The key recovery in differential attacks and the automated models

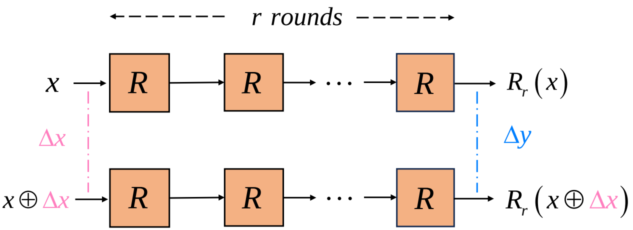
Ling Song

March 15, 2025

暨南大學 JINAN UNIVERSITY

Differential Attack

- Introduced by Biham and Shamir in 1990 [BS90].
- Find a differential $(\Delta x, \Delta y)$ of probability 2^{-p} covering a large number of rounds.
- $p < n$, where n is the block size.



- Variants:
  - Boomerang attack, rectangle attack, impossible differential attack, truncated differential attack, etc.

Variants of the Differential Attack

Boomerang distinguisher [Wag99]

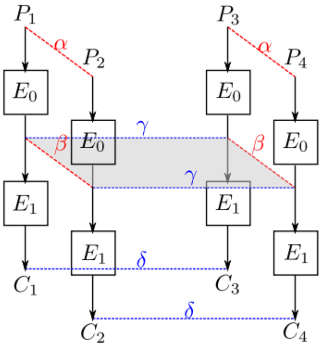
- Construct a long distinguisher using two short differentials of high probability.
- Non-random characteristic of quartets:

$$\Pr[E^{-1}(E(P) \oplus \delta) \oplus E^{-1}(E(P \oplus \alpha) \oplus \delta) = \alpha]$$

  is not negligible.
- Chosen plaintexts and chosen ciphertexts

Rectangle distinguisher [BDK01]

- Chosen-plaintext variant of the boomerang attack



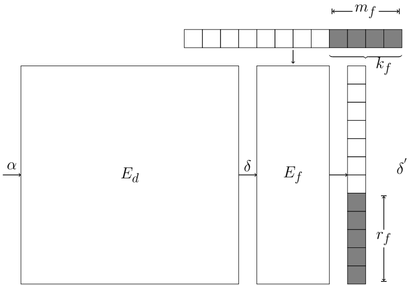
The Key Recovery Attack

- Differential distinguishers can be used to mount key recovery attacks.
- When evaluating a new block cipher using differential cryptanalysis
  - Search for distinguishers covering r rounds, where r is as large as possible.
    - A lot of work has been done.
  - Mount key recovery attacks on $r + x (x \geq 0)$ rounds on top of certain distinguishers.
    - Received much less attention.

A deep understanding of the key recovery attacks is necessary for an accurate security evaluation.

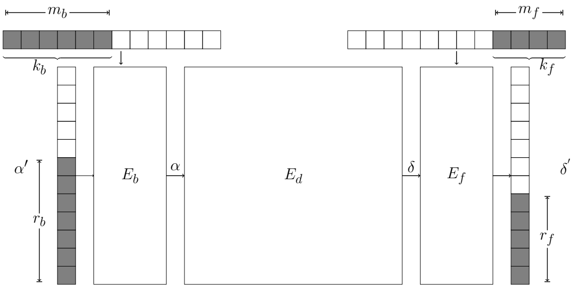
The Last-round Key Recovery Attack

- The distinguisher is of probability 2^{-p}.
- One-round E_f is appended.
- Guess k_f and decrypt one round to verify the output difference of the distinguisher.
  - ⋆ The right k_f will lead to the characteristic. The data complexity is $D = 2^{p+1}$.



Adding Rounds Before and After the Distinguisher

- Plaintext structure: a set of 2^{r_b} plaintexts
- Use 2^{p+1-r_b} structures, $D = 2^{p+1}$, and construct 2^{p+r_b} plaintext pairs.
- The number of pairs used for key recovery is $N = 2^{p+r_b+r_f-n}$.
  - No difference at $n - r_f$ ciphertext bits.



The Key Recovery Procedure

Extract key candidates

Goal

Determine the pairs for which an associated key exists that leads to the differential.

- Determine all (P, P', C, C', k_b, k_f), i.e., the (partial) key k_b, k_f can encrypts/decrypts the pair to the distinguisher.
- The right key is the candidate that has been suggested most often.

The Key Recovery Procedure

Extract key candidates

Goal

Determine the pairs for which an associated key exists that leads to the differential.

- Determine all (P, P', C, C', k_b, k_f), i.e., the (partial) key k_b, k_f can encrypts/decrypts the pair to the distinguisher.
- The right key is the candidate that has been suggested most often.

What is the time complexity of the procedure?

- lower bound: $N \cdot 2^{|k_b \cup k_f| - r_b - r_f}$ $(= \#(P, P', C, C', k_b, k_f))$.

The Key Recovery Procedure

Extract key candidates

Goal

Determine the pairs for which an associated key exists that leads to the differential.

- Determine all (P, P', C, C', k_b, k_f), i.e., the (partial) key k_b, k_f can encrypts/decrypts the pair to the distinguisher.
- The right key is the candidate that has been suggested most often.

What is the time complexity of the procedure?

- lower bound: $N \cdot 2^{|k_b \cup k_f| - r_b - r_f}$ $(= \#(P, P', C, C', k_b, k_f))$.

Parameters that affect the complexities:

- 2^{-p}, k_b, k_f, and r_b, r_f

slide from Boura's talk

Motivations

- Not much work on key recovery attacks and the optimality is not assured.
- The best distinguisher does not necessarily lead to the best key recovery attack.

Questions worth exploring

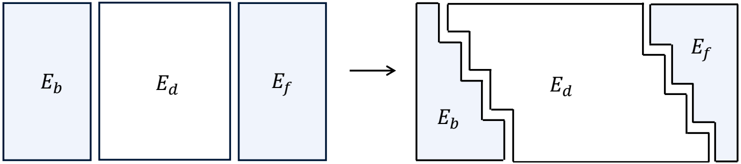
Q1: Can we propose generic key recovery algorithms for differential attacks that improve the efficiency?

Q2: Can we propose a search model that treats the distinguisher and the outer rounds as a whole?

Observation 1

Observation 1

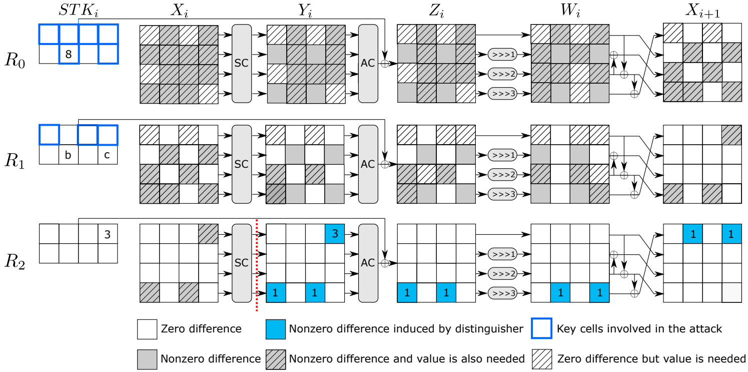
Distinguisher boundaries can be unaligned.



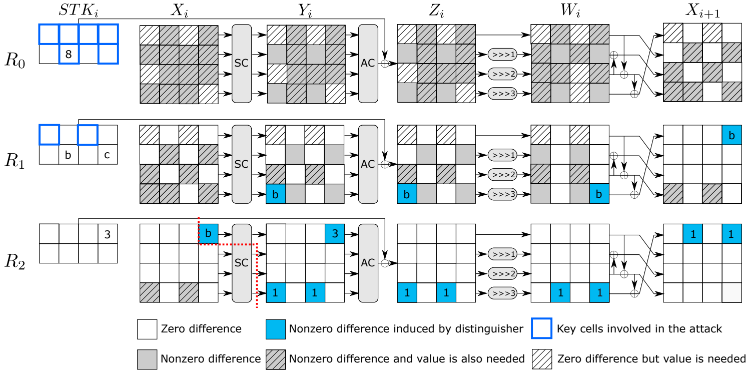
Remark: Allowing flexible boundaries expands the space of key recovery attacks.

Observation 1

Example 1: Skinny-64, aligned boundaries, $|k_b| = 9$ cells, $Pr = 2^{-p}$

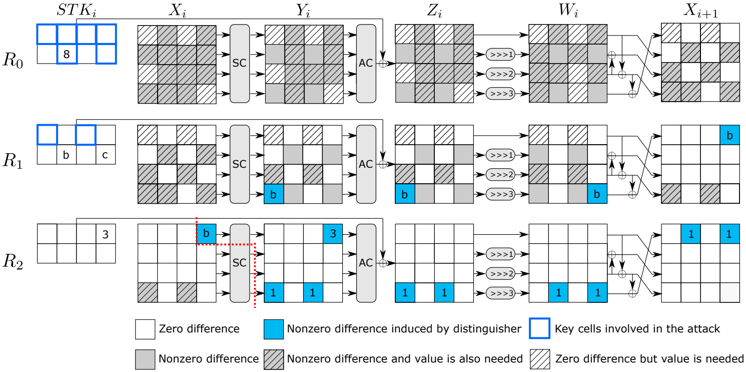


Example 2: Skinny-64, unaligned boundaries, $|k_b| = 8$ cells, $Pr = 2^{-p-2}$



Observation 1

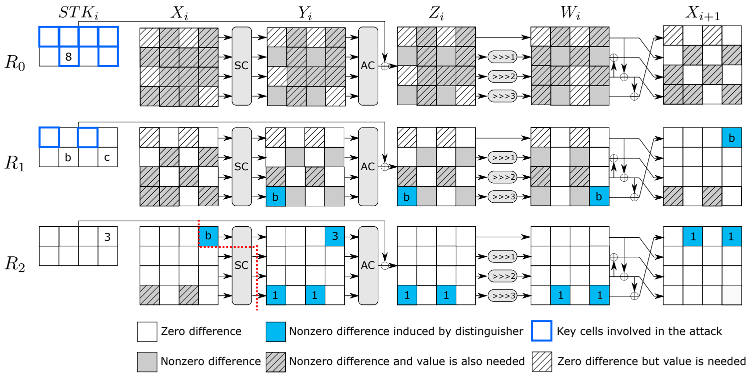
Example 2: Skinny-64, unaligned boundaries, $|k_b| = 8$ cells, $Pr = 2^{-p-2}$



A consequence of a larger space: improved complexities or covering more rounds

Observation 1

Example 2: Skinny-64, unaligned boundaries, $|k_b| = 8$ cells, $Pr = 2^{-p-2}$



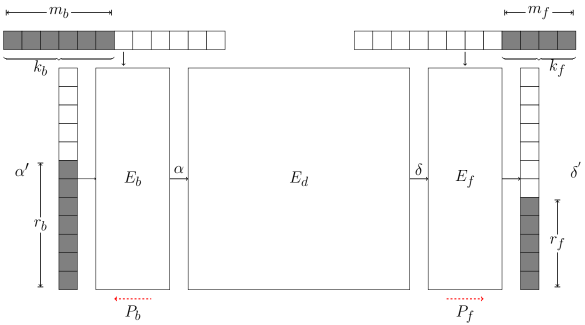
A consequence of a larger space: improved complexities or covering more rounds

Result: a new rectangle attack on Skinnye-64-256 v2, 37 rounds → 38 rounds

Observation 2

Observation 2

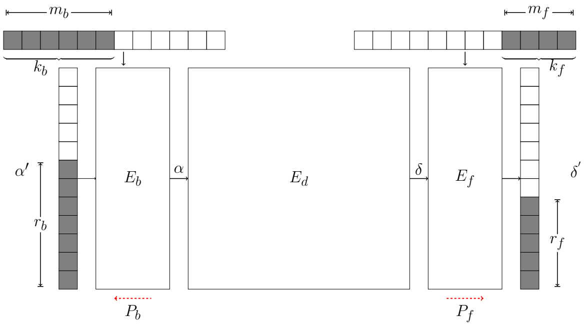
Instead of probability-1 extension, differences can propagate in the outer part with probability $< 1 \Rightarrow$ Probabilistic extension, *i.e.*, $P_b, P_f \leq 1$



Observation 2

Observation 2

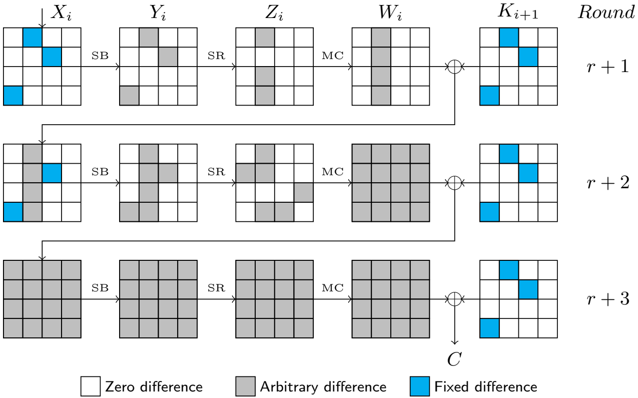
Instead of probability-1 extension, differences can propagate in the outer part with probability $< 1 \Rightarrow$ Probabilistic extension, *i.e.*, $P_b, P_f \leq 1$



Also enlarge the space of possible key recovery attacks. Benefits? Drawbacks?

Observation 2

Example 3: A toy example of classical differential attack in the related-key setting ($P_f = 1$). Suppose the distinguisher has a probability P_d.



Assume: The cipher uses AES round function, a 128-bit key with no key expansion.

Observation 2

Table: Precomputation hash tables for Example 3

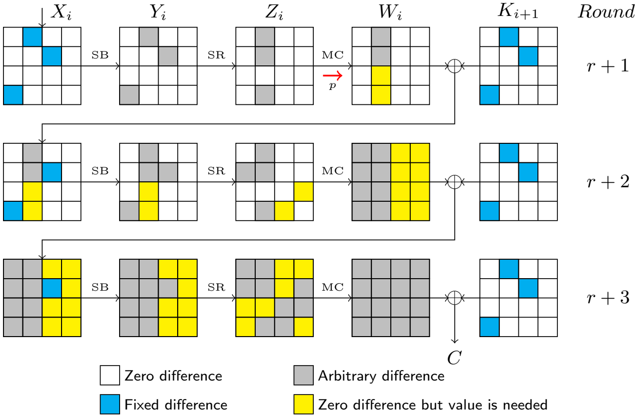
Tables	Involved key	Filters	Remaining pairs
1	$eqk[4,5,6,7]$	$\Delta Z_{r+2}[6]=0$	$2^{24}\cdot 2^{-1}\cdot D$
2	$eqk[3,9]$	$\Delta X_{r+2}[3,9]=\Delta K_{r+1}[3,9]$	$2^{24}\cdot 2^{-1}\cdot D$
3	$eqk[0,1,2]$	$\Delta Z_{r+2}[0,2,3]=0$	$2^{24}\cdot 2^{-1}\cdot D$
4	$eqk[8,10,11]$	$\Delta Z_{r+2}[8,9,10]=0$	$2^{24}\cdot 2^{-1}\cdot D$
5	$eqk[12,13,14,15]$	$\Delta Z_{r+2}[12,13,15]=\Delta Z_{r+1}[5]=0$ $\Delta X_{r+1}[3,4,9]$	$2^{-1}\cdot D$

$$\boldsymbol{D_{Example3}} = 2s\cdot P_d^{-1}$$

$$\boldsymbol{T_{Example3}} = 2^{24}\cdot s\cdot P_d^{-1}$$

Observation 2

Example 4: The toy example of differential attack in the related-key model with probabilistic extension ($P_f = 2^{-16}$)



Observation 2

Table: Precomputation hash tables for Example 4

Tables	Involved key	Filters	Remaining pairs
1	$eqk[9]$	$\Delta X_{r+3}[9] = \Delta K_{r+2}[9]$	$2^{-57} \cdot D$
2	$eqk[0,1,2,3]$	$\Delta Z_{r+2}[0,2,3] = 0$	$2^{-49} \cdot D$
3	$eqk[4,5,6,7]$	$\Delta Z_{r+2}[6] = \Delta Z_{r+1}[6] = 0$ $\Delta X_{r+2}[3,9] = \Delta K_{r+1}[3,9]$	$2^{-49} \cdot D$
4	$eqk[8, 10 \sim 15]$	$\Delta X_{r+1}[3,4,9]$	$2^{-17} \cdot D$

$\boldsymbol{D}_{\boldsymbol{Example3}} = 2s \cdot P_d^{-1}$

$\boldsymbol{T}_{\boldsymbol{Example3}} = 2^{24} \cdot s \cdot P_d^{-1}$

$\boldsymbol{D}_{\boldsymbol{Example4}} = 2s \cdot (P_d P_f)^{-1} = 2s \cdot P_d^{-1} \cdot 2^{16}$

$\boldsymbol{T}_{\boldsymbol{Example4}} = s \cdot P_d^{-1}$

Prbabilistic Extensions

- Benefits
  - Decrease the time complexity

    $T_{Example4}/T_{Example3} = s \cdot P_d^{-1}/2^{24} \cdot s \cdot P_d^{-1} = 2^{-24}$
  - Flexible boundaries

    No predefined boundaries between the inner part and outer part
  - Increase the number of filters and earlier usage.
- Drawbacks
  - Increase the data complexity (*not necessarily*)

    $Data_{Example4}/Data_{Example3} = 2s \cdot P_d^{-1} \cdot 2^{16}/2s \cdot P_d^{-1} = 2^{16}$

Observation 3

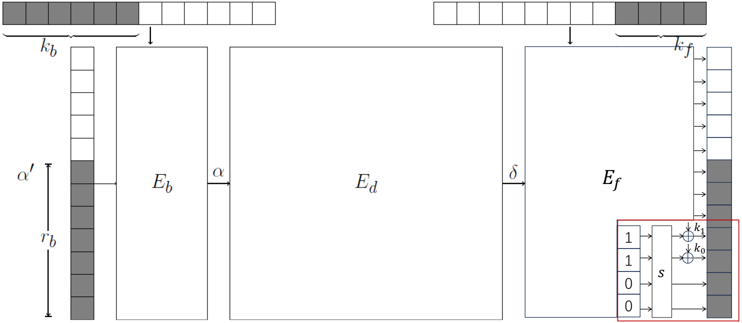
Observation 3

Pre-guessing some key bits before pairs are formed may be beneficial.

Observation 3

Observation 3

Pre-guessing some key bits before pairs are formed may be beneficial.



Observation 3

Comparison

A set of 2^{r_b} plaintexts

Forming pairs first There are $N = 2^{2r_b-1+r_f-n}$ pairs

- Guess k_0, k_1, and verify the input difference of the S-box.
  $T = 2^{2r_b-1+r_f-n+2}$, $N' = N \cdot 2^{2-4} = 2^{2r_b-1+r_f-n-2}$

Guessing key first The time for partial decryption $T_0 = 2^{r_b+2}$

- Forming pairs satisfying $n - r_f + 4$ bit conditions.
  $N = 2^{2r_b-1+r_f-n-2} = T_1$, $T = T_0 + T_1$
- Reduce the time complexity by a factor of 2^2

Observation 3

Comparison

A set of 2^{r_b} plaintexts

Forming pairs first There are $N = 2^{2r_b-1+r_f-n}$ pairs

- Guess k_0, k_1, and verify the input difference of the S-box.
  $T = 2^{2r_b-1+r_f-n+2}$, $N' = N \cdot 2^{2-4} = 2^{2r_b-1+r_f-n-2}$

Guessing key first The time for partial decryption $T_0 = 2^{r_b+2}$

- Forming pairs satisfying $n - r_f + 4$ bit conditions.
  $N = 2^{2r_b-1+r_f-n-2} = T_1$, $T = T_0 + T_1$
- Reduce the time complexity by a factor of 2^2

Guessing k_0, k_1 leads to a 4-bit filter in a differential attack, while the number of filters is doubled in a rectangle attack as there are two pairs in a quartet.

The Rectangle Key Recovery Attack

- The basic steps from data$\rightarrow$ pairs$\rightarrow$ quartets:

  1. Collect data; 2. construct pairs; 3. generate and process quartets; 4. exhaustive search.

Previous algorithms where gray parts stand for the pre-guessed key:

Alg. 1	$m_b + m_f$		Biham et al. at Eurocrypt'01 [BDK01]
Alg. 2	$m_b + m_f$		Biham et al. at FSE'02 [BDK02]
Alg. 3	m_b	m_f	Zhao et al. at DCC'20 [ZDM$^+$20]
Alg. 4	$m_b + m'_f$	$m_f - m'_f$	Dong et al. at Eurocrypt'22 [DQSW22]

The Rectangle Key Recovery Attack

The holistic key guessing strategy

With some key bits guessed in advance:

- Construct pairs on the plaintext side or ciphertext side?
  - $\Rightarrow$ On the side with more filters (discard useless pairs as early as possible)
- Which part of key bits are guessed in advance?
  - $\Rightarrow$ The part that leads to balanced compexities of the four steps (minimize the overall time complexity)
- How to find the key bits to be guessed?
  - $\Rightarrow$ Build an automated model to search for the best attacking parameters.

The Rectangle Key Recovery Attack

The holistic key guessing strategy

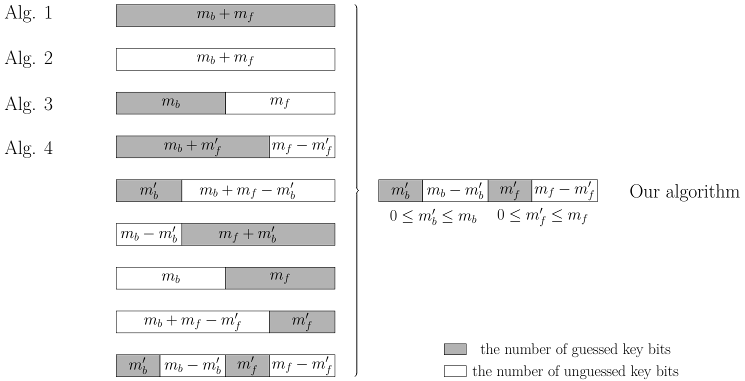
With some key bits guessed in advance:

- Construct pairs on the plaintext side or ciphertext side?
  - ⇒ On the side with more filters (discard useless pairs as early as possible)
- Which part of key bits are guessed in advance?
  - ⇒ The part that leads to balanced compexities of the four steps (minimize the overall time complexity)
- How to find the key bits to be guessed?
  - ⇒ Build an automated model to search for the best attacking parameters.

Answer to Q1: We propose a generic key recovery algorithm that supports any possible key guessing strategy for the rectangle attack [YSZ+24] and for the differential attack [SLY+24].

The Generic Key Recovery Algorithm for the Rectangle Attack

It allows to minimize the time complexity for a given distinguisher. It not only unifies four previous algorithms but also discovers five new ones.

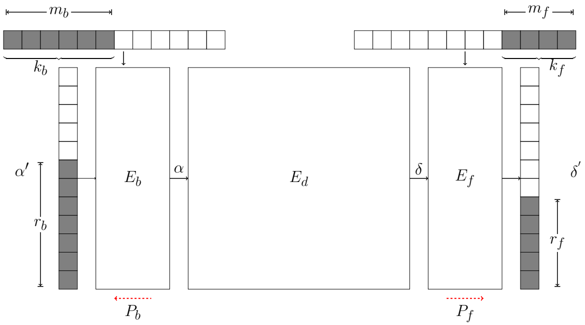


The Classical Key Recovery

Inner part Search for a distinguisher $\alpha \rightarrow \delta$ with a high probability P_d

Outer part Probability-1 extension and key recovery attacks, *i.e.*, $P_b = P_f = 1$.

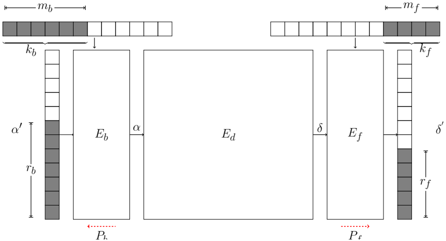
- ⋆ The inner and outer parts are often treated separately, but attempts in ID attacks to treat them together achieve remarkable results [HSE23].



One-step Model for Finding Efficient Key Recovery Attacks

Answer to Q2: Propose a search model that treats the inner and outer parts **as a whole** and searches for efficient attacks.

- Allow probabilistic extension in the outer rounds. The overall probability is $P = P_b P_d P_f \ (P_b, P_f \leq 1)$.
- The model determines the boundaries of the inner part.
- Determine the pre-guessing strategy automatically.
- Optimize the (data, time) complexities.



One-step Model for Finding Efficient Key Recovery Attacks

Core parts:

- Probabilistic extensions in the outer parts
- Determine the boundaries of the inner part automatically

Probabilistic Extensions

State labels:

- Inactive: $(x, y) = (0, 0)$ □
- Active with a fixed difference: $(x, y) = (1, 0)$ ■
- Active with an arbitrary difference: $(x, y) = (1, 1)$ ■

Probabilistic Extensions

State labels:

- Inactive: $(x, y) = (0, 0)$ □
- Active with a fixed difference: $(x, y) = (1, 0)$ ■
- Active with an arbitrary difference: $(x, y) = (1, 1)$ ■

$\boldsymbol{P_b, P_f}$**:**

- Non-linear layer (e.g., S-box), probabilistic extensions have two cases.

  case 1: ■ $\rightarrow$ ■. case 2: ■ $\rightarrow$ ■

  $\sum_i (O_i.x - O_i.y)$ to model P_f over S-boxes
- Linear layer (e.g., Mixcolumn)

  $$\begin{cases} T = 1 & \text{if } I_i.y = 1 \\ T = 0 & \text{if all } I_i.y = 0 \end{cases}$$

  $\sum_i (T - O_i.y)$ to model the truncated probability over the linear layer

Probabilistic Extensions

State labels:

- Inactive: $(x, y) = (0, 0)$ □
- Active with a fixed difference: $(x, y) = (1, 0)$ ■
- Active with an arbitrary difference: $(x, y) = (1, 1)$ ■

***P_b, P_f*:**

- Non-linear layer (e.g., S-box), probabilistic extensions have two cases.

  case 1: ■ $\rightarrow$ ■. case 2: ■ $\rightarrow$ ■

  $\sum_i (O_i.x - O_i.y)$ to model P_f over S-boxes

- Linear layer (e.g., Mixcolumn)

$$\begin{cases} T = 1 & \text{if } I_i.y = 1 \\ T = 0 & \text{if all } I_i.y = 0 \end{cases}$$

  $\sum_i (T - O_i.y)$ to model the truncated probability over the linear layer

Boundaries: the part where the value is needed for verifying the distinguisher is the outer part.

Applications

- AES, NIST standard
  - AES-192, rectangle attack, 12 $\rightarrow$ 13 rounds
  - AES-256, differential attack, 12 rounds, the time complexity $2^{206} \rightarrow 2^{144}$
  - Without probabilistic extensions, with pre-guessed keys.
- Deoxys-BC-384, ISO standard
  - Rectangle attack, 14 $\rightarrow$ 15 rounds
  - Narrowing the security margin to just 1 round
  - With probabilistic extensions, with pre-guessed keys.

Summary

Unified and generic key recovery algorithms

★ Support the holistic key guessing strategy

⇒ Cover four previous rectangle key recovery algorithms and unveil five new ones

Probabilistic extension and a one-step framework

★ Allow probabilistic differential propagation in the extended part

⇒ Overall considerations for the distinguisher and extended part

⇒ More flexible selection for attack parameters

⇒ Incorporating the unified key recovery algorithm

★ The new framework for automatically finding the best parameters for rectangle/differential attacks

↪ A series of improved results

Thanks for your attention!

References I

Eli Biham, Orr Dunkelman, and Nathan Keller, The rectangle attack—rectangling the Serpent, International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2001, pp. 340–357.

______, New results on boomerang and rectangle attacks, International Workshop on Fast Software Encryption, Springer, 2002, pp. 1–16.

Eli Biham and Adi Shamir, Differential cryptanalysis of DES-like cryptosystems, Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings (Alfred Menezes and Scott A. Vanstone, eds.), Lecture Notes in Computer Science, vol. 537, Springer, 1990, pp. 2–21.

Xiaoyang Dong, Lingyue Qin, Siwei Sun, and Xiaoyun Wang, Key guessing strategies for linear key-schedule algorithms in rectangle attacks, EUROCRYPT (2022).

Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder, Finding the impossible: Automated search for full impossible-differential, zero-correlation, and integral attacks, Advances in Cryptology – EUROCRYPT 2023 (Cham) (Carmit Hazay and Martijn Stam, eds.), Springer Nature Switzerland, 2023, pp. 128–157.

Ling Song, Huimin Liu, Qianqian Yang, Yincen Chen, Lei Hu, and Jian Weng, Generic differential key recovery attacks and beyond, Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part VII (Kai-Min Chung and Yu Sasaki, eds.), Lecture Notes in Computer Science, vol. 15490, Springer, 2024, pp. 361–391.

David A. Wagner, The boomerang attack, Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings (Lars R. Knudsen, ed.), Lecture Notes in Computer Science, vol. 1636, Springer, 1999, pp. 156–170.

Qianqian Yang, Ling Song, Nana Zhang, Danping Shi, Libo Wang, Jiahao Zhao, Lei Hu, and Jian Weng, Optimizing rectangle and boomerang attacks: A unified and generic framework for key recovery, Journal of Cryptology **37** (2024), no. 2, 1–62.

References II

Boxin Zhao, Xiaoyang Dong, Willi Meier, Keting Jia, and Gaoli Wang, Generalized related-key rectangle attacks on block ciphers with linear key schedule: applications to SKINNY and GIFT, Designs, Codes and Cryptography **88** (2020), no. 6, 1103–1126.