99]

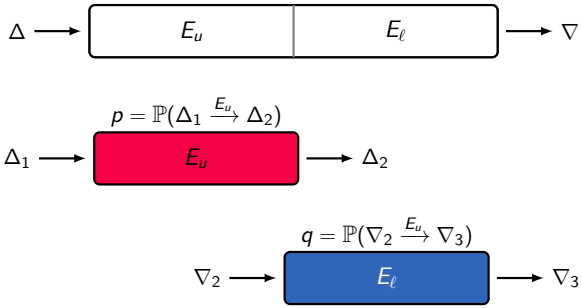


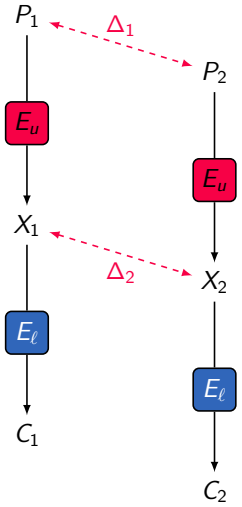
$$0 \lneq \mathbb{P}(\Delta \xrightarrow{E} \nabla) \lll 2^{-n}$$

Probability of Boomerang Distinguishers [Wag99]

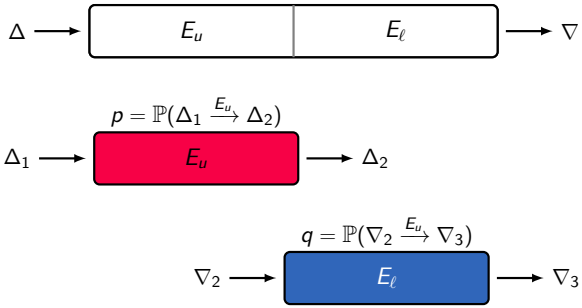


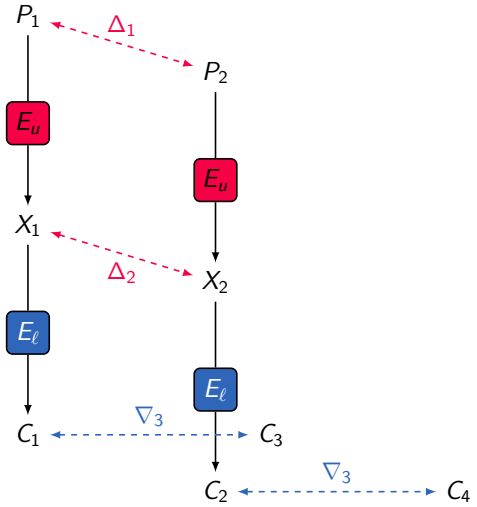
Probability of Boomerang Distinguishers [Wag99]



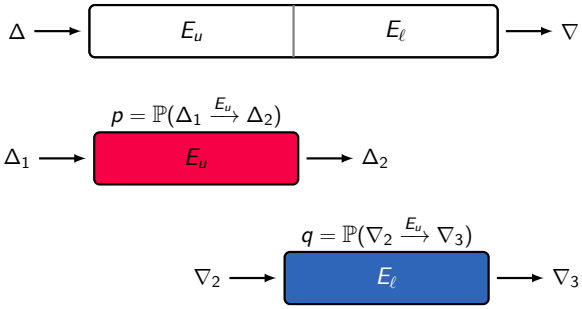


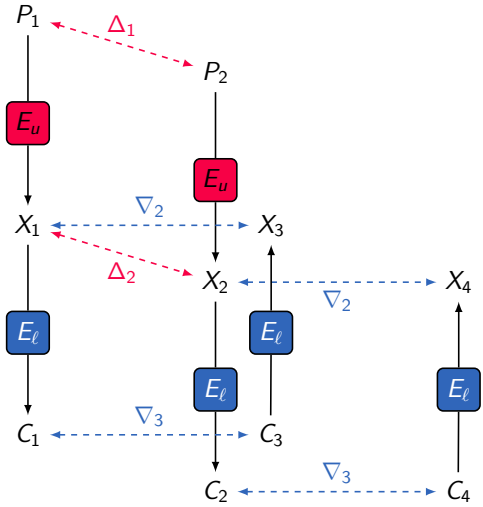
Probability of Boomerang Distinguishers [Wag99]



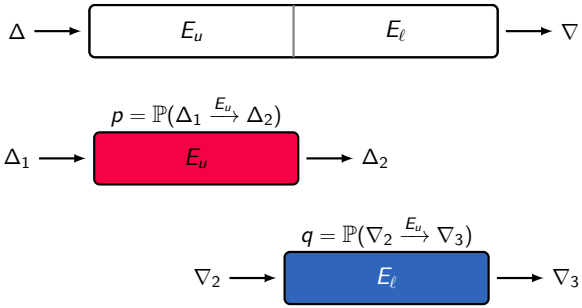


Probability of Boomerang Distinguishers [Wag99]

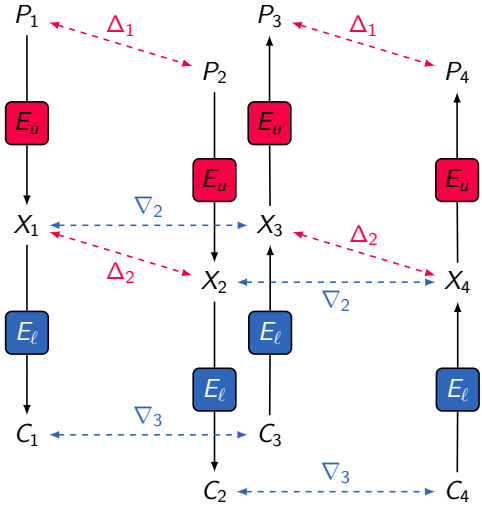




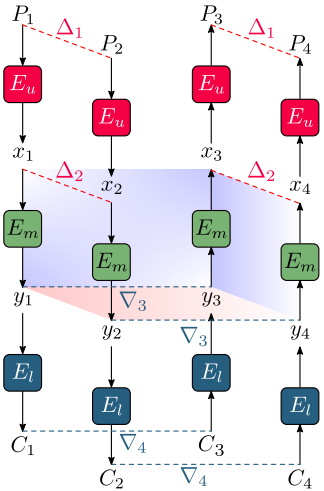
Probability of Boomerang Distinguishers [Wag99]



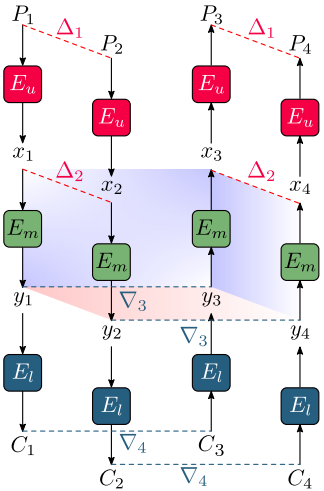
$$\mathbb{P}(P_3 \oplus P_4 = \Delta_1) = p^2 q^2$$



Sandwiching the Differentials! [DKS10; DKS14]

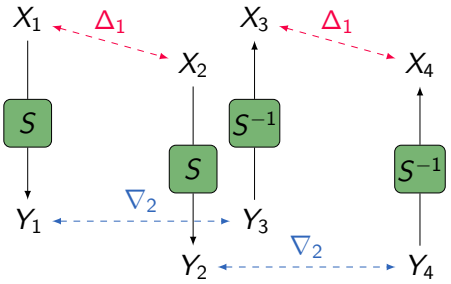


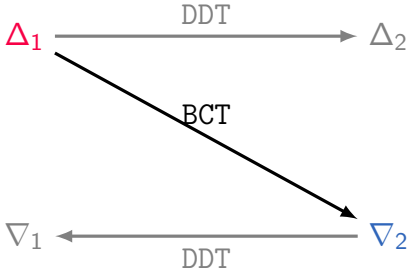
Sandwiching the Differentials! [DKS10; DKS14]



$$\mathbb{P}(P_3 \oplus P_4 = \Delta_1) \approx p^2 \times r \times q^2$$
$$r = \mathbb{P}(\Delta_2 \rightleftarrows \nabla_3)$$

Boomerang Connectivity Table (BCT) [Cid+18]

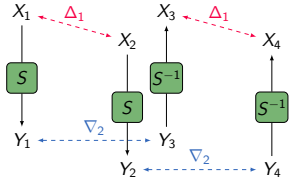


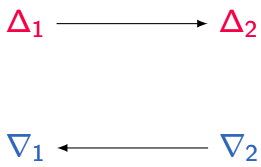


$$\mathtt{BCT}(\Delta_1, \nabla_2) \coloneqq \#\{X \in \mathbb{F}_2^n \mid S^{-1}(S(X) \oplus \nabla_2) \oplus S^{-1}(S(X \oplus \Delta_1) \oplus \nabla_2) = \Delta_1\}$$

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

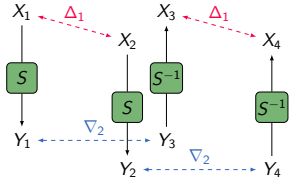
Generalized BCT Framework - I

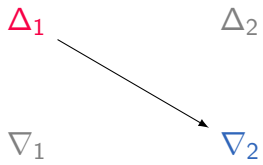




- $\mathcal{X}_{\mathtt{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}, \quad \mathtt{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\mathtt{DDT}}(\Delta_1, \Delta_2)$
- $\mathcal{X}_{\mathtt{BCT}}(\Delta_1, \nabla_2) = \{x : S^{-1}(S(x) \oplus \nabla_2) \oplus S^{-1}(S(x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

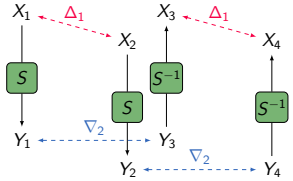
Generalized BCT Framework - I

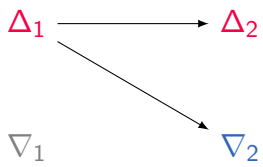




- $\mathcal{X}_{\mathtt{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}, \quad \mathtt{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\mathtt{DDT}}(\Delta_1, \Delta_2)$
- $\mathcal{X}_{\mathtt{BCT}}(\Delta_1, \nabla_2) = \{x : S^{-1}(S(x) \oplus \nabla_2) \oplus S^{-1}(S(x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

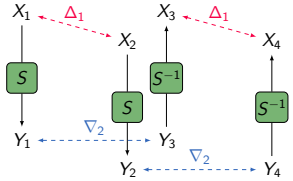
Generalized BCT Framework - I

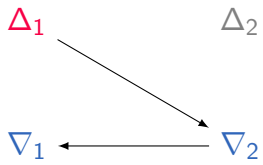




- $\mathcal{X}_{\mathtt{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}, \quad \mathtt{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\mathtt{DDT}}(\Delta_1, \Delta_2)$
- $\mathcal{X}_{\mathtt{BCT}}(\Delta_1, \nabla_2) = \{x : S^{-1}(S(x) \oplus \nabla_2) \oplus S^{-1}(S(x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

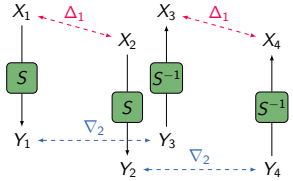
Generalized BCT Framework - I

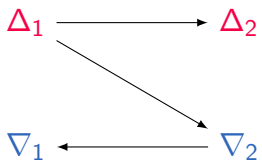




- $\mathcal{X}_{\mathtt{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}, \quad \mathtt{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\mathtt{DDT}}(\Delta_1, \Delta_2)$
- $\mathcal{X}_{\mathtt{BCT}}(\Delta_1, \nabla_2) = \{x : S^{-1}(S(x) \oplus \nabla_2) \oplus S^{-1}(S(x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

Generalized BCT Framework - I

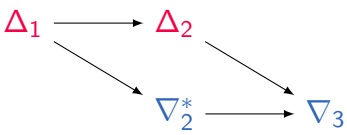


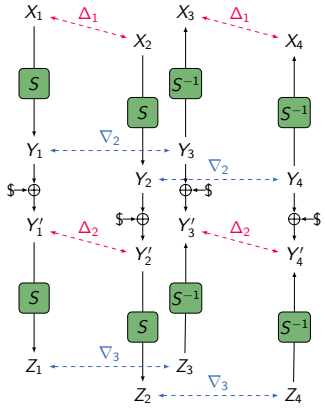


- $\mathcal{X}_{\mathtt{DDT}}(\Delta_1, \Delta_2) = \{x : S(x) \oplus S(x \oplus \Delta_1) = \Delta_2\}, \quad \mathtt{DDT}(\Delta_1, \Delta_2) = \#\mathcal{X}_{\mathtt{DDT}}(\Delta_1, \Delta_2)$
- $\mathcal{X}_{\mathtt{BCT}}(\Delta_1, \nabla_2) = \{x : S^{-1}(S(x) \oplus \nabla_2) \oplus S^{-1}(S(x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

Generalized BCT Framework (GBCT) - II

- Double Boomerang Connectivity Table (DBCT) [HB21]



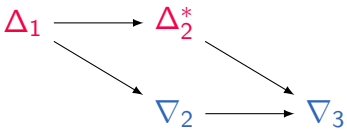


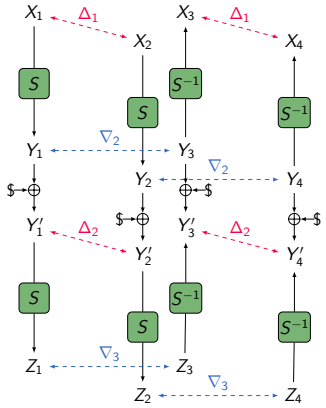
- $\mathtt{DBCT}^{\vdash}(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \mathtt{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \mathtt{LBCT}(\Delta_2, \nabla_2, \nabla_3)$
- $\mathtt{DBCT}^{\dashv}(\Delta_1, \nabla_2, \nabla_3) = \sum_{\Delta_2} \mathtt{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \mathtt{LBCT}(\Delta_2,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

Hosein Hadipour
Ph.D. Defense, Graz University of Technology

Generalized BCT Framework (GBCT) - II

- Double Boomerang Connectivity Table (DBCT) [HB21]

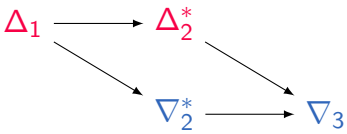


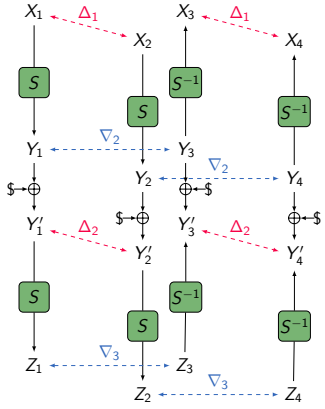


- $\mathtt{DBCT}^{\vdash}(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \mathtt{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \mathtt{LBCT}(\Delta_2, \nabla_2, \nabla_3)$
- $\mathtt{DBCT}^{\dashv}(\Delta_1, \nabla_2, \nabla_3) = \sum_{\Delta_2} \mathtt{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \mathtt{LBCT}(\Delta_2,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

Generalized BCT Framework (GBCT) - II

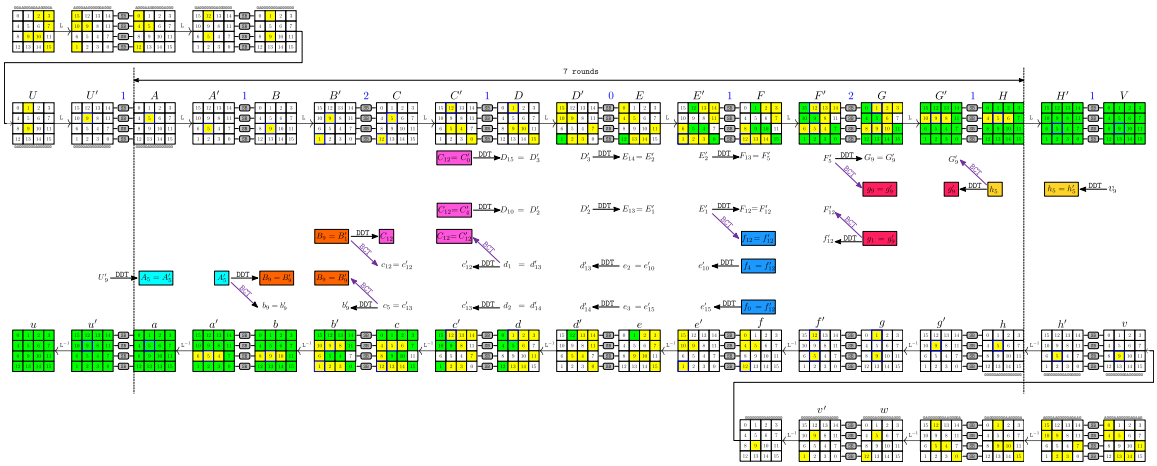
- Double Boomerang Connectivity Table (DBCT) [HB21]





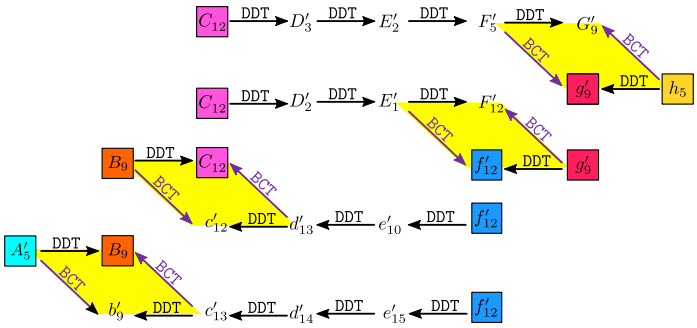
- $\mathtt{DBCT}^{\vdash}(\Delta_1, \Delta_2, \nabla_3) = \sum_{\nabla_2} \mathtt{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \mathtt{LBCT}(\Delta_2, \nabla_2, \nabla_3)$
- $\mathtt{DBCT}^{\dashv}(\Delta_1, \nabla_2, \nabla_3) = \sum_{\Delta_2} \mathtt{UBCT}(\Delta_1, \Delta_2, \nabla_2) \cdot \mathtt{LBCT}(\Delta_2,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

Application of GBCT [HB21]



Hosein Hadipour
Ph.D. Defense, Graz University of Technology

Application of GBCT [HB21]



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

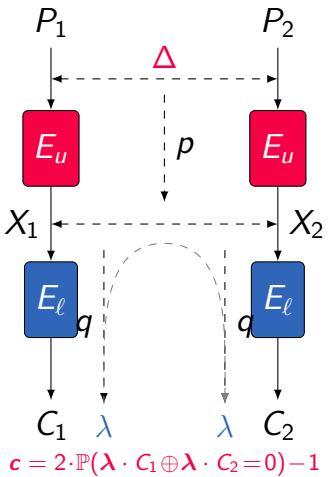
$$\Pr_{\text{total}} = \Pr(d_1 \xleftarrow{2\ \mathtt{DDT}} f'_{12}) \cdot \Pr(c_5 \xleftarrow{3\ \mathtt{DDT}} f'_{12}) \cdot \Pr(C_{12} \xrightarrow{2\ \mathtt{DDT}} E'_1) \cdot \Pr(C_{12} \xrightarrow{3\ \mathtt{DDT}} F'_5)$$

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

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

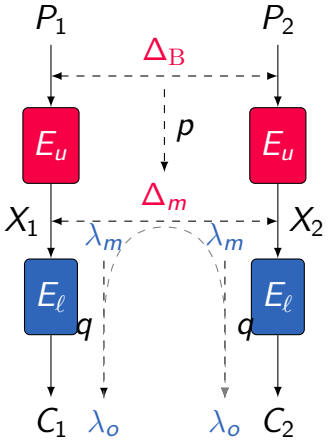
Initialize a counter list $V[z] \leftarrow 0$ for $z \in \{0,1\}$;
for $i = 0, \ldots, N-1$ **do**
  $P_1 \xleftarrow{\$} \mathbb{F}_2^n$;
  $b_1 \leftarrow \lambda \cdot E_K(P_1)$;
  $P_2 \leftarrow P_1 \oplus \Delta$;
  $b_2 \leftarrow \lambda \cdot E_K(P_2)$;
  $V[b_1 \oplus b_2] \leftarrow V[b_1 \oplus b_2] + 1$;
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



Differential-Linear Attacks

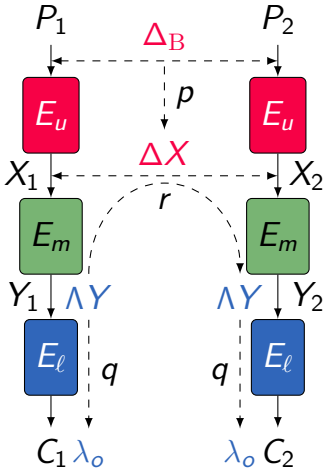
Differential-Linear (DL) Attack II [LH94]

- $p = \mathbb{P}(\Delta_{\mathrm{B}} \xrightarrow{E_u} \Delta_m)$
- $q = \mathbb{C}(\lambda_m \xrightarrow{E_\ell} \lambda_{\mathrm{F}}) = 2 \cdot \mathbb{P}(\lambda_m \cdot X \oplus \lambda_{\mathrm{F}} \cdot E_\ell(X) = 0) - 1$
- Assumptions ($\Delta X = X_1 \oplus X_2$):
  1. E_u, and E_ℓ are statistically independent
  2. $\mathbb{P}(\lambda_m \cdot \Delta X = 0) = 1/2$ when $\Delta X \neq \Delta_m$
- $\mathcal{C} = \mathbb{C}(\lambda_{\mathrm{F}} \cdot \Delta C) \approx (-1)^{\lambda_m \cdot \Delta_m} \cdot pq^2 = \pm pq^2$
- Time/Data complexity: $\mathcal{O}(\mathcal{C}^{-2})$



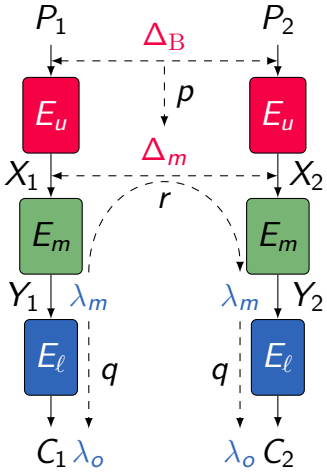
Sandwich Framework for DL Attack [BLN14; DKS14; Bar+19]

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

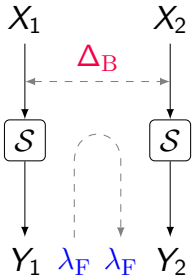


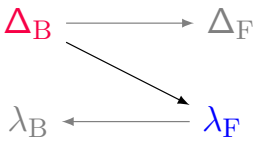
Sandwich Framework for DL Attack [BLN14; DKS14; Bar+19]

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



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



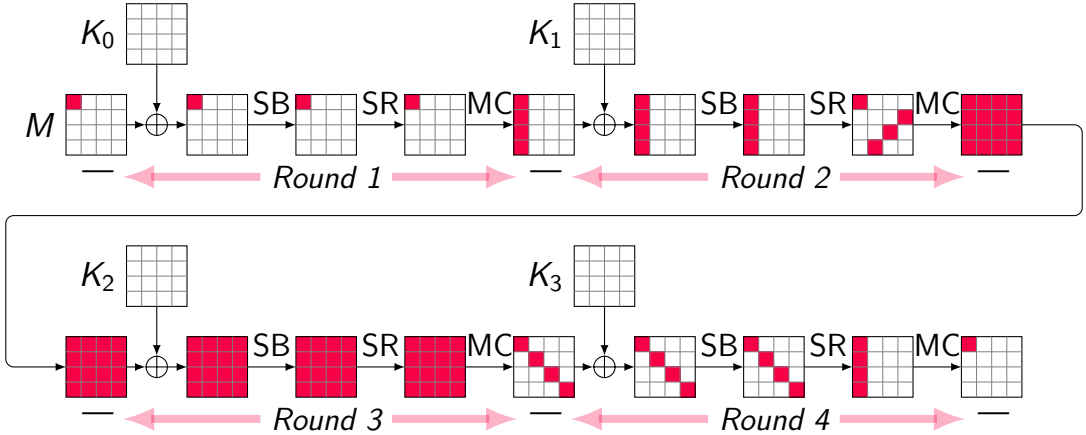


$$\mathtt{DLCT}_b(\Delta_B, \lambda_F) = \{x \in \mathbb{F}_2^n : \ \lambda_F \cdot S(x) \oplus \lambda_F \cdot S(x \oplus \Delta_B) = b\}$$

$$\mathtt{DLCT}(\Delta_B, \lambda_F) = |\mathtt{DLCT}_0(\Delta_B, \lambda_F)| - |\mathtt{DLCT}_1(\Delta_B, \lambda_F)|$$

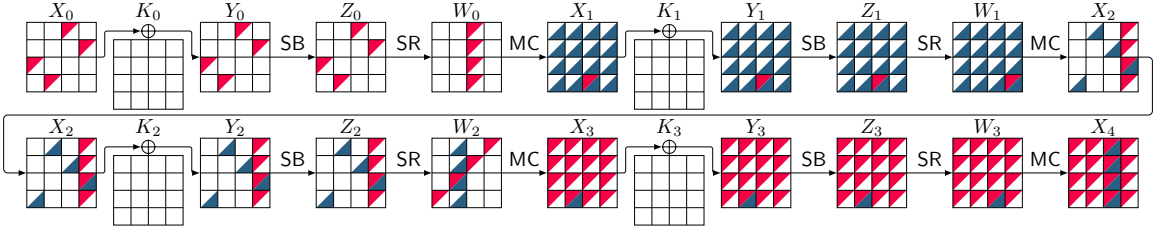
$$\mathbb{C}_{\mathtt{DLCT}}(\Delta_B, \lambda_F) = 2^{-n} \cdot \mathtt{DLCT}(\Delta_B, \lambda_F)$$

Security of AES Against Differential/Linear Attacks



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

A 4-round DL Distinguisher for AES



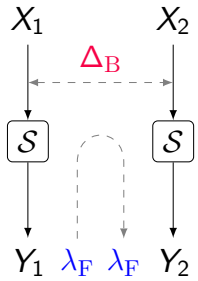
$r_u = 1, r_m = 3, r_\ell = 0,\ p = 2^{-24.00},\ r = 2^{-7.66}, q^2 = 1,\ \mathbb{C} = prq^2 = \mathbf{2^{-31.66}}$

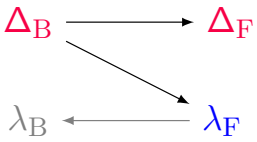
ΔX_0 00005200000000f58f000000007b0000	ΔX_1 00000000000000000000000000000b400
ΓX_4 0032000000ab00000066000000980000	-

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

Generalized DLCT Framework

Upper Differential-Linear Connectivity Table (UDLCT)



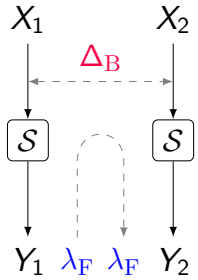


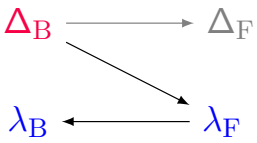
$$\texttt{UDLCT}_b(\Delta_B, \Delta_F, \lambda_F) = \{x \in \mathbb{F}_2^n : \ S(x) \oplus S(x \oplus \Delta_B) = \Delta_F \text{ and } \lambda_F \cdot \Delta_F = b\}$$

$$\texttt{UDLCT}(\Delta_B, \Delta_F, \lambda_F) = |\texttt{UDLCT}_0(\Delta_B, \Delta_F, \lambda_F)| - |\texttt{UDLCT}_1(\Delta_B, \Delta_F, \lambda_F)|$$

$$\mathbb{C}_{\texttt{UDLCT}}(\Delta_B, \Delta_F, \lambda_F) = 2^{-n} \cdot \texttt{UDLCT}(\Delta_B, \Delta_F, \lambda_F)$$

Lower Differential-Linear Connectivity Table (LDLCT)



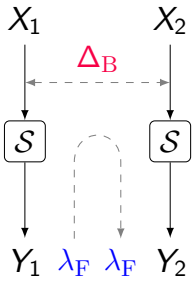


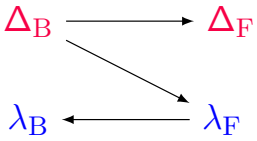
$$\texttt{LDLCT}_b(\Delta_B, \lambda_B, \lambda_F) = \{x \in \mathbb{F}_2^n : \ \lambda_B \cdot \Delta_B \oplus \lambda_F \cdot S(x) \oplus \lambda_F \cdot S(x \oplus \Delta_B) = b\}$$

$$\texttt{LDLCT}(\Delta_B, \lambda_B, \lambda_F) = |\texttt{LDLCT}_0(\Delta_B, \lambda_B, \lambda_F)| - |\texttt{LDLCT}_1(\Delta_B, \lambda_B, \lambda_F)|$$

$$\mathbb{C}_{\texttt{LDLCT}}(\Delta_B, \lambda_B, \lambda_F) = 2^{-n} \cdot \texttt{LDLCT}(\Delta_B, \lambda_B, \lambda_F)$$

Extended Differential-Linear Connectivity Table (EDLCT)



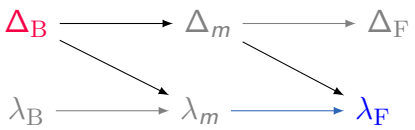


$$\mathtt{EDLCT}_b(\Delta_B, \Delta_F, \lambda_B, \lambda_F) = \{x \in \mathbb{F}_2^n : \ S(x) \oplus S(x \oplus \Delta_B) = \Delta_F \text{ and } \lambda_B \cdot \Delta_B \oplus \lambda_F \cdot \Delta_F = b\}$$

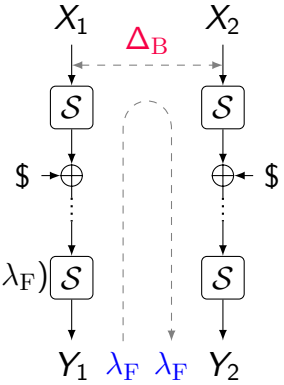
$$\mathtt{EDLCT}(\Delta_B, \Delta_F, \lambda_B, \lambda_F) = |\mathtt{EDLCT}_0(\Delta_B, \Delta_F, \lambda_B, \lambda_F)| - |\mathtt{EDLCT}_1(\Delta_B, \Delta_F, \lambda_B, \lambda_F)|$$

$$\mathbb{C}_{\mathtt{EDLCT}}(\Delta_B, \Delta_F, \lambda_B, \lambda_F) = 2^{-n} \cdot \mathtt{EDLCT}(\Delta_B, \Delta_F, \lambda_B, \lambda_F)$$

Double Differential-Linear Connectivity Table (DDLCT)

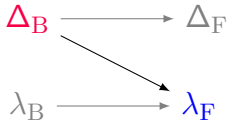


$$\texttt{DDLCT}(\Delta_B, \lambda_F) = 2^{-n} \cdot \sum_{\Delta_m} \sum_{\lambda_m} \texttt{UDLCT}(\Delta_B, \Delta_m, \lambda_m) \cdot \texttt{LDLCT}(\Delta_m, \lambda_m, \lambda_F)$$

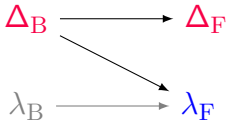


Generalized DLCT Framework (GBCT)

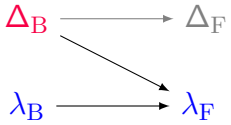
- How to formulate the correaltion for more than 1 round?



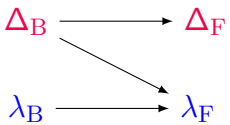
$\texttt{DLCT}(\Delta_B, \lambda_F)$



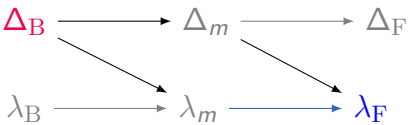
$\texttt{UDLCT}(\Delta_B, \Delta_F, \lambda_F)$



$\texttt{LDLCT}(\Delta_B, \lambda_B, \lambda_F)$

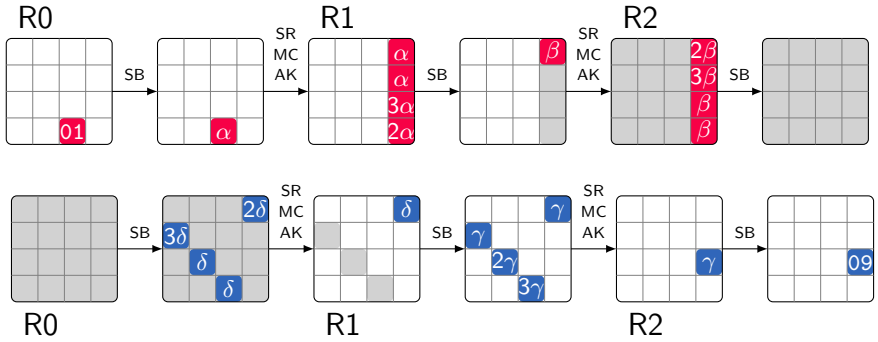


$\texttt{EDLCT}(\Delta_B, \Delta_F, \lambda_B, \lambda_F)$



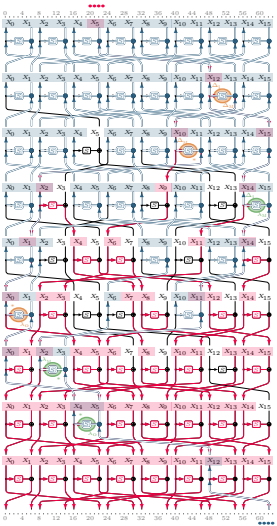
$\texttt{DDLCT}(\Delta_B, \lambda_F)$

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



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

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 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

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

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

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

$$\lambda_B = (1, ?, ?, 1) \xleftarrow{S} \lambda_F = (0, 1, 0, 0)$$

$$\lambda_B = (1, 1, ?, ?) \xleftarrow{S} \lambda_F = (1, 0, 0, 0)$$

$$\lambda_B = (0, ?, ?, ?) \xleftarrow{S} \lambda_F = (1, 1, 0, 0)$$

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

$$\Delta_B = (0,0,0,1) \xrightarrow{S} \Delta_F = (?,1,?,?)$$

$$\Delta_B = (0,1,0,0) \xrightarrow{S} \Delta_F = (1,?,?,?)$$

$$\Delta_B = (1,0,0,0) \xrightarrow{S} \Delta_F = (1,1,?,?)$$

$$\Delta_B = (1,0,0,1) \xrightarrow{S} \Delta_F = (?,0,?,?)$$

$$\Delta_B = (1,1,0,0) \xrightarrow{S} \Delta_F = (0,?,?,?)$$

$$\lambda_B = (1,?,?,1) \xleftarrow{S} \lambda_F = (0,1,0,0)$$

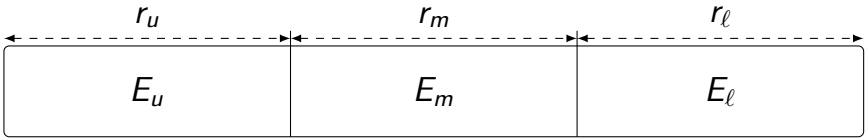
$$\lambda_B = (1,1,?,?) \xleftarrow{S} \lambda_F = (1,0,0,0)$$

$$\lambda_B = (0,?,?,?) \xleftarrow{S} \lambda_F = (1,1,0,0)$$

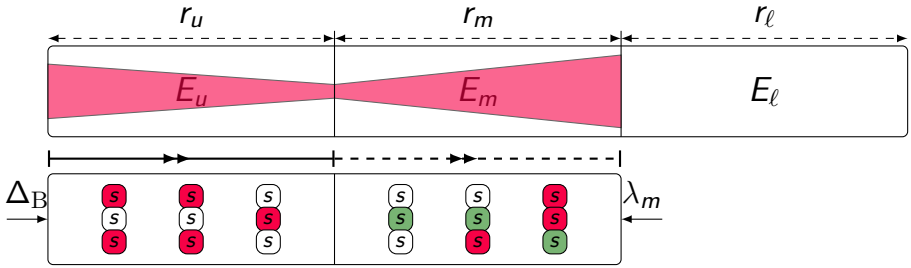
Automatic Tools to Search for DL Distinguishers

Overview of Our Method to Search for Distinguishers

Overview of Our Method to Search for Distinguishers

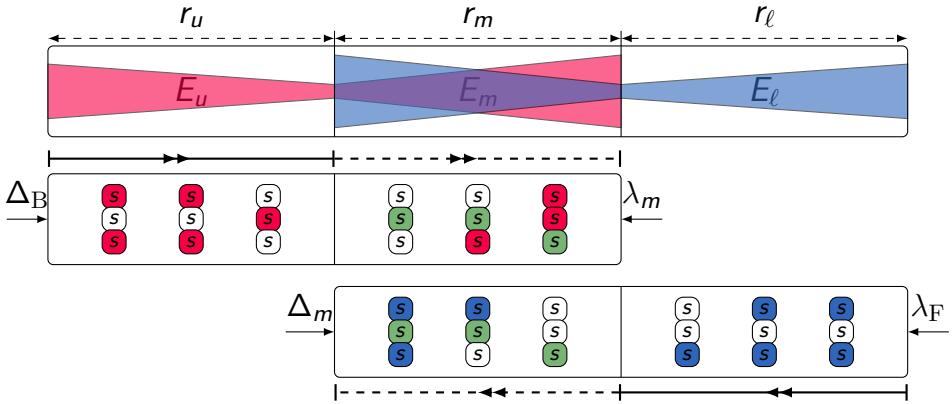


Overview of Our Method to Search for Distinguishers

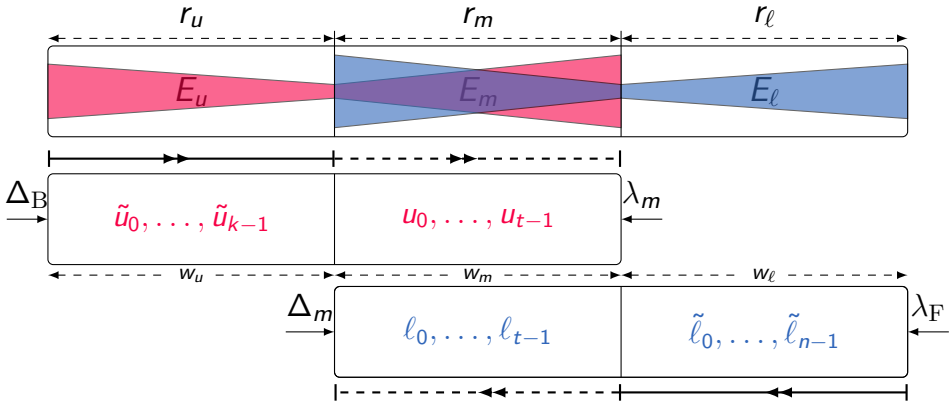


differentially active S-box linearly active S-box common active S-box

Overview of Our Method to Search for Distinguishers



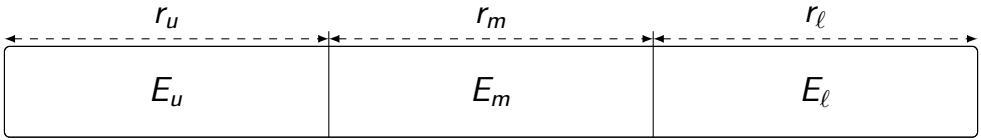
Overview of Our Method to Search for Distinguishers



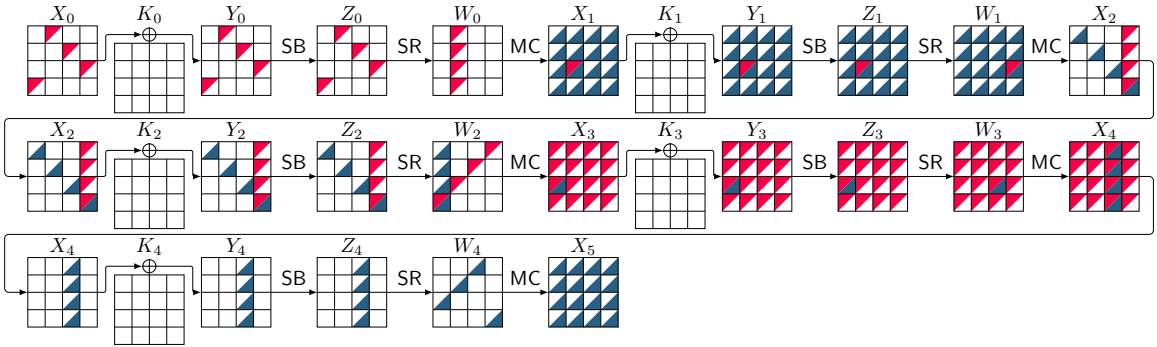
$$\min \left(\sum_{i=0}^{k-1} w_u \cdot \tilde{u}_i + \sum_{j=0}^{t-1} w_m \cdot \texttt{bool2int}\left(\ell_j + u_j = 2\right) + \sum_{k=0}^{n-1} w_\ell \cdot \tilde{\ell}_k \right)$$

Usage of Our Tool

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



Results: A 5-round DL Distinguisher for AES

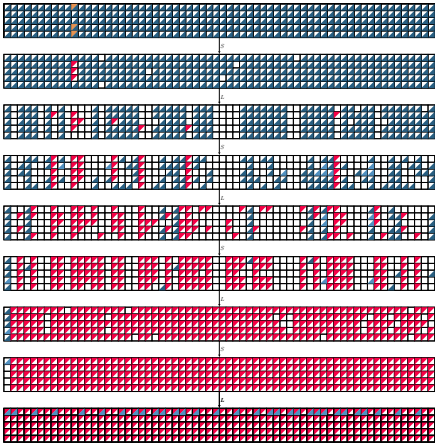


$r_0 = 1, r_m = 3, r_1 = 1,\ p = 2^{-24.00}, r = 2^{-7.66},\ q^2 = 2^{-24.00},\ prq^2 = 2^{-55.66}$

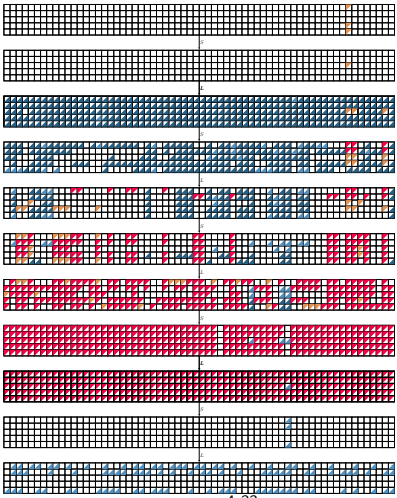
ΔX_0 001c00000000e200000000dfb3000000 ΔX_1 00000000000000000f7000000000000

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

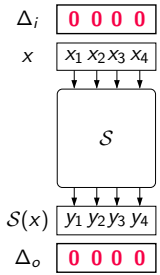
Cipher	#R	$\mathbb{C}$	💻	Ref.
Simeck-48	8	$\mathbf{1}$	✓	This work
	17	$2^{-22.37}$		[ZWH24]
	17	$\mathbf{2^{-13.89}}$	✓	This work
	18	$2^{-24.75}$		[ZWH24]
	18	$\mathbf{2^{-15.89}}$		This work
	19	$\mathbf{2^{-17.89}}$		This work
	20	$\mathbf{2^{-21.89}}$		This work

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



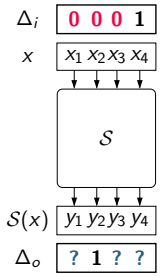
$\Delta_i \setminus \Delta_o$	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2
2	0	2	0	2	0	0	0	4	0	2	2	0	0	0	2	2
3	0	2	0	2	0	0	4	0	0	2	2	0	0	0	2	2
4	0	0	0	0	0	0	0	0	0	0	4	4	2	2	2	2
5	0	0	0	0	2	2	2	2	0	0	4	4	0	0	0	0
6	0	2	0	2	0	4	0	0	0	2	2	0	2	2	0	0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

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

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

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

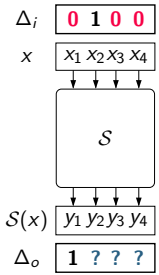
$$\Delta_i = (0,0,0,0) \xrightarrow{\mathcal{S}} \Delta_o = (0,0,0,0)$$

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

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

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

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

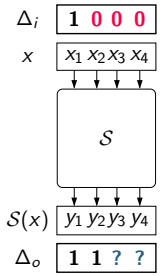
$$\Delta_i \neq (0,0,0,0) \xrightarrow{S} \Delta_o \neq (0,0,0,0)$$

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

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

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

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

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

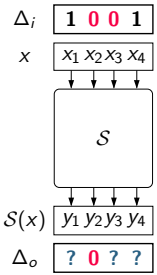
$$\Delta_i = (0,0,0,1) \xrightarrow{S} \Delta_o = (?,1,?,?)$$

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

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

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

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

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

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

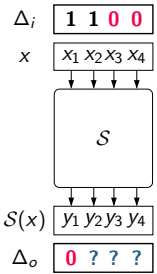
$$\Delta_i = (0,1,0,0) \xrightarrow{S} \Delta_o = (1,?,?,?)$$

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

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

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

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

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

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

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

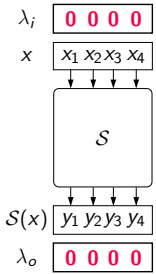
$$\Delta_i = (1,0,0,0) \xrightarrow{S} \Delta_o = (1,1,?,?)$$

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

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

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



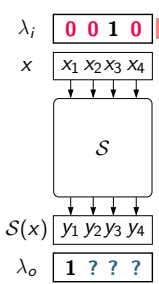
$\lambda_i \setminus \lambda_o$	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	4	-4	0	-8	-4	-4	0	0	4	-4	-8	0	4	4
2	0	0	0	0	0	0	0	0	0	8	8	0	0	8	-8	0
3	0	-8	4	4	0	0	-4	4	0	0	-4	4	-8	0	-4	-4
4	0	4	0	4	0	4	8	-4	0	4	0	4	-8	-4	0	4
5	0	4	-4	-8	0	-4	-4	0	0	4	-4	8	0	-4	-4	0
6	0	-4	8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

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

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

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

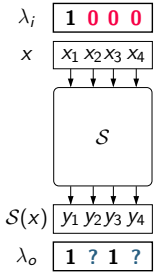
$$\lambda_i = (0,0,0,0) \xrightarrow{\mathcal{S}} \lambda_o = (0,0,0,0)$$

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

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

Deterministic Bit-Wise Linear Trails

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



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

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

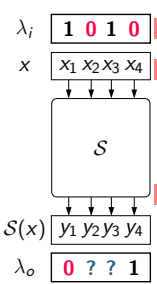
$$\lambda_i \neq (0,0,0,0) \xrightarrow{S} \lambda_o \neq (0,0,0,0)$$

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

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

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	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

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

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

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

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

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

CP Model for Deterministic Bit-Wise Trails

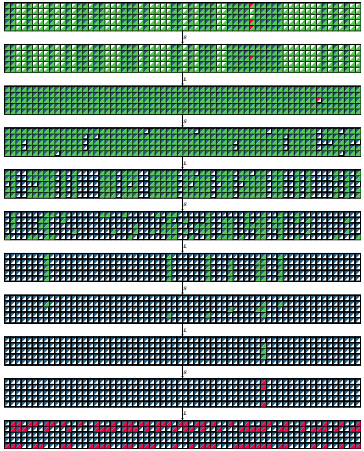
- For each bit position, we define an integer variable with domain $\{\mathbf{0}, \mathbf{1}, -\mathbf{1}\}$.
- Define CP constraints to model the propagation of deterministic bit-wise trails.

S-box

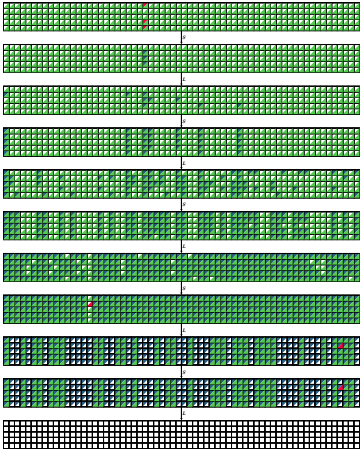
Assume that $x[i], y[i]$ are integer variables with domain $\{-1, 0, 1\}$ to encode the input and output differences at the i-th bit position, respectively. The valid deterministic differential transitions satisfy the following:

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

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: ◩/◪)