

# Finding Complete Impossible Differential Attacks on AndRX and ARX Designs

---

Debasmita Chakraborty<sup>1, 2</sup>, Hosein Hadipour<sup>1</sup>, **Hoa Nguyen**<sup>3</sup>, Maria Eichlseder<sup>1</sup>

May 30, 2024

Graz University of Technology, Graz, Austria

Indian Statistical Institute, Kolkata, India

Univ Rennes, CNRS, IRISA, Rennes, France



## Research gap

- 🕒 Lack of **automatic tool** to find full **Impossible Differential** for **AndRX** and **ARX** ciphers

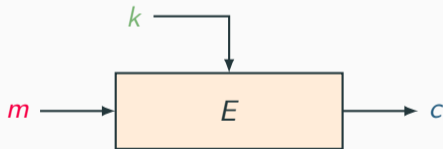
## Our contributions

- 😊 **Expanded** [Hadipour et al., 2024]'s method
  - Handling indirect contradictions
  - Adaptation for AndRX and ARX designs
- 😊 **Proposed** a unified model **to combine** both distinguisher identification and key recovery for AndRX designs

# Block Ciphers



- Encrypt plaintext  $m$  into cipher text  $c$  using key  $k$ .
- Generates a family of  $2^k$  permutations indexed by the key  $k$ .

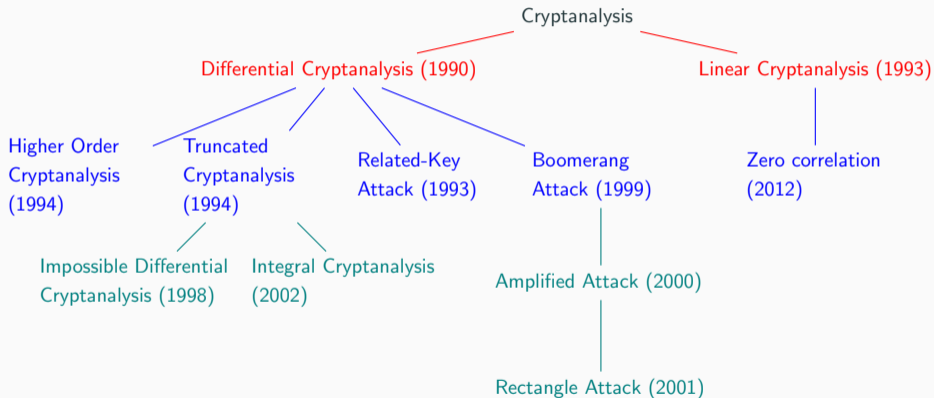


$$E : \{0, 1\}^n \times \{0, 1\}^k \rightarrow \{0, 1\}^n$$

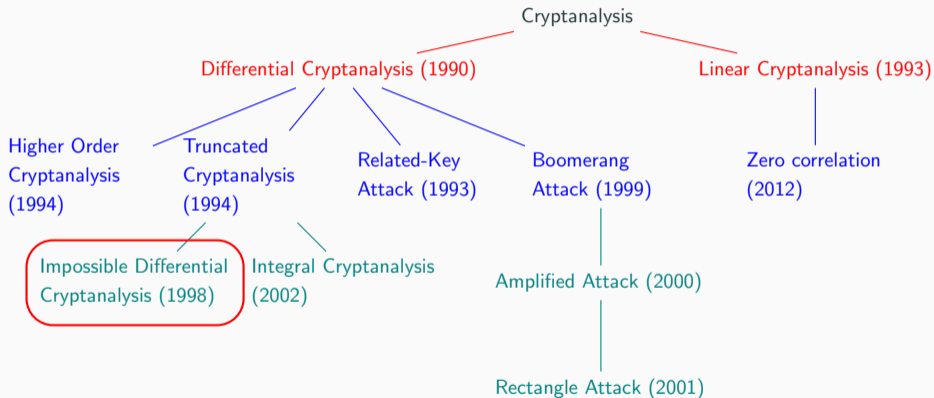
- Exhaustive search: try all  $2^k$  possible keys.
- **Secure** if no attack **faster**.
- Various other attacks.



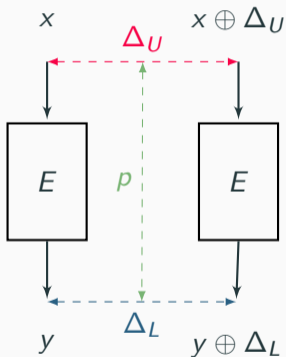
# Variants of Cryptanalysis



# Variants of Cryptanalysis



# Classical Differential cryptanalysis

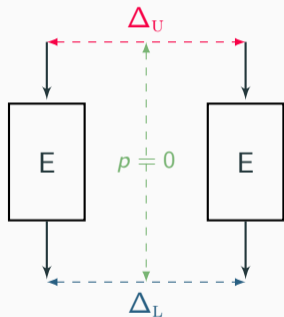


- Introduced by [Biham and Shamir, 1990]
- Given an **input** difference between two plaintexts, some **output** differences **occur more often** than others.
- A **differential** is a pair  $(\Delta_U, \Delta_L)$

# Impossible differential attacks



💡 Exploit differentials of probability 0 (never occur).



🔍 Find an impossible differential ( $\Delta_U \not\rightarrow \Delta_L$ )

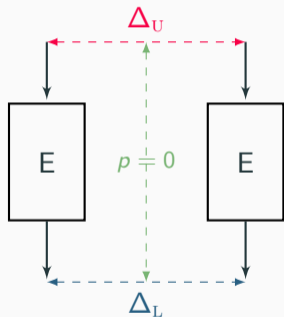
🔑 Build a key-recovery attack



# Impossible differential attacks

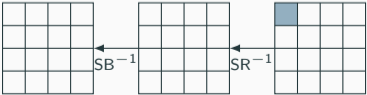
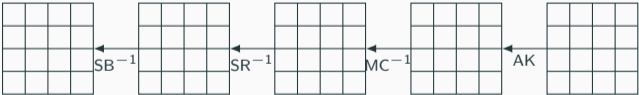
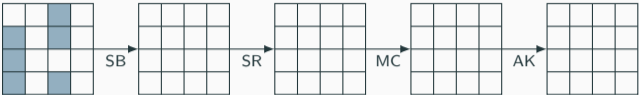


💡 Exploit differentials of probability 0 (never occur).

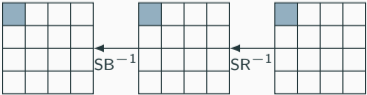
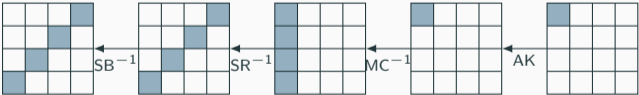
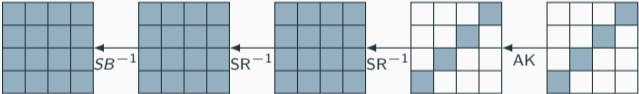
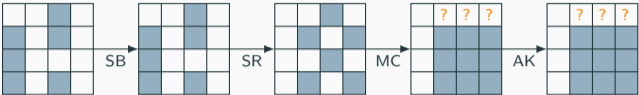


- 🔍 Find an impossible differential ( $\Delta_U \not\rightarrow \Delta_L$ )
  - Miss-in-the-middle technique [Biham et al., 1999]
  - U-method [Kim et al., 2003]
- 🔑 Build a key-recovery attack
  - Early abort technique [Lu et al., 2008]

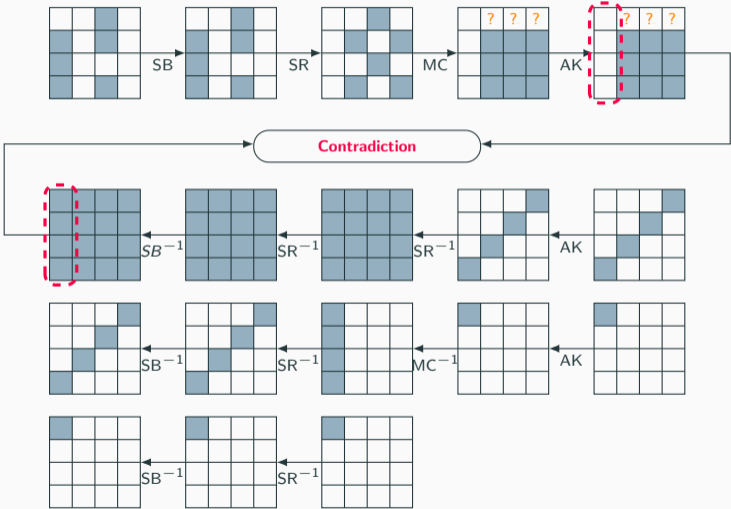
# Miss-in-the-Middle Technique



# Miss-in-the-Middle Technique



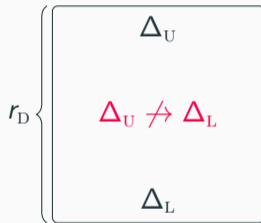
# Miss-in-the-Middle Technique



# Impossible differential attacks



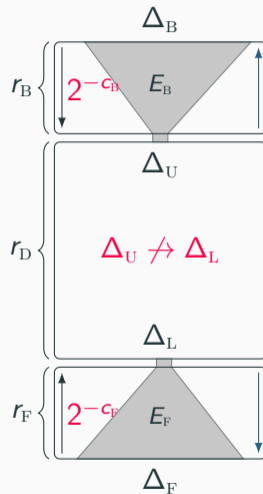
- 🔍 Find an impossible-differential  $\Delta_U \not\rightarrow \Delta_L$
- 🔑 Build a key-recovery attack



# Impossible differential attacks



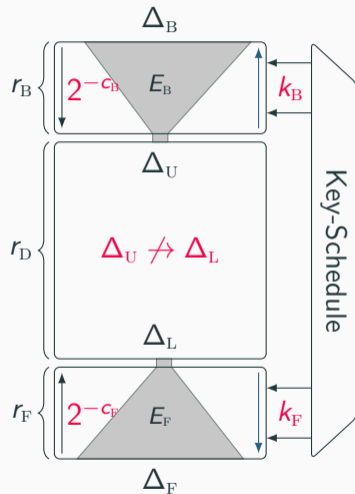
- 🔍 Find an impossible-differential  $\Delta_U \not\rightarrow \Delta_L$
- 🔑 Build a key-recovery attack
  - *Pair Generation.* Generate N pairs satisfying  $(\Delta_B, \Delta_F)$



# Impossible differential attacks



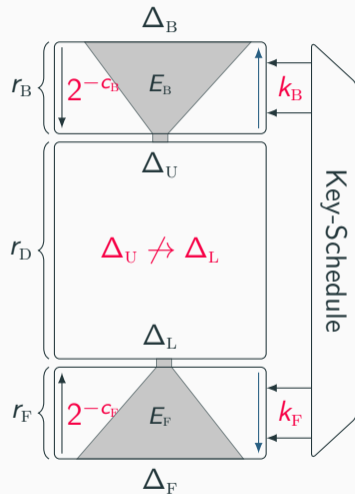
- 🔍 Find an impossible-differential  $\Delta_U \not\rightarrow \Delta_L$
- 🔑 Build a key-recovery attack
  - *Pair Generation.* Generate N pairs satisfying  $(\Delta_B, \Delta_F)$
  - *Guess-and-Filter.* For all  $k \in k_B \cup k_F$ :
    - If a pair suggests  $(\Delta_U, \Delta_L)$ , discard  $k$



# Impossible differential attacks

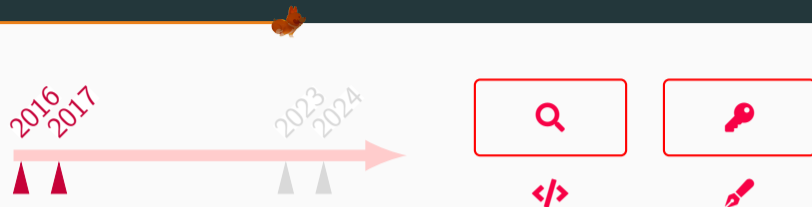


- 🔍 Find an impossible-differential  $\Delta_U \not\rightarrow \Delta_L$
- 🔑 Build a key-recovery attack
  - *Pair Generation.* Generate N pairs satisfying  $(\Delta_B, \Delta_F)$
  - *Guess-and-Filter.* For all  $k \in k_B \cup k_F$ :
    - If a pair suggests  $(\Delta_U, \Delta_L)$ , discard  $k$
  - *Exhaustive Search.* Brute force the remaining key candidates



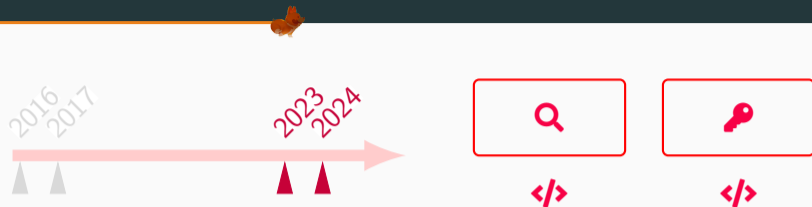


# Automatic tools for ID attacks



- Previous works:
  - CRYPTO 2016 [Derbez and Fouque, 2016]
  - Eprint 2016 [Cui et al., 2016]
  - EUROCRYPT 2017 [Sasaki and Todo, 2017]

# Automatic tools for ID attacks



- Previous works:
  - CRYPTO 2016 [Derbez and Fouque, 2016]
  - Eprint 2016 [Cui et al., 2016]
  - EUROCRYPT 2017 [Sasaki and Todo, 2017]
- New approach:
  - EUROCRYPT 2023 [Hadipour et al., 2023]: Introduce the CP approach
  - TOSC 2024 [Hadipour et al., 2024]: Extend to weakly-aligned designs

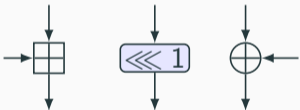


How to **automate** key recovery for complete ID attacks for **AndRX** and **ARX** ciphers?

# ARX and AndRX designs

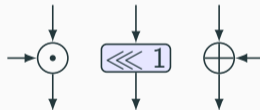


## ARX



Addition Rotation Xor

## AndRX



And Rotation Xor



- **Expanded** [Hadipour et al., 2024]'s method
  - Enhanced the model for finding the complicated contradiction
  - Adapted for AndRX and ARX designs
- **Proposed** a unified model to **combine** both distinguisher identification (🔍) and key recovery (🔑) for AndRX designs

Modeling the Distinguishers 🔍

Modeling the Key-Recovery 🔑

Applications

Summary

## Modeling the Distinguishers

---

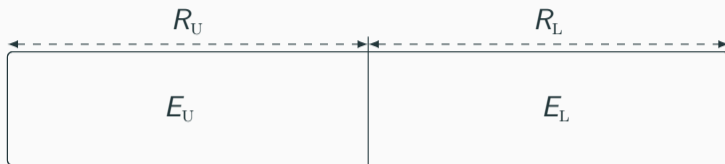
## Previous Methods to Search for Distinguishers



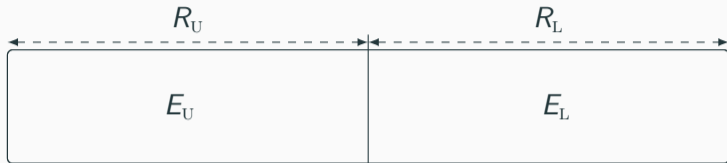
$E$



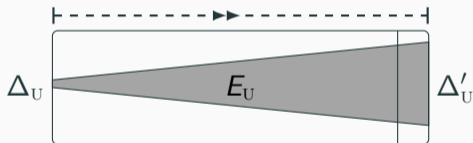
## Previous Methods to Search for Distinguishers



# Previous Methods to Search for Distinguishers



✓  $CSP_U(\Delta_U, \Delta'_U)$

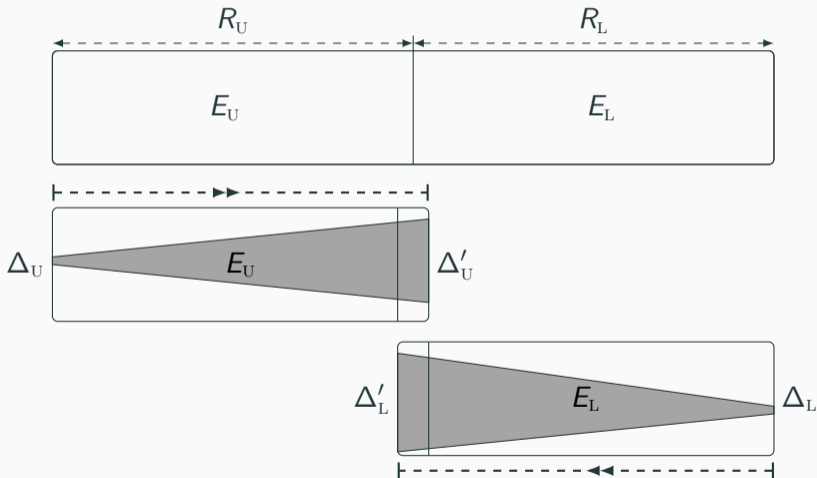


# Previous Methods to Search for Distinguishers



✓  $CSP_U(\Delta_U, \Delta'_U)$

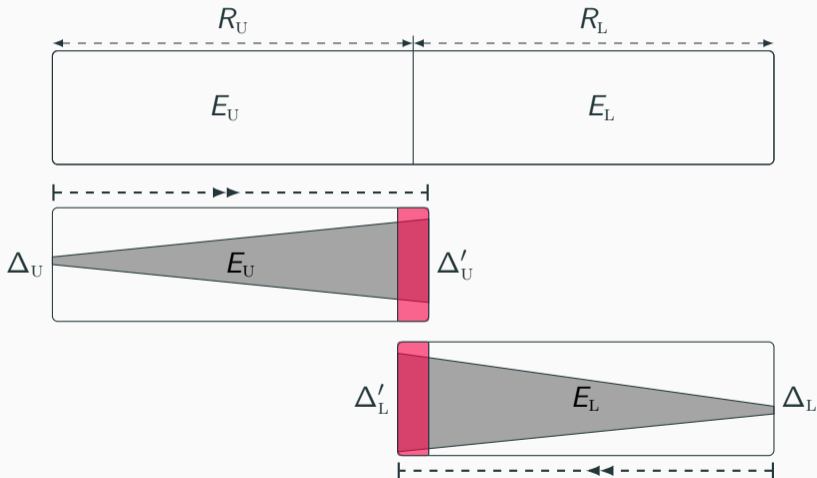
✓  $CSP_L(\Delta_L, \Delta'_L)$



# Previous Methods to Search for Distinguishers



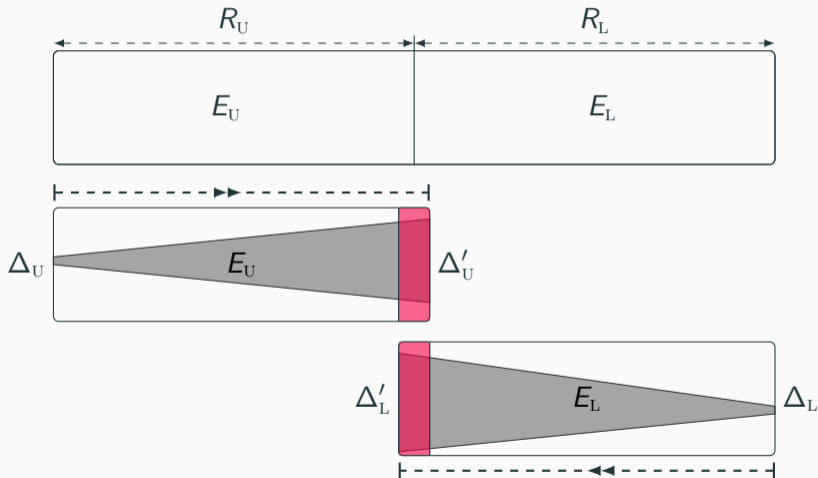
- ✓  $CSP_U(\Delta_U, \Delta'_U)$
- ✓  $CSP_L(\Delta_L, \Delta'_L)$
- ✓  $CSP_M(\Delta'_U, \Delta'_L)$



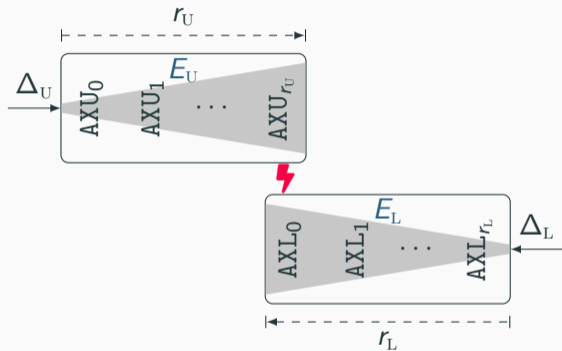
# Previous Methods to Search for Distinguishers



- ✓  $CSP_U(\Delta_U, \Delta'_U)$
- ✓  $CSP_L(\Delta_L, \Delta'_L)$
- ✓  $CSP_M(\Delta'_U, \Delta'_L)$



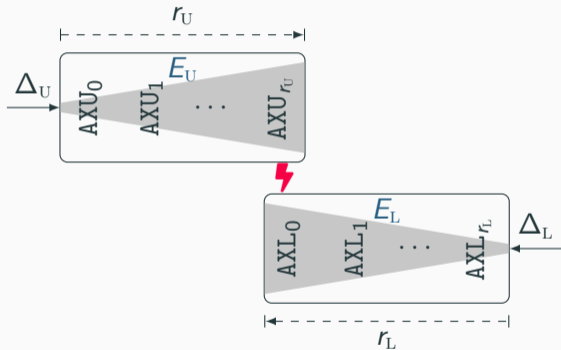
# Previous Methods to Search for Distinguishers



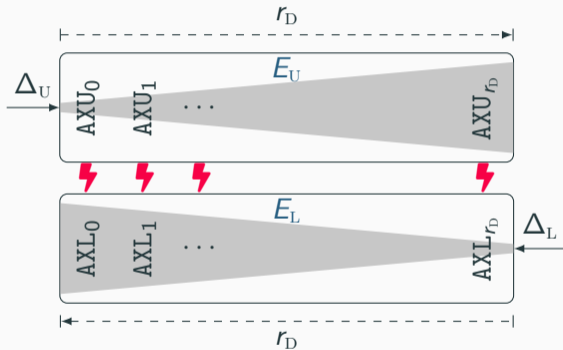
[Hadipour et al., 2023]'s Model.

# Previous Methods to Search for Distinguishers

 Find ID distinguisher for  $r_D (= r_U + r_L)$  rounds



[Hadipour et al., 2023]'s Model.



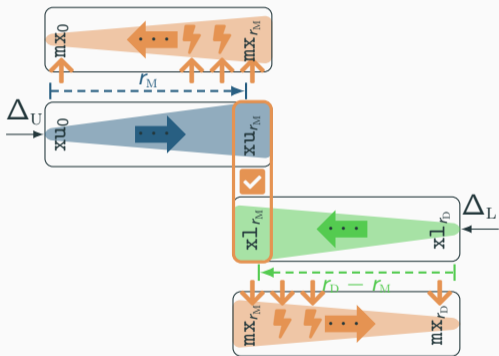
[Hadipour et al., 2024]'s Model

What if there are **no direct contradictions**?

- Some contradictions may **not** be **detectable** by direct checks.



# New CP Model to Identify Indirect Contradictions



- Focus on **indirect contradictions** described in [Sadeghi and Bagheri, 2018]
- Method Overview:
  - No direct contradiction
  - **Merge** information from both trails at a specific round
  - Propagate **merged** information in both directions



- Used to **encode the propagation** of deterministic differential/linear trails.
- **Differences** at each bit position **encoded** via an integer variable.
- Domain:  $\{-1, 0, 1\}$ .
  - 0: Fixed difference value of 0.
  - 1: Fixed difference value of 1.
  - -1: Difference value is either 0 or 1 (unknown).

# Advanced Bit-wise CP Model for Identifying ID/ZC Distinguishers



- Expand on bit-wise CP model from [Hadipour et al., 2024].
- Introduce **new rules** for AND and modular addition.
  - And
  - Full adder
  - Modular Addition
- Extends model to detect **indirect contradictions** beyond direct ones.

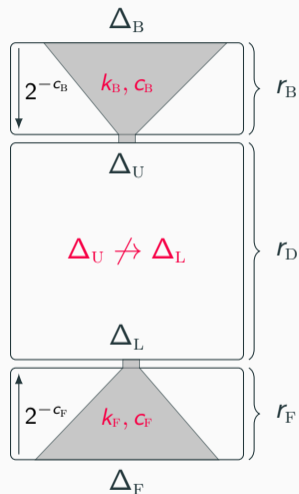
## Modeling the Key-Recovery

---

# Complexity Analysis of ID Attack [Boura et al., 2018, Boura et al., 2014]



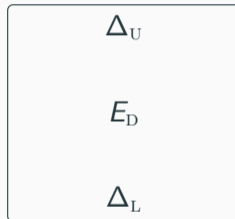
- Number of required pairs:  $N$
- Pair generation:  $T_0 = N2^{n+1-|\Delta_B|-|\Delta_F|}$
- Guess-and-filter:
  - $T_1 + T_2 = N + 2^{|k_B \cup k_F|} \frac{N}{2^{c_B+c_F}}$
  - $P = (1 - 2^{-(c_B+c_F)})^N$
- Exhaustive search:  $T_3 = P2^k$
- $T_{tot} = (T_0 + (T_1 + T_2)C_{E'} + T_3)C_E$



# Overall View of Our CP Model for Key-Recovery

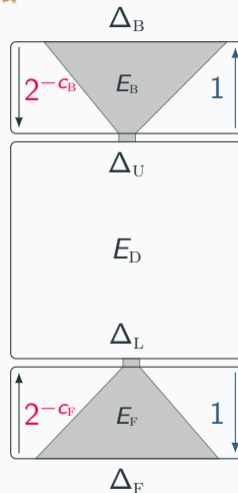


- ✓ Model the distinguisher for  $E_D (\Delta_U, \Delta_F)$



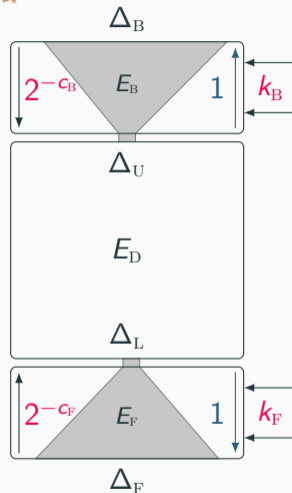
# Overall View of Our CP Model for Key-Recovery

- ✓ Model the distinguisher for  $E_D$  ( $\Delta_U, \Delta_F$ )
- ✓ Model the filters in  $E_B$ , and  $E_F$  ( $c_B, c_F, \Delta_B, \Delta_F$ )



# Overall View of Our CP Model for Key-Recovery

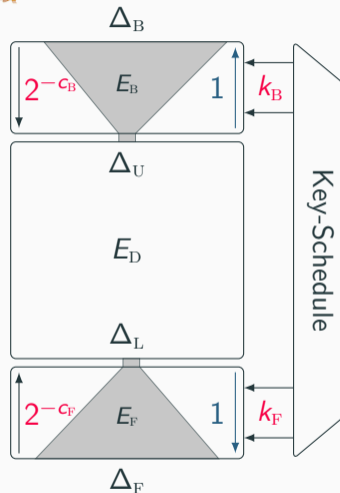
- ✓ Model the distinguisher for  $E_D$  ( $\Delta_U, \Delta_F$ )
- ✓ Model the filters in  $E_B$ , and  $E_F$  ( $c_B, c_F, \Delta_B, \Delta_F$ )
- ✓ Model the guess-and-determine in  $E_B$ , and  $E_F$
- ✓ Model Equivalent Sub-key Technique





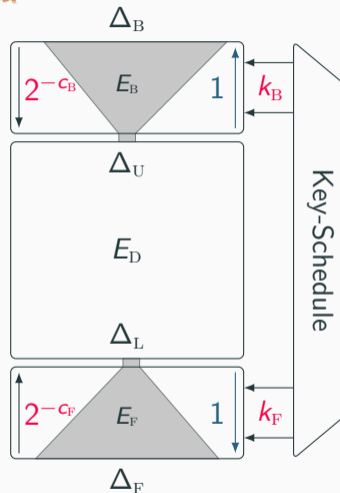
# Overall View of Our CP Model for Key-Recovery

- ✓ Model the distinguisher for  $E_D$  ( $\Delta_U, \Delta_F$ )
- ✓ Model the filters in  $E_B$ , and  $E_F$  ( $c_B, c_F, \Delta_B, \Delta_F$ )
- ✓ Model the guess-and-determine in  $E_B$ , and  $E_F$
- ✓ Model Equivalent Sub-key Technique
- ✓ Model the complexity formulas



# Overall View of Our CP Model for Key-Recovery

- ✔ Model the distinguisher for  $E_D$  ( $\Delta_U, \Delta_F$ )
- ✔ Model the filters in  $E_B$ , and  $E_F$  ( $c_B, c_F, \Delta_B, \Delta_F$ )
- ✔ Model the guess-and-determine in  $E_B$ , and  $E_F$
- ✔ Model Equivalent Sub-key Technique
- ✔ Model the complexity formulas
- ✔ Objective: Minimize the total time complexity

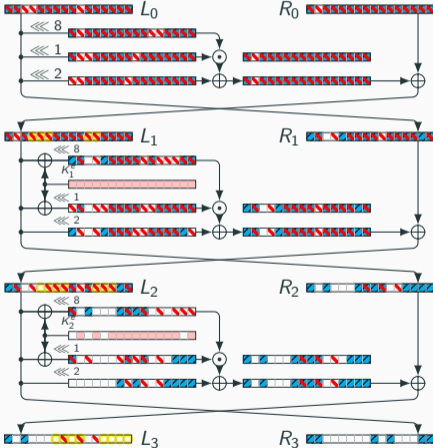
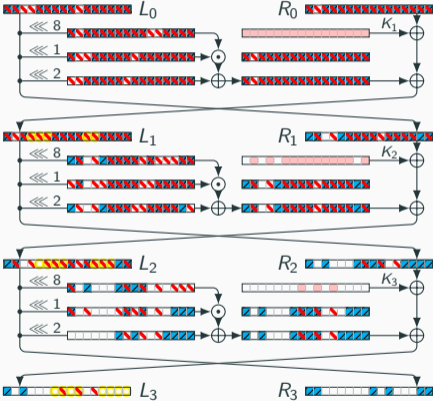


# Equivalent Subkey Technique



- Widely used in key-recovery attacks
- Reduces the number of guessed subkey bits
- Methodology:
  - Move  $K_i$  of  $Round_i$  to  $Round_{i+1}$  for  $0 \leq i \leq (r_b - 1)$
  - Move  $K_{i+1}$  of  $Round_i$  to  $Round_{i-1}$  for  $r_b + r_d \leq i \leq r_b + r_d + r_f - 1$

# Equivalent Subkey Technique

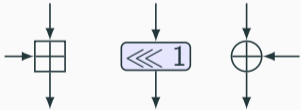


# Applications

---

## ARX

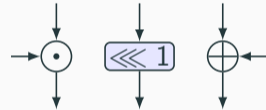
- Block ciphers: LEA, SPECK
- Stream ciphers: ChaCha
- MAC algorithms: SipHash, Chaskey



Addition Rotation Xor

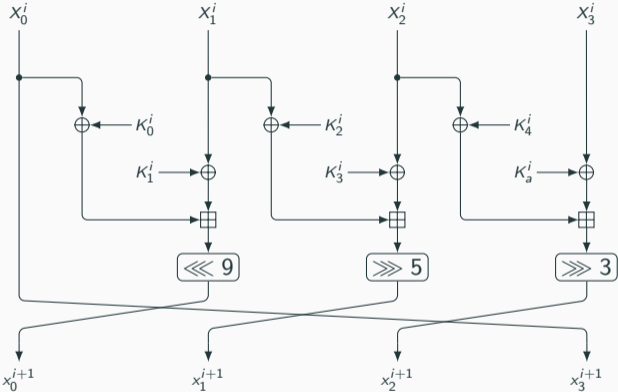
## AndRX

- SIMON and Simeck

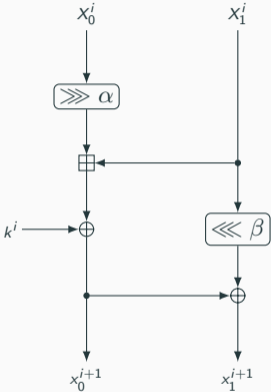


And Rotation Xor

# ARX Ciphers

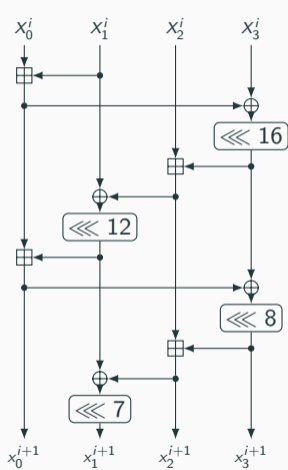


LEA

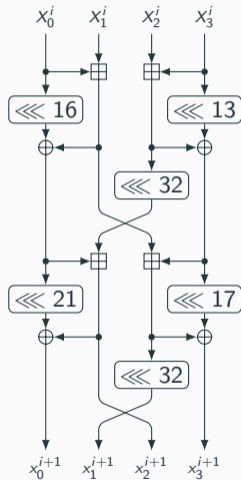


Speck

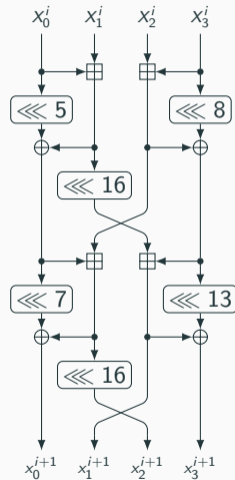
# ARX Ciphers



ChaCha



SipHash



Chaskey

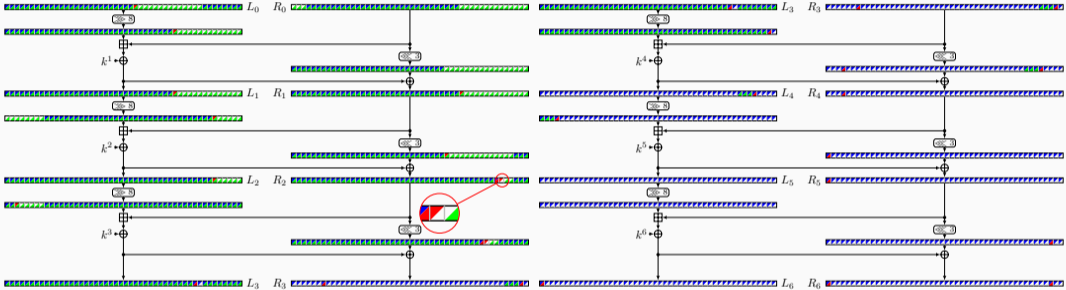








0	000000000000000000000000000000000000 *****10000000000000000000000000000000	000000000000000000000000000000000000 000000000000000000000000000000000000
1	*****10000000000000000000000000000000 000000000000000000*****1000000000	*****10000000000000000000000000000000 *****100000000000*****100000
2	***** *****	*****1 *****
2	0000000000000000000000000000000000001000 *****1	*****1000 0000000*****1
3	000000000000000000000000000000000000 100000000000000000000000000000000000	000000000000000000000000000000000000 000000000000000000000000000000000000
4	100000000000000000000000000000000000 000000000000000010000000000000000000	100000000000000000000000000000000000 100000000000000000010000000000000000

Cluster of  $2^7$  impossible-differential distinguishers for 4-round Chaskey.

# Cluster of $2^{65}$ ID distinguishers for 6-round SPECK-96



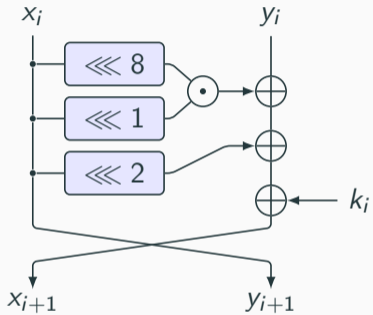
 bit difference (linear mask) 1 forward  
 unknown difference (linear mask) forward

 bit difference (linear mask) 1 backward  
 unknown difference (linear mask) backward

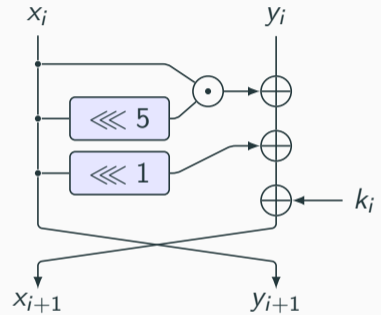
# Results



Cipher	Contradiction	#R	#Dist.	Ref.
SPECK-32	Direct	6	3	[Ren and Chen, 2019]
	Direct	6	$2^4$	This work
SPECK-48	Direct	6	20	[Ren and Chen, 2019]
	Direct	6	$2^{17}$	This work
SPECK-64	Direct	6	157	[Lee et al., 2016, Ren and Chen, 2019]
	Direct	6	$2^{33}$	This work
SPECK-96	Direct	<b>6</b>	$2^{65}$	This work
SPECK-128	Direct	<b>6</b>	$2^{97}$	This work
LEA	Direct	10	-	[Cui et al., 2016]
	Direct	10	$2^2$	This work
ChaCha	Direct	<b>5</b>	$2^{80}$	This work
SipHash	Direct	<b>4</b>	$2^{14}$	This work
Chaskey	Direct	4	15	[Saber et al., 2021]
	Direct	4	$2^7$	This work



SIMON



Simeck

# ID Attacks on SIMON



Cipher	#R	Time	Data	Mem.	Ref.
SIMON-32-64	19/20 19/20	$2^{62.56} / 2^{62.8}$ $2^{59} / 2^{62}$	$2^{32} / 2^{32}$ $2^{30.79} / 2^{31.47}$	$2^{44} / 2^{43.5}$ $2^{47.68} / 2^{44.48}$	[Boura et al., 2014, Derbez and Fouque, 2016] This work
SIMON-48-72	20 20	$2^{70.69}$ $2^{67.37}$	$2^{48}$ $2^{46.48}$	$2^{58}$ $2^{64}$	[Boura et al., 2014] This work
SIMON-48-96	21 21	$2^{94.73}$ $2^{88.47}$	$2^{48}$ $2^{45.48}$	$2^{70}$ $2^{76.49}$	[Boura et al., 2014] This work
SIMON-64-96	21 21/ 22	$2^{94.56}$ $2^{80.4} / 2^{91.81}$	$2^{64}$ $2^{62.79} / 2^{63.27}$	$2^{60}$ $2^{71.79} / 2^{84.28}$	[Boura et al., 2014] This work
SIMON-64-128	22 22/ 23	$2^{126.56}$ $2^{112.33} / 2^{124}$	$2^{64}$ $2^{62.79} / 2^{62.47}$	$2^{75}$ $2^{84.78} / 2^{99.5}$	[Boura et al., 2014] This work
SIMON-96-96	24 24	$2^{94.62}$ $2^{92}$	$2^{94}$ $2^{92.47}$	$2^{61}$ $2^{69.39}$	[Boura et al., 2014] This work
SIMON-96-144	25 25	$2^{142.59}$ $2^{124.793}$	$2^{96}$ $2^{94.793}$	$2^{77}$ $2^{84.785}$	[Boura et al., 2014] This work
SIMON-128-128	27 28	$2^{126.6}$ $2^{114.641}$	$2^{94}$ $2^{110.6}$	$2^{61}$ $2^{86}$	[Boura et al., 2014] This work
SIMON-128-192	28 29/30	$2^{190.56}$ $2^{167.42} / 2^{181}$	$2^{128}$ $2^{127.278} / 2^{127.64}$	$2^{77}$ $2^{97.278} / 2^{112.68}$	[Boura et al., 2014] This work
SIMON-128-256	30 30/31	$2^{254.68}$ $2^{235} / 2^{251}$	$2^{128}$ $2^{126.86} / 2^{124.79}$	$2^{111}$ $2^{112.87} / 2^{126.8}$	[Boura et al., 2014] This work

## Summary

---



## Conclusions

- 😊 **Expanded** the bit-wise CP model to ARX and AndRX designs
  - Integrated CP model for key recovery in AndRX designs.
- 😊 **Introduced** a novel model for direct and indirect contradictions.
- 😊 **Improved** attacks on several ciphers



## Future Works



- Apply CP models to other bit-oriented ciphers.
- Enhance optimization techniques for key recovery.



Questions?









-  Biham, E., Biryukov, A., and Shamir, A. (1999).  
**Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials.**  
In *EUROCRYPT 1999*, volume 1592 of *LNCS*, pages 12–23. Springer.
-  Biham, E. and Shamir, A. (1990).  
**Differential cryptanalysis of des-like cryptosystems.**  
In *Advances in Cryptology - CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer.



-  Boura, C., Lallemand, V., Naya-Plasencia, M., and Suder, V. (2018).  
**Making the impossible possible.**  
*Journal of Cryptology*, 31(1):101–133.
-  Boura, C., Naya-Plasencia, M., and Suder, V. (2014).  
**Scrutinizing and improving impossible differential attacks: applications to CLEFIA, Camellia, LBlock and Simon.**  
In *ASIACRYPT 2014*, volume 8873 of *LNCS*, pages 179–199. Springer.


-  Cui, T., Chen, S., Jia, K., Fu, K., and Wang, M. (2016).  
**New automatic search tool for impossible differentials and zero-correlation linear approximations.**  
IACR Cryptology ePrint Archive, Report 2016/689.
-  Derbez, P. and Fouque, P.-A. (2016).  
**Automatic search of meet-in-the-middle and impossible differential attacks.**  
In *CRYPTO 2016*, volume 9815 of *LNCS*, pages 157–184. Springer.

-  Hadipour, H., Gerhalter, S., Sadeghi, S., and Eichlseder, M. (2024).  
**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):234–325.
-  Hadipour, H., Sadeghi, S., and Eichlseder, M. (2023).  
**Finding the impossible: Automated search for full impossible-differential, zero-correlation, and integral attacks.**  
In *EUROCRYPT 2023*, volume 14007 of *LNCS*, pages 128–157. Springer.

-  Kim, J., Hong, S., Sung, J., Lee, C., and Lee, S. (2003).  
**Impossible differential cryptanalysis for block cipher structures.**  
In Johansson, T. and Maitra, S., editors, *INDOCRYPT 2003*, volume 2904 of *LNCS*, pages 82–96. Springer.
-  Lee, H., Kang, H., Hong, D., Sung, J., and Hong, S. (2016).  
**New impossible differential characteristic of SPECK64 using MILP.**  
*IACR Cryptol. ePrint Arch.*, page 1137.

-  Lu, J., Kim, J., Keller, N., and Dunkelman, O. (2008).  
**Improving the efficiency of impossible differential cryptanalysis of reduced Camellia and MISTY1.**  
In *CT-RSA 2008*, volume 4964 of *LNCS*, pages 370–386. Springer.
-  Ren, J. and Chen, S. (2019).  
**Cryptanalysis of reduced-round speck.**  
*IEEE Access*, 7:63045–63056.

-  Saberi, M., Bagheri, N., and Sadeghi, S. (2021).  
**Impossible differential and zero-correlation linear cryptanalysis of marx, marx2, chaskey and speck32.**  
*In 2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)*, pages 48–54.
-  Sadeghi, S. and Bagheri, N. (2018).  
**Improved zero-correlation and impossible differential cryptanalysis of reduced-round SIMECK block cipher.**  
*IET Inf. Secur.*, 12(4):314–325.

-  Sasaki, Y. and Todo, Y. (2017).  
**New impossible differential search tool from design and cryptanalysis aspects.**  
In *EUROCRYPT 2017*, pages 185–215, Cham. Springer International Publishing.
-  Zhang, W., Wu, W., and Feng, D. (2007).  
**New results on impossible differential cryptanalysis of reduced AES.**  
In Nam, K. and Rhee, G., editors, *Information Security and Cryptology - ICISC 2007, 10th International Conference, Seoul, Korea, November 29-30, 2007, Proceedings*, volume 4817 of *LNCS*, pages 239–250. Springer.



## Results on Simeck

Cipher	Dist.	#R	Time	Data	Mem.	Ref.
Simeck-32	11	20	$2^{61.11}$	$2^{32}$	$2^{51}$	[Zhang et al., 2007]
	11	20	<b><math>2^{55.79}</math></b>	<b><math>2^{29.79}</math></b>	<b><math>2^{50.79}</math></b>	<b>This work</b>
Simeck-48	$15^\dagger$	25	$2^{94.23}$	$2^{46}$	$2^{67}$	[Zhang et al., 2007]
	$15^\dagger$	25	<b><math>2^{93.05}</math></b>	$2^{47.05}$	$2^{68.12}$	<b>This work</b>
Simeck-64	$17^\dagger$	27	$2^{126.56}$	$2^{63}$	$2^{68}$	[Zhang et al., 2007]
	$17^\dagger$	27	<b><math>2^{126}</math></b>	$2^{63.47}$	$2^{68.45}$	<b>This work</b>

$\dagger$  : Distinguisher based on indirect contradiction.