IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2023, No. 4, pp. 112–142. DOI:10.46586/tosc.v2023.i4.112-142

Automating Collision Attacks on RIPEMD-160

Yingxin Li[1], Fukang Liu[2] and Gaoli Wang[1](✉)

[1] Shanghai Key Laboratory of Trustworthy Computing,
East China Normal University, Shanghai, China
liyx1140@163.com, glwang@sei.ecnu.edu.cn
[2] Tokyo Institute of Technology, Tokyo, Japan
liu.f.ad@m.titech.ac.jp

Abstract. As an ISO/IEC standard, the hash function RIPEMD-160 has been used to generate the Bitcoin address with SHA-256. However, due to the complex double-branch structure of RIPEMD-160, the best collision attack only reaches 36 out of 80 steps of RIPEMD-160, and the best semi-free-start (SFS) collision attack only reaches 40 steps. To improve the 36-step collision attack proposed at EUROCRYPT 2023, we explored the possibility of using different message differences to increase the number of attacked steps, and we finally identified one choice allowing a 40-step collision attack. To find the corresponding 40-step differential characteristic, we re-implement the MILP-based method to search for signed differential characteristics with SAT/SMT. As a result, we can find a colliding message pair for 40-step RIPEMD-160 in practical time, which significantly improves the best collision attack on RIPEMD-160. For the best SFS collision attack published at ToSC 2019, we observe that the bottleneck is the probability of the right-branch differential characteristics as they are fully uncontrolled in the message modification. To address this issue, we utilize our SAT/SMT-based tool to search for high-probability differential characteristics for the right branch. Consequently, we can mount successful SFS collision attacks on 41, 42 and 43 steps of RIPEMD-160, thus significantly improving the SFS collision attacks. In addition, we also searched for a 44-step differential characteristic, but the differential probability is too low to allow a meaningful SFS collision attack.

Keywords: Semi-free-start collision · collision · RIPEMD-160 · SAT/SMT

1 Introduction

As components of cryptographic primitives, hash functions are important for building secure systems. Generally, a hash function takes an arbitrarily long message as input and outputs a fixed-length hash value of size n bits. Three fundamental security properties of a hash function are collision resistance, preimage resistance and second-preimage resistance. Since 2005, many hash functions in the MD-SHA hash family have been broken, including MD4 [WLF+05], MD5 [WY05], SHA-0 [WYY05b, BCJ+05], SHA-1 [WYY05a, LP20, SBK+17, LP19] and RIPEMD-128 [LP13]. However, the security of RIPEMD-160 and SHA-2 has not been compromised. Especially, RIPEMD-160 is an ISO/IEC standard that is now used to generate the Bitcoin address with SHA-256. In this sense, further studying the security of RIPEMD-160 is meaningful.

RIPEMD-160 has a complex double-branch structure, which causes the slow progress of the collision attack. The first collision attack on RIPEMD-160 presented at ASIACRYPT 2017 [LMW17] only reached 30 steps with a time complexity of 2^{70}. Subsequently at CRYPTO 2019 [LDM+19a], two different collision attack frameworks were proposed, namely, the dense-left-and-sparse-right (DLSR) framework and the sparse-left-and-dense-right (SLDR) framework. Based on the DLSR framework, the practical 30/31-step collision

Received: 2023-09-01 Accepted: 2023-11-01 Published: 2023-12-08

attacks and the theoretic 34-step collision attack were achieved for the first time. At EUROCRYPT 2023 [LWS+23], a new strategy to choose the message differences was proposed, based on which the collision attack on 36-step RIPEMD-160 was achieved.

For the SFS collision attack on RIPEMD-160, the first SFS security analysis was presented at ISC 2012 [MNSS12], including practical examples of SFS near-collisions for 48 steps and SFS collisions for 36 steps, where the two attacks start from an intermediate step. The first major improvement was achieved at ASIACRYPT 2013 [MPS+13], where the authors presented two results: a 42-step SFS collision attack starting from an intermediate step with time complexity of $2^{75.5}$ and a 36-step SFS collision attack starting from the first step with time complexity of $2^{70.4}$. Then, a 48-step SFS collision attack starting from an intermediate step was presented at ToSC 2017 [WSL17]. Moreover, at the ASIACRYPT 2017 [LMW17], the complexity of the 36-step SFS collision attack in [MPS+13] was further improved to $2^{55.1}$. Another major progress was made at ToSC 2019 [LDM+19b], where the first practical SFS collision attack on 36/37-step RIPEMD-160 starting from the first step was achieved, and the best attack could reach 40 steps with time complexity of $2^{74.6}$. It is noted that the whole time complexity of the SFS collision attacks in [LDM+19b] is almost dominated by the probability of the right-branch differential characteristics. However, the right-branch differential characteristics are deduced by hand and whether they are optimal is unknown. Thus, it becomes important to study this problem in order to further increase the number of attacked steps.

To mount (SFS) collision attacks on RIPEMD-160, or more generally, the MD-SHA hash family, it is essential to first search for signed differential characteristics. While this problem has been efficiently solved with the guess-and-determine technique [CR06, MNS11, MNS12, MNS13, MPS+13, EMS14, DEM15], the corresponding tools are not open-source. This has motivated the authors of [LWS+23] to create a new MILP-based tool that is both open-source and easy-to-use. In particular, all the details to write this MILP-based tool are provided in [LWS+23], which makes it easy to re-implement it with other languages such as SAT/SMT.

To increase the diversity of automatic tools, we re-implement the MILP-based tool proposed at EUROCRYPT 2023 [LWS+23] with SAT/SMT. This SAT/SMT-based tool will be used in all our attacks and it has excellent performance to search for RIPEMD-160 differential characteristics. We do not treat this as a main contribution, but it enriches the available tools pool.

Our contributions. The contributions of this paper are summarized as below:

1. We shed new insight into the collision attack on RIPEMD-160. Specifically, we are able to propose the first practical colliding message pair for 40-step RIPEMD-160, improving the previously best theoretic collision attacks at EUROCRYPT 2023 [LWS+23] by 4 steps;

2. To improve the SFS attacks on RIPEMD-160, we utilize our SAT/SMT-based tool to find the most sparse differential characteristics for the right branch. In this way, we are able to find high-probability differential characteristics for the right branch, which allows us to address the above mentioned issue to improve the attacks.

3. Based on the newly found 41/42/43-step differential characteristics, we could obtain the first SFS collision attacks on 41, 42 and 43 steps of RIPEMD-160 with time complexity of $2^{59.7}$, $2^{67.3}$ and $2^{74.8}$, respectively. This is the first time to mount SFS collision attacks on more than half of the total steps (80 steps) of RIPEMD-160 starting from the first step.

We also attempted to attack 44-step RIPEMD-160, but the probability of the right-branch differential characteristic is too low to allow a successful SFS attack in the classical setting. Our results for RIPEMD-160 are summarized in Table 1.

Table 1: Summary of attacks on RIPEMD-160

Attack type	Steps	Time	Memory	References
preimage	34	$2^{158.91}$	\	[WS14]
Distinguishing	43	2^{151}	\	[WLC+20]
	52*	2^{151}	\	[WLC+20]
SFS collision	48*	$2^{76.5}$	2^{64}	[WSL17]
	36/37	*practical*	*negligible*	[LDM+19b]
	40	$2^{74.6}$	*negligible*	[LDM+19b]
	41	$2^{59.7}$	*negligible*	Sect.5.4
	42	$2^{67.3}$	*negligible*	Sect.5.4
	43	$2^{74.8}$	*negligible*	Sect.5.4
collision	30/31	*practical*	*practical*	[LP19]
	34	$2^{74.3}$	2^{32}	[LP19]
	36	$2^{64.5}$	*negligible*	[LWS+23]
	40	*practical*	*negligible*	Sect.4.3

* An attack starts at an intermediate step.

The source code to find the (SFS) collisions differential characteristics for RIPEMD-160 is available at https://github.com/Peace9911/ripemd160_attack.git

Organization. This paper is organized as follows. The notation and description of RIPEMD-160 are given in Section 2. Then, we revisit the MILP-based method to search for signed differential characteristics for RIPEMD-160 in Section 3. Next, we describe the collision attack on 40-step RIPEMD-160 in Section 4. In Section 5, we show how to improve the SFS collision attack with newly discovered differential characteristics. Finally, the paper is concluded in Section 6.

2 Preliminaries

2.1 Notations

For a better understanding of this paper, we introduce the following notations.

1. $\boxplus$ and $\boxminus$ represent modular addition and modular subtraction on 32 bits, respectively.
2. $\ll$, $\gg\!\!\gg$, $\lll$, $\oplus$, $\neg$, $\vee$ and $\wedge$ represent *shift left*, *rotate right*, *rotate left*, *exclusive or*, *not*, *or*, and *and*, respectively.
3. $x[i]$ denotes the i-th bit of x and $x[0]$ is the least significant bit.
4. δx denotes the modular difference, i.e. $\delta x = x' \boxminus x$.
5. Δx denotes the signed difference between x' and x. We use the notation as follows,

$$\Delta x[i] = \begin{cases} \texttt{n} & (x[i]=0, x'[i]=1) \\ \texttt{u} & (x[i]=1, x'[i]=0) \\ \texttt{=} & (x[i]=x'[i]) \\ \texttt{0} & (x[i]=x'[i]=0) \\ \texttt{1} & (x[i]=x'[i]=1) \end{cases} \tag{1}$$

6. ϕ_j^l and ϕ_j^r represent the 32-bit Boolean function at the left and right branches for round j, respectively.
7. K_j^l and K_j^r represent the constants used at the left and right branches for round j, respectively.
8. s_i^l and s_i^r represent the rotation constants for the left and right branches of step i, respectively.
9. π_i^l and π_i^r represent the index of the message word for the left and right branches of step i, respectively.
10. $M = (m_0, m_1, \cdots, m_{15})$ and $M' = (m'_0, m'_1, \cdots, m'_{15})$ represent two 512-bit message blocks, respectively.
11. X_i and Y_i represent the 32-bit internal state of the left and right branches updated during step i for compressing M, respectively.
12. LQ_i and RQ_i represent the 32-bit temporary states of the left and right branches updated in step i for compressing M, respectively.
13. The Hamming weight of the signed difference Δx is denoted by $\mathbf{H}(\Delta x)$ and $\mathbf{H}(\Delta x)$ is the number of indices i such that $\Delta x[i] \in \{\mathtt{n}, \mathtt{u}\}$ [LWS+23].

2.2 Description of RIPEMD-160

Dobbertin et al. proposed a 160-bit hash function RIPEMD-160 at FSE 1996 [DBP96] as a stronger hash function than RIPEMD [BP95]. The compression function of RIPEMD-160 denoted by $H(CV, M)$ uses a 512-bit message block M and a 160-bit chaining value CV as inputs and generates a 160-bit output.

For our collision attacks on RIPEMD-160, we aim to find two message blocks (M_0, M_1) and (M_0, M'_1) such that

$$H\left(H(CV_0, M_0), M_1\right) = H\left(H(CV_0, M_0), M'_1\right)$$

where $M_1 \neq M'_1$ and CV_0 is a prefixed constant.

For the SFS collision for RIPEMD-160, we need to find a pair (M, M') satisfying

$$M \neq M' \quad H(CV, M) = H(CV, M'),$$

where CV can be any 160-bit constant.

On the compression function $H(CV, M)$. The compression function H consists of 80 steps, divided into 5 rounds of 16 steps each in both branches. The input M is composed of 16 message words $(m_0, m_1, \ldots, m_{15})$ and CV is divided into five 32-bit words $(h_0, \ldots, h_4)$. Especially, we have

$$X_{-5} = h_0^{\ggg 10}, X_{-4} = h_4^{\ggg 10}, X_{-3} = h_3^{\ggg 10}, X_{-2} = h_2, X_{-1} = h_1$$
$$Y_{-5} = h_0^{\ggg 10}, Y_{-4} = h_4^{\ggg 10}, Y_{-3} = h_3^{\ggg 10}, Y_{-2} = h_2, Y_{-1} = h_1$$

For the prefixed constant CV_0, the corresponding $(h_0, \ldots, h_5)$ are as follows:

$$\begin{aligned} h_0 &= \mathtt{0x67452301}, \\ h_1 &= \mathtt{0xEFCDAB89}, \\ h_2 &= \mathtt{0x98BADCFE}, \\ h_3 &= \mathtt{0x10325476}, \\ h_4 &= \mathtt{0xC3D2E1F0}. \end{aligned}$$

Table 2: Boolean functions and round constants in RIPEMD-160

Round j	ϕ_j^l	ϕ_j^r	K_j^l	K_j^r	Function	Expression
0	*XOR*	*ONX*	0x00000000	0x50a28be6	*XOR(x, y, z)*	$x \oplus y \oplus z$
1	*IFX*	*IFZ*	0x5a827999	0x5c4dd124	*IFX(x, y, z)*	$(x \wedge y) \oplus (\neg x \wedge z)$
2	*ONZ*	*ONZ*	0x6ed9eba1	0x6d703ef3	*ONX(x, y, z)*	$x \oplus (y \vee \neg z)$
3	*IFZ*	*IFX*	0x8f1bbcdc	0x7a6d76e9	*IFZ(x, y, z)*	$(x \wedge z) \oplus (y \wedge \neg z)$
4	*ONX*	*XOR*	0xa953fd4e	0x00000000	*ONZ(x, y, z)*	$(x \vee \neg y) \oplus z$

The step function of RIPEMD-160 at step i is shown below:

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

where $i = (0, 1, 2, ..., 79)$ and $j = (0, 1, 2, 3, 4)$. The details of the Boolean functions and round constants for RIPEMD-160 are displayed in Table 2. The other parameters can be found in the specification [DBP96].

After iterating the step function for 80 steps, the 160-bit output $(h'_0, h'_1, \ldots, h'_4)$ is computed as follows:

$$\begin{aligned} h'_0 &= h_1 \boxplus X_{78} \boxplus Y_{77}^{\lll 10}, \\ h'_1 &= h_2 \boxplus X_{77}^{\lll 10} \boxplus Y_{76}^{\lll 10}, \\ h'_2 &= h_3 \boxplus X_{76}^{\lll 10} \boxplus Y_{75}^{\lll 10}, \\ h'_3 &= h_4 \boxplus X_{75}^{\lll 10} \boxplus Y_{79}, \\ h'_4 &= h_0 \boxplus X_{79} \boxplus Y_{78}. \end{aligned}$$

3 Finding RIPEMD-160 Differential Characteristics

Recently, at EUROCRYPT 2023 [LWS+23], a novel MILP-based method to search for signed differential characteristics has been proposed. The main motivation behind that work [LWS+23] is to create an easy-to-use and open-source tool for the MD-SHA hash family. To increase the diversity of such open-source tools, we re-implement the MILP-based method with SAT/SMT, i.e. all constraints will be described with conjunctive normal form (CNF) rather than linear inequalities. The implementation details will be omitted from this paper as they generally follow the pseudo-code given for the MILP-based method in [LWS+23].

3.1 The Automatic Method in [LWS+23]

For completeness, we first briefly revisit the technique in [LWS+23] to search for RIPEMD-160 differential characteristics with MILP. In our new implementation with SAT/SMT, we are following the same idea.

Specifically, the form of the step function of RIPEMD-160 can be described as below:

$$d_{i+5} = (d_{i+1}^{\lll 10}) \boxplus (F(d_{i+4}, d_{i+3}, d_{i+2}^{\lll 10}) \boxplus (d_i^{\lll 10}) \boxplus m \boxplus c_i)^{\lll s},$$

where $(d_i, \ldots, d_{i+5}, m)$ are all 32-bit variables, c is a 32-bit constant, $s \in [0, 31]$ is an integer and F is a Boolean function.

Denote the signed difference of $(d_i, \ldots, d_{i+5}, m)$ by $(\Delta d_i, \ldots, \Delta d_{i+5}, \Delta m)$. Then, each of $(\Delta d_i, \ldots, \Delta d_{i+5}, \Delta m)$ can be represented with a vector of size 32. In this sense, it is only required to study the following step function because the rotation ($\lll 10$) only affects the order of variables:

$$a_5 = a_1 \boxplus (F(a_4, a_3, a_2) \boxplus a_0 \boxplus m \boxplus c)^{\lll s}. \tag{2}$$

With some intermediate 32-bit variables $(b_0, \ldots, b_5)$, Equation 2 can be further decomposed as:

$$\begin{aligned} b_0 &= m \boxplus c, \\ b_1 &= F(a_4, a_3, a_2), \\ b_2 &= b_0 \boxplus b_1, \\ b_3 &= b_2 \boxplus a_0, \\ b_4 &= b_3^{\lll s}, \\ b_5 &= a_1 \boxplus b_4, \\ a_5 &= b_5. \end{aligned}$$

In [LWS+23], the authors described how to model the signed difference transitions through the step function, i.e. how to use constraints to describe the propagation:

$$(\Delta a_0, \ldots, \Delta a_4, \Delta m) \to \Delta a_5.$$

In particular, the model can be briefly summarized as follows:

- Model the deterministic signed difference addition

  $$\Delta z = \Delta x \boxplus \Delta y.$$

  Specifically, although we indeed have many possible Δz for a given $(\Delta x, \Delta y)$, we only consider one valid Δz. This is based on the feature of the step function of the MD-SHA hash family, and it indeed also follows the way to deduce such a differential characteristic by hand.

- Model the signed difference transitions for the Boolean function F, i.e.

  $$(\Delta a_4, \Delta a_3, \Delta a_2) \to \Delta b_1.$$

  This is captured by the so-called *fast filtering model* for F in [LWS+23]

- Model the signed difference transitions for $\Delta z = 0 \boxplus \Delta z'$, i.e. this is called *modelling the expansion of the modular difference*. In other words, for a given $\Delta z'$, how to compute all possible Δz such that they correspond to the same modular difference.

- Model the update $a_5 = a_1 \boxplus b_3^{\lll s}$. The authors [LWS+23] introduced two different ways to model it, i.e. *the first strategy* and *the second strategy*, such that the model can handle as many cases as possible.

However, simply using the above models is insufficient because contradictions easily occur, especially in the Boolean function. Hence, they introduced the so-called monitoring variable, which can be used to monitor whether contradictions occur in the difference transitions through the Boolean functions over different steps. Roughly speaking, by using three additional variables (a_4, a_3, a_2) and constructing another model only to capture the relations between $(\Delta a_4, \Delta a_3, \Delta a_2, \Delta b_1)$ and (a_4, a_3, a_2), it is possible to detect the contradictions in the Boolean functions over different steps. In [LWS+23], if (a_4, a_3, a_2) is involved, it is called the *full model* for F.

Another place where contradictions occur is at the operation

$$a_5 = a_1 \boxplus b_3^{\lll s},$$

especially when the conditions on (a_5, a_1) are dense. This is a special operation in RIPEMD-160 and makes it more difficult to find valid signed differential characteristics. Detecting the contradictions in this operation is a bit complex and we refer the interested readers to [LWS+23] for more details.

4 New Collision Attacks on RIPEMD-160

With the automatic tool at hand, we first show how to use it to significantly improve the collision attacks on RIPEMD-160. In particular, the currently best collision attack [LWS+23] only reaches 36 out of 80 steps of RIPEMD-160, and it has a time complexity of $2^{64.5}$. In what follows, we show a practical collision attack on 40-step RIPEMD-160 and give the corresponding colliding message pair. This is the first time to practically violate the collision resistance of half of the full-round RIPEMD-160. Note that the currently best SFS collision attack on RIPEMD-160 only reaches 40 steps with a time complexity of $2^{74.6}$ [LDM+19b]. Consequently, our new collision attack also updates the best SFS collision attack on RIPEMD-160.

4.1 Choosing New Message Differences

Our new collision attack relies on a new way to choose the message differences. First, we revisit the collision attack on 36-step RIPEMD-160 in [LWS+23], and we generalize their way to construct the 36-step differential characteristic. As shown in Figure 1, there are 3 places to construct local collisions:

- the first local collision spans from X_{i_0} to X_{i_1} where $i_0 < i_1 < 16$;
- the second local collision spans from X_{i_2} to X_{i_3} where $16 < i_2 < i_3 < 32$;
- the third local collision spans from Y_{i_4} to Y_{i_5} where $0 < i_4 < i_5 < 32$;

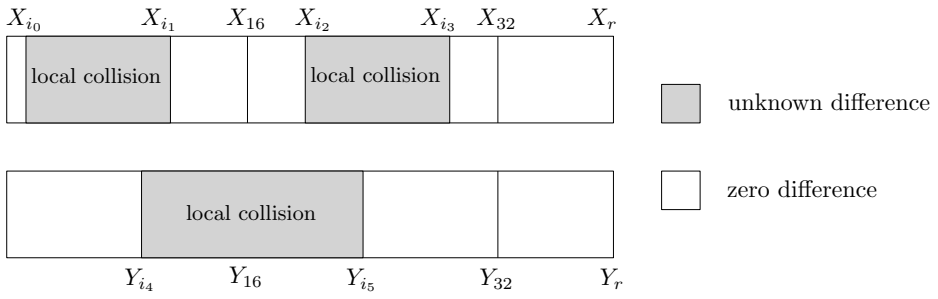

Figure 1: The pattern of the RIPEMD-160 differential characteristic.

In [LWS+23], the differences are injected in (m_0, m_6, m_9), which results in

$$i_0 = 0,\ i_1 = 9,\ i_2 = 21,\ i_3 = 26,\ i_4 = 3,\ i_5 = 29.$$

Since (m_0, m_6, m_9) are not used to update the internal states X_i and Y_i where $32 \leq i \leq 35$, a 36-step collision-generating differential characteristic can possibly be constructed by injecting message differences in these 3 message words. For such a way to construct a differential characteristic, the authors of [LWS+23] also proposed an efficient message

modification technique. In brief, the cost to fulfill all differential conditions on $(X_i)_{0\leq i\leq 9}$ and $(Y_i)_{0\leq i\leq 12}$ can be amortized, and the degrees of freedom in (Y_{13}, Y_{15}) can be utilized to fulfill the remaining uncontrolled differential conditions. Roughly speaking, the number of differential conditions on $(X_i)_{16\leq i\leq 35}$ and $(Y_i)_{16\leq i\leq 35}$ dominate the whole time complexity of the attack.

Based on the above analysis, if we aim to mount a collision attack based on a differential characteristic of a similar shape, we need to ensure

- the differential characteristic of the second local collision should be as sparse as possible;
- the message words to inject differences should be used to update the internal states (X_i, Y_i) as late as possible where $i \geq 32$;

Our strategy for the second local collision. To mount a collision attack on $r+1$ ($r \geq 36$) steps of RIPEMD-160, we first utilize our SAT/SMT-based tool to find a better choice of the message differences. Specifically, we can build a very simple model to search for the differential characteristic of the second local collision where the message words to inject differences are not allowed to update X_i and Y_i where $32 \leq i \leq r$. In this way, we find 3 possible ways to construct the second local collision, as shown in Table 3.

Table 3: Three ways to construct the second local collision

message words	(i_2, i_3)	number of conditions	r	attacked steps
(m_0, m_6, m_8, m_{11})	$(21, 31)$	8	37	38
$(m_0, m_2, m_{11}, m_{12})$	$(24, 30)$	8	39	40
(m_0, m_{12}, m_{13})	$(18, 25)$	44	41	42

Although the results indicate that a 42-step collision attack is possible, we could not find an efficient message modification for the corresponding differential characteristic due to a large number of differential conditions in the second local collision. Hence, we target the 40-step collision attack by injecting message differences in $(m_0, m_2, m_{11}, m_{12})$.

4.2 Finding the 40-Step Differential Characteristic

To ensure the second local collision and the minimal number of differential conditions on it, $(\delta m_0, \delta m_2, \delta m_{11}, \delta m_{12})$ should satisfy:

$$\delta m_0 = 0 \boxminus 2^i,\ \delta m_2 = 0 \boxminus 2^{(i+4)\%32},\ \delta m_{11} = 2^{(i+22)\%32},\ \delta m_{12} = 2^{(i+30)\%32},$$

where $0 \leq i \leq 31$.

To further optimize the whole time complexity of the collision attack, i.e., we expect that $\sum_{i=15}^{31} \Delta\mathbf{H}(Y_i)$ is also as small as possible, after several trials, we eventually determined the following message differences:

$$\delta m_0 = 0 \boxminus 2^{17},\ \delta m_2 = 0 \boxminus 2^{21},\ \delta m_{11} = 2^7,\ \delta m_{12} = 2^{15}.$$

With the above message differences, we give a high-level description of how to utilize our tool to search for the corresponding 40-step differential characteristic, as shown below:

Step 1: Find a valid solution of $(\Delta X_i)_{0\leq i\leq 12}$ to ensure $(\delta X_i = 0)_{8\leq i\leq 12}$, and we minimize $\sum_{i=0}^{7} \mathbf{H}(\Delta X_i)$.

Step 2: Find a valid solution of $(\Delta Y_i)_{11\leq i\leq 31}$.

Step 3: Choose a sparse differential characteristic manually for $(\Delta Y_i)_{3\le i\le 5}$ and fix it. Find a valid solution of $(\Delta Y_i)_{6\le i\le 10}$ to connect $(\Delta Y_i)_{3\le i\le 5}$ and $(\Delta Y_i)_{11\le i\le 31}$.

To improve the efficiency of the message modification technique, we have tried three strategies for the **Step 2**, as detailed below:

Strategy 1: Directly find a valid solution of $(\Delta Y_i)_{11\le i\le 31}$ to ensure $(\delta Y_i = 0)_{25\le i\le 31}$, and we minimize $\sum_{i=11}^{31} \mathbf{H}(\Delta Y_i)$.

Strategy 2: First, find a valid solution of $(\Delta Y_i)_{16\le i\le 31}$ such that $(\delta Y_i = 0)_{25\le i\le 31}$, and we minimize $\sum_{i=16}^{31} \mathbf{H}(\Delta Y_i)$.

Then, find a valid solution of $(\Delta Y_i)_{11\le i\le 15}$ to connect $(\Delta Y_i)_{16\le i\le 31}$, and we minimize $\sum_{i=11}^{15} \mathbf{H}(\Delta Y_i)$.

Strategy 3: First, find a valid solution of $(\Delta Y_i)_{15\le i\le 31}$ such that $(\delta Y_i = 0)_{25\le i\le 31}$, and we minimize $\sum_{i=15}^{31} \mathbf{H}(\Delta Y_i)$.

Then, find a valid solution of $(\Delta Y_i)_{11\le i\le 14}$ to connect $(\Delta Y_i)_{15\le i\le 31}$, and we minimize $\sum_{i=11}^{14} \mathbf{H}(\Delta Y_i)$.

It is found that we can benefit more from Strategy 3. The corresponding 40-step differential characteristic is displayed in Table 4. Some extra conditions for the differential characteristic are shown in Table 5.

4.3 Finding Conforming Message Pairs

For our 40-step collision attack, two message blocks (M_0, M_1) will be used. Specifically, our goal is to find a tuple (M_0, M_1, M_1') where $M_1 \neq M_1'$ such that

$$CV_1 = H(CV_0, M_0), \quad H(CV_1, M_1) = H(CV_1, M_1').$$

This is mainly because in our 40-step differential characteristic, there are some conditions on CV_1. The general procedure to find the conforming message pair for the 40-step differential characteristic is summarized as follows:

Step 1: Find a valid M_0 such that the conditions on CV_1 can hold, i.e., the conditions on $(X_{-5}, X_{-4}, X_{-3}, X_{-2}, X_{-1})$ in the 40-step differential characteristic can hold.

Step 2: Similar to [MZ06], use an SAT/SMT model to describe the value transitions for RIPEMD-160. By adding the differential conditions on the internal states to the model, we can then find a valid solution of $(X_i)_{0\le i\le 9}$ and $(Y_i)_{0\le i\le 12}$ satisfying all the corresponding conditions by solving the model. For convenience, this solution is called the starting point[1] for the collision attack.

Step 3: Reuse the degrees of freedom of (Y_{10}, Y_{11}) to generate more starting points. Specifically, although (Y_{10}, Y_{11}) have been fixed at **Step** 2, we can traverse all their possible values, recompute (X_8, X_9, Y_{12}), and check whether the conditions on them hold. In this way, we can reduce the workload of the SAT/SMT solver to generate many such solutions at **Step** 1−2. For each starting point, move to the next step. Return to **Step** 1 if all starting points are used up.

Step 4: Traverse all possible values of Y_{13}, compute Y_{14} and check the differential conditions on $(Y_{14}, LQ_{10}, LQ_{11})$. If they hold, move to the next step.

[1] Abusing the notation, in our SFS collision, we will also define the starting point as the solution of different internal states.

Table 4: The 40-step differential characteristic for RIPEMD-160, where $\delta m_0 = 0 \boxminus 2^{17}, \delta m_2 = 0 \boxminus 2^{21}, \delta m_{11} = 2^7, \delta m_{12} = 2^{15}$

i	ΔX_i	m_l^i	i	ΔY_i	m_r^i
-5	================================		-5	================================	
-4	================================		-4	================================	
-3	================================		-3	================================	
-2	================================		-2	================================	
-1	================================		-1	================================	
0	unnn============================	0	0	================================	5
1	===============nuuuu=n==========	1	1	================================	14
2	u=uun=u==========n==u====un=nnnn	2	2	============0==================	7
3	=====nnn===unun==u==u====nn=====	3	3	0==u============================	0
4	u=u==uu==u=================n=nu	4	4	0====1==========0==1=n==1===010	9
5	===========nuuu=n======u=n=====	5	5	101====u==0=00=1===0000=100u0000	2
6	==u====nuu======================	6	6	0110=1===nnuu1nuuuuuuuuuu10100=0	11
7	===============unnnnnnnnnn=====	7	7	1unnnnn11000unn00unn10nunn11=110	4
8	================================	8	8	=1011nu001nu111nuu=unnn0101nuuuu	13
9	================================	9	9	00u==nu00u010===1000101u=0101n0=	6
10	================================	10	10	111====0==u=n10=0u01=1n01=010==1	15
11	================================	11	11	0=0=n1=0=10n0===u====n1=1===0===	8
12	================================	12	12	11u===10=0=1u=0======1==0=1u===0	1
13	================================	13	13	==0======1==0=n=10=0====1=10===n	10
14	================================	14	14	==1====0========u1=1==========u	3
15	================================	15	15	=======1=n=============n=======	12
16	================================	7	16	============================u==	6
17	================================	4	17	================================	11
18	================================	13	18	================================	3
19	================================	1	19	====================0==========	7
20	================================	10	20	====================1==========	0
21	================================	6	21	==========u===================	13
22	================================	15	22	================================	5
23	================================	3	23	==========1===================	10
24	========n=======================	12	24	==========1============010000=	14
25	==u======0=======================	0	25	=u======================111111=	15
26	==0============================1	9	26	==============nuuuuu==========	8
27	======================1=======	5	27	======1=========================	12
28	================================	2	28	======0=========================	4
29	================================	14	29	================================	9
30	================================	11	30	================================	1
31	================================	8	31	================================	2
32	================================	3	32	================================	15
33	================================	10	33	================================	5
34	================================	14	34	================================	1
35	================================	4	35	================================	3
36	================================	9	36	================================	7
37	================================	15	37	================================	14
38	================================	8	38	================================	6
39	================================	1	39	================================	9

$Y_{15}[10] = Y_{14}[10], Y_{15}[27] = Y_{14}[27], Y_{16}[10] = Y_{15}[10], Y_{16}[25] = Y_{15}[25]$
$Y_{17}[0] = Y_{16}[0], Y_{17}[17] = Y_{16}[17], Y_{18}[12] = Y_{17}[12], Y_{23}[30] = Y_{22}[30],$
$Y_{27}[8] = Y_{26}[8], Y_{28}[i] = Y_{27}[i] (i \in \{21, 22, 23, 24, 26\})$
$X_{23}[22] = X_{22}[12], X_{24}[29] = X_{23}[19]$

Table 5: Some extra conditions for the 40-step differential characteristic

Conditions on LQ_i and RQ_i:
$(LQ_i \boxplus in_i^l)^{\lll s_i^l} = LQ_i{}^{\lll s_i^l} \boxplus out_i^l$
$(RQ_i \boxplus in_i^r)^{\lll s_i^r} = RQ_i{}^{\lll s_i^r} \boxplus out_i^r$

i	in_i^l	out_i^l	π_l^i	s_l^i	i	in_i^r	out_i^r	π_r^i	s_r^i
0	0xfffe0000	0xf0000000	0	11	0	0x0	0x0	5	8
1	0x50000000	0x1400	1	14	1	0x0	0x0	14	9
2	0x6fdeac00	0x560037ef	2	15	2	0x0	0x0	7	9
3	0x86006f5b	0x6f5b860	3	12	3	0xfffe0000	0xf0000000	0	11
4	0x4ace0003	0x59c00049	4	5	4	0x10000000	0x200	9	13
5	0x9fffb13f	0xffb13fa0	5	8	5	0xffdffe00	0xfeffffff0	2	15
6	0x4fbec08d	0xdf6046a8	6	7	6	0x1000090	0x480080	11	15
7	0xe1948f3f	0x291e7fc4	7	9	7	0xffefef90	0xfdfdf200	4	5
8	0xd31ffffc	0xfffde99	8	11	8	0x204107c	0x2083e01	13	7
9	0xffffd80c	0xfb018000	9	13	9	0xfc3807e	0xe1c03f08	6	7
10	0x1fc0000	0x7f	10	14	10	0xfedfe5c2	0xdfe5c1ff	15	8
11	0x1	0x8000	11	15	11	0x81020911	0x10488408	8	11
12	0x0	0x0	12	6	12	0x101e7800	0x9e000408	1	14
13	0x0	0x0	13	7	13	0x217f7f8	0xfdfe0086	10	14
14	0x0	0x0	14	9	14	0xfff60f78	0x60f77fff	3	12
15	0x0	0x0	15	8	15	0x7f08c001	0xc2300060	12	6
16	0x0	0x0	7	7	16	0x3e100020	0x2000407c	6	9
17	0x0	0x0	4	6	17	0xdfffc000	0xf7fffc00	11	13
18	0x0	0x0	13	8	18	0x8000400	0x2000400	3	15
19	0x0	0x0	1	13	19	0xfdfffc00	0xfffdffff	7	7
20	0x0	0x0	10	11	20	0x1	0x1000	0	12
21	0x0	0x0	6	9	21	0xfffff000	0xfff00000	13	8
22	0x0	0x0	15	7	22	0x0	0x0	5	9
23	0x0	0x0	3	15	23	0x0	0x0	10	11
24	0x8000	0x400000	12	7	24	0x0	0x0	14	7
25	0xfffe0000	0xe0000000	0	12	25	0x0	0x0	15	7
26	0x0	0x0	9	15	26	0x80000000	0x800	8	12
27	0x0	0x0	5	9	27	0x0	0x0	12	7
28	0xffe00000	0xffffffff	2	11	28	0x0	0x0	4	6
29	0x1	0x80	14	7	29	0x2000000	0x100	9	15
30	0x0	0x0	11	13	30	0xffffff00	0xffe00000	1	13
31	0x0	0x0	8	12	31	0x0	0x0	2	11
32	0x0	0x0	3	11	32	0x0	0x0	15	9
33	0x0	0x0	10	13	33	0x0	0x0	5	7
34	0x0	0x0	14	6	34	0x0	0x0	1	15
35	0x0	0x0	4	7	35	0x0	0x0	3	11
36	0x0	0x0	9	14	36	0x0	0x0	7	8
37	0x0	0x0	15	9	37	0x0	0x0	14	6
38	0x0	0x0	8	13	38	0x0	0x0	6	6
39	0x0	0x0	1	15	39	0x0	0x0	9	14

Step 5: Traverse all possible values of Y_{15} and compute the corresponding m_{12}. Then, all message words $(m_i)_{0\leq i\leq 15}$ are fixed. Check the remaining uncontrolled differential conditions. If all of them hold, a colliding message pair is found and exits. Otherwise, move to **Step** 4.

Based on the above procedure, we found the first colliding message pair for 40-step RIPEMD-160, as shown in Table 6. The whole procedure takes about 16 hours with 115 threads. A theoretic analysis of the time complexity is given below.

Table 6: The colliding message pair (M_0, M_1) and (M_0, M_1') for 40 steps of RIPEMD-160

M_0	4b1de304	f52a5a3e	bbd7d814	6454a1d6	a5571007	6c4151f5	8970f768	32c48fd1
	54c428ea	113b00cf	3db1bb85	1d2b2de6	89157118	89157118	d22f990b	6db9f321
M_1	0a179ed0	582e9fee	8c68cd3d	0d120a6e	de43af57	df2e7a6f	2b40967e	df302947
	ee7f066f	d7b7707d	9f1cc8a9	eaecfcb8	0b449f1a	ec058b69	996ee0d2	994ef6b1
M_1'	0a159ed0	582e9fee	8c48cd3d	0d120a6e	de43af57	df2e7a6f	2b40967e	df302947
	ee7f066f	d7b7707d	9f1cc8a9	eaecfd38	0b451f1a	ec058b69	996ee0d2	994ef6b1
hash	a76b7982	e39826f9	52eb6b63	6b48ecdd	4ddca6c5			

Complexity evaluation. There are only 4 bit conditions on $(X_i)_{-5\leq i\leq -1}$ and hence Step 1 takes about time 2^4. Furthermore, it is found that the cost to find a starting point by simply using the SAT/SMT solver is equivalent to about $2^{32.6}$ calls of RIPEMD-160. We found in total 100 such starting points with the SAT/SMT solver. Then, for each such starting point, we reuse the degrees of freedom in (Y_{10}, Y_{11}) to generate more starting points. We randomly chose 80 out of 100 starting points and generated in total about 1000 starting points with the degrees of freedom in (Y_{10}, Y_{11}) in a few minutes on a single core. Based on each of these starting points, we further utilize the degree of freedom in (Y_{13}, Y_{15}) to satisfy the remaining uncontrolled differential conditions. Note that the time complexity to fulfill the conditions on $(Y_{14}, LQ_{10}, LQ_{11})$ can be amortized because there are sufficiently many free bits in Y_{15}. The total time complexity is almost dominated by the conditions on (X_i, Y_i) where $i \geq 16$, which hold with a probability of about $2^{-49.9}$. Theoretically, the time complexity of our attack is $2^{49.9}$.

5 Improved SFS Collision Attacks on RIPEMD-160

After improving the collision attacks on RIPEMD-160 by 4 steps, we feel interested whether it is possible to further utilize this automatic tool to improve the SFS collision attack on RIPEMD-160. In particular, we aim to improve the SFS collision attack published at ToSC 2019 [LDM+19b], where the authors could only attack at most 40 steps of RIPEMD-160 with their technique.

5.1 Finding New Differential Characteristic for SFS Collision Attacks

Our improved SFS collision attack on RIPEMD-160 still follows the attack framework proposed in [LDM+19b], as shown in Figure 2.

In this framework, the message difference is only injected at the message word m_{12}, and the right-branch differential characteristic should be as sparse as possible because the time complexity of the SFS collision attack is almost dominated by its probability. However, in the previous SFS collision attacks on RIPEMD-160 [LDM+19b], the right-branch differential characteristics are deduced by hand and whether they are optimal is unknown. In particular, in their 40-step SFS collision attack, the probability of the right-branch differential characteristic is $2^{-74.6}$, which makes it infeasible to further extend the attack for more steps.

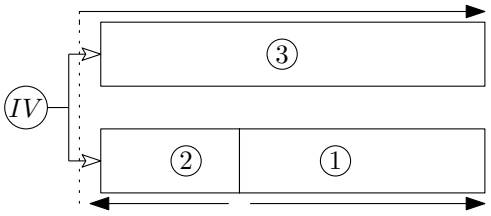

Figure 2: SFS collision attack framework

Intuitively, if better right-branch differential characteristics can be found, the SFS collision attack can be improved. Hence, we feel interested whether it is possible to find such differential characteristics with the new automatic tool to increase the number of attacked steps because it is relatively easy to solve optimization problems with these tools.

In the following, we give the details of how we use the automatic tool to find left/branch differential characteristics for $t+1$ steps of RIPEMD-160 where $t \geq 40$.

Finding right-branch differential characteristics. Finding $(\Delta Y_{15}, \cdots, \Delta Y_t)$ is simply done with the SAT/SMT-based tool. Specifically, for each possible t, we set the objective function as minimizing $\sum_{i=15}^{i=t} \mathbf{H}(\Delta Y_i)$.

The authors of [LDM+19b] pointed out that the conditions on the right branch will influence the whole time complexity. To ensure a valid attack, the probability of the right-branch differential characteristic should be higher than 2^{-80}. Experimental results indicate that the minimal values of $\sum_{i=15}^{t} \mathbf{H}(\Delta Y_i)$ are 15, 18 and 20 for $t = 40$, $t = 41$, and $t = 42$, respectively, and the corresponding right-branch differential characteristics hold with probability higher than 2^{-80}. However, when $t = 43$, the minimal value of $\sum_{i=15}^{i=43} \mathbf{H}(\Delta Y_i)$ is 22, and the corresponding differential characteristic holds with a probability smaller than 2^{-80}. Therefore, we could possibly perform SFS collision attacks on 41, 42 and 43 steps of RIPEMD-160 under the attack framework [LDM+19b]. This is because the final time complexity is indeed affected by several factors and they should be analyzed in a more careful way.

Finding left-branch differential characteristics. After determining the right-branch differential characteristics, we need to find the corresponding left-branch differential characteristics such that the differences $(\Delta X_{t-5}, \ldots, \Delta X_t)$ should cancel the differences $(\Delta Y_{t-5}, \ldots, \Delta Y_t)$ to allow an SFS collision attack on $t+1$ steps of RIPEMD-160. For this purpose, we first find the solutions of $(\Delta X_{12}, \ldots, \Delta X_{20})$ and $(\Delta X_{35}, \ldots, \Delta X_t)$, respectively, and make them as sparse as possible. This can be achieved by setting the objective functions of the SAT/SMT-based tool as minimizing $\sum_{i=12}^{20} \mathbf{H}(\Delta X_i)$ and $\sum_{i=35}^{t} \mathbf{H}(\Delta X_i)$, respectively. Then, we find a valid solution of $(\Delta X_{21}, \ldots, \Delta X_{34})$ to connect $(\Delta X_{12}, \cdots, \Delta X_{20})$ and $(\Delta X_{35}, \cdots, \Delta X_t)$.

The new differential characteristics. The corresponding 41/42/43/44-step differential characteristics are given in Table 7, Table 8, Table 9 and Table 10, respectively.

5.2 The General Message Modification Technique

We mainly use a similar strategy proposed in [LDM+19b] to perform the message modification, as the shape of the 41/42/43-step differential characteristics is almost the same as the 40-step one in [LDM+19b].

Specifically, it consists of two phases, and a graphic illustration for the strategy is given in Figure 3.

Table 7: The 41-step differential characteristic, where $\delta m_{12} = 2^{15}$.

i	ΔX_i	m_l^i	i	ΔY_i	m_r^i
-5	================================		-5	================================	
-4	================================		-4	================================	
-3	================================		-3	================================	
-2	================================		-2	================================	
-1	================================		-1	================================	
0	================================	0	0	================================	5
1	================================	1	1	================================	14
2	================================	2	2	================================	7
3	================================	3	3	================================	0
4	================================	4	4	================================	9
5	================================	5	5	================================	2
6	================================	6	6	================================	11
7	================================	7	7	================================	4
8	================================	8	8	================================	13
9	================================	9	9	================================	6
10	================================	10	10	================================	15
11	================================	11	11	================================	8
12	==========n=====================	12	12	================================	1
13	==nu============================	13	13	====================0==========	10
14	=n========0===============u=====	14	14	====================1==========	3
15	10=0========00====u=====110u====	15	15	==========n====================	12
16	n1==============1=0====1=1=0====	7	16	================================	6
17	0=======1====1===11====0=n=1====	4	17	==========1====================	11
18	1==1====0========0====1n=0======	13	18	==========1====================	3
19	==1u====u======1=u====00=1======	1	19	u===============================	7
20	===0====0====100=0====n1========	10	20	================================	0
21	===1===1=====1=n=1====0==1====1=	6	21	1==============================0	13
22	=1=====0====1n00===0==1==1==1===	15	22	1==============================1	5
23	==10=1=u=0==00=1==1====10u======	3	23	====================un=========	10
24	===11010=n00n1====u====u=0======	12	24	================================	14
25	=1110n11=0==0=01==1=100010=01110	0	25	====================001=========	15
26	01111100100111101=11000=101000100	9	26	====================111=========	8
27	1000011100unnnnn0nuuuu0uuuunuu01	5	27	==========nnn===================	12
28	nnnnnu=uuu111100nuu1nuuuuuuuuu=u	2	28	================================	4
29	101unn=1010uuun001=nuuu000nnu010	14	29	===========1====================	9
30	=1111n0110110=000011010u1u110110	11	30	==1=============================	1
31	=000001n==100100u=1==000=0u0==u0	8	31	==u=============================	2
32	011=1=11====n00=n===1u111n11100=	3	32	==1=============================	15
33	01n=n0n===000=10u===00=1=1======	10	33	==1======0====================	5
34	001=0=0===n===n=1====u=u1=010===	14	34	========n=============1=======	1
35	======n===1===1======1=1===u====	4	35	========1=============u======1	3
36	1===0=0====1=n==================	9	36	========1=============10=====0	7
37	============0===0=======0======	15	37	========0================n=====1	14
38	====================0===u====0=	8	38	========n=====01=========1=====n	6
39	====================u===1====u=	1	39	========1=====u0==============1	9
40	=============n=================	2	40	===u============================	11

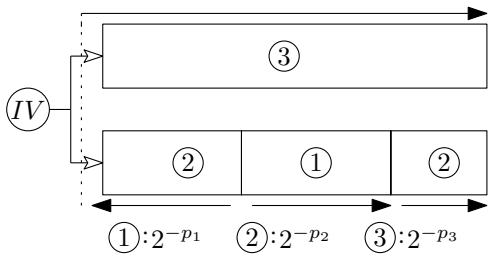

Figure 3: The general message modification technique

Table 8: The 42-step differential characteristic, where $\delta m_{12} = 2^{15}$.

i	ΔX_i	π_l^i	i	ΔY_i	π_r^i
-5	================================		-5	================================	
-4	================================		-4	================================	
-3	================================		-3	================================	
-2	================================		-2	================================	
-1	================================		-1	================================	
0	================================	0	0	================================	5
1	================================	1	1	================================	14
2	================================	2	2	================================	7
3	================================	3	3	================================	0
4	================================	4	4	================================	9
5	================================	5	5	================================	2
6	================================	6	6	================================	11
7	================================	7	7	================================	4
8	================================	8	8	================================	13
9	================================	9	9	================================	6
10	================================	10	10	================================	15
11	================================	11	11	================================	8
12	==========n=====================	12	12	================================	1
13	==un============================	13	13	====================0===========	10
14	=n========1===============n=====	14	14	====================1===========	3
15	00=0=========1====n=====110n====	15	15	==========n====================	12
16	u0==============1=0====1=1=0====	7	16	================================	6
17	0=======1====1===11====1=u=1====	4	17	==========1====================	11
18	1==1====0========0====0n=0======	13	18	==========1====================	3
19	==1n====n======1=n====00=1======	1	19	u===============================	7
20	===0====0===01=0=0====u0========	10	20	================================	0
21	===1===11====1=u=1====0==1====1=	6	21	1==============================0	13
22	=======0====0n=0===0==1==0====0=	15	22	1==============================1	5
23	====01=n=0==00=1========n====n=	3	23	====================un=========	10
24	===1=0=0=n==u0==========0====0=	12	24	================================	14
25	1==00u=0=0==0==1====100001===11=	0	25	====================001=========	15
26	==0n=0===1=1110011010=1=0110=1=0	9	26	====================111=========	8
27	01011100100nuuuu=10nu=n=1=u10n=0	5	27	==========nnn==================	12
28	1nuu1=nuu=1011un0n000001n000=0=n	2	28	================================	4
29	uu01u110n1n=1001n0u1=unnnnnn=10u	14	29	============0===================	9
30	000n00110=011001101u010nn1001011	11	30	==1=============================	1
31	=1n0n=0n01010011uu10=1=00u===00=	8	31	==u====================0=======	2
32	1u111011uu1u=00n00=n=0==010==0==	3	32	==1====================u=======	15
33	u100=uu001=1=111===0=0==n11==n10	10	33	==1====================0=======	5
34	01==1010=0=u==0====n==110u=00100	14	34	===0==================1=======	1
35	====u======0===1===1==0==01==n==	4	35	===n==========1=========u=======	3
36	====0=0=======00=========n===0==	9	36	===1==========u=========11======	7
37	======u========1=========1===1==	15	37	===1==========0====0====10======	14
38	======1=======00===1==1==0==1===	8	38	==============1====n==0==1======	6
39	====1=======1==1=01u========0===	1	39	====1====1====u====1==n==n======	9
40	====n1======u==u===1============	2	40	====u====0====1=======1==1======	11
41	====n============================	7	41	============================n===	8

Table 9: The 43-step differential characteristic, where $\delta m_{12} = 2^{15}$.

i	ΔX_i	m_l^i	i	ΔY_i	m_r^i
-5	================================		-5	================================	
-4	================================		-4	================================	
-3	================================		-3	================================	
-2	================================		-2	================================	
-1	================================		-1	================================	
0	================================	0	0	================================	5
1	================================	1	1	================================	14
2	================================	2	2	================================	7
3	================================	3	3	================================	0
4	================================	4	4	================================	9
5	================================	5	5	================================	2
6	================================	6	6	================================	11
7	================================	7	7	================================	4
8	================================	8	8	================================	13
9	================================	9	9	================================	6
10	================================	10	10	================================	15
11	================================	11	11	================================	8
12	==========n=====================	12	12	================================	1
13	==nu============================	13	13	====================0===========	10
14	=n========0===============u=====	14	14	====================1===========	3
15	10=0==============u=====110n====	15	15	==========n====================	12
16	u1==============1=0====1=1=0====	7	16	================================	6
17	0=======1====1===11====0=n=1====	4	17	==========1==================	11
18	1==1====0========0====1n=0======	13	18	==========1==================	3
19	===u====u======1=n0===00=1======	1	19	u===============================	7
20	===0====0====1=0=1====u1========	10	20	================================	0
21	=======1n====1=n=1====0==1====1=	6	21	1==============================0	13
22	=======01===1n=0===0==1=====0===	15	22	1==============================1	5
23	=====1=n10==00=1==0==========0=	3	23	====================un=========	10
24	===1=0=0=n==u=====u==========0=	12	24	================================	14
25	=010=n11=0==0===10111010000101n=	0	25	====================001========	15
26	==1==10110000110101O0=1110=00010	9	26	====================111========	8
27	100=00nuuuuuuuuuuuuu11uuu0uun00	5	27	==========nnn==================	12
28	10u1nn1n0n00u0nu1u01110100=uun=1	2	28	================================	4
29	un1000u1000nnu10u01nuunuu0=11u1n	14	29	============1===================	9
30	1110n10n=1n1001u010100011n0000=n	11	30	==1=============================	1
31	1101110u=0000100===0=u=0010==0=0	8	31	==u=============================	2
32	1=0=01=1===1u0=0===u=0==uu1=11=u	3	32	==1=============================	15
33	0=n0=n1=0==01u11==110==000u=u0=0	10	33	==1======0======================	5
34	==10=u==n1==01u0==u=u110u101u=1=	14	34	=========n==============1=======	1
35	0=n=11==1011=n1=1=n=10n=1u=0n=11	4	35	=========1==============u=======	3
36	1=010=0=1=0n00u===00u=1=10==1=uu	9	36	=========1==============10======	7
37	===0==n=0110n=1===0=1=n=0=====10	15	37	========0================u=====1	14
38	==1===1=1u0=1=======0=0=====1==u	8	38	========n=====0==========1=====n	6
39	==0=====u1====0============0==u	1	39	===0====1====1u======1====1===11	9
40	========1=====n======1===0====10	2	40	===u====0=====1=10===0========0=	11
41	==============1======0===n====0=	7	41	===0====n=======uu==============	8
42	========u=======nn==============	0	42	====================n=========n	12

Table 10: The 44-step differential characteristic, where $\delta m_{12} = 2^{15}$.

i	ΔX_i	π_l^i	i	ΔY_i	π_r^i
-5	================================		-5	================================	
-4	================================		-4	================================	
-3	================================		-3	================================	
-2	================================		-2	================================	
-1	================================		-1	================================	
0	================================	0	0	================================	5
1	================================	1	1	================================	14
2	================================	2	2	================================	7
3	================================	3	3	================================	0
4	================================	4	4	================================	9
5	================================	5	5	================================	2
6	================================	6	6	================================	11
7	================================	7	7	================================	4
8	================================	8	8	================================	13
9	================================	9	9	================================	6
10	================================	10	10	================================	15
11	================================	11	11	================================	8
12	==========n=====================	12	12	================================	1
13	==un============================	13	13	====================0===========	10
14	=u========1===============n=====	14	14	====================1===========	3
15	00=0==============n=====110n====	15	15	==========n====================	12
16	u01=============1=0====1=1=0====	7	16	================================	6
17	1=======1====0===1=====11u=1====	4	17	==========1====================	11
18	1==0===0========0====0un0======	13	18	==========1====================	3
19	===n===========1=n====0001===0==	1	19	u===============================	7
20	===0=========110=0====u=1====n==	10	20	================================	0
21	===1===1======0u=0====0==1===0==	6	21	1============================0	13
22	=======1====1=u0===1==1==0===1==	15	22	1============================1	5
23	=====0=n=1==0=01===0=====n======	3	23	====================un========	10
24	====10=0=n==u======n=====0======	12	24	================================	14
25	001=0u10=00=0==1===1=00110=11110	0	25	====================001========	15
26	1=0==1==00011001010011001=0=0100	9	26	====================111========	8
27	n1001101000nuuuu1nuuuuuun0n0nn1=	5	27	==========nnn===================	12
28	nun1uuuun=n11110u010nu11011011n0	2	28	================================	4
29	0nuu1111n1n1nn1nu=1nn01=unuu==uu	14	29	============1===================	9
30	1010=01=0110n0=011=00u0111n01101	11	30	==1=============================	1
31	0=10=u0=n0n101101u==01=1111=1=0n	8	31	==u=============================	2
32	==11=10=1=10==u100==uu===u=1u=10	3	32	==1=============================	15
33	1=1n1u=1==10=10==01000==00=01=n1	10	33	==1======0====================	5
34	===1u0=u=10==1=11uu10u==n100n=1=	14	34	========n=============1=======	1
35	===u10101u=1==1=100=n0==00=111=1	4	35	=========1==============u=======	3
36	===0=0u01010==00u=1=1n=====u101=	9	36	=======011========1=====10======	7
37	=====11=n=01====11n==0=====11nn=	15	37	========1================u===111	14
38	=====u0=1==0=====001=0=====u001=	8	38	========n===============10==nuu	6
39	=====1==0========1=0=u====01==0=	1	39	===0====1====1===========u==111	9
40	===0=============0===1====n0====	2	40	===u====0=======00========1===01	11
41	===n==00========1=========1===0=	7	41	===0====u=======uu===1==========	8
42	===1==nn========0=============n=	0	42	========1=======11===n==========	12
43	====================u==========	6	43	===========n====================	2

$Y_{17}[31] = Y_{16}[31], Y_{20}[9] = Y_{21}[9], Y_{25}[20, 19] = Y_{24}[20, 19], Y_{29}[31, 30, 29] = Y_{28}[31, 30, 29]$,
$Y_{33}[7] \vee \neg Y_{32}[7] = 1, Y_{36}[0] \vee \neg Y_{35}[0] = 1, Y_{37}[17] \vee \neg Y_{36}[17] = 1$,
$Y_{39}[16] \vee \neg Y_{38}[16] = 1, Y_{40}[10] \vee \neg Y_{39}[10] = 1, Y_{40}[11] \vee \neg Y_{39}[11] = 1$,
$Y_{40}[12] \vee \neg Y_{39}[12] = 1, Y_{42}[6] \vee \neg Y_{41}[6] = 1$

Phase 1: Find a valid solution of $(X_{12}, \ldots, X_{40})$. For convenience, this solution is called a starting point for the SFS collision attack. For this starting point, all message words except m_7 are fixed. We present partial information of the message expansion, as illustrated in Figure 4.

Phase 2: Verify the remaining uncontrolled parts by exhausting all valid values of X_{11}. The underlying reason is that after X_{11} is fixed, m_7 will be fixed for each starting point. This phase can be more efficient via an early-abort strategy. Repeat this phase with another starting point if all valid values of X_{11} are used.

The details of Phase 1 and Phase 2 are specified below.

Efficiently generating more starting points. We observe that in the 41/42/43-step differential characteristics, the conditions on $(X_{24}, \ldots, X_{38})$ are dense. With this observation in mind, we can generate an initial starting point as follows. For better understanding, we refer the readers to Figure 4 when reading this part.

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

X_{17}	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}
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_{32}	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	m_2

Figure 4: Partial information of the message expansion of RIPEMD-160

Step 1: Find a solution for $(X_{24}, \ldots, X_{38})$ such that all the conditions on them hold. This can be similarly done with the model to describe the value transitions. After this step, the message words

$$(m_{11}, m_8, m_3, m_{10}, m_{14}, m_9, m_{15})$$

are fixed.

Step 2: Then, we utilize the available degrees of freedom in (m_2, m_5) to fulfill the conditions on (X_{23}, X_{22}, X_{21}). It can be found that there are only a few conditions on (X_{23}, X_{22}, X_{21}) and hence there are many possible choices of (m_2, m_5).

Step 3: Next, we use (m_0, m_{12}) to fulfill the conditions on $(X_{20}, X_{19}, X_{18}, X_{17})$ and there are also many possible values of (m_0, m_{12}) due to the sparsity in these 4 states.

Step 4: Then, we use m_6 to fulfill the conditions on (X_{16}, X_{15}), and use m_1 to fulfill the conditions on (X_{14}, X_{39}, X_{40}).

Step 5: Finally, we use m_{13} to fulfill the conditions on (X_{13}, X_{12}).

The main reason to give such a detailed procedure to find the initial starting point is to better understand the available number of initial starting points, which will be important to the 43-step attack. Especially, from what follows, it will become clear that we are interested in the possible number of solutions for $(X_{15}, \ldots, X_{38})$ because we can efficiently generate new starting points from it. Due to the sparsity in (X_{15}, X_{23}), i.e. the sufficiently many available degrees of freedom in $(m_2, m_5, m_0, m_{12}, m_6)$, we can expect to generate

many solutions of $(X_{15}, \ldots, X_{38})$ with the above method, and this number will be much larger than 2^{32} by simply counting the conditions on (X_{23}, X_{22}, X_{21}).

After the initial starting point is generated, with the technique in [LDM+19b], we can generate more starting points from it in a much more efficient way:

Step 1: Keep (X_{15}, X_{16}, X_{17}) unchanged. Randomly choose a valid value of (X_{13}, X_{14}) and recompute X_{12} as follows:

$$X_{12} = ((X_{17} \boxminus X_{13}^{\lll 10})^{\ggg s_{17}^l} \boxminus IFX(X_{16}, X_{15}, X_{14}^{\lll 10}) \boxminus m_4 \boxminus K_1^l)^{\ggg 10}$$

Then, check the conditions on $(X_{12}, LQ_{16}, LQ_{17}, LQ_{18})$. If they do not hold, randomly choose a new valid value of (X_{13}, X_{14}) and repeat until they hold.

Step 2: Modify m_{13} and m_1 to keep X_{18} and X_{19} unchanged:

$$\begin{aligned} m_{13} &= (X_{18} \boxminus X_{14}^{\lll 10})^{\ggg s_{18}^l} \boxminus IFX(X_{17}, X_{16}, X_{15}^{\lll 10}) \boxminus X_{13}^{\lll 10} \boxminus K_1^l, \\ m_1 &= (X_{19} \boxminus X_{15}^{\lll 10})^{\ggg s_{19}^l} \boxminus IFX(X_{18}, X_{17}, X_{16}^{\lll 10}) \boxminus X_{14}^{\lll 10} \boxminus K_1^l. \end{aligned}$$

In this way, $(X_{15}, \ldots, X_{38})$ will be kept the same as in the initial starting point. However, (X_{39}, X_{40}) should be updated as X_{39} is computed from m_1. Therefore, we need to further check whether the conditions on (X_{39}, X_{40}) hold. If not, we need to move to Step 1 to use a different value of (X_{13}, X_{14}).

Verifying the uncontrolled part. For Phase 2, we utilize the available degrees of freedom in m_7 to fulfill the remaining uncontrolled conditions. The details are as follows:

Step 1: Assume that there are n_1 bit conditions on X_{11}. In this way, we can exhaust 2^{32-n_1} possible values of X_{11} in total. For each possible value of X_{11}, compute m_7 as follows:

$$m_7 = (X_{16} \boxminus X_{12}^{\lll 10})^{\ggg 7} \boxminus IFX(X_{15}, X_{14}, X_{13}^{\lll 10}) \boxminus X_{11}^{\lll 10} \boxminus K_0^l.$$

In this way, all message words are fixed. Then, we first verify the remaining uncontrolled conditions on the left branch via the early-abort strategy. Specifically, we first check the conditions on $(X_{41}, \ldots, X_t)$ since X_{41} is updated with m_7. Then, compute backward until X_8 and check the conditions on

$$(X_{11}, X_{10}, LQ_{12}, LQ_{13}, LQ_{14}, LQ_{15}).$$

If these conditions hold, move to the next step. Otherwise, choose another possible value for X_{11} and repeat.

Step 2: Compute backwards to obtain $(X_{-5}, \ldots, X_{-1})$. Then, compute all the internal states on the right branch. If the conditions on the right branch do not hold, move to Step 1. Otherwise, an SFS collision is found.

Differences between this work and [LDM+19b]. The general idea of this work is the same as [LDM+19b]. However, the right-branch differential characteristics are not optimal in [LDM+19b] as they are deduced by hand, i.e. by experience. We addressed this issue by using the recently proposed MILP/SAT/SMT-based tools to search for optimal differential characteristics for the right branch. As the number of attacked steps increases, the general message modification also slightly differs from [LDM+19b]. Specifically, to efficiently generate more starting points, we further need to check the conditions on (X_{39}, X_{40}). When verifying the remaining uncontrolled conditions on the left branch, we also additionally need to check the conditions on $(X_{41}, \ldots, X_t)$. It is unclear whether these 2 extra probabilistic parts will affect the whole time complexity, and hence it should be carefully analyzed.

5.3 Evaluating the Time Complexity

First, we emphasize that generating the initial starting point can be finished in practical time. In our attacks, it is expected that only a few initial starting points are sufficient because we can generate many more starting points from one initial starting point in an efficient way. Hence, the time complexity to generate the initial starting points is negligible.

Second, we evaluate the cost to generate new starting points from the initial starting point. In this procedure, we will exhaust all possible values of (X_{13}, X_{14}) and then check the conditions on

$$(X_{12}, LQ_{16}, LQ_{17}, LQ_{18}, X_{39}, X_{40}, LQ_{39}, LQ_{40}).$$

when $t > 40$. When $t = 40$, we only need to check the following conditions

$$(X_{12}, LQ_{16}, LQ_{17}, LQ_{18}, X_{39}, LQ_{39}, LQ_{40})$$

since we only need to ensure the modular difference of X_{40} in this case. Denote the probability of these conditions by 2^{-p_1} and the number of bit conditions on (X_{13}, X_{14}) by n_2. Then, we can expect to generate $2^{64-n_2-p_1}$ new starting points from the initial starting point. The time complexity to generating each new starting point is 2^{p_1}.

Third, we estimate the cost to verify the remaining uncontrolled conditions on the left branch. As already stated, we denote the number of conditions on X_{11} by n_1 and there will be 2^{32-n_1} possible values for X_{11}. For each starting point, we first verify the conditions on

$$(X_{11}, X_{10}, LQ_{15}, LQ_{14}, LQ_{13}, LQ_{12}, X_{41}, \ldots, X_{t-1}, LQ_{41}, \ldots, LQ_t).$$

Note that we only need to ensure the modular difference of X_t and therefore we only care about whether LQ_t satisfies its condition. Denote the probability of these conditions by 2^{-p_2}. In this way, for each starting point, we expect to find

$$2^{32-n_1-p_2}$$

many values of X_{11} such that all the conditions on the left branch hold. The cost to find each such solution is then estimated as 2^{p_2} times of evaluations of $4 + (t - 41 + 1) = t - 36$ steps of the step function.

Finally, we need to verify the conditions on the right branch. Denote the probability of the right-branch differential characteristic by 2^{-p_3}. In this way, we need in total

$$2^{p_3-(32-n_1-p_2)}$$

starting points. This indicates that we need

$$T_1 = \max\{1, 2^{p_3-(32-n_1-p_2)-(64-n_2-p_1)}\}$$

initial starting points. Denote the time complexity to generate one initial starting point by T_s. In this way, the whole time complexity of the attack is then estimated as

$$T_{\text{final}} = T_1 \cdot T_s + 2^{p_3-(32-n_1-p_2)} \cdot 2^{p_1} + \frac{t-36}{2(t+1)} \cdot 2^{p_2+p_3} + \frac{t+12}{2(t+1)} \cdot 2^{p_3}.$$

5.4 Application to 41/42/43-Step Differential Characteristics

Apart from the conditions specified in Table 7, Table 8, Table 9, we will further list some other conditions that will affect the whole time complexity.

On 41-step RIPEMD-160. As discussed above, we list the conditions in Table 11 that affect the performance of SFS collision attacks and are not present in Table 7. All the conditions on (LQ_i, RQ_i) for $(0 \leq i \leq t)$ are also listed in Table 15 for completeness. Consequently, for the 41-step differential characteristic in Table 7, we have

$$n_1 = 3,\ n_2 = 8,\ p_1 = 8.2,\ p_2 = 1.9,\ p_3 = 57.2.$$

Since

$$p_3 - (32 - n_1 - p_2) - (64 - n_2 - p_1) = (p_1 + p_2 + p_3 + n_1 + n_2) - 96 < 0,$$

we only need one initial starting point. Consequently, the whole time complexity of the SFS collision attack on 41-step RIPEMD-160 is about $2^{57.2}$.

Table 11: Extra conditions influencing the attack for the 41-step differential characteristic

	Conditions	Pro.
Y_{17}	$Y_{17}[31] = Y_{16}[31]$	2^{-1}
Y_{21}	$Y_{20}[9] = Y_{21}[9]$	2^{-1}
Y_{25}	$Y_{25}[19] = Y_{24}[19], Y_{25}[20] = Y_{24}[20]$	2^{-2}
Y_{29}	$Y_{29}[29] = Y_{28}[29], Y_{29}[30] = Y_{28}[30], Y_{29}[31] = Y_{28}[31]$	2^{-3}
Y_{33}	$Y_{33}[7] \vee \neg Y_{32}[7] = 1$	$2^{-0.5}$
Y_{37}	$Y_{37}[17] \vee \neg Y_{36}[17] = 1$	$2^{-0.5}$
RQ_{15}	$(RQ_{15} \boxplus \texttt{0x8000})^{\lll 6} = RQ_{15}^{\lll 6} \boxplus \texttt{0x200000}$	≈ 1
RQ_{27}	$(RQ_{27} \boxplus \texttt{0x8000})^{\lll 7} = RQ_{27}^{\lll 7} \boxplus \texttt{0x400000}$	≈ 1
RQ_{34}	$(RQ_{34} \boxplus \texttt{0x80})^{\lll 15} = RQ_{34}^{\lll 15} \boxplus \texttt{0x400000}$	≈ 1
RQ_{37}	$(RQ_{37} \boxplus \texttt{0x1})^{\lll 6} = RQ_{37}^{\lll 6} \boxplus \texttt{0x40}$	≈ 1
RQ_{38}	$(RQ_{38} \boxplus \texttt{0x20000})^{\lll 6} = RQ_{38}^{\lll 6} \boxplus \texttt{0x800000}$	$2^{-0.1}$
RQ_{40}	$(RQ_{40} \boxplus \texttt{0xffff0000})^{\lll 12} = RQ_{40}^{\lll 12} \boxplus \texttt{0xf0000000}$	$2^{-0.1}$
$2^{-p_3} = 2^{-49-8.2} = 2^{-57.2}$		
X_{12}	$X_{12}[20] = X_{13}[30], X_{12}[27] = X_{13}[5], X_{12}[19] = X_{14}[29], X_{12}[18] \neq X_{14}[28]$	2^{-4}
LQ_{18}	$(LQ_{18} \boxplus \texttt{0x80})^{\lll 8} = LQ_{18}^{\lll 8} \boxplus \texttt{0x8000}$	$2^{-0.1}$
LQ_{39}	$(LQ_{39} \boxplus \texttt{0x77ec0000})^{\lll 15} = LQ_{39}^{\lll 15} \boxplus \texttt{0x3bf6}$	2^{-1}
LQ_{40}	$(LQ_{40} \boxplus \texttt{0xffffc008})^{\lll 14} = LQ_{40}^{\lll 14} \boxplus \texttt{0xf0020000}$	$2^{-0.1}$
$2^{-p_1} = 2^{-4-3-1-0.1-0.1} = 2^{-8.2}$		
X_{11}	$X_{11}[18] \neq X_{12}[28], X_{11}[19] \neq X_{12}[29], X_{11}[11] = X_{13}[21]$	2^{-3}
X_{10}	$X_{10}[11] = X_{11}[21]$	2^{-1}
LQ_{12}	$(LQ_{12} \boxplus \texttt{0x8000})^{\lll 6} = LQ_{12}^{\lll 6} \boxplus \texttt{0x200000}$	$2^{-0.1}$
LQ_{13}	$(LQ_{13} \boxplus \texttt{0x200000})^{\lll 7} = LQ_{13}^{\lll 7} \boxplus \texttt{0x10000000}$	$2^{-0.1}$
LQ_{14}	$(LQ_{14} \boxplus \texttt{0xf0200000})^{\lll 9} = LQ_{14}^{\lll 9} \boxplus \texttt{0x3fffffe0}$	$2^{-0.6}$
LQ_{15}	$(LQ_{15} \boxplus \texttt{0xeffffffe0})^{\lll 8} = LQ_{15}^{\lll 8} \boxplus \texttt{0xffffdff0}$	$2^{-0.1}$
$n_1 = 3, \quad 2^{-p_2} = 2^{-1.9}$		
X_{14}	$X_{14}[31] = X_{13}[31], X_{14}[3] = X_{13}[13], X_{14}[26] = X_{13}[4]$	2^{-3}
$n_2 = 3 + 5 = 8$		

On 42-step RIPEMD-160 Similarly, we list the extra conditions in Table 12 that affect the performance of SFS collision attacks and are not present in Table 8. The conditions on all (LQ_i, RQ_i) can be referred to Table 16. Consequently, we can obtain

$$n_1 = 3,\ n_2 = 8,\ p_1 = 16.7,\ p_2 = 2.0,\ p_3 = 67.3.$$

Since

$$p_3 - (32 - n_1 - p_2) - (64 - n_2 - p_1) = 1.0,$$

it means we need to generate $T_1 = 2$ initial starting points and hence the whole time complexity of the 42-step SFS collision attack is about $2^{67.3}$.

Table 12: Extra conditions influencing the attack for the 42-step differential characteristic

	Conditions	Prob.
Y_{17}	$Y_{17}[31] = Y_{16}[31]$	2^{-1}
Y_{21}	$Y_{21}[9] = Y_{20}[9]$	2^{-1}
Y_{25}	$Y_{25}[19] = Y_{24}[19], Y_{25}[20] = Y_{24}[20]$	2^{-2}
Y_{29}	$Y_{29}[29] = Y_{28}[29], Y_{29}[30] = Y_{28}[30], Y_{29}[31] = Y_{28}[31]$	2^{-3}
Y_{33}	$Y_{33}[17] \vee \neg Y_{32}[17] = 1$	$2^{-0.5}$
Y_{37}	$Y_{38}[27] \vee \neg Y_{37}[27] = 1$	$2^{-0.5}$
RQ_{15}	$(RQ_{15} \boxplus \texttt{0x8000})^{\lll 6} = RQ_{15}^{\lll 6} \boxplus \texttt{0x200000}$	≈ 1
RQ_{27}	$(RQ_{27} \boxplus \texttt{0x8000})^{\lll 7} = RQ_{27}^{\lll 7} \boxplus \texttt{0x400000}$	≈ 1
RQ_{35}	$(RQ_{35} \boxplus \texttt{0x20000})^{\lll 11} = RQ_{35}^{\lll 11} \boxplus \texttt{0x10000000}$	≈ 1
RQ_{38}	$(RQ_{38} \boxplus \texttt{0x40})^{\lll 6} = RQ_{38}^{\lll 6} \boxplus \texttt{0x1000}$	$2^{-0.1}$
RQ_{39}	$(RQ_{39} \boxplus \texttt{0x8000000})^{\lll 14} = RQ_{39}^{\lll 14} \boxplus \texttt{0x200}$	$2^{-0.1}$
RQ_{41}	$(RQ_{40} \boxplus \texttt{0x400000})^{\lll 13} = RQ_{41}^{\lll 13} \boxplus \texttt{0x8}$	$2^{-0.1}$
$2^{-p_3} = 2^{-67.3}$		
X_{12}	$X_{12}[20] \neq X_{13}[30], X_{12}[27] = X_{13}[5], X_{12}[19] = X_{14}[29], X_{12}[18] \neq X_{14}[28]$	2^{-4}
LQ_{18}	$(LQ_{18} \boxplus \texttt{0xffffff80})^{\lll 8} = LQ_{18}^{\lll 8} \boxplus \texttt{0xffff8000}$	$2^{-0.1}$
LQ_{39}	$(LQ_{39} \boxplus \texttt{0xc0400000})^{\lll 15} = LQ_{39}^{\lll 15} \boxplus \texttt{0xfffe020}$	$2^{-0.5}$
LQ_{40}	$(LQ_{40} \boxplus \texttt{0x1fd8})^{\lll 14} = LQ_{40}^{\lll 14} \boxplus \texttt{0x7f60000}$	$2^{-0.1}$
$2^{-p_1} = 2^{-12-4-0.7} = 2^{-16.7}$		
X_{11}	$X_{11}[18] \neq X_{12}[28], X_{11}[19] \neq X_{12}[29], X_{11}[11] = X_{13}[21]$	2^{-3}
X_{10}	$X_{10}[11] \neq X_{11}[21]$	2^{-1}
LQ_{12}	$(LQ_{12} \boxplus \texttt{0x8000})^{\lll 6} = LQ_{12}^{\lll 6} \boxplus \texttt{0x200000}$	$2^{-0.1}$
LQ_{13}	$(LQ_{13} \boxplus \texttt{0xffe00000})^{\lll 7} = LQ_{13}^{\lll 7} \boxplus \texttt{0xf0000000}$	$2^{-0.1}$
LQ_{14}	$(LQ_{14} \boxplus \texttt{0x10200000})^{\lll 9} = LQ_{14}^{\lll 9} \boxplus \texttt{0x40000020}$	$2^{-0.6}$
LQ_{15}	$(LQ_{15} \boxplus \texttt{0x10000020})^{\lll 8} = LQ_{15}^{\lll 8} \boxplus \texttt{0x2010}$	$2^{-0.1}$
LQ_{41}	$(LQ_{41} \boxplus \texttt{0x8080000})^{\lll 8} = LQ_{41}^{\lll 8} \boxplus \texttt{0x8000008}$	$2^{-0.1}$
$n_1 = 3, \quad 2^{-p_2} = 2^{-2.0}$		
X_{14}	$X_{14}[31] = X_{13}[31], X_{14}[3] = X_{13}[13], X_{14}[26] = X_{13}[4]$	2^{-3}
$n_2 = 8$		

Table 13: Extra conditions influencing the attack for the 43-step differential characteristic

	Conditions	Prob.
Y_{17}	$Y_{17}[31] = Y_{16}[31]$	2^{-1}
Y_{21}	$Y_{21}[9] = Y_{20}[9]$	2^{-1}
Y_{25}	$Y_{25}[19] = Y_{24}[19], Y_{25}[20] = Y_{24}[20]$	2^{-2}
Y_{29}	$Y_{29}[29] = Y_{28}[29], Y_{29}[30] = Y_{28}[30], Y_{29}[31] = Y_{28}[31]$	2^{-3}
Y_{33}	$Y_{33}[7] \vee \neg Y_{32}[7] = 1$	$2^{-0.5}$
Y_{36}	$Y_{36}[0] \vee \neg Y_{35}[0] = 1$	$2^{-0.5}$
Y_{37}	$Y_{37}[17] \vee \neg Y_{36}[17] = 1$	$2^{-0.5}$
Y_{39}	$Y_{39}[16] \vee \neg Y_{38}[16] = 1$	$2^{-0.5}$
Y_{41}	$Y_{41}[27] \vee \neg Y_{40}[27] = 1$	$2^{-0.5}$
RQ_{15}	$(RQ_{15} \boxplus \texttt{0x8000})^{\lll 6} = RQ_{15}^{\lll 6} \boxplus \texttt{0x200000}$	≈ 1
RQ_{27}	$(RQ_{27} \boxplus \texttt{0x8000})^{\lll 7} = RQ_{27}^{\lll 7} \boxplus \texttt{0x400000}$	≈ 1
RQ_{34}	$(RQ_{35} \boxplus \texttt{0x80})^{\lll 15} = RQ_{34}^{\lll 15} \boxplus \texttt{0x400000}$	≈ 1
RQ_{37}	$(RQ_{37} \boxplus \texttt{0xffffffff})^{\lll 6} = RQ_{37}^{\lll 6} \boxplus \texttt{0xffffffc0}$	≈ 1
RQ_{38}	$(RQ_{38} \boxplus \texttt{0x20000})^{\lll 6} = RQ_{38}^{\lll 6} \boxplus \texttt{0x800000}$	$2^{-0.1}$
RQ_{39}	$(RQ_{39} \boxplus \texttt{0x8000000})^{\lll 14} = RQ_{39}^{\lll 14} \boxplus \texttt{0x200}$	$2^{-0.1}$
RQ_{40}	$(RQ_{40} \boxplus \texttt{0xffff0000})^{\lll 12} = RQ_{40}^{\lll 12} \boxplus \texttt{0xf0000000}$	$2^{-0.1}$
RQ_{41}	$(RQ_{41} \boxplus \texttt{0x402})^{\lll 13} = RQ_{41}^{\lll 13} \boxplus \texttt{0x804000}$	$2^{-0.1}$
RQ_{42}	$(RQ_{42} \boxplus \texttt{0xf8000000})^{\lll 5} = RQ_{42}^{\lll 5} \boxplus \texttt{0xffffffff}$	≈ 1
$2^{p_3} = 2^{-62-9.9} = 2^{-71.9}$		
X_{12}	$X_{12}[20] \neq X_{13}[30], X_{12}[27] = X_{13}[5], X_{12}[19] = X_{14}[29], X_{12}[18] \neq X_{14}[28]$	2^{-4}
X_{39}	$X_{39}[19] \vee \neg X_{38}[19] = 1$	$2^{-0.5}$
X_{40}	$X_{40}[10] \vee \neg X_{39}[10] = 1$	$2^{-0.5}$
LQ_{18}	$(LQ_{18} \boxplus \texttt{0x80 })^{\lll 8} = LQ_{18}^{\lll 8} \boxplus \texttt{0x8000}$	$2^{-0.1}$
LQ_{39}	$(LQ_{39} \boxplus \texttt{0xbefdddf2})^{\lll 15} = LQ_{39}^{\lll 15} \boxplus \texttt{0xeef8df7f}$	$2^{-0.6}$
LQ_{40}	$(LQ_{40} \boxplus \texttt{0x2fff2088})^{\lll 14} = LQ_{40}^{\lll 14} \boxplus \texttt{0xc8220c00}$	$2^{-0.7}$
$2^{-p_1} = 2^{-11-4-1-0.1-0.6-0.7} = 2^{-17.4}$		
X_{11}	$X_{11}[18] \neq X_{12}[28], X_{11}[19] \neq X_{12}[29], X_{11}[11] = X_{13}[21]$	2^{-3}
X_{10}	$X_{10}[11] \neq X_{11}[21]$	2^{-1}
LQ_{12}	$(LQ_{12} \boxplus \texttt{0x8000})^{\lll 6} = LQ_{12}^{\lll 6} \boxplus \texttt{0x200000}$	$2^{-0.1}$
LQ_{13}	$(LQ_{13} \boxplus \texttt{0x200000})^{\lll 7} = LQ_{13}^{\lll 7} \boxplus \texttt{0x10000000}$	$2^{-0.1}$
LQ_{14}	$(LQ_{14} \boxplus \texttt{0xf0200000})^{\lll 9} = LQ_{14}^{\lll 9} \boxplus \texttt{0x3fffffe0}$	$2^{-0.6}$
LQ_{15}	$(LQ_{15} \boxplus \texttt{0xfffffe0})^{\lll 8} = LQ_{15}^{\lll 8} \boxplus \texttt{0xffffe010}$	$2^{-0.1}$
LQ_{41}	$(LQ_{41} \boxplus \texttt{0x37dff800})^{\lll 8} = LQ_{41}^{\lll 8} \boxplus \texttt{0xdf80038}$	$2^{-0.6}$
LQ_{42}	$(LQ_{42} \boxplus \texttt{0x2007fc06})^{\lll 13} = LQ_{42}^{\lll 13} \boxplus \texttt{0xff80c401}$	$2^{-0.2}$
$n_1 = 3, \quad 2^{-p_2} = 2^{-1-4-0.1-0.1-0.6-0.1-0.6-0.2} = 2^{-6.7}$		
X_{14}	$X_{14}[31] = X_{13}[31], X_{14}[3] = X_{13}[13], X_{14}[26] = X_{13}[4]$	2^{-3}
$n_2 = 8$		

On 43-step RIPEMD-160. The extra conditions for the SFS collision attack on 43-step RIPEMD-160 are given in Table 13, and the conditions on all (LQ_i, RQ_i) are shown in Table 17. Therefore, we have

$$n_1 = 3,\ n_2 = 8,\ p_1 = 17.4,\ p_2 = 6.7,\ p_3 = 71.9.$$

Since

$$p_3 - (32 - n_1 - p_2) - (64 - n_2 - p_1) = 11,$$

it implies that we need to generate $T_1 = 2^{11}$ initial starting points and hence the whole time complexity of the 43-step SFS collision attack is about $2^{74.8}$. As already mentioned, we can generate much more than 2^{11} different initial starting points and this is not a problem for our 43-step attack.

On the initial starting points. As already mentioned, the initial starting points can be efficiently found. For evidence, we provide the initial starting points for the 41/42/43-step SFS collision attacks in Table 14. Indeed, in our experiments, we could generate one initial starting point in 30 seconds. Hence, the time complexity $T_1 \cdot T_s$ is negligible given that we only need a few initial starting points.

Table 14: Solution for $X_i (12 \leq i \leq 40)$

	41-step	42-step	43-step
X_{12}	0000011000n01110011000111110100 11	0011000100n00110011000000100000	0100001101n10111011000110000 1100
X_{13}	10nu00111011111011101100000 1111	11un00110001000101110000100 11011	10nu10100111001010001111100000 111
X_{14}	1n10000111000000011000010u00100	0n00001100110100111000101 0n01101	1n11101011010100110000011u01001
X_{15}	10001100111000111 1u01000110u0010	00000000111101000 1n11011110n1101	10100110000011110 1u11111110n1100
X_{16}	n11111101001111110110010100 0100	u00011011001000110011011010000 110	u10110110001001111000001110000 10
X_{17}	00100000100111101111110 01n110000	01110011100101001111111111u110010	01100000110111111111011100n111100
X_{18}	10010000001111001000101n10100000	11010010010000110010010n10010100	11011111010100110011101n10000001
X_{19}	101u0110u01100010u00000011000011	001n0100n11111011n11010011010010	001u0000u01111111n00010011011101
X_{20}	00000001001011000001 01n101000001	11101110000001100000100u001101111	01001111010011000100 11u110001000
X_{21}	0101110101101 10n1101100001000011	01110011110101 1u0101110011011010	01010111n111010n01101101111000010
X_{22}	1100010001101n0000101111011001001	0100110000100n101100001110001100	0111010011111n0010101111111000010
X_{23}	1110011u0001001110110101 0u000010	0000010n0001001101100111 0n0100n1	0010110n10110001100011000111 0100
X_{24}	111110100n00n11011u1010u00000000	000100100n00u0101000111110111100	110100001n10u00101u1110000011001
X_{25}	01110n1110100001001010001000111 0	11100u0010110011100110000 1101111	00101n1110000011101110100001 01n0
X_{26}	0111110010011101011000010 1000100	000n1000010111001101011101100110	0111010110000110101001111000 0010
X_{27}	1000011100unnnnn0nuuuu0uuuunuu01	01011100100nuuuu110nu0n110u10n10	100000nuuuuuuuuuuuuu11uuu0uun00
X_{28}	nnnnnu1uuu111100nuu1nuuuuuuuuu1u	1nuu11nuu01011un0n000001n000001n	10u1nn1n0n00u0nu1u011010000uun11
X_{29}	101unn01010uuun0011nuuu000nnu010	uu01u110n1n11001n0u11unnnnnn010u	un1000u1000nnu10u01nuuuu0011u1n
X_{30}	01111n0110110000001101 0u1u11011 0	000n0011000110011101u010nn1001011	1110n10n11n1001u010100011n00001n
X_{31}	0000001n01100100u0110000 10u011u0	01n0n00n01010011uu1011100u011000	1101110u0000010000101u0001011000
X_{32}	011011110111n001n1001u111n111001	1u111011uu1u100n000n000101010011	1101010111111u010011u0001uu11111u
X_{33}	01n0n0n000000110u001001101100001	u1000uu00111111101000000n1101n10	00n00n1001101u111011011000u0u010
X_{34}	0011000011n100n010001u0u10010001	01101010101u0100111n10110u100100	01101u10n11101u001u0u110u101u010
X_{35}	101010n1111011101100111101 0u0001	0000u011001001011001000010100n10	01n1110110110n1110n110n11u10n111
X_{36}	1001000000110n101101110110011000	0101010111010100101110110n001001	10010100100n00u01000u010101011uu
X_{37}	0101011101110000000011000001000	100011u1111100010001010111100111	100010n00110n110000011n100011010
X_{38}	0101000001001010100100110u111000	1101101010001100100111110010 1001	001010111u011000101001011010110u
X_{39}	1110100101110111111001u01010110u1	1001100000010110110 1u111011000010	01000010u11111010100111011100 01u
X_{40}	1000001011001 1n10110011000001010	0011n1100101u11u011111000011001	1101110111110 1n00100010000001110

m_0 =0xc408cf53,m_1 =0x291e9252,	m_0 = 0xee3603a1, m_1 = 0x8241ac1d,	m_0 = 0x5cfdf423,m_1 = 0xa4524a12,
m_2 =0xd523de93,m_3 =0x8ba64003,	m_2 = 0xe4c6b917,m_3 = 0x859fd0e6,	m_2 = 0x8ae6913d,m_3 = 0x3281ae27,
m_4 =0xe5b58a85,m_5 =0xb99d6243,	m_4 = 0xd21a1098,m_5 = 0xc9441e8b,	m_4 = 0x14487cb0,m_5 = 0xb429816b,
m_6 =0x54e6af34,m_8 =0x27c09f39,	m_6 = 0xc08c06b8,m_8 = 0xc662c263,	m_6 = 0x75b453bc,m_8 = 0x8cd71d13,
m_9 =0xe8300da6,m_{10} =0xa4fff494,	m_9 = 0x8b684af9,m_{10} = 0x877fd7e7,	m_9 = 0x4a7429e7,m_{10} = 0x8beaeaf4,
m_{11} =0x3efb77d4,m_{12} =0x4adaef9,	m_{11} = 0xfb418343,m_{12} = 0x5b3b73a8.	m_{11} = 0xd501b848,m_{12} = 0x6af25b11.
m_{13} =0x9204280d,m_{14} =0x8ae1c0fa,	m_{13} = 0x66be3488,m_{14} = 0x27eebf4.	m_{13} = 0x1499e711,m_{14} = 0x58157e02.
m_{15} =0xbc002a65.	m_{15} = 0xf1f6cd2e.	m_{15} = 0xc0cd54b4.

6 Conclusion

With the automatic SAT/SMT-based tools, we have significantly improved the (SFS) collision attacks on RIPEMD-160. In particular, we found the practical colliding message pair for 40-step RIPEMD-160 for the first time, and it practically breaks half of full-round RIPEMD-160 after more than 20 years of its publication. For the SFS collision attack, by

searching for new right-branch differential characteristics with minimal Hamming weight, we successfully improved the best attack 3 steps. It is interesting to investigate whether it is possible to further improve the (SFS) collision attacks on RIPEMD-160 by using other strategies different from [LDM+19b, LWS+23] since we seem to have reached the best of the strategies proposed in these two papers. In addition, it may be possible to mount a valid SFS collision attack on 44-step RIPEMD-160 in the quantum setting based on our 44-step differential characteristic. However, this is not our interest as the most important step in the dedicated quantum collision attack is still to search for a high-probability differential characteristic, and we have addressed this issue in this work.

Acknowledgments

We thank the reviewers for helping improve the quality of this paper. Yingxin Li and Gaoli Wang are supported by the National Key Research and Development Program of China (2022YFB2701900), the National Natural Science Foundation of China (No. 62072181) and the "Digital Silk Road" Shanghai International Joint Lab of Trustworthy Intelligent Software (No. 22510750100). Fukang Liu is supported by Grant-in-Aid for Research Activity Start-up (Grant No. 22K21282).

References

[BCJ+05] Eli Biham, Rafi Chen, Antoine Joux, Patrick Carribault, Christophe Lemuet, and William Jalby. Collisions of SHA-0 and reduced SHA-1. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 36–57. Springer, 2005.

[BP95] Antoon Bosselaers and Bart Preneel, editors. *Integrity Primitives for Secure Information Systems, Final Report of RACE Integrity Primitives Evaluation RIPE-RACE 1040*, volume 1007 of *Lecture Notes in Computer Science*. Springer, 1995.

[CR06] Christophe De Cannière and Christian Rechberger. Finding SHA-1 characteristics: General results and applications. In *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2006.

[DBP96] Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. RIPEMD-160: A strengthened version of RIPEMD. In Dieter Gollmann, editor, *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 21-23, 1996, Proceedings*, volume 1039 of *Lecture Notes in Computer Science*, pages 71–82. Springer, 1996.

[DEM15] Christoph Dobraunig, Maria Eichlseder, and Florian Mendel. Analysis of SHA-512/224 and SHA-512/256. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 612–630. Springer, 2015.

[EMS14] Maria Eichlseder, Florian Mendel, and Martin Schläffer. Branching heuristics in differential collision search with applications to SHA-512. In Carlos Cid and

Christian Rechberger, editors, *Fast Software Encryption - 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers*, volume 8540 of *Lecture Notes in Computer Science*, pages 473–488. Springer, 2014.

[LDM+19a] Fukang Liu, Christoph Dobraunig, Florian Mendel, Takanori Isobe, Gaoli Wang, and Zhenfu Cao. Efficient collision attack frameworks for RIPEMD-160. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 117–149. Springer, 2019.

[LDM+19b] Fukang Liu, Christoph Dobraunig, Florian Mendel, Takanori Isobe, Gaoli Wang, and Zhenfu Cao. New semi-free-start collision attack framework for reduced RIPEMD-160. *IACR Trans. Symmetric Cryptol.*, 2019(3):169–192, 2019.

[LMW17] Fukang Liu, Florian Mendel, and Gaoli Wang. Collisions and semi-free-start collisions for round-reduced RIPEMD-160. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 158–186. Springer, 2017.

[LP13] Franck Landelle and Thomas Peyrin. Cryptanalysis of full RIPEMD-128. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 228–244. Springer, 2013.

[LP19] Gaëtan Leurent and Thomas Peyrin. From collisions to chosen-prefix collisions application to full SHA-1. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 527–555. Springer, 2019.

[LP20] Gaëtan Leurent and Thomas Peyrin. SHA-1 is a shambles: First chosen-prefix collision on SHA-1 and application to the PGP web of trust. In Srdjan Capkun and Franziska Roesner, editors, *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, pages 1839–1856. USENIX Association, 2020.

[LWS+23] Fukang Liu, Gaoli Wang, Santanu Sarkar, Ravi Anand, Willi Meier, Yingxin Li, and Takanori Isobe. Analysis of RIPEMD-160: new collision attacks and finding characteristics with MILP. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 189–219. Springer, 2023.

[MNS11] Florian Mendel, Tomislav Nad, and Martin Schläffer. Finding SHA-2 characteristics: Searching through a minefield of contradictions. In Dong Hoon Lee

and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 288–307. Springer, 2011.

[MNS12] Florian Mendel, Tomislav Nad, and Martin Schläffer. Collision attacks on the reduced dual-stream hash function RIPEMD-128. In Anne Canteaut, editor, *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *Lecture Notes in Computer Science*, pages 226–243. Springer, 2012.

[MNS13] Florian Mendel, Tomislav Nad, and Martin Schläffer. Improving local collisions: New attacks on reduced SHA-256. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 262–278. Springer, 2013.

[MNSS12] Florian Mendel, Tomislav Nad, Stefan Scherz, and Martin Schläffer. Differential attacks on reduced RIPEMD-160. In Dieter Gollmann and Felix C. Freiling, editors, *Information Security - 15th International Conference, ISC 2012, Passau, Germany, September 19-21, 2012. Proceedings*, volume 7483 of *Lecture Notes in Computer Science*, pages 23–38. Springer, 2012.

[MPS+13] Florian Mendel, Thomas Peyrin, Martin Schläffer, Lei Wang, and Shuang Wu. Improved cryptanalysis of reduced RIPEMD-160. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*, volume 8270 of *Lecture Notes in Computer Science*, pages 484–503. Springer, 2013.

[MZ06] Ilya Mironov and Lintao Zhang. Applications of SAT solvers to cryptanalysis of hash functions. In *SAT*, volume 4121 of *Lecture Notes in Computer Science*, pages 102–115. Springer, 2006.

[SBK+17] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The first collision for full SHA-1. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 570–596. Springer, 2017.

[WLC+20] Gaoli Wang, Fukang Liu, Binbin Cui, Florian Mendel, and Christoph Dobraunig. Improved (semi-free-start/near-) collision and distinguishing attacks on round-reduced RIPEMD-160. *Des. Codes Cryptogr.*, 88(5):887–930, 2020.

[WLF+05] Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, and Xiuyuan Yu. Cryptanalysis of the hash functions MD4 and RIPEMD. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2005.

[WS14] Gaoli Wang and Yanzhao Shen. (pseudo-) preimage attacks on step-reduced HAS-160 and RIPEMD-160. In Sherman S. M. Chow, Jan Camenisch, Lucas Chi Kwong Hui, and Siu-Ming Yiu, editors, *Information Security - 17th International Conference, ISC 2014, Hong Kong, China, October 12-14, 2014. Proceedings*, volume 8783 of *Lecture Notes in Computer Science*, pages 90–103. Springer, 2014.

[WSL17] Gaoli Wang, Yanzhao Shen, and Fukang Liu. Cryptanalysis of 48-step RIPEMD-160. *IACR Trans. Symmetric Cryptol.*, 2017(2):177–202, 2017.

[WY05] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2005.

[WYY05a] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer, 2005.

[WYY05b] Xiaoyun Wang, Hongbo Yu, and Yiqun Lisa Yin. Efficient collision search attacks on SHA-0. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2005.

A Additional Conditions for Differential Characteristic

The conditions on (LQ_i, RQ_i) are given in Table 15, Table 16, and Table 17, respectively.

Table 15: Some extra conditions for the 41-step differential characteristic

Conditions on LQ_i and RQ_i:
$(LQ_i \boxplus in_i^l)^{\lll s_i^l} = LQ_i{}^{\lll s_i^l} \boxplus out_i^l$
$(RQ_i \boxplus in_i^r)^{\lll s_i^r} = RQ_i{}^{\lll s_i^r} \boxplus out_i^r$

i	in_i^l	out_i^l	π_l^i	s_l^i	i	in_i^r	out_i^r	π_r^i	s_r^i
0	0x0	0x0	0	11	0	0x0	0x0	5	8
1	0x0	0x0	1	14	1	0x0	0x0	14	9
2	0x0	0x0	2	15	2	0x0	0x0	7	9
3	0x0	0x0	3	12	3	0x0	0x0	0	11
4	0x0	0x0	4	5	4	0x0	0x0	9	13
5	0x0	0x0	5	8	5	0x0	0x0	2	15
6	0x0	0x0	6	7	6	0x0	0x0	11	15
7	0x0	0x0	7	9	7	0x0	0x0	4	5
8	0x0	0x0	8	11	8	0x0	0x0	13	7
9	0x0	0x0	9	13	9	0x0	0x0	6	7
10	0x0	0x0	10	14	10	0x0	0x0	15	8
11	0x1	0x0	11	15	11	0x0	0x0	8	11
12	0x8000	0x200000	12	6	12	0x0	0x0	1	14
13	0x200000	0x10000000	13	7	13	0x0	0x0	10	14
14	0xf0200000	0x3fffffe0	14	9	14	0x0	0x0	3	12
15	0xeffffffe0	0xffffdff0	15	8	15	0x8000	0x200000	12	6
16	0x0	0x0	7	7	16	0x0	0x0	6	9
17	0x0	0x0	4	6	17	0x0	0x0	11	13
18	0x80	0x8000	13	8	18	0x0	0x0	3	15
19	0xffff8000	0xf0000000	1	13	19	0x0	0x0	7	7
20	0x0	0x0	10	11	20	0x0	0x0	0	12
21	0x0	0x0	6	9	21	0x0	0x0	13	8
22	0x0	0x0	15	7	22	0x0	0x0	5	9
23	0x40000	0x2	3	15	23	0x0	0x0	10	11
24	0x7fbe	0x3fdf00	12	7	24	0x0	0x0	14	7
25	0x0	0x0	0	12	25	0x0	0x0	15	7
26	0xffffe000	0xf0000000	9	15	26	0x0	0x0	8	12
27	0x14000001	0x228	5	9	27	0x8000	0x400000	12	7
28	0x3a5884	0xd2c42002	2	11	28	0x0	0x0	4	6
29	0x2ff7cc04	0xfbe60218	14	7	29	0x0	0x0	9	15
30	0xf6002000	0x3fffec0	11	13	30	0x0	0x0	1	13
31	0xfde04f6f	0x4f6efde	8	12	31	0x0	0x0	2	11
32	0x8edff10e	0xff886c77	3	11	32	0x0	0x0	15	9
33	0x7c8fb7	0x91f6e010	10	13	33	0x0	0x0	5	7
34	0xc0009bec	0x26faf0	14	6	34	0x80	0x400000	1	15
35	0xd8080110	0x40087ec	4	7	35	0x0	0x0	3	11
36	0xffff784c	0xde130000	9	14	36	0x0	0x0	7	8
37	0xac010000	0x1ffff58	15	9	37	0x1	0x40	14	6
38	0xfe03c0a0	0x7813ffc0	8	13	38	0x20000	0x800000	6	6
39	0x77ec0000	0x3bf6	1	15	39	0x0	0x0	9	14
40	0xffffc008	0xf0020000	2	14	40	0xffff0000	0xf0000000	11	12

Table 16: Some extra conditions for the 42-step differential characteristic

Conditions on LQ_i and RQ_i:
$(LQ_i \boxplus in_i^l)^{\lll s_i^l} = LQ_i^{\lll s_i^l} \boxplus out_i^l$
$(RQ_i \boxplus in_i^r)^{\lll s_i^r} = RQ_i^{\lll s_i^r} \boxplus out_i^r$

i	in_i^l	out_i^l	π_l^i	s_l^i	i	in_i^r	out_i^r	π_r^i	s_r^i
0	0x0	0x0	0	11	0	0x0	0x0	5	8
1	0x0	0x0	1	14	1	0x0	0x0	14	9
2	0x0	0x0	2	15	2	0x0	0x0	7	9
3	0x0	0x0	3	12	3	0x0	0x0	0	11
4	0x0	0x0	4	5	4	0x0	0x0	9	13
5	0x0	0x0	5	8	5	0x0	0x0	2	15
6	0x0	0x0	6	7	6	0x0	0x0	11	15
7	0x0	0x0	7	9	7	0x0	0x0	4	5
8	0x0	0x0	8	11	8	0x0	0x0	13	7
9	0x0	0x0	9	13	9	0x0	0x0	6	7
10	0x0	0x0	10	14	10	0x0	0x0	15	8
11	0x1	0x0	11	15	11	0x0	0x0	8	11
12	0x8000	0x200000	12	6	12	0x0	0x0	1	14
13	0xffe00000	0xf0000000	13	7	13	0x0	0x0	10	14
14	0x10200000	0x40000020	14	9	14	0x0	0x0	3	12
15	0x10000020	0x2010	15	8	15	0x8000	0x200000	12	6
16	0x0	0x0	7	7	16	0x0	0x0	6	9
17	0x0	0x0	4	6	17	0x0	0x0	11	13
18	0xffffff80	0xffff8000	13	8	18	0x0	0x0	3	15
19	0x8000	0x10000000	1	13	19	0x0	0x0	7	7
20	0x0	0x0	10	11	20	0x0	0x0	0	12
21	0x0	0x0	6	9	21	0x0	0x0	13	8
22	0x0	0x0	15	7	22	0x0	0x0	5	9
23	0x0	0x0	3	15	23	0x0	0x0	10	11
24	0x8000	0x400000	12	7	24	0x0	0x0	14	7
25	0x0	0x0	0	12	25	0x0	0x0	15	7
26	0x0	0x0	9	15	26	0x0	0x0	8	12
27	0xf0000001	0x1e0	5	9	27	0x8000	0x400000	12	7
28	0x10060fe8	0x307f4080	2	11	28	0x0	0x0	4	6
29	0xfe7140c0	0x38a05fff	14	7	29	0x0	0x0	9	15
30	0x8a008000	0xffff140	11	13	30	0x0	0x0	1	13
31	0xfc024d7b	0x24d7afc0	8	12	31	0x0	0x0	2	11
32	0x77b845e2	0xc22f0bbe	3	11	32	0xc0000000	0xffffff80	15	9
33	0x3d0fc402	0xf88047a2	10	13	33	0x0	0x0	5	7
34	0xa83e	0x2a0f80	14	6	34	0x0	0x0	1	15
35	0xbff601ff	0xfb00ff60	4	7	35	0x20000	0x10000000	3	11
36	0x50cef00	0x3bc00143	9	14	36	0x0	0x0	7	8
37	0xbfefef9	0xfdfdf218	15	9	37	0x0	0x0	14	6
38	0x1fe08	0x3fc10000	8	13	38	0x40	0x1000	6	6
39	0xc0400000	0xffffe020	1	15	39	0x8000000	0x200	9	14
40	0x1fd8	0x7f60000	2	14	40	0x0	0x0	11	12
41	0x8080000	0x8000008	7	8	41	0x400000	0x8	8	13

Table 17: Some extra conditions for the 43-step differential characteristic

Conditions on LQ_i and RQ_i:
$(LQ_i \boxplus in_i^l)^{\lll s_i^l} = LQ_i{}^{\lll s_i^l} \boxplus out_i^l$
$(RQ_i \boxplus in_i^r)^{\lll s_i^r} = RQ_i{}^{\lll s_i^r} \boxplus out_i^r$

i	in_i^l	out_i^l	π_l^i	s_l^i	i	in_i^r	out_i^r	π_r^i	s_r^i
0	0x0	0x0	0	11	0	0x0	0x0	5	8
1	0x0	0x0	1	14	1	0x0	0x0	14	9
2	0x0	0x0	2	15	2	0x0	0x0	7	9
3	0x0	0x0	3	12	3	0x0	0x0	0	11
4	0x0	0x0	4	5	4	0x0	0x0	9	13
5	0x0	0x0	5	8	5	0x0	0x0	2	15
6	0x0	0x0	6	7	6	0x0	0x0	11	15
7	0x0	0x0	7	9	7	0x0	0x0	4	5
8	0x0	0x0	8	11	8	0x0	0x0	13	7
9	0x0	0x0	9	13	9	0x0	0x0	6	7
10	0x0	0x0	10	14	10	0x0	0x0	15	8
11	0x0	0x0	11	15	11	0x0	0x0	8	11
12	0x8000	0x200000	12	6	12	0x0	0x0	1	14
13	0x200000	0x10000000	13	7	13	0x0	0x0	10	14
14	0xf0200000	0x3fffffe0	14	9	14	0x0	0x0	3	12
15	0xffffffe0	0xffffe010	15	8	15	0x8000	0x200000	12	6
16	0x0	0x0	7	7	16	0x0	0x0	6	9
17	0x0	0x0	4	6	17	0x0	0x0	11	13
18	0x80	0x8000	13	8	18	0x0	0x0	3	15
19	0xffff8000	0xf0000000	1	13	19	0x0	0x0	7	7
20	0x0	0x0	10	11	20	0x0	0x0	0	12
21	0x4000	0x800000	6	9	21	0x0	0x0	13	8
22	0x0	0x0	15	7	22	0x0	0x0	5	9
23	0x840000	0x42	3	15	23	0x0	0x0	10	11
24	0x7fc0	0x3fe000	12	7	24	0x0	0x0	14	7
25	0x0	0x0	0	12	25	0x0	0x0	15	7
26	0xffffe000	0xf0000000	9	15	26	0x0	0x0	8	12
27	0x14000003	0x628	5	9	27	0x8000	0x400000	12	7
28	0xfd61b718	0xdb8bfeb	2	11	28	0x0	0x0	4	6
29	0xd97c26f9	0xbe137c6d	14	7	29	0x0	0x0	9	15
30	0x20848f8	0x91f0041	11	13	30	0x0	0x0	1	13
31	0xbfefee75	0xfee74bff	8	12	31	0x0	0x0	2	11
32	0xf1439f08	0x1cf83f8a	3	11	32	0xc0000000	0xffffff80	15	9
33	0x66feaf50	0xd5ea0ce0	10	13	33	0x0	0x0	5	7
34	0x5201f34d	0x807cd354	14	6	34	0x80	0x400000	1	15
35	0x98402844	0x201421cc	4	7	35	0x0	0x0	3	11
36	0xeff48144	0x2050fbfd	9	14	36	0x0	0x0	7	8
37	0xb8090451	0x1208a170	15	9	37	0xffffffff	0xffffffc0	14	6
38	0x684311	0x862200d	8	13	38	0x20000	0x800000	6	6
39	0xbefdddf2	0xeef8df7f	1	15	39	0x0	0x0	9	14
40	0x2fff2088	0xc8220c00	2	14	40	0xffff0000	0xf0000000	11	12
41	0x37dff800	0xdff80038	7	8	41	0x402	0x804000	8	13
42	0x2007fc06	0xff80c401	0	13	42	0xf8000000	0xffffffff	12	5