New Semi-Free-Start Collision Attack Framework for Reduced RIPEMD-160

Fukang Liu[1,4] Christoph Dobraunig[2] Florian Mendel[3]
Takanori Isobe[4,5] Gaoli Wang[1] Zhenfu Cao[1,6]

[1]East China Normal University, China
[2]Radboud University, The Netherlands
[3]Infineon Technologies AG, Germany
[4]University of Hyogo, Japan
[5]NICT, Japan
[6]Cyberspace Security Research Center, Shenzhen, China

Oct. 23, 2020

Cryptanalysis of MD-SHA Hash Family

Major breakthrough (full-round collision attacks):

1. MD4 (practical, Dobbertin, FSE'96)
2. RIPEMD (practical, Wang et al., EUROCRYPT'05)
3. MD5 (practical, Wang et al., EUROCRYPT'05)
4. SHA-0 (practical, Biham et al., EUROCRYPT'05)
5. SHA-1 (theoretical, Wang et al., CRYPTO'05)
6. SHA-1 (practical, Stevens et al., CRYPTO'17)

Full-round theoretical semi-free-start collision attacks:

1. RIPEMD-128 (Landelle et al., EUROCRYPT'13)

Cryptanalysis of MD-SHA Hash Family

Developed techniques:

1. start-from-the-middle (Dobbertin, FSE'96)
2. advanced message modification (Wang et al., EUROCRYPT'05)
3. modular differential characteristic (Wang et al., EUROCRYPT'05)
4. neutral bits (Biham et al., EUROCRYPT'05)
5. boomerangs (Joux et al., CRYPTO'07)

Cryptanalysis of MD-SHA Hash Family

Automatic tools for collision-generating differential characteristics:

1. guess-and-determine method
   (Cannière, ASIACRYPT'06)
2. meet-in-the-middle method
   (Stevens et al., EUROCRYPT'07)
3. improved guess-and-determine method
   (Mendel et al., ASIACRYPT'11)
4. improved guess-and-determine method
   (Eichlseder et al., FSE'14)

Cryptanalysis of Reduced RIPEMD-160

Table: Collision attacks on reduced RIPEMD-160

Steps	Time	Memory	Ref.
30/80	$2^{35.9}$	2^{32}	CRYPTO'19
31/80	$2^{41.5}$	2^{32}	CRYPTO'19
33/80	$2^{67.1}$	2^{32}	CRYPTO'19
34/80	$2^{74.3}$	2^{32}	CRYPTO'19

Cryptanalysis of Reduced RIPEMD-160

Table: Semi-free-start collision attacks on reduced RIPEMD-160

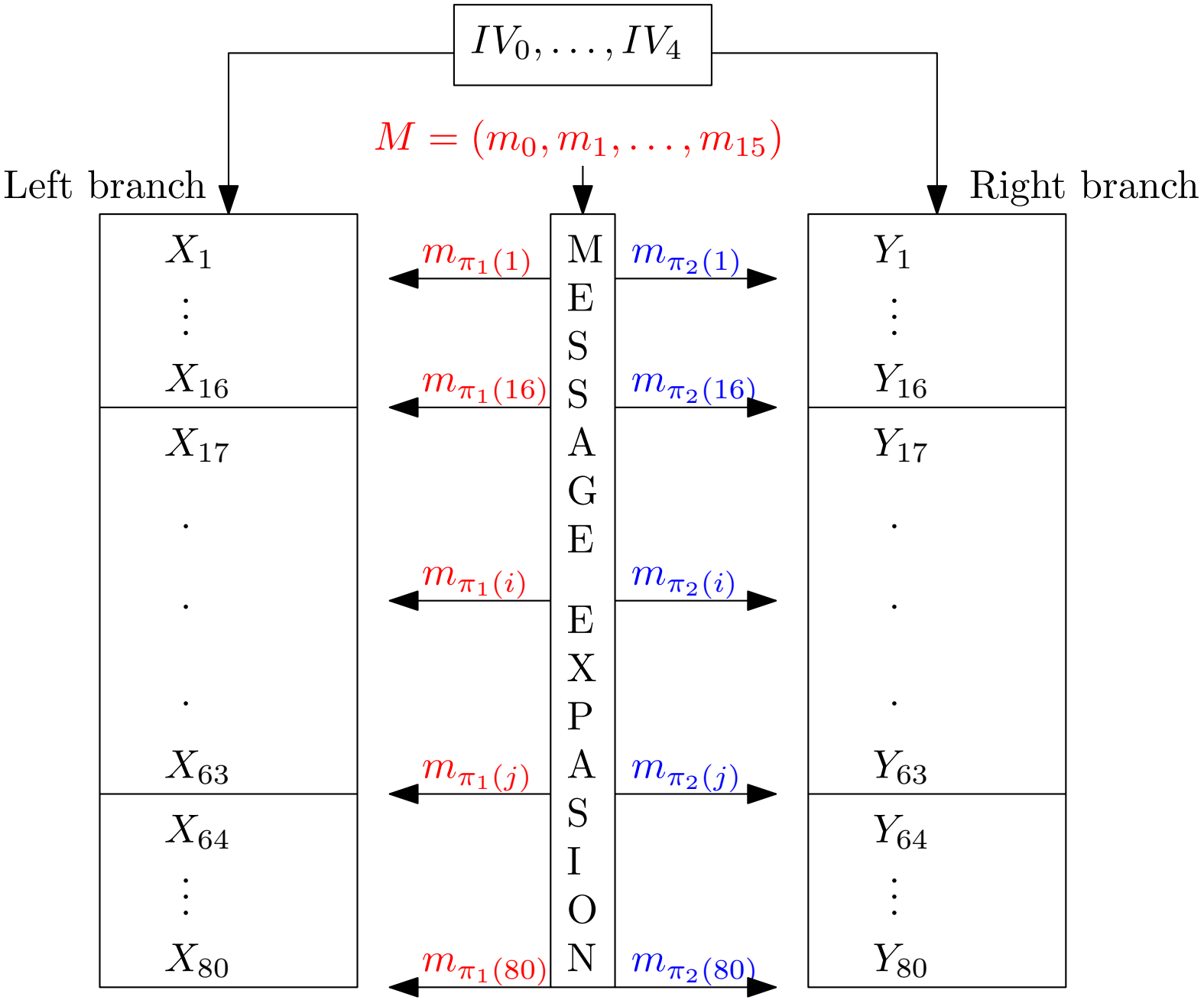
Steps	Time	Memory	Ref.
attaks starting from an intermediate step			
36/80	low	negligible	ISC'12
42/80	$2^{75.5}$	2^{64}	ASIACRYPT'13
48/80	$2^{76.4}$	2^{64}	ToSC 2017
attacks starting from the first step			
36/80	$2^{70.4}$	2^{64}	ASIACRYPT'13
36/80	$2^{55.1}$	2^{32}	ASIACRYPT'17
36/80	2^{41}	negligible	this work
37/80	2^{49}	negligible	this work
38/80	2^{52}	negligible	this work
40/80	$2^{74.6}$	negligible	this work

Cryptanalysis of RIPEMD-160

What we learned from the cryptanalysis of RIPEMD-160:

- **Progress has been made.**
- It is far from the full-round collision attack!!!

RIPEMD-160


$IV_0, \ldots, IV_4$
$M = (m_0, m_1, \ldots, m_{15})$
Left branch
Right branch
X_1
X_{16}
X_{17}
X_{63}
X_{64}
X_{80}
Y_1
Y_{16}
Y_{17}
Y_{63}
Y_{64}
Y_{80}
$m_{\pi_1(1)}$
$m_{\pi_1(16)}$
$m_{\pi_1(i)}$
$m_{\pi_1(j)}$
$m_{\pi_1(80)}$
$m_{\pi_2(1)}$
$m_{\pi_2(16)}$
$m_{\pi_2(i)}$
$m_{\pi_2(j)}$
$m_{\pi_2(80)}$
MESSAGE EXPASION

RIPEMD-160

Step function:

$$
\begin{aligned}
X_i = &\quad X_{i-4}^{\lll 10} \boxplus (X_{i-5}^{\lll 10} \boxplus \Phi_j^l(X_{i-1}, X_{i-2}, X_{i-3}^{\lll 10}) \boxplus m_{\pi_1(i)} \boxplus K_j^l)^{\lll s_i^l}, \\
Y_i = &\quad Y_{i-4}^{\lll 10} \boxplus (Y_{i-5}^{\lll 10} \boxplus \Phi_j^r(Y_{i-1}, Y_{i-2}, Y_{i-3}^{\lll 10}) \boxplus m_{\pi_2(i)} \boxplus K_j^r)^{\lll s_i^r},
\end{aligned}
$$

Finalization:

$$
\begin{aligned}
h_0' &= h_1 \boxplus X_{79} \boxplus Y_{78}^{\lll 10}, \\
h_1' &= h_2 \boxplus X_{78}^{\lll 10} \boxplus Y_{77}^{\lll 10}, \\
h_2' &= h_3 \boxplus X_{77}^{\lll 10} \boxplus Y_{76}^{\lll 10}, \\
h_3' &= h_4 \boxplus X_{76}^{\lll 10} \boxplus Y_{80}, \\
h_4' &= h_0 \boxplus X_{80} \boxplus Y_{79}.
\end{aligned}
$$

Previous Cryptanalysis of Reduced RIPEMD-160

- Procedure:
  - Step 1: Fix a solution for the dense part.
  - Step 2: Utilize free message words to merge both branches.
  - Step 3: Verify the remaining probabilistic part.

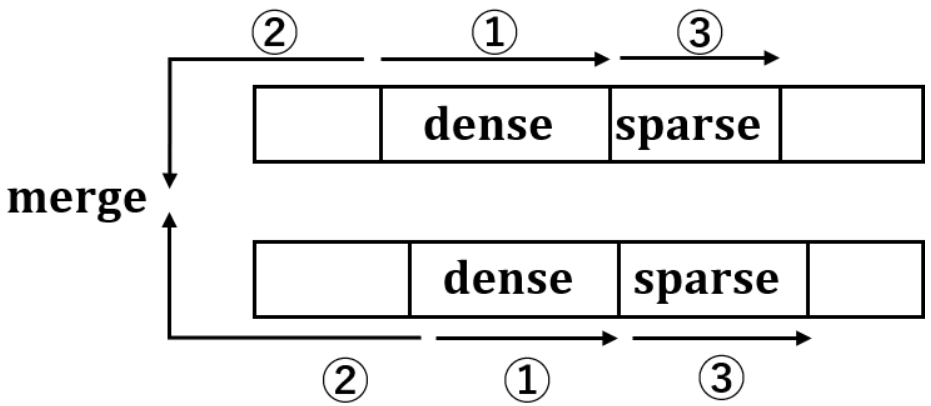


Figure: Previous framework for SFS start collision attacks

New Cryptanalysis of Reduced RIPEMD-160

- The common procedure to find collisions:
  - Step 1: Construct a differential characteristic.
  - Step 2: Fulfill the differential conditions.

★ Technical contribution of this paper:

- Efficient methods to fulfill the differential conditions.

Constructing a Differential Characteristic

- Reuse the pattern of the differential characteristic (CRYPTO'19)

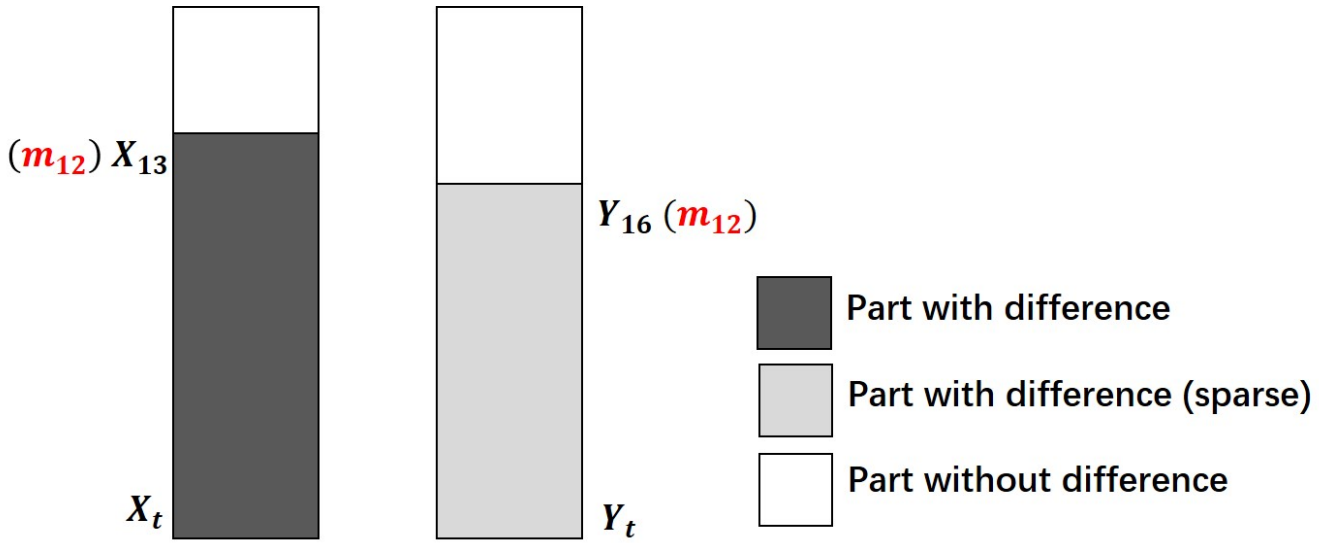


Figure: Attack on t steps of RIPEMD-160 by inserting difference at m_{12}

Fulfilling Differential Conditions

Observation on the message expansion:

X_{13}	X_{14}	X_{15}	X_{16}	X_{17}
m_{12}	m_{13}	m_{14}	m_{15}	m_7

X_{18}	X_{19}	X_{20}	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}	X_{27}	X_{28}	X_{29}	X_{30}	X_{31}	X_{32}
m_4	m_{13}	m_1	m_{10}	m_6	m_{15}	m_3	m_{12}	m_0	m_9	m_5	m_2	m_{14}	m_{11}	m_8

X_{33}	X_{34}	X_{35}	X_{36}	X_{37}	X_{38}	X_{39}	X_{40}
m_3	m_{10}	m_{14}	m_4	m_9	m_{15}	m_8	m_1

Figure: Partial information of the message expansion of RIPEMD-160

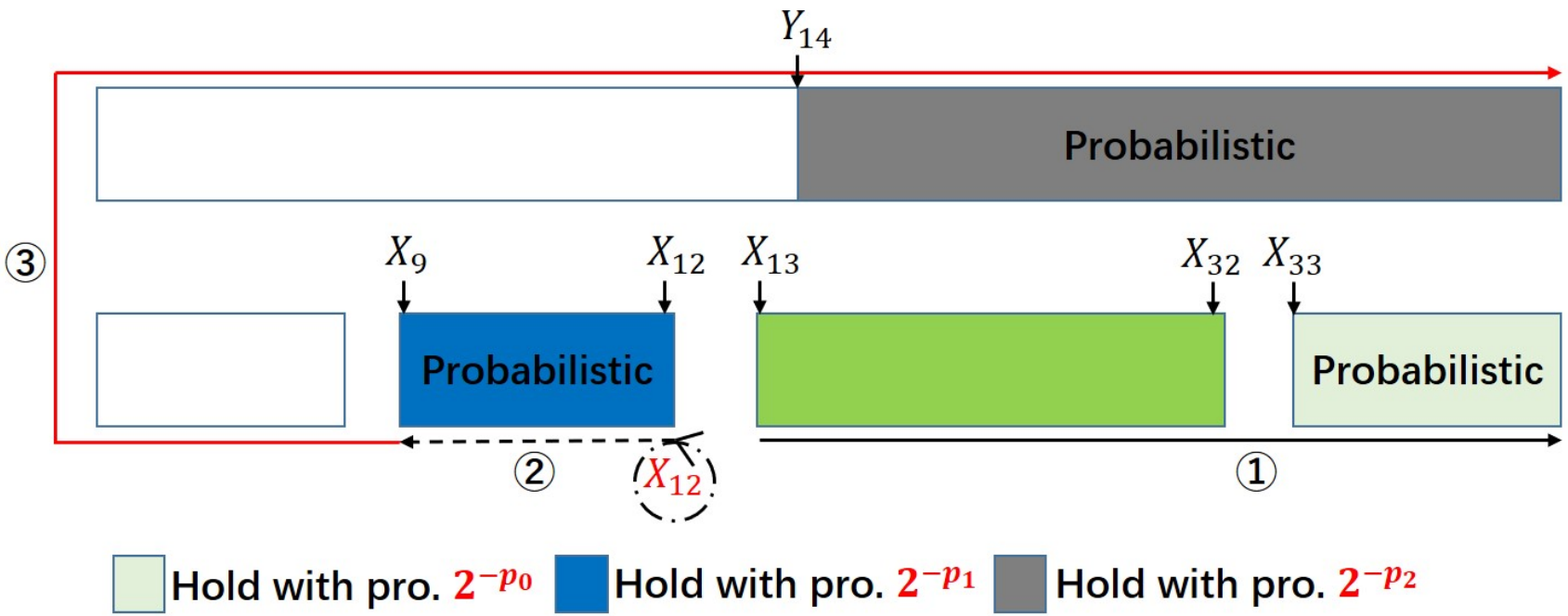
Observation

For the left branch, X_{17} is updated with m_7 in the second round. Besides, m_7 is used to update X_{42} in the third round.

Fulfilling Differential Conditions

The overall attack procedure to find t-step semi-free-start collisions:

1. Step 1: Find a starting point, i.e. a solution for $(X_{13}, \ldots, X_t)$.
2. Step 2: Filter invalid X_{12}, i.e. valid solutions for $(X_9, \ldots, X_{12})$.
3. Step 3: Verify the differential conditions on the right branch.



Fulfilling Differential Conditions

Efficiently re-generate a starting point from an existing one:

■ Strategy 1:

- Step 1: Modify $(X_{13}, \ldots, X_{15})$.
- Step 2: Update (m_4, m_{13}, m_1) to keep $(X_{16}, \ldots, X_{35})$ stay the same.
- Step 3: Recompute $(X_{36}, \ldots, X_t)$ and check their conditions.

X_{13}	X_{14}	X_{15}	X_{16}	X_{17}
m_{12}	m_{13}	m_{14}	m_{15}	m_7

X_{18}	X_{19}	X_{20}	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}	X_{27}	X_{28}	X_{29}	X_{30}	X_{31}	X_{32}
m_4	m_{13}	m_1	m_{10}	m_6	m_{15}	m_3	m_{12}	m_0	m_9	m_5	m_2	m_{14}	m_{11}	m_8

X_{33}	X_{34}	X_{35}	X_{36}	X_{37}	X_{38}	X_{39}	X_{40}
m_3	m_{10}	m_{14}	m_4	m_9	m_{15}	m_8	m_1

Fulfilling Differential Conditions

Efficiently re-generate a starting point from an existing one:

■ Strategy 2:

- Step 1: Modify (X_{14}, X_{15})
- Step 2: Compute X_{13} using $(m_4, X_{14}, \ldots, X_{18})$ and check conditions.
- Step 3: Update (m_{13}, m_1) to keep $(X_{16}, \ldots, X_{39})$ stay the same.

X_{13}	X_{14}	X_{15}	X_{16}	X_{17}
m_{12}	m_{13}	m_{14}	m_{15}	m_7

X_{18}	X_{19}	X_{20}	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}	X_{27}	X_{28}	X_{29}	X_{30}	X_{31}	X_{32}
m_4	m_{13}	m_1	m_{10}	m_6	m_{15}	m_3	m_{12}	m_0	m_9	m_5	m_2	m_{14}	m_{11}	m_8

X_{33}	X_{34}	X_{35}	X_{36}	X_{37}	X_{38}	X_{39}	X_{40}
m_3	m_{10}	m_{14}	m_4	m_9	m_{15}	m_8	m_1

Fulfilling Differential Conditions

Efficiently re-generate a starting point from an existing one:

■ Strategy 1 V.S. Strategy 2:

- 1: A few conditions on $(X_{36}, \ldots, X_t)$ (Use Strategy 1)
- 2: Many conditions on $(X_{36}, \ldots, X_t)$ (Use Strategy 2)

X_{13}	X_{14}	X_{15}	X_{16}	X_{17}
m_{12}	m_{13}	m_{14}	m_{15}	m_7

X_{18}	X_{19}	X_{20}	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}	X_{27}	X_{28}	X_{29}	X_{30}	X_{31}	X_{32}
m_4	m_{13}	m_1	m_{10}	m_6	m_{15}	m_3	m_{12}	m_0	m_9	m_5	m_2	m_{14}	m_{11}	m_8

X_{33}	X_{34}	X_{35}	X_{36}	X_{37}	X_{38}	X_{39}	X_{40}
m_3	m_{10}	m_{14}	m_4	m_9	m_{15}	m_8	m_1

Fulfilling Differential Conditions

Efficiently re-generate a starting point from an existing one:

- Benefits:
  - The cost almost has no influence on the whole complexity.
  - The whole complexity is dominated by the right branch.

!!! Regenerate a starting point only when X_{12} is fully traversed.

Desirable Differential Characteristics

To use our attack framework, a characteristic should satisify

- A few conditions on $(X_9, \ldots, X_{12})$.
- Not too many conditions on (X_{13}, X_{14}, X_{15}) .
- Not too many conditions on the right branch.

- The characteristic should be
- As sparse as possible in $(X_{13}, \ldots, X_{17})$.
- As sparse as possible in $(X_{36}, \ldots, X_t)$.
- As sparse as possible on the right branch.

The 36-Step Differential Characteristic

i	X	$\pi_1(i)$	Y	$\pi_2(i)$
13	----------n----------	12	----------	1
14	--nu----------	13	----------0----------	10
15	-n----------u-----	14	----------1----------	3
16	-0-0----------u-----110n----	15	----------n----------	12
17	n0----------1-1-0----1-110----	7	----------	6
18	0-------1-------01-----1nu-0--0-	4	----------1----------	11
19	---011011-011111u0----0n001--0-1	13	----------1----------	3
20	11010000------101n1-00-011010000	1	u----------	7
21	nnnn1nn101011111-1011nu110100u10	10	----------	0
22	1001-01nu1-01u0n1--1n1nuunu nnu1-	6	1----------0	13
23	nn110u11n10nu100n001nnn10u11nnu1	15	1----------1	5
24	1101nu0-10uun01nu1n0000nn1101u10	3	----------un----------	10
25	-1uu1nn1-n011001n0u01uuu1n101uuu	12	----------	14
26	1101011u1un0u10-u01uuuuuu001-010	0	----------0-01----------	15
27	0u11u0010011n1-1uuun001111000111	9	----------1-11----------	8
28	11000100n11nnn0n11100n-10n0n0nn1	5	---------n-un----------	12
29	01n00nu000u01nnu-01uuu-unununn11n	2	----------	4
30	n1u01n10u010011n000110-00000un1u	14	---------1--1----------	9
31	-1n100-nnn01-0101011-11-nnn0-10u	11	--1----------1	1
32	10n010-000--0-111110010-100--10u	8	--u----------n	2
33	--n----------001----0--1------u	3	--1----------1	15
34	--0----------0------1	10	------0--0----------	5
35	----------	14	------u--n----------	1
36	------n--u----------	4	----------n--u------	3

Practical SFS Collisions for 36/37-Step RIPEMD-160

Table: SFS collision for 36 steps of RIPEMD-160

$h_0 \sim h_4$	809825f7 d2a55861 6bd86be7 fc58a6cb 11f6a005
M	6c2c8526 dc3084cc 16188d15 c6c5da57 73f15b99 f7a7a97a a7cbbf38 53a4b30 b6477677 47f24a3e b1bdf3b5 78aaa252 69a579f0 72b32f35 bb877480 5caa647e
M'	6c2c8526 dc3084cc 16188d15 c6c5da57 73f15b99 f7a7a97a a7cbbf38 53a4b30 b6477677 47f24a3e b1bdf3b5 78aaa252 69a5f9f0 72b32f35 bb877480 5caa647e
hash value	88f79fa4 c9973719 dcf0ff7f 15cef816 a9d702a5

Table: SFS collision for 37 steps of RIPEMD-160

$h_0 \sim h_4$	51c683bc e9cd8258 75924d6d b31d5b2b 9f1418b8
M	2a3e3e5d 2f3acda8 c5ab4a9c dc1f16ce 695a6d71 848cc0fe f11aa5a3 65da8473 9e6914b7 fe96a9cf da48b5c6 59b4296f 14a47a10 c0870c31 3b3e4837 7f4d5b3f
M'	2a3e3e5d 2f3acda8 c5ab4a9c dc1f16ce 695a6d71 848cc0fe f11aa5a3 65da8473 9e6914b7 fe96a9cf da48b5c6 59b4296f 14a4fa10 c0870c31 3b3e4837 7f4d5b3f
hash value	4ba88e59 fe3d1b6d 92324a6e 124af3ea e0206481

Comparison

■ Advantage:

- A new simple efficient framework is proposed.
- The memory complexity is further reduced.

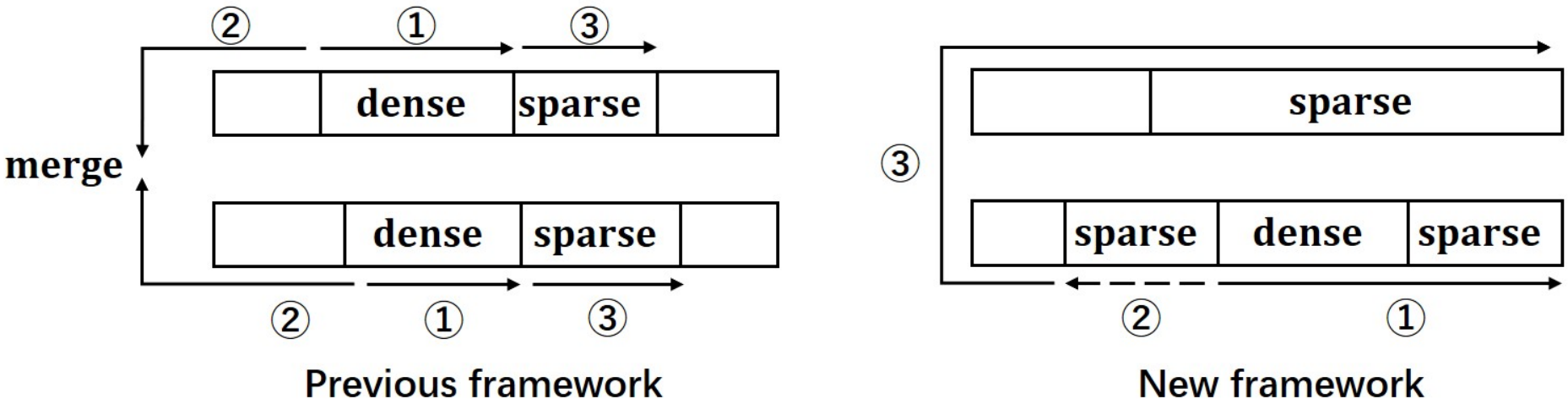


Figure: Comparison between our framework and previous frameworks

Summary

- Results:
  - Practical SFS collision attacks on 36/37-step RIPEMD-160.
  - Theoretical SFS collision attacks up to 40 steps.

Steps	Time	Memory	Ref.
36/80	2^{41}	negligible	this work
37/80	2^{49}	negligible	this work
38/80	2^{52}	negligible	this work
40/80	$2^{74.6}$	negligible	this work

Table: Semi-free-start collision attacks on reduced RIPEMD-160

Thank you