# Parallel SAT Framework to Find Clustering of Differential Characteristics and Its Applications

Kosei Sakamoto[1,3] Ryoma Ito[2] Takanori Isobe[2,3]

[1] Mitsubishi Electric Corporation
[2] National Institute of Information and Communications Technology
[3] University of Hyogo

SAC 2023

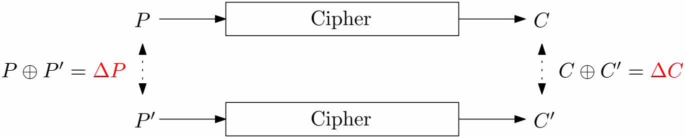
# Contents

# Differential Cryptanalysis

- Most popular attack to symmetric-key primitives
- Exploiting a pair of input and output differences with a high probability
  - Security: Prob($\Delta P \rightarrow \Delta C$)

    Ex) Prob($\Delta P \rightarrow \Delta C$) $> 2^{-64}$ on a 64-bit cipher

    $\rightarrow$ differential distinguisher




$\Delta P \rightarrow \Delta C$ with high Prob.

# Motivation & Background

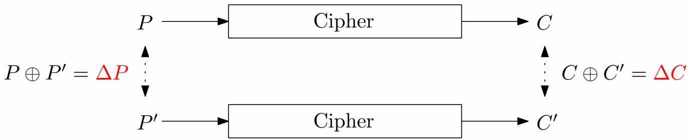
## ■ Differential Cryptanalysis

- ☐ Most popular attack to symmetric-key primitives
- ☐ Exploiting a pair of input and output differences with a high probability
  - ➢ Security: Prob($\Delta P \rightarrow \Delta C$)

    Ex) Prob($\Delta P \rightarrow \Delta C$) $> 2^{-64}$ on a 64-bit cipher

    $\rightarrow$ differential distinguisher

However, such a pair is hard to find...



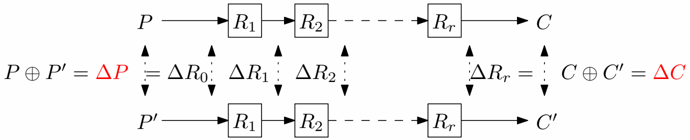

$\Delta P \rightarrow \Delta C$ with high Prob.

## Differential characteristic

- Sequence of differences over a cipher
  - Prob.($C$): Product of probabilities on each round

$$\text{Prob.}(C) = \prod_{i=1}^{r} \text{Prob.}(\Delta R_{i-1} \rightarrow \Delta R_i)$$

  - Weight: $-log_2(\text{Prob.}(C))$

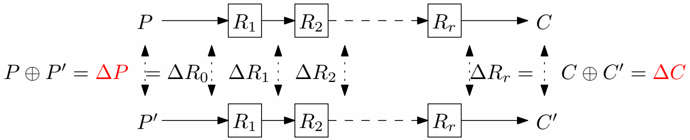



Differential characteristic
$C = (\Delta P \rightarrow \Delta R_1 \rightarrow \Delta R_2 \rightarrow \cdots \rightarrow \Delta C)$

# Differential characteristic

## Goal for designers

- Bounds Max(Prob.($C$)) below $2^{-b}$, $b$: size of block



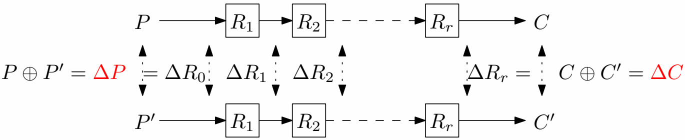

Differential characteristic
$C = (\Delta P \rightarrow \Delta R_1 \rightarrow \Delta R_2 \rightarrow \cdots \rightarrow \Delta C)$

# 1.2 Background



## Differential characteristic

- Goal for designers
  - Bounds Max(Prob.($C$)) below $2^{-b}$, $b$: size of block
- Goal for attackers
  - Finds a differential characteristic with as high a probability as possible
  - No need to consider internal differences ($\Delta R_1, \Delta R_2, \ldots, \Delta R_{r-1}$)

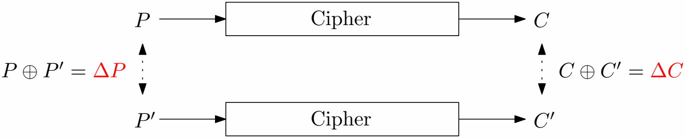



Differential characteristic
$$C = (\Delta P \to \Delta R_1 \to \Delta R_2 \to \cdots \to \Delta C)$$

# Motivation & Background

## Differential

- Pair of the input and output differences (No information about the internal differences)




Differential $(\Delta P, \Delta C)$

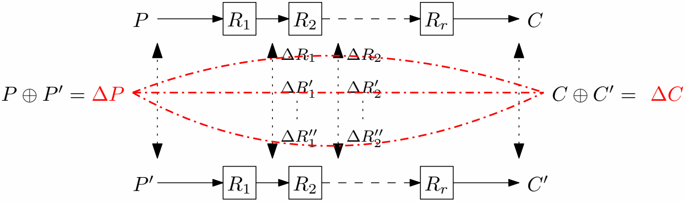
# 1.3 Motivation & Background



## Differential

- Pair of the input and output differences (No information about the internal differences)
- **Clustering effect**
  - We can see it as a bunch of differential characteristics
  - Prob.$((\Delta P, \Delta C))$: Sum of probabilities of all differential characteristics

**Prob.(differential) $>$ Prob.(differential characteristic)**




Differential $(\Delta P, \Delta C)$

# Motivation & Background



- Automatic search tools for differential characteristics
  - MILP/CP/SAT-based tools
  - SAT is the problem that checks if a given Boolean formula can turn TRUE or False
    1. Propagation of the differences in a cipher (given as clauses)
    2. Sum of all variables to express weight (given as clauses)
    - ➢ **Check existence of the propagation of differences under** $\Sigma_{i=0}^{r\cdot n-1} w_i \leq k$
    - ➢ If there is no such a propagation $\rightarrow$ increment $k$ and repeat this procedure

$$f_b = \underbrace{\underbrace{\overbrace{(x_0 \vee x_1 \vee \overline{x_2})}^{\text{Clauses}} \wedge (x_2 \vee x_3 \vee \overline{x_4}) \wedge (x_0 \vee x_2 \vee \overline{x_4})\wedge, \dots,}_{\text{Propagation of differences}} \underbrace{\wedge(x_7 \vee x_8 \vee \overline{x_9})}_{\Sigma_{i=0}^{r\cdot n-1} w_i \leq k}}_{\text{CNF}}$$

# Motivation & Background

## ■ Automatic search tools for differential

### □ SAT-based tools

1. Find the optimal differential characteristic
2. Fix the input and output differences
3. Search the differential characteristics under the fixed differences (clustering effect)
4. Calculate the probability

Useful for constructing a diferential with a high probability

# Motivation & Background

■ Automatic search tools for differential

□ SAT-based tools

1. Find the optimal differential characteristic
2. Fix the input and output differences
3. Search the differential characteristics under the fixed differences (clustering effect)
4. Calculate the probability

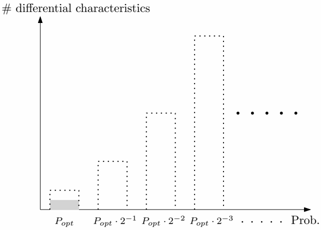
Useful for constructing a diferential with a high probability

**However,**
**It is just a straightforward extension of tools for differential characteristic**
**Not optimized for finding a differential with a high probability**

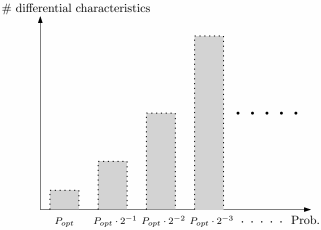
**We need to optimize these tools for differentials**

# Our approach

- We develop Sun et al's SAT-based tool [SWW21] to find a good differential
  - We optimize the evaluation of clustering effect for the multi-thread environment
    - This optimization is enable to evaluate the wide range of differential characteristics which are the seed of differentials




General approach



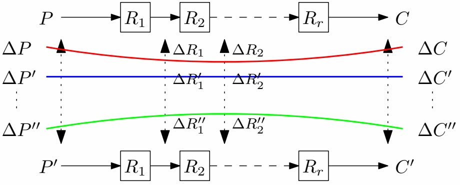

Our approach

# Concept of Our Framework

■ We evaluate clustering effect for multiple differential characteristics with a high probability

## Procedure of our framework

1. Find the differential characteristics with a high probability having the different input and output differences (not only optimal differential characteristics)




$(\Delta P, \Delta R_1, \Delta R_2, \ldots, \Delta C)$

$(\Delta P', \Delta R'_1, \Delta R'_2, \ldots, \Delta C')$

$\vdots$

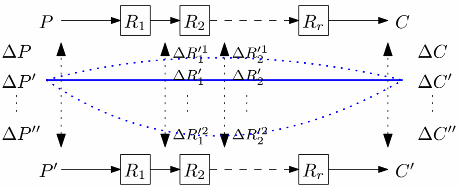
$(\Delta P'', \Delta R''_1, \Delta R''_2, \ldots, \Delta C'')$

# Concept of Our Framework

■ We evaluate clustering effect for multiple differential characteristics with a high probability

## Procedure of our framework

1. Find the differential characteristics with a high probability having the different input and output differences (not only optimal differential characteristics)
2. Evaluate clustering effect for the found differential characteristics




$(\Delta P', \Delta R_1', \Delta R_2', \ldots, \Delta C')$

$(\Delta P', \Delta R_1'^1, \Delta R_2'^1, \ldots, \Delta C')$

$\vdots$

$(\Delta P', \Delta R_1'^2, \Delta R_2'^2, \ldots, \Delta C')$

$\rightarrow (\Delta P', \Delta C')$

# Concept of Our Framework

■ We evaluate clustering effect for multiple differential characteristics with a high probability

## Procedure of our framework

1. Find the differential characteristics with a high probability having the different input and output differences (not only optimal differential characteristics)
2. Evaluate clustering effect for the found differential characteristics
3. Calculate probabilities for all differentials and find the highest one

$$(\Delta P, \Delta C) \quad \text{with Prob. } 2^{-p}$$

$$(\Delta P', \Delta C') \quad \boxed{\text{with Prob. } 2^{-p+2}}$$

$$\vdots \qquad\qquad \vdots$$

$$(\Delta P'', \Delta C'') \quad \text{with Prob. } 2^{-p+1}$$

# Concept of Our Framework

- We evaluate clustering effect for multiple differential characteristics with a high probability

## Procedure of our framework

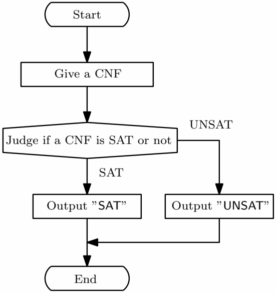
1. Find the differential characteristics with a high probability having the different input and output differences (not only optimal differential characteristics)
2. Evaluate clustering effect for the found differential characteristics
3. Calculate probabilities for all differentials and find the highest one

$$
\begin{array}{ll}
(\Delta P, \Delta C) & \text{with Prob. } 2^{-p} \\
(\Delta P', \Delta C') & \boxed{\text{with Prob. } 2^{-p+2}} \\
\vdots & \vdots \\
(\Delta P'', \Delta C'') & \text{with Prob. } 2^{-p+1}
\end{array}
$$

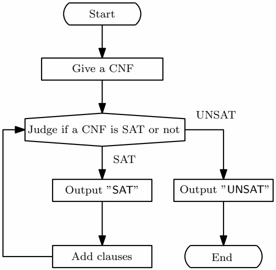
Conducting these evaluation is difficult due to a high computational cost

# Incremental SAT

- We fully leverage **an Incremental SAT**
  - Solving a general SAT multiple times with a small modification
    - Much more efficient than solving general SAT multiple times
  - Used to evaluate clustering effect in many works



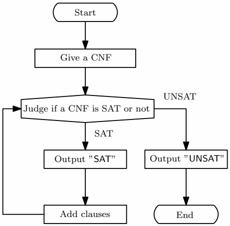

General SAT




Incremental SAT

# Incremental SAT

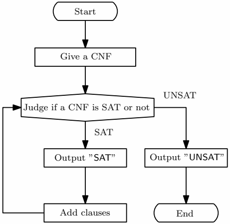
- We apply an incremental SAT to
  - Enumerate all differential characteristics with a certain weight having the different input and output differences
    - Adding a new clause to eliminate the same input and output differences whenever finding a differential characteristic




Incremental SAT

# Incremental SAT

- We apply an incremental SAT to
  - Enumerate all differential characteristics with a certain weight having the different input and output differences
    - Adding a new clause to eliminate the same input and output differences whenever finding a differential characteristic
  - Evaluate clustering effect
    - Adding a new clause to fix the input and output differences
    - Adding a new clause to eliminate the same internal propagation whenever finding a differential characteristics




Incremental SAT

# Optimizing for Multi-Thread Environment

**Question**

Solving a single incremental SAT on multi threads is really efficient?

- In case of a general SAT
  - Solving it on multi threads has a positive impact on runtime [EME22]
- In case of an incremental SAT
  - We evaluate runtime of several setting satisfying following equation:

$$P_{deg} = \frac{T_m}{T_s}$$

$P_{deg}$: Degree of parallelization to solve multiple incremental SAT

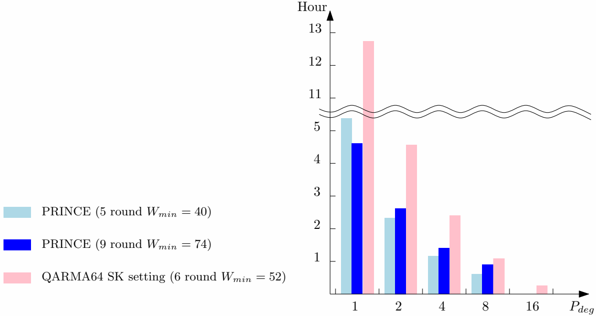
$T_m$: The total number of threads assigned for our evaluations

$T_s$: The number of threads assigned to solve a single incremental SAT

# Optimizing for Multi-Thread Environment

## Results on PRINCE and QARMA64

- PRINCE : $T_m = 8$, $(P_{deg}, T_s) = (1, 8), (2, 4), (4, 2), (8, 1)$
- QARMA64: $T_m = 16$, $(P_{deg}, T_s) = (1, 16), (2, 8), (4, 4), (8, 2), (16, 1)$




Solving multiple incremental SAT on each thread

>

Solving a single incremental SAT on multi threads

# Optimizing for Multi-Thread Environment



## Observations

- Increasing the degree of parallelization is greatly useful to improve runtime
- Assigning many threads to solve a single incremental SAT does not improve runtime
- In the same degree of parallelization, Assigning many threads to solve a single incremental SAT is worsen than assigning a single threads in terms of runtime
  - $T_m = 8, P_{deg} = 8, T_s = 1$ on the 6 round QARMA64: 35m15s
  - $T_m = 16, P_{deg} = 8, T_s = 2$ on the 6 round QARMA64: 1h6m4s

# Optimizing for Multi-Thread Environment

## Observations

- Increasing the degree of parallelization is greatly useful to improve runtime
- Assigning many threads to solve a single incremental SAT does not improve runtime
- In the same degree of parallelization, Assigning many threads to solve a single incremental SAT is worsen than assigning a single threads in terms of runtime
  - $T_m = 8, P_{deg} = 8, T_s = 1$ on the 6 round QARMA64: 35m15s
  - $T_m = 16, P_{deg} = 8, T_s = 2$ on the 6 round QARMA64: 1h6m4s

**Conclusion**

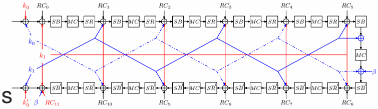
Assigning a single incremental SAT problem to each thread is more advantageous

We decide to assign an independent incremental SAT to each thread

# Application to PRINCE and QARMA

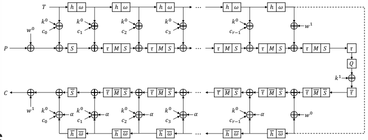
## ■ PRINCE

- ☐ 64-bit block cipher based on SPN
- ☐ Reflection construction for low-latency applications



## ■ QARMA

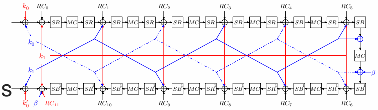
- ☐ There are two variant called QARMA64/128
- ☐ 64(128)-bit tweakable block cipher based on SPN
- ☐ Reflection construction for low-latency applications



# Application to PRINCE and QARMA

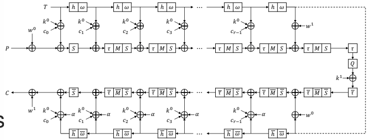
■ PRINCE

- 64-bit block cipher based on SPN
- Reflection construction for low-latency applications



■ QARMA

- There are two variant called QARMA64/128
- 64(128)-bit tweakable block cipher based on SPN
- Reflection construction for low-latency applications



## Why PRINCE and QARMA?

- Low-latency primitives tend to be weak at differential cryptanalysis
  - Mantis and SPEEDY are broken by differential cryptanalysis [BDBN22, DEKM16]
  - The best attack to PRINCE is differential cryptanalysis [CFG+14]
- To investigate the impact of the different design strategy in a linear layer on the behavior of differentials

# Results on PRINCE

- Distinguishing attack on 7 rounds
  - The known best one is on 6 rounds [CFG+14]

PRINCE

| Rounds | 4 (1+2+1) | | | | | 5 (1+2+2/2+2+1) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $W_{min}$ | **32** | 33 | 34 | 35 | 36 | **39** | 40 | 41 | 42 | 43 |
| Prob. | $2^{-30.868}$ | $2^{-31.861}$ | $2^{-32.587}$ | $2^{-33.333}$ | $2^{-32.979}$ | $2^{-38.810}$ | $2^{-39.385}$ | $2^{-40.017}$ | $2^{-40.607}$ | $2^{-40.837}$ |
| # differentials | 477452 | 3792944 | 4929816 | 5537848 | 5547896 | 576 | 12512 | 113840 | 598592 | 2231756 |
| Time | 6h06m57s | 48h48m43s | 47h34m17s | 47h35m06s | 48h01m15s | 1m21s | 26m09s | 4h08m26s | 23h14m24s | 48h03m32s |
| Rounds | 6 (2+2+2) | | | | | 7 (2+2+3/3+2+2) | | | | |
| $W_{min}$ | **44** | 45 | 46 | 47 | 48 | **56** | 57 | 58 | 59 | 60 |
| Prob. | $2^{-43.907}$ | $2^{-44.907}$ | $2^{-45.195}$ | $2^{-46.111}$ | $2^{-46.374}$ | $2^{-55.771}$ | $2^{-55.887}$ | $2^{-56.810}$ | $2^{-57.37}$ | $2^{-57.990}$ |
| # differentials | 64 | 512 | 1984 | 6592 | 25968 | 5632 | 100976 | 835456 | 205272 | 212280 |
| Time | 51s | 4m21s | 17m57s | 1h07m16s | 4h46m53s | 5h07m16s | 90h40m16s | 48h00m00s | 73h03m01s | 71h43m12s |
| Rounds | 8 (3+2+3) | | | | | 9 (3+2+4/4+2+3) | | | | |
| $W_{min}$ | **66** | 67 | 68 | 69 | 70 | **74** | 75 | 76 | 77 | 78 |
| Prob. | $2^{-64.389}$ | $2^{-65.384}$ | $2^{-66.303}$ | $2^{-66.970}$ | $2^{-67.075}$ | $2^{-73.888}$ | $2^{-74.881}$ | $2^{-74.970}$ | $2^{-75.970}$ | $2^{-76.166}$ |
| # differentials | 256 | 3584 | 46736 | 18352 | 24056 | 64 | 544 | 3400 | 26592 | 13968 |
| Time | 1h55m50s | 24h34m09s | 290h41m48s | 47h32m37s | 48h4m28s | 34m49s | 5h11m49s | 32h10m51s | 235h42m42s | 48h04m53s |

∗ Environment: Apple M1 MAX with 64 GB of main memory

# Results on QARMA64

MITSUBISHI ELECTRIC
*Changes for the Better*

- Distinguishing attack on 7 rounds (SK setting)
  - The known best one is on 6 rounds [YQC18] (SK setting)
- Distinguishing attack on 10 rounds (RT setting)
  - The known best one is on 9 rounds [ADG+19] (RT setting)

QARMA64 under the SK setting

| Rounds | 6 (2+2+2) | | | 7 (2+2+3/3+2+2) | | | 8 (3+2+3) | | |
|---|---|---|---|---|---|---|---|---|---|
| $W_{min}$ | **52** | 53 | 54 | **64** | 65 | 66 | **72** | 73 | 74 |
| Prob. | $2^{-45.741}$ | $2^{-46.019}$ | $2^{-46.112}$ | $2^{-60.278}$ | $2^{-60.111}$ | $2^{-58.921}$ | $2^{-64.845}$ | $2^{-64.503}$ | $2^{-64.693}$ |
| # differentials | 1024 | 18048 | 315360 | 512 | 16896 | 313280 | 400 | 21904 | 333776 |
| Time | 35m15s | 19h47m31s | 109h51m44s | 48m19s | 39h48m41s | 186h21m10s | 15h47m58s | 53h01m41s | 508h11m56s |

QARMA64 under the RT setting

| Rounds | 6 (2+2+2) | | | 7 (2+2+3/3+2+2) | | | 8 (3+2+3) | | |
|---|---|---|---|---|---|---|---|---|---|
| $W_{min}$ | **14** | 15 | 16 | **28** | 29 | 30 | **36** | 37 | 38 |
| Prob. | $2^{-14.000}$ | $2^{-14.913}$ | $2^{-15.193}$ | $2^{-27.541}$ | $2^{-28.000}$ | $2^{-28.286}$ | $2^{-36.000}$ | $2^{-36.679}$ | $2^{-36.679}$ |
| # differentials | 17 | 202 | 2571 | 84 | 3030 | 48840 | 20 | 840 | 18509 |
| Time | 36s | 1m44s | 13m33s | 5m35s | 1h15m24s | 15h28m20s | 11m16s | 30m22s | 10h18m25s |
| Rounds | 9 (3+2+4/4+2+3) | | | 10 (4+2+4) | | | 11 (4+2+5/5+2+4) | | |
| $W_{min}$ | **52** | 53 | 54 | **62** | 63 | 64 | **77** | 78 | 79 |
| Prob. | $2^{-51.415}$ | $2^{-51.415}$ | $2^{-52.246}$ | $2^{-60.831}$ | $2^{-60.831}$ | $2^{-60.831}$ | $2^{-77.000}$ | $2^{-77.415}$ | $2^{-77.509}$ |
| # differentials | 8 | 688 | 11290 | 273 | 4822 | 49585 | 64 | 7616 | 18424 |
| Time | 6h32m25s | 10h27m32s | 49h31m02s | 96h12m59s | 114h45m17s | 303h33m25s | 596h07m26s† | 1317h17m08s† | 1317h16m57s† |

∗ Environment: Intel Xeon Gold 6258R CPU (2.70 GHz) with 256 GB of main memory.

# Results on QARMA128

MITSUBISHI ELECTRIC
*Changes for the Better*

- Distinguishing attack on 10 rounds (SK setting)
  - The known best one is on 6 rounds [YQC18] (SK setting)
- Distinguishing attack on 12 rounds (RT setting)
  - The known best one is on 8 rounds [LHW19] (RT setting)

QARMA128 under the SK setting

| Rounds | 6 (2+2+2) | | | 7 (2+2+3/3+2+2) | | | 8 (2+2+4/4+2+2) | | |
|---|---|---|---|---|---|---|---|---|---|
| $W_{min}$ | **60** | 61 | 62 | **76** | 77 | 78 | **87** | 88 | 89 |
| Prob. | $2^{-54.494}$ | $2^{-54.521}$ | $2^{-54.581}$ | $2^{-71.930}$ | $2^{-72.321}$ | $2^{-72.614}$ | $2^{-84.850}$ | $2^{-85.093}$ | $2^{-85.539}$ |
| # differentials | 1312 | 98984 | 391352 | 516 | 32880 | 31960 | 16 | 708 | 14300 |
| Time | 15h27m17s | 499h19m12s | 1316h25m40s† | 40h57m50s | 530h05m58s | 430h44m47s | 57h59m37s | 92h7m23s | 693h25m04s |

| Rounds | 9 (3+2+4/4+2+3) | | | 10 (3+2+5/5+2+3) | | |
|---|---|---|---|---|---|---|
| $W_{min}$ | **106** | 107 | 108 | **125** | 126 | 127 |
| Prob. | $2^{-104.285}$ | $2^{-103.616}$ | $2^{-103.255}$ | $2^{-121.549}$ | $2^{-121.667}$ | $2^{-122.304}$ |
| # differentials | 240 | 561 | 1172 | 12 | 54 | 31 |
| Time | 249h25m14s† | 1004h00m44s† | 1004h00m32s† | 794h25m35s† | 794h25m23s† | 794h25m13s† |

QARMA128 under the RT setting

| Rounds | 7 (2+2+3/3+2+2) | | | 8 (3+2+3) | | | 9 (3+2+4/4+2+3) | | |
|---|---|---|---|---|---|---|---|---|---|
| $W_{min}$ | **28** | 29 | 30 | **42** | 43 | 44 | **64** | 65 | 66 |
| Prob. | $2^{-28.000}$ | $2^{-27.415}$ | $2^{-28.000}$ | $2^{-42.000}$ | $2^{-42.415}$ | $2^{-42.187}$ | $2^{-63.679}$ | $2^{-64.415}$ | $2^{-64.679}$ |
| # differentials | 32 | 2144 | 64368 | 64 | 5248 | 203200 | 1815 | 6870 | 26105 |
| Time | 38m43s | 4h51m52s | 48h32m23s | 21h17m20s | 52h32m19s | 470h54m17s | 1154h39m26s† | 1154h39m16s† | 1154h39m05s† |

| Rounds | 10 (4+2+4) | | | 11 (4+2+5/5+2+4) | | | 12 (5+2+5) | | |
|---|---|---|---|---|---|---|---|---|---|
| $W_{min}$ | **80** | 81 | 82 | **100** | 101 | 102 | **125** | 126 | 127 |
| Prob. | $2^{-78.005}$ | $2^{-79.005}$ | $2^{-78.408}$ | $2^{-96.466}$ | $2^{-97.929}$ | $2^{-96.521}$ | $2^{-120.024}$ | $2^{-123.499}$ | $2^{-124.084}$ |
| # differentials | 2 | 72 | 51 | 9 | 6 | 2 | 3 | 3 | 2 |
| Time | 978h51m03s† | 1316h34m33s† | 1316h33m53s† | 794h24m09s† | 794h23m59s† | 1036h39m39s† | 794h16m56s† | 1036h44m17s† | 1036h44m02s† |

∗ Environment: Intel Xeon Gold 6258R CPU (2.70 GHz) with 256 GB of main memory.

# Comparison with PRINCE and QARMA

- Gap of the probability between differential characteristics and differentials
  - PINRCE on 8 rounds
    - Optimal differential characteristic: $2^{-66}$
    - Best found Differential : $2^{-64.389}$
    - Gap: $2^{1.611}$
  - QARMA64 on 8 rounds (SK setting)
    - Optimal differential characteristic: $2^{-72}$
    - Best found Differential : $2^{-64.845}$
    - Gap: $2^{7.155}$

# Comparison with PRINCE and QARMA

■ Gap of the probability between differential characteristics and differentials

- PINRCE on 8 rounds
  - Optimal differential characteristic: $2^{-66}$
  - Best found Differential : $2^{-64.389}$
  - Gap: $2^{1.611}$
- QARMA64 on 8 rounds (SK setting)
  - Optimal differential characteristic: $2^{-72}$
  - Best found Differential : $2^{-64.845}$
  - Gap: $2^{7.155}$

Behavior of this gap is different between PRINCE and QARMA

**Question**

Where does this difference come from?

# Comparison with PRINCE and QARMA

## Non-linear layer (S-box)

- PRINCE
  - 4-bit S-box, MDP/ALB = $2^{-2}$, full diffusion property
- QARMA64
  - 4-bit S-box, MDP/ALB = $2^{-2}$, full diffusion property

# Comparison with PRINCE and QARMA

## Non-linear layer (S-box)

### PRINCE

- 4-bit S-box, MDP/ALB = $2^{-2}$, full diffusion property

### QARMA64

- 4-bit S-box, MDP/ALB = $2^{-2}$, full diffusion property

## **Linear layer** (matrix and permutation)

### PRINCE

- Designed to ensure 16 active S-boxes in consecutive four rounds

  Matrix: Constructed by 2 different $16 \times 16$ matrices

### QARMA64

- Designed based on an almost MDS matrix suitable for hardware implementation

  Matrix: Constructed by a single $16 \times 16$ matrix

# Comparison with PRINCE and QARMA

- Macro perspective
  - PRINCE
    - Round function can be viewed constructed by **2 different super S-boxes**
  - QARMA64
    - Round function can be viewed constructed by **a single super S-box**
- Micro perspective
  - PRINCE
    - Each output nibble in a matrix comes from **four input nibbles**
  - QARMA64
    - Each output nibble in a matrix comes from **three input nibbles**

We investigate the impact of these different properties

# Comparison with PRINCE and QARMA

- We change the matrix in PRINCE to:

$$M_{e1} = diag(\widehat{M}^{(0)}, \widehat{M}^{(0)}, \widehat{M}^{(0)}, \widehat{M}^{(0)})$$

$$M_{e2} = diag(circ(0, \rho^1, \rho^2, \rho^1), circ(0, 1, \rho^2, 1), circ(0, 1, \rho^2, 1), circ(0, \rho^1, \rho^2, \rho^1))$$

$$M_{e3} = diag(circ(0, \rho^1, \rho^2, \rho^1), circ(0, \rho^1, \rho^2, \rho^1), circ(0, \rho^1, \rho^2, \rho^1), circ(0, \rho^1, \rho^2, \rho^1))$$

- $M_{e1}$
  - Macro: Single super S-box
  - Micro: Output nibble in a matrix comes from four input nibbles
- $M_{e2}$
  - Macro: Two different super S-boxes
  - Micro: Output nibble in a matrix comes from three input nibbles
- $M_{e3}$
  - Macro: Single super S-box
  - Micro: Output nibble in a matrix comes from three input nibbles

# Comparison with PRINCE and QARMA

MITSUBISHI ELECTRIC
Changes for the Better

## Results on $M_{e1}$, $M_{e2}$, $M_{e3}$

- Original matrix and $M_{e1}$ has a good resistance against clustering effect
- Macro perspective is different but Macro perspective is same
  - **Output nibble in a matrix comes from four input nibbles**
  - Ankele and Kölbl reported that clustering effect easily happen in MIDORI and SKINNY [AK18]

PRINCE (6 (2+2+2) rounds) $T_w = 1$, $T_c = 10$

| Matrix | Original | $M_{e1}$ | $M_{e2}$ | $M_{e3}$ |
|---|---|---|---|---|
| $W_{min}$ | **44** | **40** | **44** | **42** |
| Prob. | $2^{-43.907}$ | $2^{-38.526}$ | $2^{-38.616}$ | $2^{-37.458}$ |
| Gap (Prob./$2^{-W_{min}}$) | $2^{0.093}$ | $2^{1.474}$ | $2^{5.384}$ | $2^{4.542}$ |
| # differentials | 64 | 256 | 8 | 272 |

PRINCE (6 (2+2+2) rounds) $T_w = 1$, $T_c = 10$

| Matrix \ Weight | | $W_{min}$ | $W_{min}+1$ | $W_{min}+2$ | $W_{min}+3$ | $W_{min}+4$ | $W_{min}+5$ | $W_{min}+6$ | $W_{min}+7$ | $W_{min}+8$ | $W_{min}+9$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| # DC† | Original | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| | $M_{e1}$ | 2 | 0 | 0 | 0 | 11 | 0 | 0 | 0 | 23 | 0 |
| | $M_{e2}$ | 1 | 2 | 7 | 16 | 55 | 116 | 452 | 848 | 2152 | 3498 |
| | $M_{e3}$ | 1 | 0 | 5 | 2 | 56 | 38 | 358 | 210 | 1719 | 1102 |

# Key Recovery

MITSUBISHI ELECTRIC
*Changes for the Better*

- Key-recovery attacks to QARMA
  - First key-recovery attack to QARMA by differential cryptanalysis

| Cipher (Setting†) | Attacked # Rounds | Type‡ | Outer whitening | Time | Data | Memory | Validity$ | Reference |
|---|---|---|---|---|---|---|---|---|
| QARMA-64 (SK) | 10 (3+2+5) | MITM | No | $2^{70.1}$ | $2^{53}$ | $2^{116}$ | ✓ | [ZD16] |
| | 10 (3+2+5) | ID | Yes | $2^{119.3}$ | $2^{61}$ | $2^{72}$ | × | [YQC18] |
| | 11 (3+2+6) | ID | Yes | $2^{120.4}$ | $2^{61}$ | $2^{116}$ | × | [YQC18] |
| QARMA-64 (RT) | 10 (2+2+6) | ID | Yes | $2^{125.8}$ | $2^{62}$ | $2^{37}$ | × | [ZD19] |
| | 10 (4+2+4) | TD | Yes | $2^{83.53}$ | $2^{47.06}$ | $2^{80}$ | × | Our |
| | 10 (3+2+5) | TD | Yes | $2^{75.13}$ | $2^{47.12}$ | $2^{72}$ | ✓ | Our |
| | 10 (3+2+5) | SS | Yes | $2^{59.0}$ | $2^{59.0}$ | $2^{29.6}$ | ✓ | [LHW19] |
| | 11 (4+2+5) | TD | Yes | $2^{111.16}$ | $2^{34.26}$ | $2^{108}$ | × | Our |
| | 11 (4+2+5) | ID | No | $2^{64.92}$ | $2^{58.38}$ | $2^{63.38}$ | ✓ | [LZG+20] |
| | 12 (3+2+7) | ZC/I | Yes | $2^{66.2}$ | $2^{48.4}$ | $2^{53.7}$ | ✓ | [ADG+19] |
| QARMA-128 (SK) | 10 (3+2+5) | MITM | No | $2^{141.7}$ | $2^{105}$ | $2^{232}$ | ✓ | [ZD16] |
| | 10 (3+2+5) | ID | Yes | $2^{237.3}$ | $2^{122}$ | $2^{144}$ | × | [YQC18] |
| | 11 (3+2+6) | ID | Yes | $2^{241.8}$ | $2^{122}$ | $2^{232}$ | × | [YQC18] |
| QARMA-128 (RT) | 11 (4+2+5) | TDIB | Yes | $2^{126.1}$ | $2^{126.1}$ | $2^{71}$ | ✓ | [LHW19] |
| | 11 (4+2+5) | ID | No | $2^{137.0}$ | $2^{111.38}$ | $2^{120.38}$ | ✓ | [LZG+20] |
| | 11 (7+2+2) | TD | Yes | $2^{104.60}$ | $2^{124.05}$ | $2^{48}$ | ✓ | Our |
| | 12 (7+2+3) | TD | Yes | $2^{154.53}$ | $2^{108.52}$ | $2^{144}$ | × | Our |
| | 12 (3+2+7) | MITM | Yes | $2^{156.06}$ | $2^{88}$ | $2^{154}$ | ✓ | [LZG+20] |
| | 13 (8+2+3) | TD | Yes | $2^{238.02}$ | $2^{106.63}$ | $2^{240}$ | × | Our |

‡ TD: Truncated Differential, MITM: Meet-in-the-Middle,

$ The designer claims that the multiplication of time and data complexities for QARMA-64 and QARMA-128 should be less than $2^{128-\epsilon}$ and $2^{256-\epsilon}$ for a small $\epsilon$ (e.g., $\epsilon = 2$), respectively. The symbol '✓' indicates that the attack is feasible within the designer's security claim and the symbol '×' indicates otherwise.

# Summary

- Design an efficient SAT-based tool for constructing good differentials
  - Develop Sun et al's SAT-based tool for constructing differentials optimized for the multi-thread environment
- Improve the distinguishing attack to PRINCE and QARMA
  - Find a new differential distinguishers
- Investigate the differential behavior on PRINCE and QARMA
  - Show the different design concept having the impact on the differential behavior
- Give the key-recovery attack to QARMA
  - Give the key-recovery attack to QARMA by differential cryptanalysis for the first time

# References I


- Ralph Ankele, Christoph Dobraunig, Jian Guo, Eran Lambooij, Gregor Leander, and Yosuke Todo, *Zero-correlation attacks on tweakable block ciphers with linear tweakey expansion*, IACR Trans. Symmetric Cryptol. **2019** (2019), no. 1, 192–235.
- Ralph Ankele and Stefan Kölbl, *Mind the gap - A closer look at the security of block ciphers against differential cryptanalysis*, SAC, Lecture Notes in Computer Science, vol. 11349, Springer, 2018, pp. 163–190.
- Christina Boura, Nicolas David, Rachelle Heim Boissier, and María Naya-Plasencia, *Better steady than speedy: Full break of SPEEDY-7-192*, IACR Cryptol. ePrint Arch. (2022), 1351.
- Anne Canteaut, Thomas Fuhr, Henri Gilbert, María Naya-Plasencia, and Jean-René Reinhard, *Multiple differential cryptanalysis of round-reduced PRINCE*, FSE, Lecture Notes in Computer Science, vol. 8540, Springer, 2014, pp. 591–610.

# References II


- Christoph Dobraunig, Maria Eichlseder, Daniel Kales, and Florian Mendel, *Practical key-recovery attack on MANTIS5*, IACR Trans. Symmetric Cryptol. **2016** (2016), no. 2, 248–260.
- Johannes Erlacher, Florian Mendel, and Maria Eichlseder, *Bounds for the security of ascon against differential and linear cryptanalysis*, IACR Trans. Symmetric Cryptol. **2022** (2022), no. 1, 64–87.
- Muzhou Li, Kai Hu, and Meiqin Wang, *Related-tweak statistical saturation cryptanalysis and its application on QARMA*, IACR Trans. Symmetric Cryptol. **2019** (2019), no. 1, 236–263.
- Ya Liu, Tiande Zang, Dawu Gu, Fengyu Zhao, Wei Li, and Zhiqiang Liu, *Improved cryptanalysis of reduced-version QARMA-64/128*, IEEE Access **8** (2020), 8361–8370.
- Ling Sun, Wei Wang, and Meiqin Wang, *Accelerating the search of differential and linear characteristics with the SAT method*, IACR Trans. Symmetric Cryptol. **2021** (2021), no. 1, 269–315.

# References III


- Dong Yang, Wen-Feng Qi, and Hua-Jin Chen, *Impossible differential attack on QARMA family of block ciphers*, IACR Cryptol. ePrint Arch. (2018), 334.
- Rui Zong and Xiaoyang Dong, *Meet-in-the-middle attack on QARMA block cipher*, IACR Cryptol. ePrint Arch. (2016), 1160.
- ______, *Milp-aided related-tweak/key impossible differential attack and its applications to qarma, joltik-bc*, IEEE Access **7** (2019), 153683–153693.