Analysis of BLAKE2

Jian Guo[†] **Pierre Karpman**[†⋆] Ivica Nikolić[†] Lei Wang[†]
Shuang Wu[†]

[†]Nanyang Technological University, Singapore
[⋆]École normale supérieure de Rennes, France

The Cryptographer's Track at the RSA Conference, San Francisco
2014–02–28

The BLAKE hash function family

- One of the five SHA-3 finalists
- Purely ARX round function inspired from ChaCha
- Local wide-pipe compression function in a HAIFA iteration mode
- Four digest sizes: BLAKE-224/256 & BLAKE-384/512
- Very fast in software
- Widely believed to be very secure

BLAKE specifications (compression function)

- Bijectively transforms a $4 \times 4 \times 32/64$-bit state with a $16 \times 32/64$-bit message
- (Uses four parallel applications of a 'G function')
- The output is compressed to form the chaining value
- Initial state:

$$\begin{pmatrix} v_0 & v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 & v_7 \\ v_8 & v_9 & v_{10} & v_{11} \\ v_{12} & v_{13} & v_{14} & v_{15} \end{pmatrix} \leftarrow \begin{pmatrix} h_0 & h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 & h_7 \\ s_0 \oplus c_0 & s_1 \oplus c_1 & s_2 \oplus c_2 & s_3 \oplus c_3 \\ t_0 \oplus c_4 & t_0 \oplus c_5 & t_1 \oplus c_6 & t_1 \oplus c_7 \end{pmatrix}$$

BLAKE specifications (compression function)

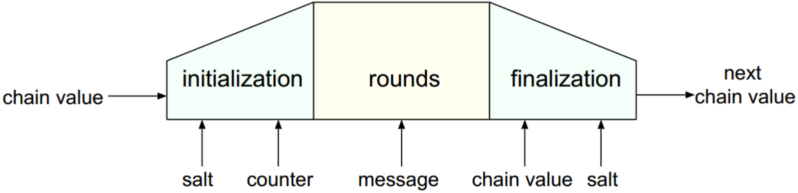


Figure : BLAKE compression function structure (Aumasson *& al.*, 2010)

BLAKE specifications (G function)

- Feistel-like function with four branches
- $\mathbf{G}_{i,j}(a, b, c, d)$ computes:

$1{:}a \leftarrow a + b + (m_i \oplus c_j)$	$5{:}a \leftarrow a + b + (m_j \oplus c_i)$
$2{:}d \leftarrow (d \oplus a) \ggg 32/16$	$6{:}d \leftarrow (d \oplus a) \ggg 16/8$
$3{:}c \leftarrow c + d$	$7{:}c \leftarrow c + d$
$4{:}b \leftarrow (b \oplus c) \ggg 25/12$	$8{:}b \leftarrow (b \oplus c) \ggg 11/7$

BLAKE specifications (G function)

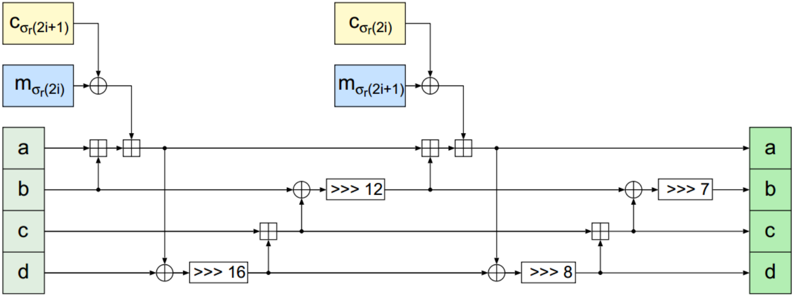


Figure : Diagram of the BLAKE-224/256 G function (Aumasson *& al.*, 2010)

BLAKE specifications (round structure)

- One round alternates a column & a diagonal step
- BLAKE-224/256 use 14 rounds; BLAKE-384/512 use 16

BLAKE specifications (round structure)

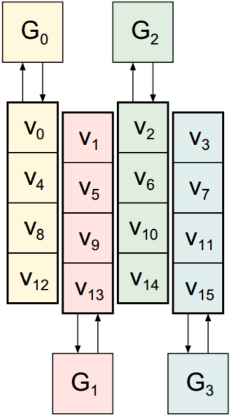


Figure : BLAKE column step (Aumasson *& al.*, 2010)

BLAKE specifications (round structure)

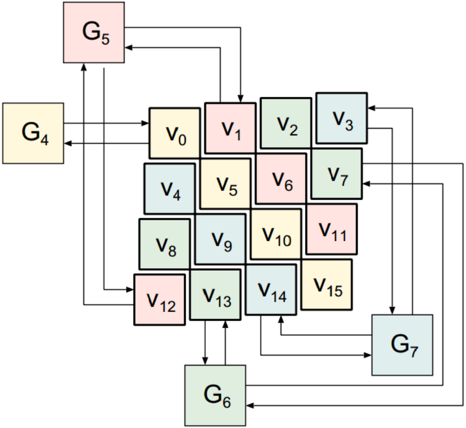


Figure : BLAKE diagonal step (Aumasson *& al.*, 2010)

BLAKE evolves into BLAKE2

- BLAKE2 is an even faster evolution of BLAKE (Aumasson & *al.*, ACNS 2013)
- Already popular
- Some changes made to the G function; initialisation; # of rounds
- No specific security analysis provided

BLAKE2 specifications (compression function)

- Initial state:

$$\begin{pmatrix} v_0 & v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 & v_7 \\ v_8 & v_9 & v_{10} & v_{11} \\ v_{12} & v_{13} & v_{14} & v_{15} \end{pmatrix} \leftarrow \begin{pmatrix} h_0 & h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 & h_7 \\ c_0 & c_1 & c_2 & c_3 \\ t_0 \oplus c_4 & t_1 \oplus c_5 & f_0 \oplus c_6 & f_1 \oplus c_7 \end{pmatrix}$$

- $\Longrightarrow$ Less freedom for the attacker (salt goes somewhere else)
- BLAKE2s uses 10 rounds; BLAKE2b uses 12

BLAKE2 specifications (G function)

- $\mathbf{G}_{i,j}(a,b,c,d)$ computes:

1: $a \leftarrow a + b + m_i$	5: $a \leftarrow a + b + m_j$
2: $d \leftarrow (d \oplus a) \ggg 32/16$	6: $d \leftarrow (d \oplus a) \ggg 16/8$
3: $c \leftarrow c + d$	7: $c \leftarrow c + d$
4: $b \leftarrow (b \oplus c) \ggg 24/12$	8: $b \leftarrow (b \oplus c) \ggg 63/7$

- Self-difference only in the message words
- 'Similar' rotations for BLAKE2s & BLAKE2b

Soooo.... what can we do?

Figure : Calvin & Hobbes (Watterson, 1985–1995)

Rotational distinguishers for the (keyed) permutation

- Introduced by (Khovratovich & Nikolić, FSE 2010)
- Distinguish a function F by $F(x) \lll r = F(x \lll r)$
- Exploits the absence of constants & 'small' number of '+' ops in **G**
- $\Pr[\mathbf{G}(a, b, c, d, m_i, m_j) \lll 1 = \mathbf{G}(a \lll 1, b \lll 1, c \lll 1, d \lll 1, m_i \lll 1, m_j \lll 1)] = 2^{6 \cdot (-1.4)}$ (th.) / $2^{-9.1}$ (exp.)
- $\Longrightarrow$ distinguish BLAKE2b's permutation in $\approx 2^{-876}$!!
- Not applicable to the compression/hash function

Fixed point partial collision for the compression function chosen IV

- Try to find a valid (iterative) differential pair for a fixed point of **G**
- $\Longrightarrow$ Iterates for free, for any # rounds
- ‽ Only 2^{64} trials available to find the pair

- Non-trivial fixed-points for **G** : $\approx 2^{64}$, each costs $\approx 2^{25}$ to find
- Search for differential characteristics unsuccessful
- Use rotationals again!
- Total cost of $\approx 2^{61}$ $\Rightarrow$ partial collisions on 304 chosen bits

Impossible differentials for all the BLAKE & BLAKE2

- New prob. 1 differential paths for BLAKE-224/256, BLAKE-384/512, BLAKE2s, BLAKE2b
- 0.5 + 2.5 forward path; 3.5 backward path
- $\Longrightarrow$ 6.5-round miss-in-the-middle ID for all (keyed) permutations
- Improves the best known results on BLAKE

Forward path (BLAKE-224/256 & BLAKE2s)

- Starts with a diff. in the MSB of m_{13} & v_2 @ round 3
- Non-trivial prob. 1 diff. @ round 5.5:

v_0: `????????????????????????????x---`

v_3: `????????????????????x-----------`

v_7: `???x---?????????????????????????`

v_{11}: `????????????????????????????x---`

v_{12}: `????x---????????????????????????`

v_{15}: `--------????????????????????x---`

Forward path (BLAKE-224/256 & BLAKE2s)

0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	14	10	4	8	9	15	13	6	1	12	0	2	11	7	5	3
2	11	8	12	0	5	2	15	13	10	14	3	6	7	1	9	4
3	7	9	3	1	13	12	11	14	2	6	5	10	4	0	15	8
4	9	0	5	7	2	4	10	15	14	1	11	12	6	8	3	13
5	2	12	6	10	0	11	8	3	4	13	7	5	15	14	1	9
6	12	5	1	15	14	13	4	10	0	7	6	3	9	2	8	11
7	13	11	7	14	12	1	3	9	5	0	15	4	8	6	2	10
8	6	15	14	9	11	3	0	8	12	2	13	7	1	4	10	5
9	10	2	8	4	7	6	1	5	15	11	9	14	3	12	13	0

Figure : Difference propagation in the forward path
(■ means no diff.; ■ means corrected diff.; ■ means controlled diff.)

Backward path (BLAKE-224/256 & BLAKE2s)

- Starts with @ the inverse of round 8 with:

  v_4 : `x---------------0-----------n---`

  v_9 : `---n-------x---x-------x-------x`

  v_{14} : `----n---n-------n1------n---0---`

  v_3 : `----n---n-------00------n-------`

- Non-trivial prob. 1 diff. @ round 5.5:

  v_0: `????????????????????????x-------`

  v_3: `--------????????????????x-------`

  v_7: `????????????????????????x-------`

  v_{12}: `????????????????????????x-------`

  v_{15}: `--------------------------------`

Backward path (BLAKE-224/256 & BLAKE2s)

0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	14	10	4	8	9	15	13	6	1	12	0	2	11	7	5	3
2	11	8	12	0	5	2	15	13	10	14	3	6	7	1	9	4
3	7	9	3	1	13	12	11	14	2	6	5	10	4	0	15	8
4	9	0	5	7	2	4	10	15	14	1	11	12	6	8	3	13
5	2	12	6	10	0	11	8	3	4	13	7	5	15	14	1	9
6	12	5	1	15	14	13	4	10	0	7	6	3	9	2	8	11
7	13	11	7	14	12	1	3	9	5	0	15	4	8	6	2	10
8	6	15	14	9	11	3	0	8	12	2	13	7	1	4	10	5
9	10	2	8	4	7	6	1	5	15	11	9	14	3	12	13	0

Figure : Difference propagation in the backward path
(□ means no diff.; □ means corrected diff.; □ means controlled diff.)

Impossible differentials : last details

- Contradiction between the paths in *e.g.*:

  v_{15}: `--------????????????????????x---` (forward)

  $\neq$

  v_{15}: `--------------------------------` (backward)

- One 0.5-round forward extension using (MSB, 0, MSB, MSB $\oplus$ MSB $\lll$ 64/32) $\rightarrow$ (MSB, 0, 0, 0)
- Similar paths for BLAKE-384/512 & BLAKE2b

Differential analysis

- Focus on yet unattacked models: compression & hash function of BLAKE2b
- Builds on previous analysis on BLAKE-256 (Guo & Matusiewicz, 2009), (Dunkelman & Khovratovich, 2011)
- The rotations on BLAKE2b are 'similar' to the ones of BLAKE-256 (all rotations div. by 8 or close to be, 3 out of 4 div. by 16 or close to be)
- BLAKE2b has a bigger state $\Longrightarrow$ lower probs. possible

Differential analysis (cont.)

- Automated search for rotation-friendly characteristics
- With diffs:
  - $\delta = \overline{04}$
  - $2 \times \delta = \overline{08}$
  - $3 \times \delta = \overline{0c}$
- $\Longrightarrow$ characteristic of prob. 2^{-344} on 3-round hash function / 2^{-367} on 4-round compression function
- And:
  - $\nabla = \overline{0004}$
  - $2 \times \nabla = \overline{0008}$
  - $3 \times \nabla = \overline{000c}$
- $\Longrightarrow$ characteristic of prob. 2^{-198} on 2-round hash function / 2^{-336} on 3-round compression function

Conclusion

- Building blocks of BLAKE2 quite more vulnerable than ones of BLAKE (rotational diffs., fixed points, etc.)
- Not so much a concern in practice
- The stronger initialisation makes attacks on the compression & hash function harder

Summary of results

Framework	Type	# Rounds	Complexity
BLAKE2s perm.	imp. diff.	6.5	—
	rotational	7	2^{511}
BLAKE2b perm.	imp. diff.	6.5	—
	rotational	**12**	2^{876}
	differential	5.5	2^{928}
BLAKE2s cf. ch. IV	collision	**10**	2^{64}
BLAKE2b cf. ch. IV	partial collision	**12**	2^{61}
	2^{64} weak preimages	**12**	1
BLAKE2b cf.	differential	4.5	2^{495}
BLAKE2b	differential	3.5	2^{480}

RSA CONFERENCE 2014

FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Share.
Learn.
Secure.

Capitalizing on Collective Intelligence

An Automated Evaluation Tool for Improved Rebound Attack: New Distinguishers and Proposals of ShiftBytes Parameters for Grøstl

SESSION ID: CRYP-F01

Yu Sasaki[1], Yuuki Tokushige[2], Lei Wang[3], Mitsugu Iwamoto[2] and Kazuo Ohta[2]

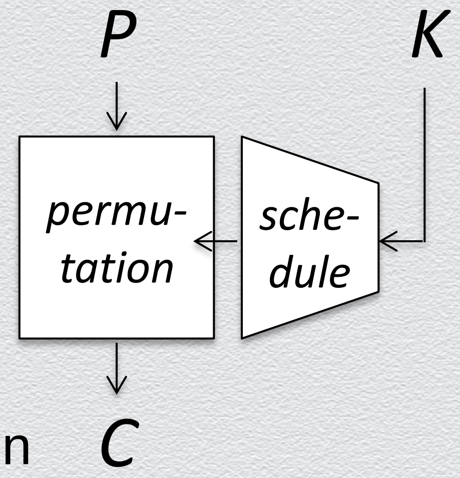
1. NTT Secure Platform Laboratories, Japan
2. University of Electro-Communications, Japan
3. Nanyang Technological University, Singapore

AES Based Design is Very Popular

- AES is one of the most successful designs
  - Special instruction in recent CPUs
  - Trustable security
  - Accumulated knowledge of implementation techniques
  - Accumulated knowledge of Side-Channel Attack countermeasures
- Many cryptographic primitives are designed based on AES even now

- The analysis on AES based primitives is important.

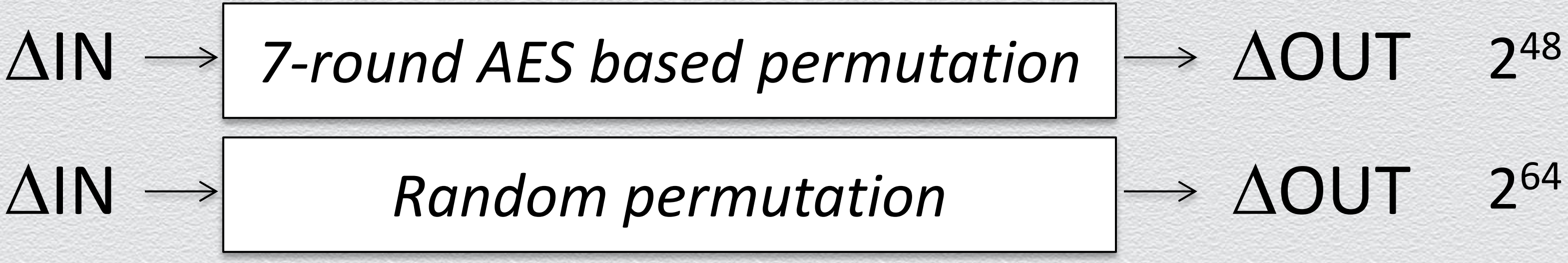
AES Permutation



- Two parts of the AES block-cipher
  - Key schedule
  - **Permutation** *Good design!!*
- Many primitives can be built by using AES permutation
  - **Hash function**
  - Stream cipher
  - Authenticated encryption
  - Even-Mansour based block-cipher

Rebound Attack

- Proposed by Mendel *et al.* at FSE 2009
- Particular differences ΔIN and ΔOUT are easily satisfied for the 7-round AES based permutation



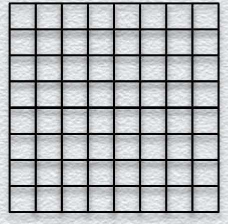
- Extended to 8 rounds by Gilbert and Peyrin at FSE 2010.
- Extension to 9 rounds was an open problem for a while.

Improved Rebound Attack

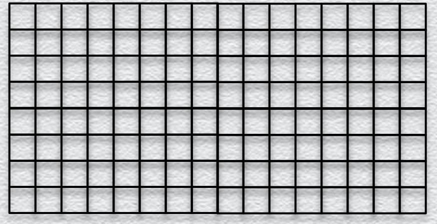
- Finally, extended to 9-rounds by Jean *et al.* at FSE 2012.
  - Simple if internal state is square
  - Complicated if internal state is rectangle
- Attack validity can be confirmed
- Attack optimality cannot be confirmed
- Those lead to the following three issues.

Internal state of AES-based permutation

Whirlpool
8 × 8



Grøstl-512
8 × 16



Issues to Discuss

- Optimality of the previous attack on Grøstl-512
- Applications to other AES-based primitives
- “ShiftRows” relate to the attack efficiency. Is there any other ShiftRows that can resist the improved rebound attack?

Our Approach

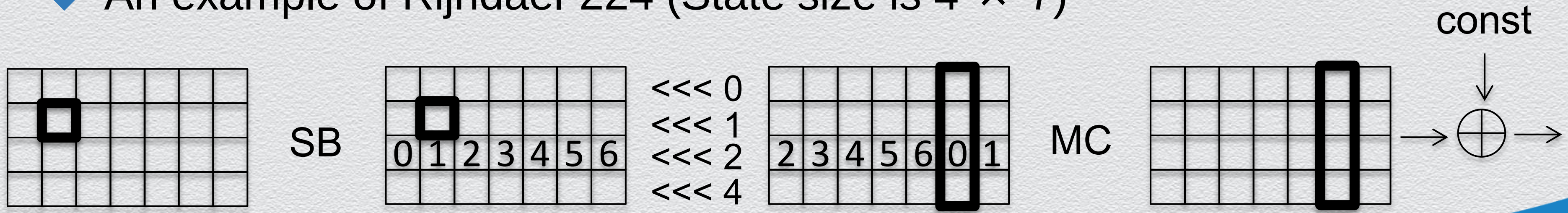
- Solve the three issues by developing an automated evaluation tool.
- **Input:** Internal state size and ShiftRows parameter
- **Output:** Optimized procedure and its complexity

- Results
  - The first 9-round distinguisher on Rijndael-192 and Rijndael-224
  - Show the optimality of the previous distinguisher on Grøstl-512.
  - Propose new stronger ShiftRows for Grøstl-512.

Technical Details: How to find an optimal attack?

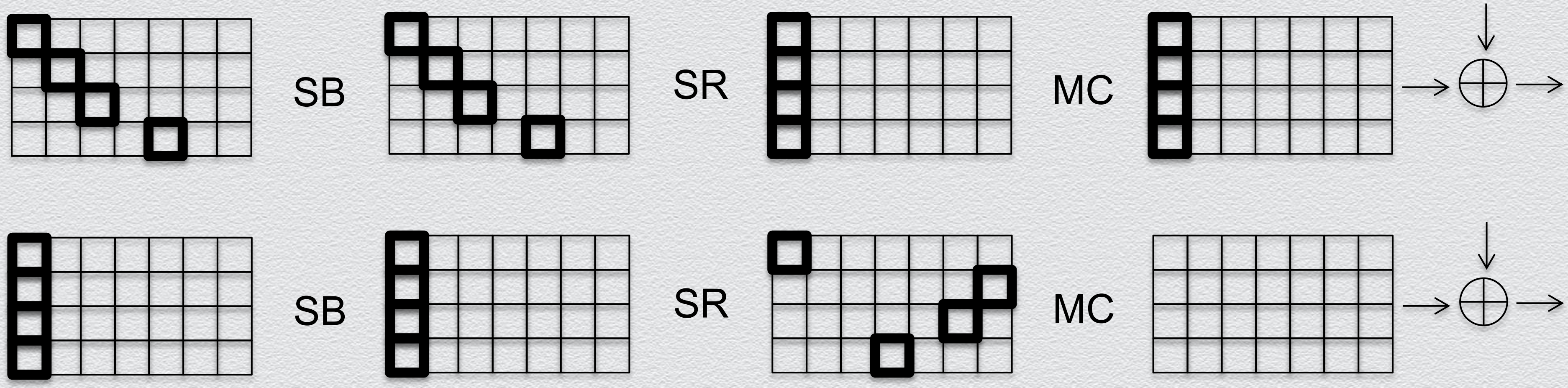
Specification of AES-based Permutation

- Iteration of the following four operations:
  - SubBytes (word-wise S-box application)
  - ShiftRows (row-wise word-positions rotation)
  - MixColumns (Column-wise diffusion by applying an MDS matrix)
  - AddConst (word-wise XOR with constant)
- An example of Rijndael-224 (State size is 4 × 7)



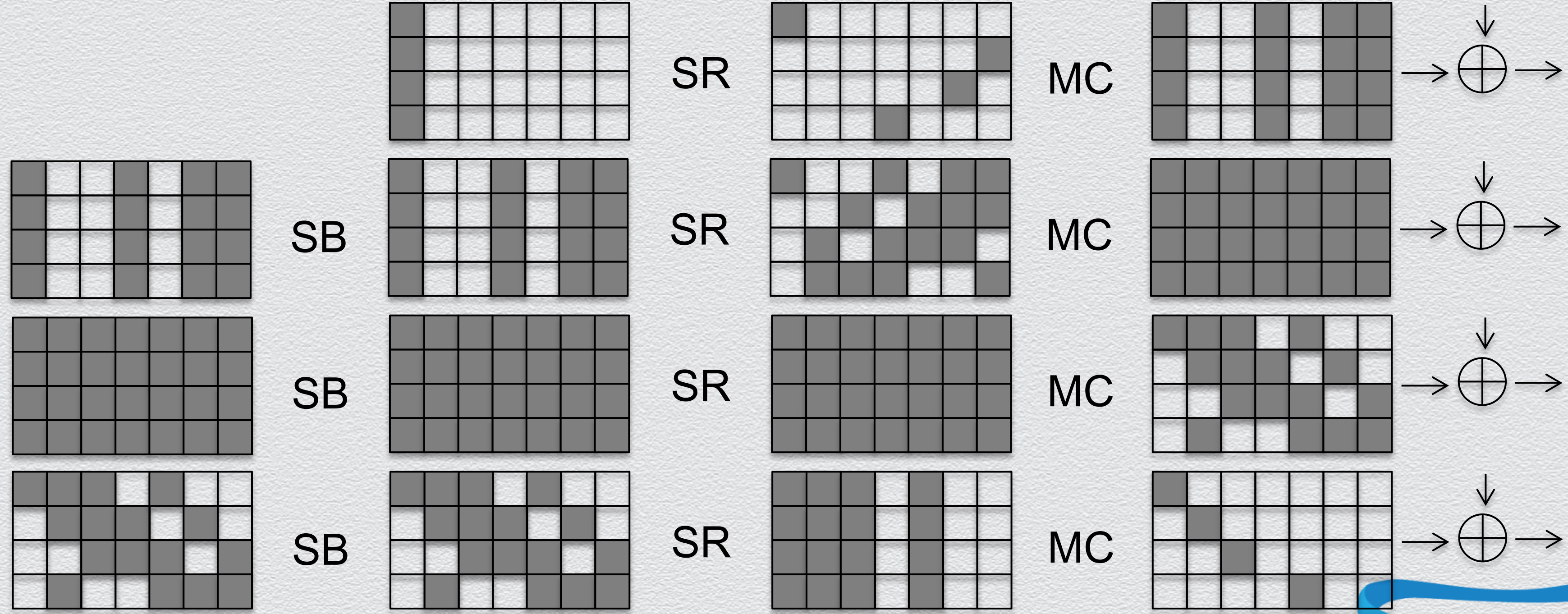
Super-Sbox Technique

- 1 AES-round + SB + SR can be computed column-wise, and can be regarded as big S-boxes.



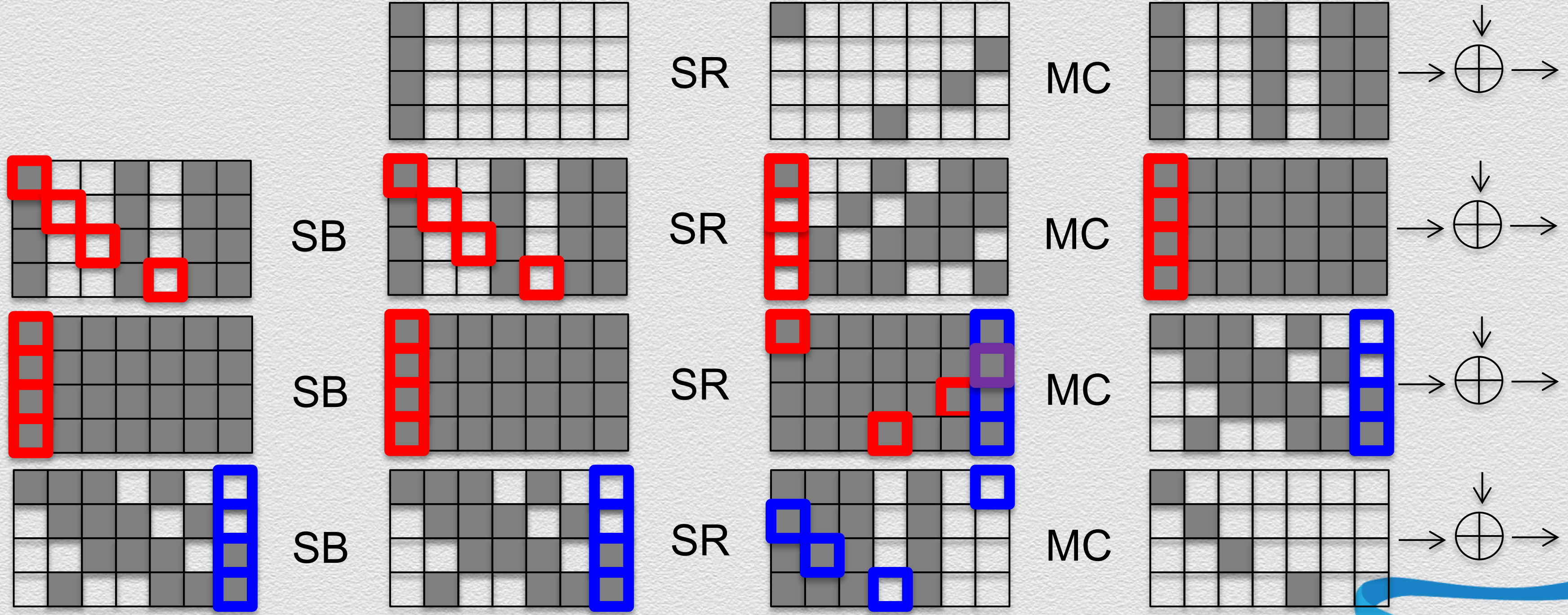
Core of Improved Rebound Attack

- Find a pair of values to satisfy the following truncated difference.



Super-Sbox Matching

- Construct Super-Sboxes in two directions, and find a match in middle.
- What is the best Super-Sboxes order to efficiently find a match?



Overall Framework

Need to detect the best analysis order of Super-Sboxes.

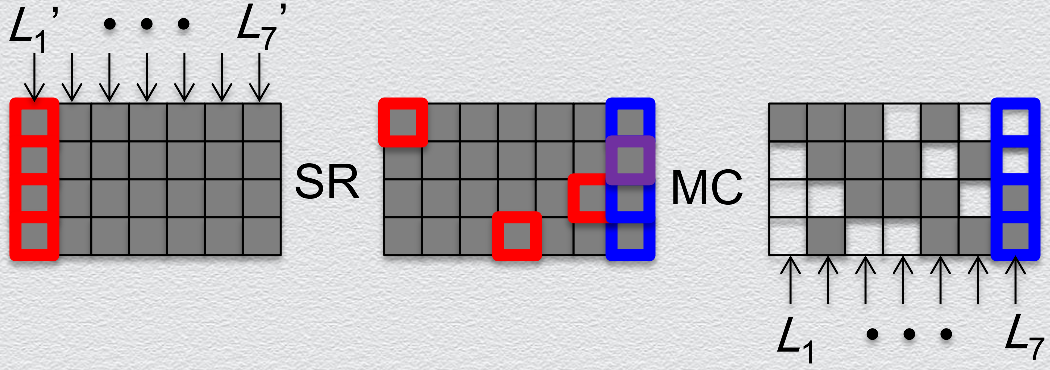
1. Find which Super-Sboxes interact each other.

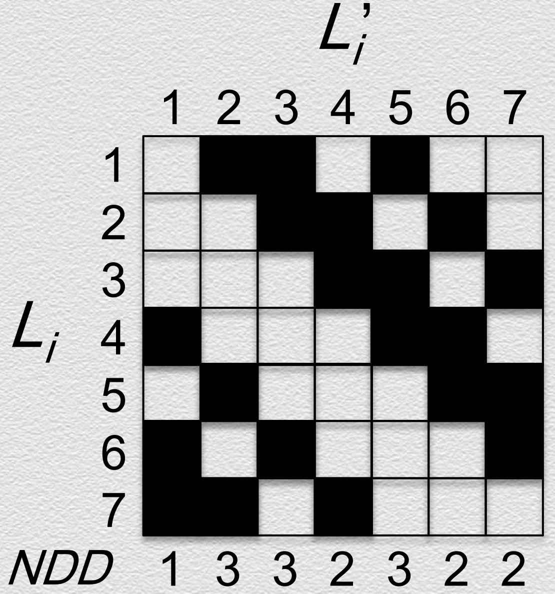
   Intersection Table Generation

2. Try all possible orders of Super-Sboxes. For each order, find the attack comlexity.

   Guess-and-Determine

Intersection Table Generation





- L_1' interacts with L_1, L_4, L_6, L_7 → empty
- L_1' does not interact with L_2, L_3, L_5 → black
- Each L_i can take limited number of differences. (after the MC operation)

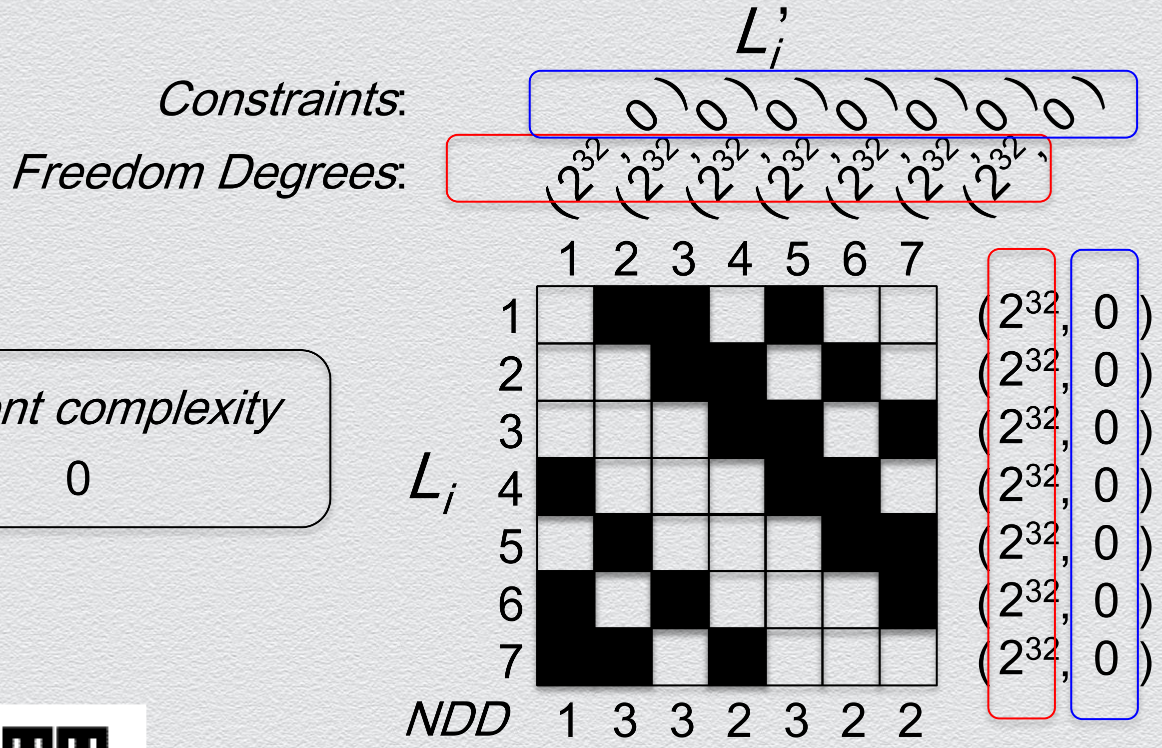
  A possible number of differences is given in *NDD*.

Guess-and-Determine

- Guess phase:
  - Exhaustively guess the value and diff of a target Super-Sbox.
  - The guessed value and diff become constraints to other Super-Sboxes.
- Determine phase:
  - For the increased constrains, reduce the freedom degrees of all Super-Sboxes.
  - The determine phase is iterated until no information is updated.

Demonstration for Rijndael-224

- Each Super-Sbox has 2^{32} choices. Constraints are initialized to 0.

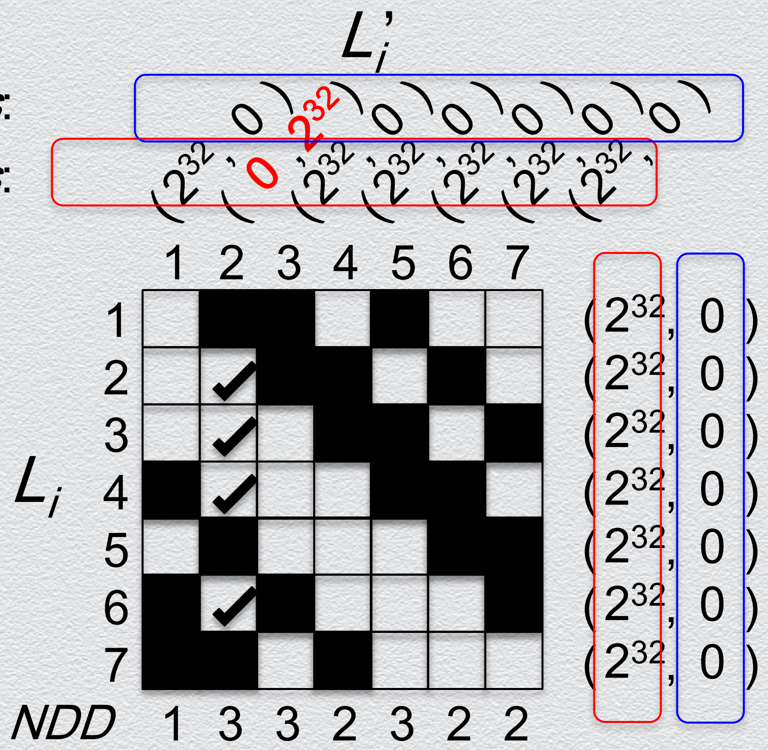


Current complexity
0

NTT

Demonstration for Rijndael-224

- 1^{st} guess : Choose the value and difference of L_2'.

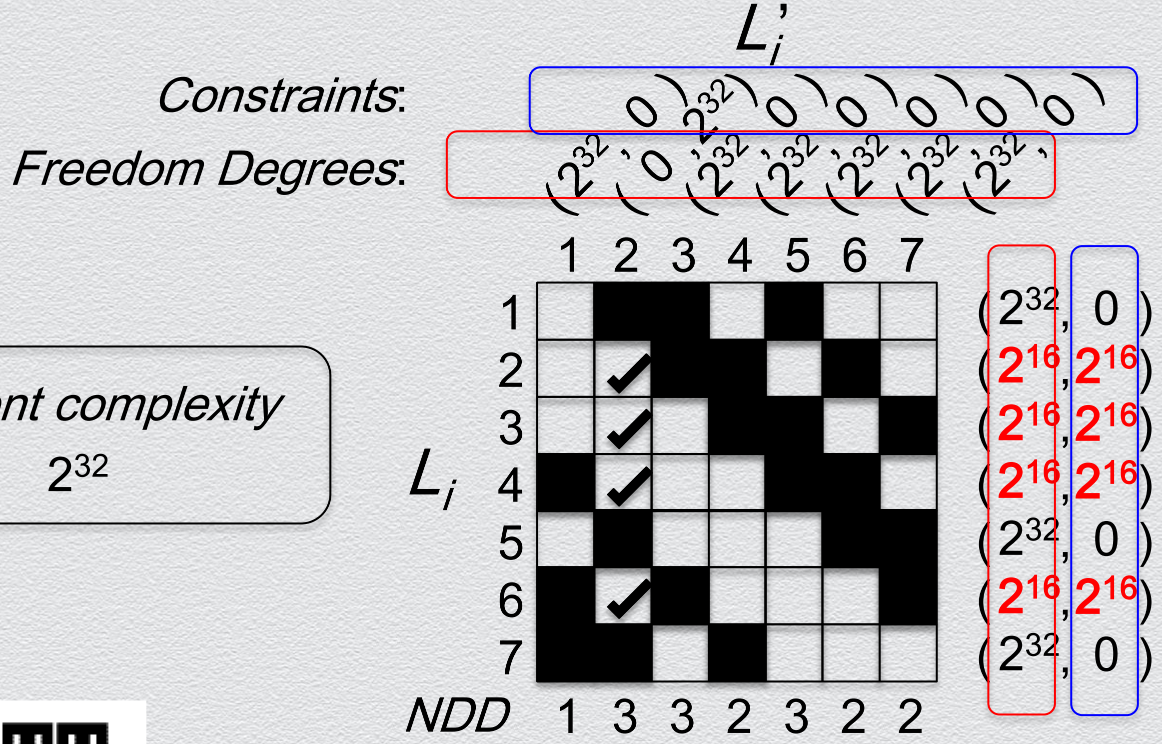


Current complexity

2^{32}

Demonstration for Rijndael-224

- 1^{st} determine: Update constrains for other Super-Sboxes.

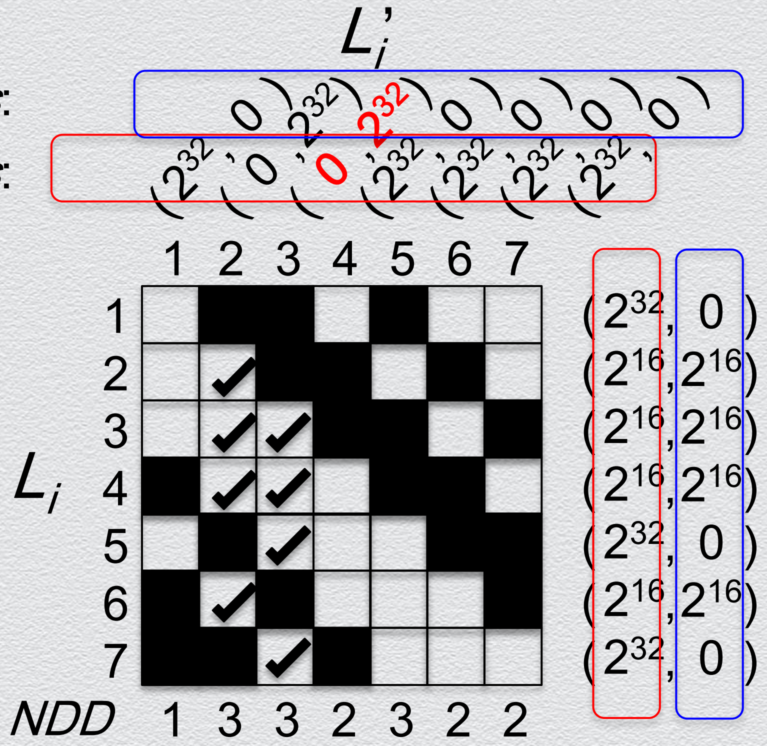


Current complexity

2^{32}

Demonstration for Rijndael-224

- 2nd guess: Choose the value and difference of L_3'.

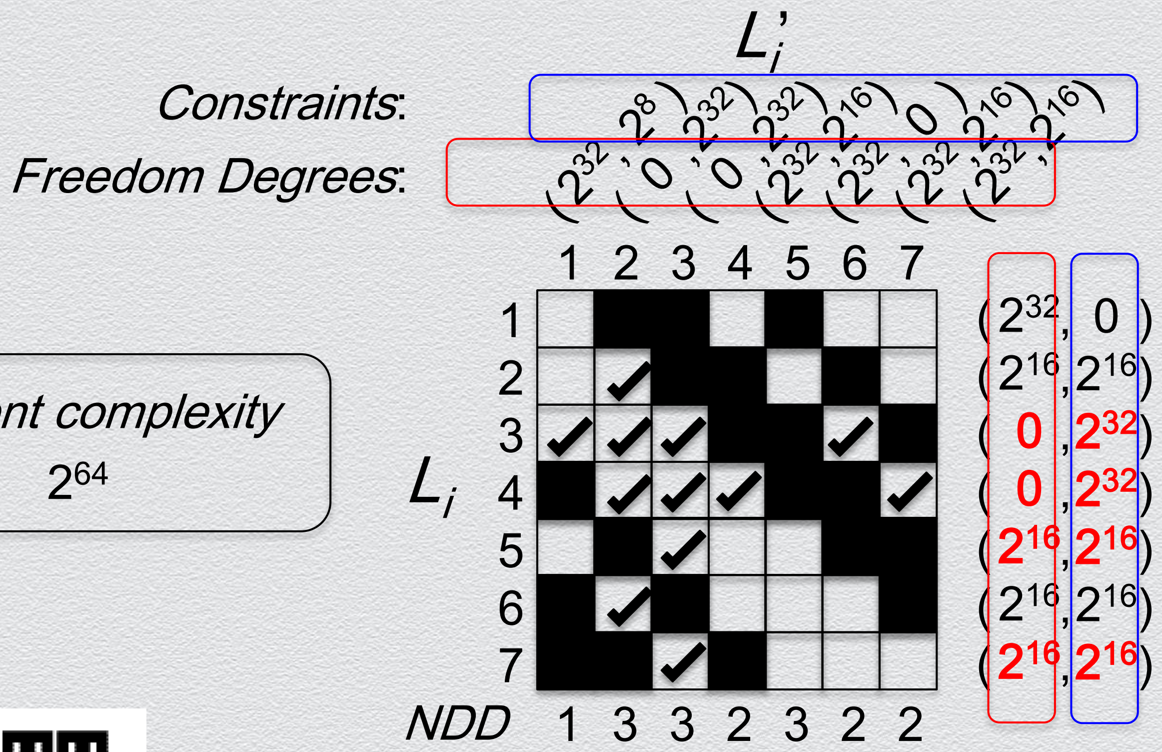


Current complexity

2^{64}

Demonstration for Rijndael-224

- 2^{nd} determine: Update constrains for other Super-Sboxes.

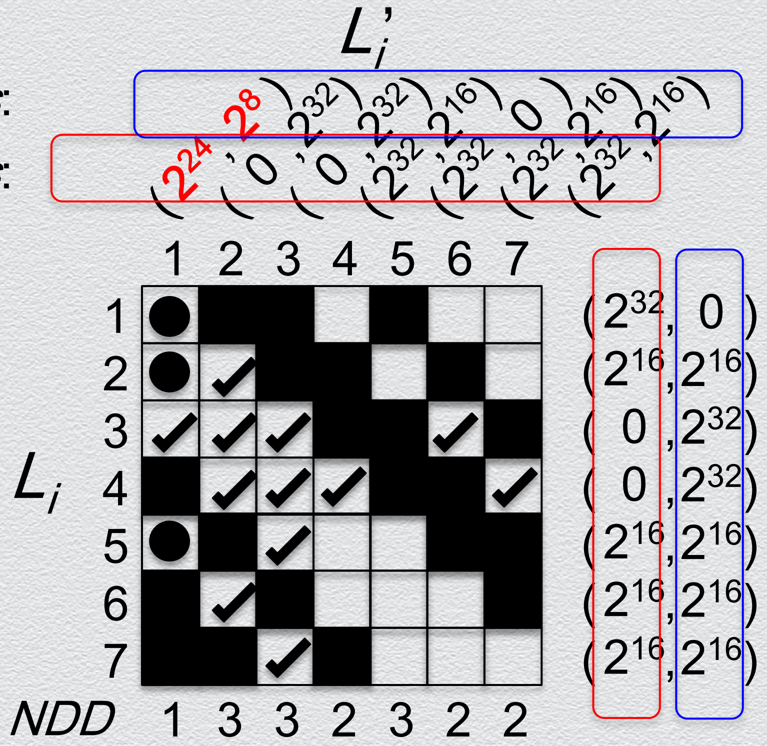


Current complexity

2^{64}

Demonstration for Rijndael-224

- 2^{nd} determine: Fix differences of Super-Sboxes if constraints > NDD.

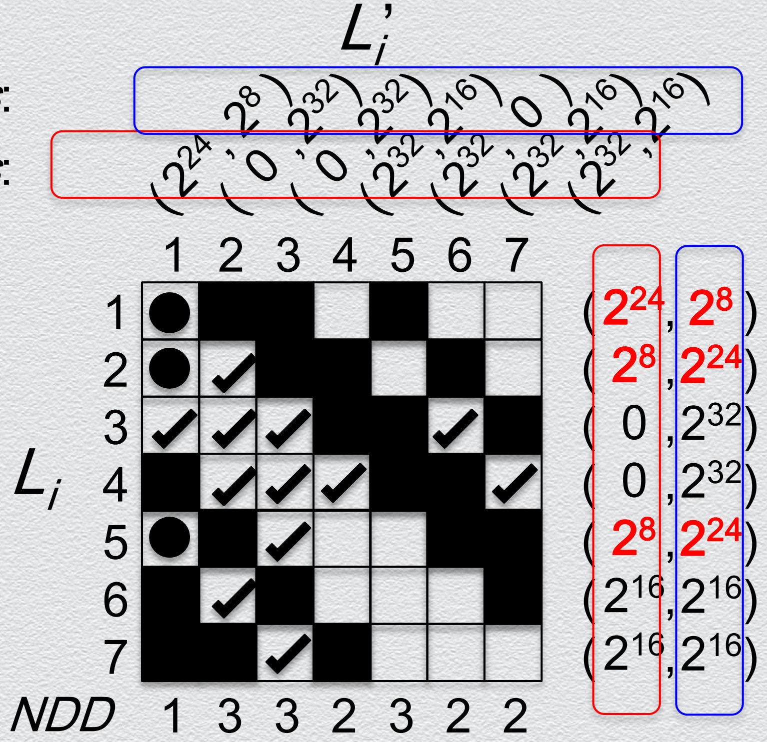


Current complexity

2^{64}

Demonstration for Rijndael-224

- 2^{nd} determine: Further update constraints as long as it is possible.

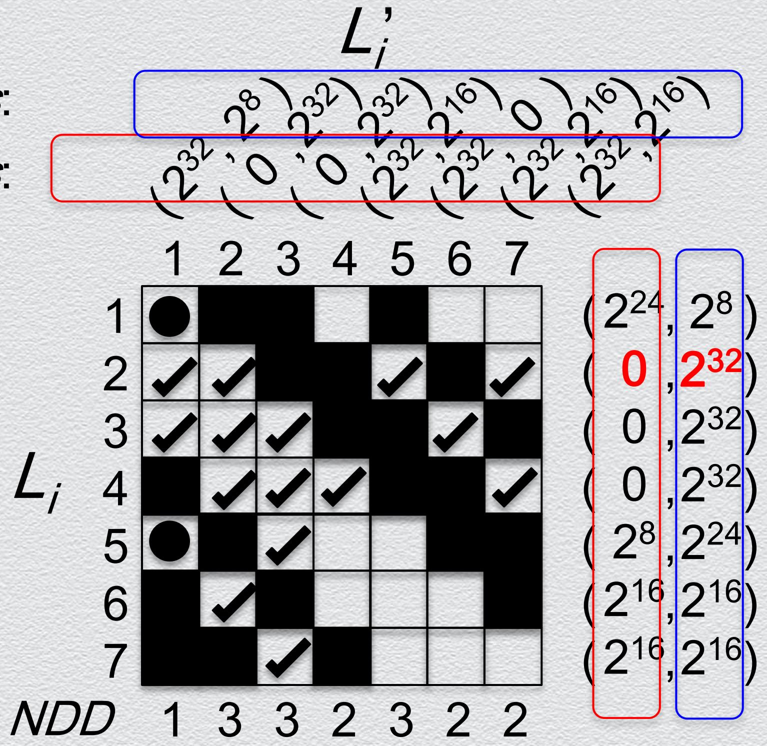


Current complexity

2^{64}

Demonstration for Rijndael-224

- 3rd guess: Choose the value and difference of L_2.

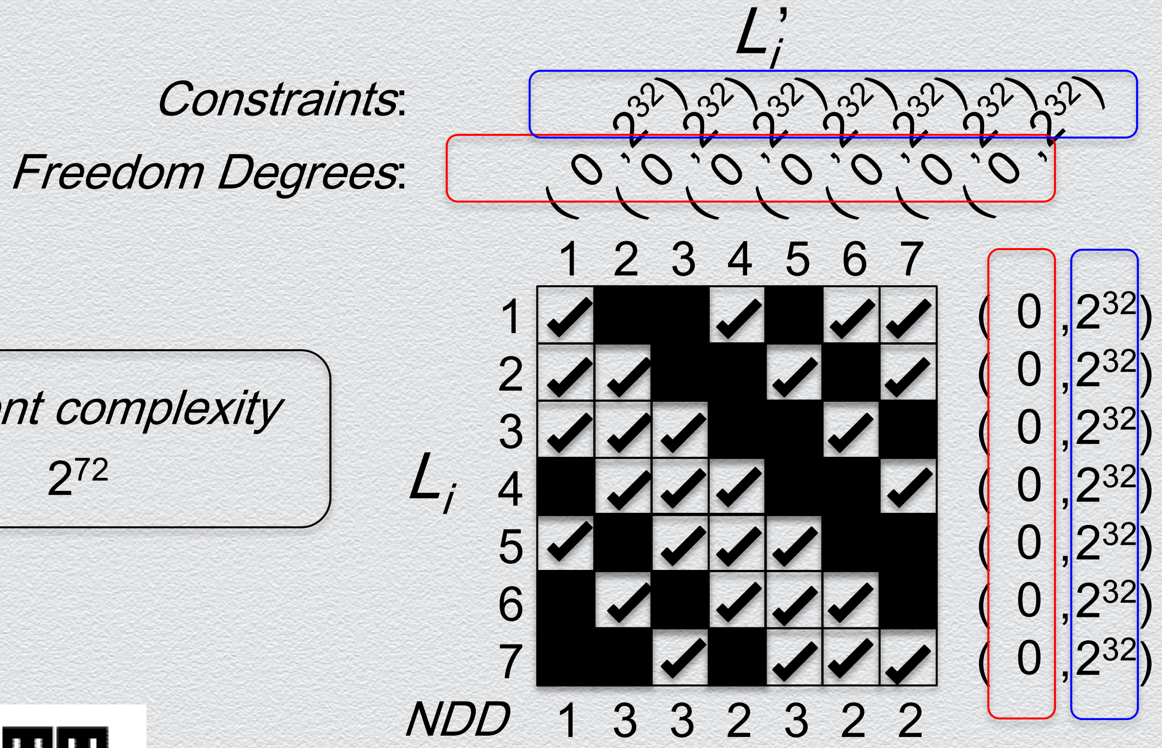


Current complexity

2^{72}

Demonstration for Rijndael-224

- 3rd determine: All states will be determined.



Current complexity

2^{72}

Summary of Our Tool

- The above demonstration shows that if Super-Sboxes are analyzed in the order of $L_2' \rightarrow L_3' \rightarrow L_2$, the attack complexity is 2^{72}.
- Our automated tool allows us to check all Super-Sboxes orders.
- Among all the choices, we found that the best choice achieves 2^{72}.
- This is the first 9-round attack on Rijndael-224.

- Easily applied to other AES-based permutation.
- Easily applied to other ShiftRows parameters.

Application Results

Summary of Results (for Wide-block Rijndael)

Target	State size	Previous	Ours	Different ShiftRows (Original Weak Strong)
Rijndael-160	4×5	8 rounds	N/A	
Rijndael-192	4×6	8 rounds	9 rounds	$(2^{112}, 2^{112}, 2^{112})$
Rijndael-224	4×7	8 rounds	9 rounds	$(2^{120}, 2^{104}, 2^{120})$
Rijndael-256	4×8	9 rounds	N/A	

Summary of Results (Grøstl-512 Permutation)

Target	State size	Previous	Complexity	Optimality
Grøstl-512	8×16	9 rounds	2^{392}	✓

Complexity for random permutation: 2^{441}

Results for Different ShiftRows

Complexity:	2^{336}	2^{360}	2^{392}	2^{424}	2^{448}	2^{456}	2^{464}
#Parameters:	32	128	320	928	512	256	128

Those parameters resist the attack.

NTT

Examples of 128 New ShiftRows Parameters

Table 4. 128 New ShiftBytes Parameters for the Grøstl-512 Permutation

Class 1	Class 7	Class 13
(0 , 1 , 2 , 3 , 4 , 7 , 9 ,12)	(0 , 1 , 2 , 3 , 6 , 7 , 9 ,14)	(0 , 1 , 2 , 5 , 6 , 8 , 9 ,11)
(0 , 1 , 2 , 3 , 6 , 8 ,11 ,15)	(0 , 1 , 2 , 5 , 6 , 8 ,13 ,15)	(0 , 1 , 3 , 4 , 6 ,11 ,12 ,13)
(0 , 1 , 2 , 5 , 7 ,10 ,14 ,15)	(0 , 1 , 3 , 8 ,10 ,11 ,12 ,13)	(0 , 1 , 3 , 8 , 9 ,10 ,13 ,14)
(0 , 1 , 4 , 6 , 9 ,13 ,14 ,15)	(0 , 1 , 4 , 5 , 7 ,12 ,14 ,15)	(0 , 1 , 4 , 5 , 7 , 8 ,10 ,15)
(0 , 2 , 5 , 9 ,10 ,11 ,12 ,13)	(0 , 2 , 3 , 4 , 5 , 8 , 9 ,11)	(0 , 2 , 3 , 5 ,10 ,11 ,12 ,15)
(0 , 3 , 5 , 8 ,12 ,13 ,14 ,15)	(0 , 2 , 7 , 9 ,10 ,11 ,12 ,15)	(0 , 2 , 7 , 8 , 9 ,12 ,13 ,15)
(0 , 3 , 7 , 8 , 9 ,10 ,11 ,14)	(0 , 3 , 4 , 6 ,11 ,13 ,14 ,15)	(0 , 3 , 4 , 6 , 7 , 9 ,14 ,15)
(0 , 4 , 5 , 6 , 7 , 8 ,11 ,13)	(0 , 5 , 7 , 8 , 9 ,10 ,13 ,14)	(0 , 5 , 6 , 7 ,10 ,11 ,13 ,14)

Concluding Remarks

- Developed a complexity evaluation tool for improved rebound attack.
- It can find an optimized attack procedure and complexity
- Applications
  - The first 9-round distinguisher on Rijndael-192
  - The first 9-round distinguisher on Rijndael-224
  - Optimality of the previous 9-round distinguisher on Grøstl-512 permutation
  - New stronger ShiftRows parameters for Grøstl-512 permutation

Thank you for your attention !!

Practical collision attack on 40-step RIPEMD-128

Gaoli Wang[1,2]

[1]Donghua University, Shanghai, China
[2]State Key Laboratory of Information Security
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

RSA Conference Cryptographers' Track (CT-RSA 2014)
San Francisco, America

February, 2014

General security notions

h is a hash function that takes an n-bit initial value IV and an m-bit message block M as inputs, and outputs another n-bit chaining value.

- **Collision:** two messages $M_1 \neq M_2$ satisfy: $h(IV, M_1) = h(IV, M_2)$.
- **Near-collision:** A k-bit ($k < n$) near-collision: two messages $M_1 \neq M_2$ satisfy:

$$HW(h(IV, M_1) \oplus h(IV, M_2)) = n - k,$$

  where HW denotes the Hamming distance.
- **Semi-free-start Collision:** $M_1 \neq M_2$ satisfy:

$$h(CV, M_1) = h(CV, M_2),$$

  where $CV = IV$ does not always hold.
- **Free-start Collision:** $CV_1 \neq CV_2, M_1 \neq M_2$ satisfy:

$$h(CV_1, M_1) = h(CV_2, M_2).$$

General security notions

h is a hash function that takes an n-bit initial value IV and an m-bit message block M as inputs, and outputs another n-bit chaining value.

- **Collision:** two messages $M_1 \neq M_2$ satisfy: $h(IV, M_1) = h(IV, M_2)$.
- **Near-collision:** A k-bit ($k < n$) near-collision: two messages $M_1 \neq M_2$ satisfy:

$$HW(h(IV, M_1) \oplus h(IV, M_2)) = n - k,$$

where HW denotes the Hamming distance.

- **Semi-free-start Collision:** $M_1 \neq M_2$ satisfy:

$$h(CV, M_1) = h(CV, M_2),$$

where $CV = IV$ does not always hold.

- **Free-start Collision:** $CV_1 \neq CV_2, M_1 \neq M_2$ satisfy:

$$h(CV_1, M_1) = h(CV_2, M_2).$$

General security notions

h is a hash function that takes an n-bit initial value IV and an m-bit message block M as inputs, and outputs another n-bit chaining value.

- **Collision:** two messages $M_1 \neq M_2$ satisfy: $h(IV, M_1) = h(IV, M_2)$.
- **Near-collision:** A k-bit ($k < n$) near-collision: two messages $M_1 \neq M_2$ satisfy:

$$HW(h(IV, M_1) \oplus h(IV, M_2)) = n - k,$$

  where HW denotes the Hamming distance.
- **Semi-free-start Collision:** $M_1 \neq M_2$ satisfy:

$$h(CV, M_1) = h(CV, M_2),$$

  where $CV = IV$ does not always hold.
- **Free-start Collision:** $CV_1 \neq CV_2, M_1 \neq M_2$ satisfy:

$$h(CV_1, M_1) = h(CV_2, M_2).$$

General security notions

h is a hash function that takes an n-bit initial value IV and an m-bit message block M as inputs, and outputs another n-bit chaining value.

- **Collision:** two messages $M_1 \neq M_2$ satisfy: $h(IV, M_1) = h(IV, M_2)$.
- **Near-collision:** A k-bit ($k < n$) near-collision: two messages $M_1 \neq M_2$ satisfy:

  $$HW(h(IV, M_1) \oplus h(IV, M_2)) = n - k,$$

  where HW denotes the Hamming distance.
- **Semi-free-start Collision:** $M_1 \neq M_2$ satisfy:

  $$h(CV, M_1) = h(CV, M_2),$$

  where $CV = IV$ does not always hold.
- **Free-start Collision:** $CV_1 \neq CV_2, M_1 \neq M_2$ satisfy:

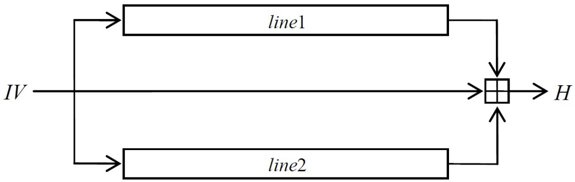
  $$h(CV_1, M_1) = h(CV_2, M_2).$$

Summary of Attacks on RIPEMD-128

Attack	Steps	Generic	Complexity	Reference
collision	32	2^{64}	2^{28}	Wang et al., Journal of Software in China 2008
collision	38	2^{64}	2^{14}	Mendel et al., FSE 2012
collision	40	2^{64}	2^{35}	NEW
near collision	44	$2^{47.8}$	2^{32}	Mendel et al., FSE 2012
free-start collision	48	2^{64}	2^{40}	Mendel et al., FSE 2012
preimage	33	2^{128}	$2^{124.5}$	Ohtahara et al., INSCRYPT 2010
preimage	35^*	2^{128}	2^{121}	Ohtahara et al., INSCRYPT 2010
preimage	36^*	2^{128}	$2^{126.5}$	Wang et al., CT-RSA 2011
distinguishing	48	2^{76}	2^{70}	Mendel et al., FSE 2012
distinguishing	45	2^{42}	2^{27}	Sasaki et al., ACNS 2012
distinguishing	47	2^{42}	2^{39}	Sasaki et al., ACNS 2012
distinguishing	48	–	2^{53}	Sasaki et al., ACNS 2012
distinguishing	52	–	2^{107}	Sasaki et al., ACNS 2012
distinguishing	64	2^{128}	$2^{105.4}$	Landelle, et al., EUROCRYPT 2013
semi-free-start collision	64	2^{64}	$2^{61.57}$	Landelle, et al., EUROCRYPT 2013

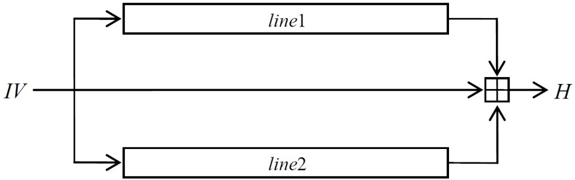
* The attack starts from an intermediate step.

The Hash Function RIPEMD-128



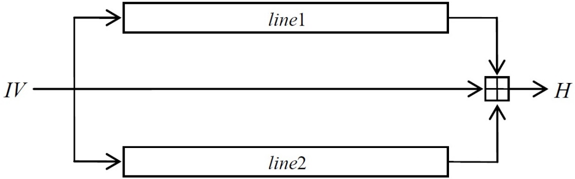
- Proposed by Hans Dobbertin, Antoon Bosselaers and Bart Preneel, Standardized by ISO/IEC and was used in HMAC in RFC
- Merkle-Damgård design
  - Message block size: 512 bits
  - state (chaining variable): 128 bits
  - 64 steps
- A double-branch hash function — —the compression function consists of two parallel operations denoted by line1 operation and line2 operation, respectively.

The Hash Function RIPEMD-128



- Proposed by Hans Dobbertin, Antoon Bosselaers and Bart Preneel, Standardized by ISO/IEC and was used in HMAC in RFC
- Merkle-Damgård design
  - Message block size: 512 bits
  - state (chaining variable): 128 bits
  - 64 steps
- A double-branch hash function ——the compression function consists of two parallel operations denoted by line1 operation and line2 operation, respectively.

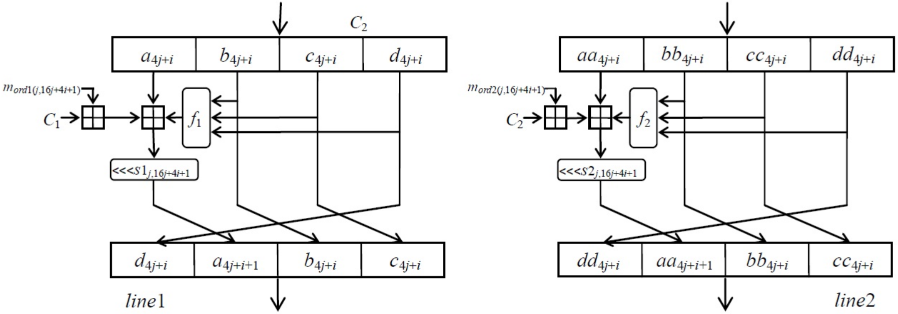
The Hash Function RIPEMD-128



- Proposed by Hans Dobbertin, Antoon Bosselaers and Bart Preneel, Standardized by ISO/IEC and was used in HMAC in RFC
- Merkle-Damgård design
  - Message block size: 512 bits
  - state (chaining variable): 128 bits
  - 64 steps
- A double-branch hash function ——the compression function consists of two parallel operations denoted by line1 operation and line2 operation, respectively.

State Update Transformation

- Operations: $+$ mod 2^{32}, rotation, logical functions



Logical functions

Logical functions in RIPEMD-128:

$$\begin{aligned} F(X,Y,Z) &= X \oplus Y \oplus Z \\ G(X,Y,Z) &= (X \wedge Y) \vee (\neg X \wedge Z) \\ H(X,Y,Z) &= (X \vee \neg Y) \oplus Z \\ I(X,Y,Z) &= (X \wedge Z) \vee (Y \wedge \neg Z) \end{aligned}$$

Round	Line1 operation	Line2 operation
0 (Steps 1-16)	$F(X,Y,Z)$	$I(X,Y,Z)$
1 (Steps 17-32)	$G(X,Y,Z)$	$H(X,Y,Z)$
2 (Steps 33-48)	$H(X,Y,Z)$	$G(X,Y,Z)$
3 (Steps 49-64)	$I(X,Y,Z)$	$F(X,Y,Z)$

The classical collision attacks for Hash functions

Wang's method [Wang, CRYPTO 2005, EUROCRYPT 2005]

1. Choose proper difference of message. Find a concrete differential characteristic which holds with high probability without round 1.
2. Derive a set of sufficient conditions which ensure the differential characteristic hold.
3. Modify the message to fulfill most of the sufficient conditions on chaining variables.

The classical collision attacks for Hash functions

Wang's method [Wang, CRYPTO 2005, EUROCRYPT 2005]

1. Choose proper difference of message. Find a concrete differential characteristic which holds with high probability without round 1.
2. Derive a set of sufficient conditions which ensure the differential characteristic hold.
3. Modify the message to fulfill most of the sufficient conditions on chaining variables.

The classical collision attacks for Hash functions

Wang's method [Wang, CRYPTO 2005, EUROCRYPT 2005]

1. Choose proper difference of message. Find a concrete differential characteristic which holds with high probability without round 1.
2. Derive a set of sufficient conditions which ensure the differential characteristic hold.
3. Modify the message to fulfill most of the sufficient conditions on chaining variables.

Logical functions - Absorption property

Round	Line1 operation	Line2 operation
0 (Steps 1-16)	$F(X,Y,Z)$	$I(X,Y,Z)$
1 (Steps 17-32)	$G(X,Y,Z)$	$H(X,Y,Z)$
2 (Steps 33-48)	$H(X,Y,Z)$	$G(X,Y,Z)$
3 (Steps 49-64)	$I(X,Y,Z)$	$F(X,Y,Z)$

- $F(X,Y,Z) = X \oplus Y \oplus Z$: the absorption property of $F(X,Y,Z)$ does not hold
- In the practical collision attack on the first 32-step RIPEMD-128 [Wang, Journal of Software in China 2008]
  - the differential characteristic of Line1 operation almost keeps away from $F(X,Y,Z)$
  - by choosing $\Delta m_{14} \neq 0, \Delta m_i = 0 (0 \leq i \leq 15, i \neq 14)$

Logical functions - Absorption property

Round	Line1 operation	Line2 operation
0 (Steps 1-16)	$F(X,Y,Z)$	$I(X,Y,Z)$
1 (Steps 17-32)	$G(X,Y,Z)$	$H(X,Y,Z)$
2 (Steps 33-48)	$H(X,Y,Z)$	$G(X,Y,Z)$
3 (Steps 49-64)	$I(X,Y,Z)$	$F(X,Y,Z)$

- $F(X,Y,Z) = X \oplus Y \oplus Z$: the absorption property of $F(X,Y,Z)$ does not hold
- In the practical collision attack on the first 32-step RIPEMD-128 [Wang, Journal of Software in China 2008]
  - the differential characteristic of Line1 operation almost keeps away from $F(X,Y,Z)$
  - by choosing $\Delta m_{14} \neq 0, \Delta m_i = 0 (0 \leq i \leq 15, i \neq 14)$

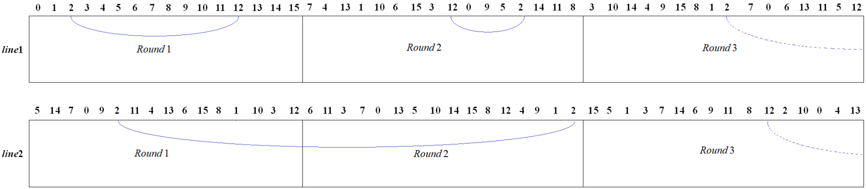
Logical functions - Absorption property

- In the practical collision attack on the first 38-step RIPEMD-128 [Mendel, FSE 2012]
  - take advantage of the property of $F(X, Y, Z)$
  - construct a differential characteristic, the difference starts from the first step of line1 operation
  - by choosing $\Delta m_0 \neq 0, \Delta m_6 \neq 0, \Delta m_i = 0 (1 \leq i \leq 15, i \neq 6)$
- In the practical collision attack on the first 40-step RIPEMD-128 [NEW]
  - take advantage of the property of $F(X, Y, Z)$
  - choosing a different message difference than in [Mendel, FSE 2012]

Logical functions - Absorption property

- In the practical collision attack on the first 38-step RIPEMD-128 [Mendel, FSE 2012]
  - take advantage of the property of $F(X, Y, Z)$
  - construct a differential characteristic, the difference starts from the first step of line1 operation
  - by choosing $\Delta m_0 \neq 0, \Delta m_6 \neq 0, \Delta m_i = 0 (1 \leq i \leq 15, i \neq 6)$
- In the practical collision attack on the first 40-step RIPEMD-128 [NEW]
  - take advantage of the property of $F(X, Y, Z)$
  - choosing a different message difference than in [Mendel, FSE 2012]

Collision attack on the first 40-step RIPEMD-128: Step 1. Choosing the message difference



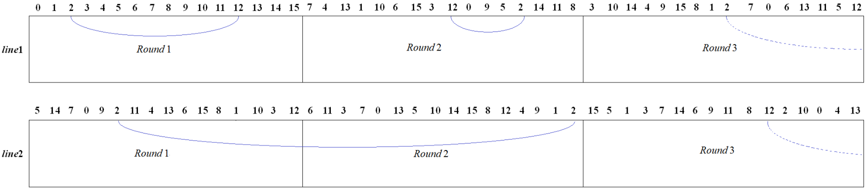
Goals:

- form a local collision in the second round of Line1 operation
- characteristics hold with high pr. after message modification

Choice:

- $\Delta m_2 \neq 0, \Delta m_{12} \neq 0, \Delta m_i = 0 (0 \leq i \leq 15, i \neq 2, 12)$
- non-linear characteristics are in the first round of Line1 operation, and in the rounds 1-2 of Line2 operation

Collision attack on the first 40-step RIPEMD-128: Step 1. Choosing the message difference



Goals:

- form a local collision in the second round of Line1 operation
- characteristics hold with high pr. after message modification

Choice:

- $\Delta m_2 \neq 0, \Delta m_{12} \neq 0, \Delta m_i = 0 (0 \leq i \leq 15, i \neq 2, 12)$
- non-linear characteristics are in the first round of Line1 operation, and in the rounds 1-2 of Line2 operation

Step 1. Differential Characteristic for Line1 Operation

Step	Message M	$Shift$	Δm_i	The output for M'
1	m_0	11		a_1
2	m_1	14		d_1
3	m_2	15	2^8	$c_1[-1, -2, 3, -24, \ldots, -32]$
4	m_3	12		$b_1[4, \ldots, 10, -11, 12, -13, \ldots, -22, 23]$
5	m_4	5		$a_2[1, -2, \ldots, -11, 12, \ldots, 21, -22, \ldots, -32]$
6	m_5	8		d_2
7	m_6	7		c_2
8	m_7	9		$b_2[2, \ldots, 10, -11, -12]$
9	m_8	11		$a_3[-2, \ldots, -11, 12]$
10	m_9	13		d_3
11	m_{10}	14		c_3
12	m_{11}	15		b_3
13	m_{12}	6	-2	a_4
...	...	...	...	...
25	m_{12}	7	-2	$a_7[-9]$
26	m_0	12		d_7
27	m_9	15		c_7
28	m_5	9		b_7
29	m_2	11	2^8	a_8
...	...	...	...	...
40	m_1	15		b_{10}

Step 1. Differential Characteristic for Line2 Operation

Step	Message M	Shift	Δm_i	The output for M'
6	m_2	15	2^8	$dd_2[-1,-2,-3,4,-24,\ldots,-32]$
7	m_{11}	15		$cc_2[17,18-19]$
8	m_4	5		$bb_2[8,\ldots,15,-16,-24]$
9	m_{13}	7		$aa_3[-31]$
10	m_6	7		$dd_3[8,-23,26,\ldots,31,-32]$
11	m_{15}	8		$cc_3[7,8,-25]$
12	m_8	11		$bb_3[2,5]$
13	m_1	14		$aa_4[7,-9,-12]$
14	m_{10}	14		$dd_4[-5,7,-9]$
15	m_3	12		$cc_4[-5]$
16	m_{12}	6	-2	bb_4
17	m_6	9		$aa_5[-21]$
18	m_{11}	13		$dd_5[-20,-21]$
19	m_3	15		$cc_5[-20]$
20	m_7	7		bb_5
21	m_0	12		aa_6
22	m_{13}	8		$dd_6[-29]$
23	m_5	9		$cc_6[-29]$
24	m_{10}	11		bb_6
25	m_{14}	7		aa_7
26	m_{15}	7		dd_7
27	m_8	12		$cc_7[-9]$
28	m_{12}	7	-2	$bb_7[-9]$
29	m_4	6		aa_8
30	m_9	15		dd_8
31	m_1	13		cc_8
32	m_2	11	2^8	bb_8
...	...	...	...	...
40	m_9	14		bb_{10}

Step 2. A Set of Sufficient Conditions for the Characteristic of Line1

Step	Variable	Conditions on the Chaining Variable
2	d_1	$d_{1,i} = a_{1,i}(i = 1, 2, 3, 31), d_{1,i} \neq a_{1,i}(i = 24, ..., 30, 32)$
3	c_1	$c_{1,3} = 0, c_{1,i} = 1(i = 1, 2, 24, ..., 32), c_{1,i} = d_{1,i}(i = 7, ..., 10, 12, 17, ..., 22), c_{1,i} \neq d_{1,i}(i = 4, 5, 6, 11, 13, ..., 16, 23)$
4	b_1	$b_{1,i} = 0(i = 4, ..., 10, 12, 23), b_{1,i} = 1(i = 11, 13, ..., 22), b_{1,i} = d_{1,i}(i = 1, 2, 24, ..., 27, 29, ..., 32), b_{1,i} \neq d_{1,i}(i = 3, 28)$
5	a_2	$a_{2,i} = 0(i = 1, 12, ..., 21), a_{2,i} = 1(i = 2, ..., 11, 22, ..., 32)$
6	d_2	$d_{2,i} = b_{1,i}(i = 1, 3), d_{2,i} \neq b_{1,i}(i = 2, 24, ..., 32)$
7	c_2	$c_{2,i} = d_{2,i}(i = 1, ..., 10, 13, ..., 21, 24)$ $c_{2,i} \neq d_{2,i}(i = 11, 12, 22, 23, 25, ..., 32)$
8	b_2	$b_{2,i} = 0(i = 2, ..., 10), b_{2,i} = 1(i = 11, 12)$
9	a_3	$a_{3,12} = 0, a_{3,i} = 1(i = 2, ..., 11)$
11	c_3	$c_{3,i} = d_{3,i}(i = 2, ..., 10, 12), c_{3,11} \neq d_{3,11}$
24	b_6	$b_{6,9} = c_{6,9}$
25	a_7	$a_{7,9} = 1$
26	d_7	$d_{7,9} = 0$
27	c_7	$c_{7,9} = 1$

Step 2. Sufficient Conditions for Charac. of Line2

Step	Variable	Conditions on the Chaining Variable
4	bb_1	$bb_{1,i} = 0(i = 1, 3, 4, 24, \ldots, 32), bb_{1,2} = 1$
5	aa_2	$aa_{2,i} = 0(i = 3, 17, 18), aa_{2,i} = 1(i = 1, 2, 4, 19, 24, \ldots, 32)$
6	dd_2	$dd_{2,i} = 0(i = 4, 8, \ldots, 16), dd_{2,i} = 1(i = 1, 2, 3, 17, 18, 19, 24, \ldots, 32)$
7	cc_2	$cc_{2,i} = 0(i = 16, 17, 18, 24, 26, \ldots, 32), cc_{2,i} = 1(i = 8, \ldots, 15, 19)$
8	bb_2	$bb_{2,i} = 0(i = 8, \ldots, 15, 19, 23, 26, \ldots, 32), bb_{2,i} = 1(i = 16, 24)$ $bb_{2,i} = cc_{2,i}(i = 1, 2, 3, 4, 25)$
9	aa_3	$aa_{3,i} = 0(i = 7, 23, 27), aa_{3,i} = 1(i = 8, 19, 25, 26, 28, \ldots, 32), aa_{3,i} = bb_{2,i}(i = 17, 18)$
10	dd_3	$dd_{3,i} = 0(i = 2, 5, 8, 25, \ldots, 31), dd_{3,i} = 1(i = 7, 23, 32), dd_{3,i} = aa_{3,i}(i = 9, \ldots, 16, 24)$
11	cc_3	$cc_{3,i} = 0(i = 7, 8, 12), cc_{3,i} = 1(i = 2, 5, 9, 25, 26, 30, 31)$
12	bb_3	$bb_{3,i} = 0(i = 2, 5, 8, 25, 26, 30, 31), bb_{3,i} = 1(i = 7, 12), bb_{3,i} = cc_{3,i}(i = 23, 27, 28, 29)$ $bb_{3,32} \neq cc_{3,32}$
13	aa_4	$aa_{4,i} = 0(i = 5, 7), aa_{4,i} = 1(i = 8, 9, 12, 25)$
14	dd_4	$dd_{4,7} = 0, dd_{4,i} = 1(i = 5, 9), dd_{4,2} = aa_{4,2}$
15	cc_4	$cc_{4,i} = 0(i = 7, 9), cc_{4,5} = 1, cc_{4,12} = dd_{4,12}$
16	bb_4	$bb_{4,i} = 0(i = 5, 21)$
17	aa_5	$aa_{5,20} = 0, aa_{5,21} = 1$
18	dd_5	$dd_{5,i} = 1(i = 20, 21)$
19	cc_5	$cc_{5,21} = 0, cc_{5,20} = 1$
20	bb_5	$bb_{5,20} = 0$
21	aa_6	$aa_{6,29} = 0$
22	dd_6	$dd_{6,29} = 1$
23	cc_6	$cc_{6,29} = 1$
24	bb_6	$bb_{6,29} = 0$
26	dd_7	$dd_{7,9} = 0$
27	cc_7	$cc_{7,9} = 1$
28	bb_7	$bb_{7,9} = 1$
29	aa_8	$aa_{8,9} = 0$

Step 3. Message modification

- Freedom: m_0 to m_{15}
- In steps 1-15 of the two branches, after message modification:
  - all the corrected conditions: hold with probability 2^{-3}
  - being corrected with pr. $3/4$: 3 conditions
  - being corrected with pr. $5/8$: 1 conditions
  - being not corrected: 29 conditions
  - Thus, all the conditions: hold with pr. 2^{-35}
  - These equivalent 35 conditions can be satisfied by searching m_0 to m_{15} except m_{12}
- The other steps except 1-15 of two branches, being not corrected:
  - Line1: 4 conditions
  - Line2: 17 conditions
  - These 21 conditions be satisfied by searching m_{12}

Step 3. Message modification

- Freedom: m_0 to m_{15}
- In steps 1-15 of the two branches, after message modification:
  - all the corrected conditions: hold with probability 2^{-3}
  - being corrected with pr. $3/4$: 3 conditions
  - being corrected with pr. $5/8$: 1 conditions
  - being not corrected: 29 conditions
  - Thus, all the conditions: hold with pr. 2^{-35}
  - These equivalent 35 conditions can be satisfied by searching m_0 to m_{15} except m_{12}
- The other steps except 1-15 of two branches, being not corrected:
  - Line1: 4 conditions
  - Line2: 17 conditions
  - These 21 conditions be satisfied by searching m_{12}

Step 3. Message modification

- Freedom: m_0 to m_{15}
- In steps 1-15 of the two branches, after message modification:
  - all the corrected conditions: hold with probability 2^{-3}
  - being corrected with pr. $3/4$: 3 conditions
  - being corrected with pr. $5/8$: 1 conditions
  - being not corrected: 29 conditions
  - Thus, all the conditions: hold with pr. 2^{-35}
  - These equivalent 35 conditions can be satisfied by searching m_0 to m_{15} except m_{12}
- The other steps except 1-15 of two branches, being not corrected:
  - Line1: 4 conditions
  - Line2: 17 conditions
  - These 21 conditions be satisfied by searching m_{12}

Collision search algorithm

1. in the message modification, add some conditions on: $b_{0,i} = 1\ (i = 1, 2, 3, 27)$, $b_{0,i} = 0\ (i = 7, ..., 10, 13, ..., 24)$,
   Thus, search the first block N such that the hash value of N satisfies $b_{0,i} = 1\ (i = 1, 2, 3, 27)$ and $b_{0,i} = 0\ (i = 7, ..., 10, 13, ..., 24)$.
2. Choose $m_i\ (0 \leq i \leq 15, i \neq 12)$, do message modification, check whether all the conditions in steps 1-15 of the two branches hold.
3. Choose m_{12}, check whether the two hash values are equal.

Therefore, the total complexity is

$$2^{35} + 2^{21}$$

calls to the 40-step RIPEMD-128.

Collision search algorithm

1. in the message modification, add some conditions on: $b_{0,i} = 1$ $(i = 1, 2, 3, 27)$, $b_{0,i} = 0$ $(i = 7, ..., 10, 13, ..., 24)$,
   Thus, search the first block N such that the hash value of N satisfies $b_{0,i} = 1$ $(i = 1, 2, 3, 27)$ and $b_{0,i} = 0$ $(i = 7, ..., 10, 13, ..., 24)$.
2. Choose m_i $(0 \leq i \leq 15, i \neq 12)$, do message modification, check whether all the conditions in steps 1-15 of the two branches hold.
3. Choose m_{12}, check whether the two hash values are equal.

Therefore, the total complexity is

$$2^{35} + 2^{21}$$

calls to the 40-step RIPEMD-128.

Collision search algorithm

1. in the message modification, add some conditions on: $b_{0,i} = 1$ $(i = 1, 2, 3, 27)$, $b_{0,i} = 0$ $(i = 7, ..., 10, 13, ..., 24)$,
   Thus, search the first block N such that the hash value of N satisfies $b_{0,i} = 1$ $(i = 1, 2, 3, 27)$ and $b_{0,i} = 0$ $(i = 7, ..., 10, 13, ..., 24)$.
2. Choose m_i $(0 \leq i \leq 15, i \neq 12)$, do message modification, check whether all the conditions in steps 1-15 of the two branches hold.
3. Choose m_{12}, check whether the two hash values are equal.

Therefore, the total complexity is

$$2^{35} + 2^{21}$$

calls to the 40-step RIPEMD-128.

Collision search algorithm

1. in the message modification, add some conditions on: $b_{0,i} = 1$ $(i = 1, 2, 3, 27)$, $b_{0,i} = 0$ $(i = 7, ..., 10, 13, ..., 24)$,
   Thus, search the first block N such that the hash value of N satisfies $b_{0,i} = 1$ $(i = 1, 2, 3, 27)$ and $b_{0,i} = 0$ $(i = 7, ..., 10, 13, ..., 24)$.
2. Choose m_i $(0 \leq i \leq 15, i \neq 12)$, do message modification, check whether all the conditions in steps 1-15 of the two branches hold.
3. Choose m_{12}, check whether the two hash values are equal.

Therefore, the total complexity is

$$2^{35} + 2^{21}$$

calls to the 40-step RIPEMD-128.

Collision search algorithm

1. in the message modification, add some conditions on: $b_{0,i} = 1$ $(i = 1, 2, 3, 27)$, $b_{0,i} = 0$ $(i = 7, ..., 10, 13, ..., 24)$,
   Thus, search the first block N such that the hash value of N satisfies $b_{0,i} = 1$ $(i = 1, 2, 3, 27)$ and $b_{0,i} = 0$ $(i = 7, ..., 10, 13, ..., 24)$.
2. Choose m_i $(0 \leq i \leq 15, i \neq 12)$, do message modification, check whether all the conditions in steps 1-15 of the two branches hold.
3. Choose m_{12}, check whether the two hash values are equal.

Therefore, the total complexity is

$$2^{35} + 2^{21}$$

calls to the 40-step RIPEMD-128.

A collision example and Conclusion

A collision example for 40-step RIPEMD-128:

N	664504b6 cee9d90	d6e949ba c5078e4b	2176407d 84bae5bc	85426fc1 99f3f4ae	5ec28995 d7403dc6	c3d318b 917fa14c	787db431 85155db5	ae2c13fb fd9311e6
M	a7e4a89f ffe74d8e	6278156c 6df2c5f7	2a535118 a3ffdbfd	90eba965 53e156d4	670841b2 54f75d	ea6f8dcb f0d3a13f	800766d9 7eef12b9	d0bfa5c6 ef317f76
M'	a7e4a89f ffe74d8e	6278156c 6df2c5f7	2a535218 a3ffdbfd	90eba965 53e156d4	670841b2 54f75b	ea6f8dcb f0d3a13f	800766d9 7eef12b9	d0bfa5c6 ef317f76
H	a76df6ab	43ae1a6e	171d9fda	da03925e				

Conclusion:

- Find high-probability characteristics, implement message modifications.
- present a collision instance for 40-step RIPEMD-128.

A collision example and Conclusion

A collision example for 40-step RIPEMD-128:

N	664504b6 cee9d90	d6e949ba c5078e4b	2176407d 84bae5bc	85426fc1 99f3f4ae	5ec28995 d7403dc6	c3d318b 917fa14c	787db431 85155db5	ae2c13fb fd9311e6
M	a7e4a89f ffe74d8e	6278156c 6df2c5f7	2a535118 a3ffdbfd	90eba965 53e156d4	670841b2 54f75d	ea6f8dcb f0d3a13f	800766d9 7eef12b9	d0bfa5c6 ef317f76
M'	a7e4a89f ffe74d8e	6278156c 6df2c5f7	2a535218 a3ffdbfd	90eba965 53e156d4	670841b2 54f75b	ea6f8dcb f0d3a13f	800766d9 7eef12b9	d0bfa5c6 ef317f76
H	a76df6ab	43ae1a6e	171d9fda	da03925e				

Conclusion:

- Find high-probability characteristics, implement message modifications.
- present a collision instance for 40-step RIPEMD-128.

Thank you for your attention!