Probabilistic Extensions: A One-Step Framework for Finding Rectangle Attacks and Beyond

Ling Song, Qianqian Yang, **Yincen Chen**, Lei Hu, Jian Weng

EUROCRYPT 2024

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

暨南大學
JINAN UNIVERSITY

Outline

Outline

Preliminaries

Probabilistic Extensions

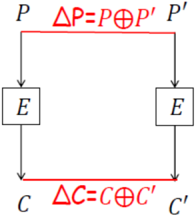
- Basic idea
- Framework for finding the best attack

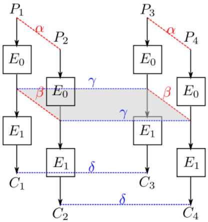
The Split-and-Bunch Technique

Comparison and Application

Summary

Preliminaries





Differential attack

- To exploit the non-random relation between input difference and output difference.

Boomerang attack

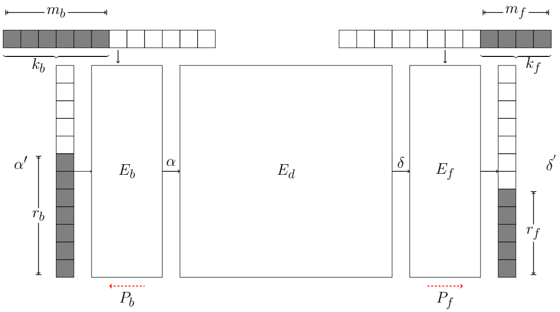
- To construct a long differential utilizing two short ones of high probability.

Rectangle attack (Chosen-plaintext variant of boomerang attack)

- More common for key recovery attacks.

Preliminaries

Outline and notations for classical rectangle key recovery attacks



- k_f': Part of k_f to be guessed;
- $m_f' = |k_f'|$;
- r_f': The condition can be verified under the guess of k_f' for a ciphertext;
- $m_f^* = m_f - m_f'$;
- $r_f^* = r_f - r_f'$

Preliminaries

Basic ideas and intuitions

- Classical rectangle attack

Inner part Search for a distinguisher with a high probability

Outer part Probability-1 extension and key recovery attacks

⋆ The inner and outer parts are treated **separately**

- Generalized rectangle attack

⋆ Treat the inner and outer parts **as a whole**

- A unified key recovery algorithm
- Take the minimum time complexity as the search target

Outline

Preliminaries

Probabilistic Extensions

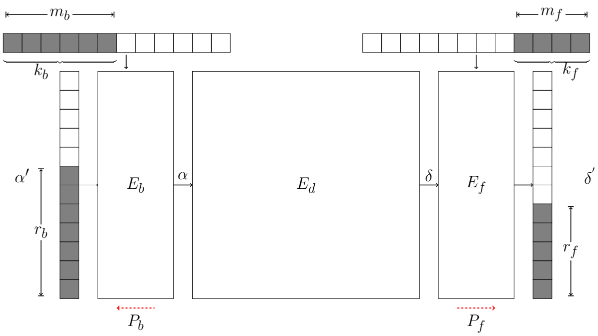
- Basic idea
- Framework for finding the best attack

The Split-and-Bunch Technique

Comparison and Application

Summary

Probabilistic Extensions



Question 1: Can the differential propagate in the outer part with probability $< 1 \Rightarrow$ Probabilistic extension?

- ▲ Benefits?
- ▼ Obstacles?

Probabilistic Extensions

Example 1: A toy example of classical differential attack in the related-key model ($P_f = 1$)

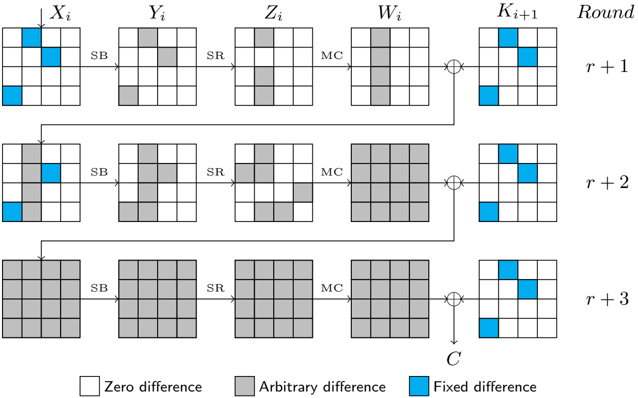


Table: Precomputation hash tables for Example 1

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	$eqk[12,13,14,15]$	$\Delta Z_{r+2}[12,13,15] = \Delta Z_{r+1}[5] = 0$ $\Delta X_{r+1}[3,4,9]$	$2^{-1} \cdot D$

$$\boldsymbol{D_{Example1}} = 2s \cdot P_d^{-1}$$
$$\boldsymbol{T_{Example1}} = 2^{24} \cdot s \cdot P_d^{-1}$$

Probabilistic Extensions

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

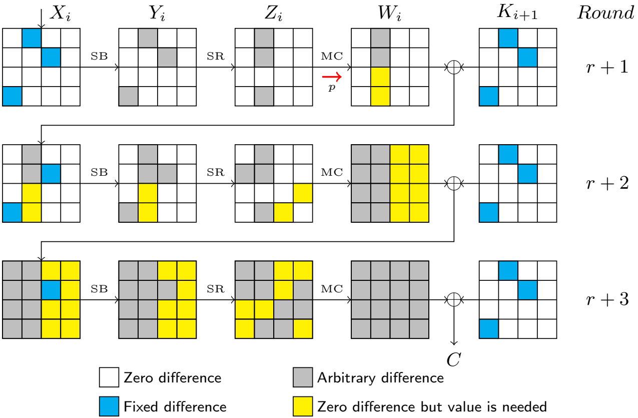


Table: Precomputation hash tables for Example 2

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	$eqk[8,10 \sim 15]$	$\Delta X_{r+1}[3,4,9]$	$2^{-17} \cdot D$

$$\boldsymbol{D_{Example2}} = 2s \cdot (P_d P_f)^{-1} = 2s \cdot P_d^{-1} \cdot 2^{16}$$
$$\boldsymbol{T_{Example2}} = s \cdot P_d^{-1}$$

Question 1: Can the differential propagate in the outer part with probability $< 1 \Rightarrow$ Probabilistic extension?

- ▲ Benefits
  - Decrease the time complexity
    $T_{Example2}/T_{Example1} = s \cdot P_d^{-1}/2^{24} \cdot s \cdot P_d^{-1} = 2^{-24}$
  - Flexible boundaries
    No predefined boundaries between the inner part and outer part
  - Increase the number of filters and earlier usage.
- ▼ Obstacles
  - Increase the data complexity (*not necessarily*)
    $Data_{Example2}/Data_{Example1} = 2s \cdot P_d^{-1} \cdot 2^{16}/2s \cdot P_d^{-1} = 2^{16}$

Question 2: How do we consider the inner part and outer part together and search for the optimal attack?

- The holistic probabilities ($P = P_b P_d P_f$)
- Boundaries where key recovery starts
- Combine with the unified key recovery algorithm [SZY+22]

- The **new** framework for rectangle attack

Data complexity:

- $y \cdot 2^{r_b} = \sqrt{s} 2^{n/2+1}/P$, where $P = P_b P_d P_f$

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 Extension

- The **new** framework for rectangle attack

Boundaries and P_b, P_f:

- Non-linear layer (eg. S-box)

  case 1: $\blacksquare \rightarrow \blacksquare$

  case 2: $\blacksquare \rightarrow \blacksquare$

  $\sum_i (O_i.x - O_i.y)$

- Linear layer (eg. Mixcolumn)

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

  $\sum_i (T - O_i.y)$

Guess-and-determine: guess the key and obtain filters.

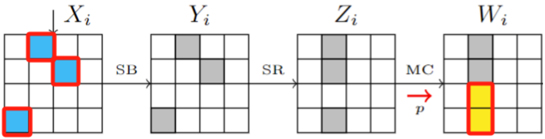
Constraints for the complexities: constraints for the data and memory complexities, and minimize the time complexity.

Outline

The Split-and-Bunch Technique

Table: Precomputation hash tables for Example 2

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	$\boldsymbol{eqk[8,10 \sim 15]}$	$\boldsymbol{\Delta X_{r+1}[3,4,9]}$	$\boldsymbol{2^{-17} \cdot D}$



Guess $eqk[8,10 \sim 15]$ to determine $\boldsymbol{W_{r+1}[6,7]} \overset{MC^{-1}}{\rightarrow} Z_i[4,5,7]$ $\overset{SB^{-1} \circ SR^{-1}}{\longrightarrow}$ determine $\boldsymbol{X_{r+1}[3,4,7]}$

$\Rightarrow$ From hash tables 3 to 4, the time complexity increases by $\mathbf{2^{32}}$

The Split-and-Bunch Technique

Question 3: Can filters be obtained with less consumption?

$\Rightarrow$ Does the 7-byte key $eqk[8, 10 \sim 15]$ have to be traversed?

$$\Downarrow$$

Example 3: Traverse $W_{r+1}[6, 7]$ instead of $eqk[8, 10 \sim 15]$

The Split-and-Bunch Technique

Observation

Traverse $W_{r+1}[6,7]$ instead of $eqk[8, 10 \sim 15]$

- The number of suggestions for the correct key is the same.
- For a wrong pair, the number of suggestions for the incorrect key is equal to expanding the number of pairs by a factor of 2^{16}.

Ensuring the correct key is not overlooked, and the split-and-bunch technique brings **advantages** to attack.

The Split-and-Bunch Technique

Table: Precomputation hash tables for Example 3

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	$\boldsymbol{W_{r+1}[6,7]}$	$\Delta X_{r+1}[3,4,9]$	$2^{-57} \cdot D$

$$\boldsymbol{D_{Example3}} = 2s \cdot P_d^{-1} \cdot 2^{16}$$
$$\boldsymbol{T_{Example3}} = 2^{-32} \cdot s \cdot P_d^{-1}$$

Advantage: $T_{Example3} / T_{Example2} = 2^{-32}$

Outline

Comparison and Application

More compatible

- ⋆ Our framework includes the unified key recovery algorithm.

More flexible

- ⋆ No predefined boundaries between the inner and outer parts.

Better attack effects

- ⋆ Allow probabilistic extension, set the overall time complexity as the objective function.
- ⋆ Previous rectangle attacks can be improved to some extent using our new idea and technique.

Comparison and Application

Table: Summary of the results

Cipher	Rounds	Data	Memory	Time	Approach	Setting	Ref.
Deoxys-BC-384	14	$2^{125.2}$	2^{140}	2^{260}	Rect.	RTK	[DQSW22]
	14	$\mathbf{2^{115.7}}$	2^{160}	$2^{260.59}$	Rect.	RTK	This work
	14	$2^{115.7}$	2^{128}	$2^{242.7}$	Rect.	RTK	This work
	15	$\mathbf{2^{115.7}}$	$\mathbf{2^{128}}$	$\mathbf{2^{371.7}}$	**Rect.**	**RTK**	**This work**
SKINNY-128-256	26	$2^{126.53}$	$2^{128.44}$	$2^{254.4}$	Rect.	RTK	[DQSW22]
	26	$2^{126.53}$	2^{136}	$2^{241.38}$	Rect.	RTK	[SZY+22]
	26	$2^{121.93}$	2^{136}	$2^{219.93}$	Rect.	RTK	This work
ForkSkinny-128-256	28	$2^{118.88}$	$2^{118.88}$	$2^{224.76}$	Rect.	RTK	[DQSW22]
	28	$2^{123.89}$	$2^{123.89}$	$2^{212.89}$	Rect.	RTK	This work
CRAFT	23	2^{74}	2^{51}	2^{94}	D	WK&ST	[LR22]
	26	2^{73}	2^{60}	2^{105}	D	WK&WT	[LR22]
	20	$2^{62.89}$	2^{49}	$2^{120.43}$	ZC	SK&ST	[HSE23]
	21	$2^{60.99}$	2^{100}	$2^{106.53}$	ID	SK&ST	[HSE23]
	19	$2^{60.99}$	2^{68}	$2^{94.59}$	D	SK&WT	[GSS+20]
	21	$2^{60.99}$	2^{92}	$2^{87.60}$	D	SK&WT	This work
	23	$\mathbf{2^{60.99}}$	$\mathbf{2^{120}}$	$\mathbf{2^{111.46}}$	**D**	**SK&WT**	**This work**

Outline

Summary

Probabilistic extension

★ Allow probabilistic differential propagation in the extended part

$\Rightarrow$ Overall considerations for the distinguisher and extended part

$\Rightarrow$ More flexible selection for attack parameters

$\Rightarrow$ Incorporating the unified key recovery algorithm

★ The new framework for automatically finding the best parameters for rectangle attack and beyond

Split-and-bunch technique

★ Compress intricate connections between key and state

$\Rightarrow$ Further reducing the time complexity of the attack

$\hookrightarrow$ A series of improved results

Thank you!

Q & A

References

- Xiaoyang Dong, Lingyue Qin, Siwei Sun, and Xiaoyun Wang, Key guessing strategies for linear key-schedule algorithms in rectangle attacks, EUROCRYPT (2022).
- Hao Guo, Siwei Sun, Danping Shi, Ling Sun, Yao Sun, Lei Hu, and Meiqin Wang, Differential attacks on CRAFT exploiting the involutory s-boxes and tweak additions, ToSC (2020).
- Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder, Finding the impossible: Automated search for full impossible-differential, zero-correlation, and integral attacks, EUROCRYPT (2023).
- Gregor Leander and Shahram Rasoolzadeh, Weak tweak-keys for the CRAFT block cipher, ToSC (2022).
- Ling Song, Nana Zhang, Qianqian Yang, Danping Shi, Jiahao Zhao, Lei Hu, and Jian Weng, Optimizing rectangle attacks: A unified and generic framework for key recovery, ASIACRYPT (2022).