New Collision Attacks on Round-Reduced SHA-512

Yingxin Li[1],Fukang Liu[2], Gaoli Wang[1(✉)], Haifeng Qian[1], Keting Jia[3], Xiangyu Kong[1]

[1]East China Normal University
[2]Tokyo Institute of Technology
[3]Tsinghua University

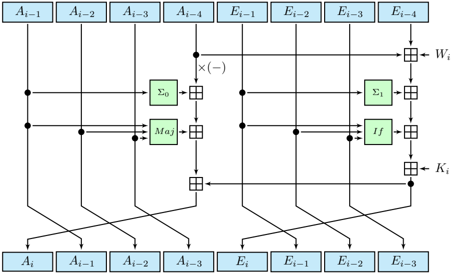
August, 2025

Overview

SHA-2

- A popular hash function family standardized by NIST.
- Strengthening SHA-1 (more complex compression function).
- Two main versions: SHA-256 and SHA-512.
- Used worldwide.

Compression Functions of SHA-512



- Step function

$$
\begin{aligned}
E_i &= A_{i-4} \boxplus E_{i-4} \boxplus \Sigma_1(E_{i-1}) \boxplus \mathrm{IF}(E_{i-1}, E_{i-2}, E_{i-3}) \boxplus K_i \boxplus W_i, \\
A_i &= E_i \boxminus A_{i-4} \boxplus \Sigma_0(A_{i-1}) \boxplus \mathrm{MAJ}(A_{i-1}, A_{i-2}, A_{i-3}).
\end{aligned}
$$

Compression Functions of SHA-512

- Boolean functions $\Sigma_0, \Sigma_1, \mathrm{IF}$ and MAJ are given by

$$\begin{aligned}
\mathrm{IF}(x, y, z) &= (x \wedge y) \oplus (x \wedge z) \oplus z, \\
\mathrm{MAJ}(x, y, z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z), \\
\Sigma_0(x) &= (x \ggg 28) \oplus (x \ggg 34) \oplus (x \ggg 39), \\
\Sigma_1(x) &= (x \ggg 14) \oplus (x \ggg 18) \oplus (x \ggg 41).
\end{aligned}$$

Compression Functions of SHA-512

■ Message expansion

The message expansion of SHA-512 splits the 1024-bit message block M_j into 16 words m_i, $i = 0, \cdots, 15$, and expands them into 80 expanded message words W_i

$$W_i = \begin{cases} m_i & 0 \leq i \leq 15, \\ \sigma_1(W_{i-2}) \boxplus W_{i-7} \boxplus \sigma_0(W_{i-15}) \boxplus W_{i-16} & 16 \leq i \leq 79. \end{cases}$$

The functions $\sigma_0(x)$ and $\sigma_1(x)$ are given by

$$\begin{aligned} \sigma_0(x) &= (x \ggg 1) \oplus (x \ggg 8) \oplus (x \gg 7), \\ \sigma_1(x) &= (x \ggg 19) \oplus (x \ggg 61) \oplus (x \gg 6). \end{aligned}$$

Key Progress in Collision Attacks on SHA-2

Expanded Message Words	Version	Step	Types	Ref.
$(W_7, W_8, W_{12}, W_{15}, W_{17})$	SHA-256 SHA-512	27 27	Practical Practical	Asiacrypt 2011 Asiacrypt 2015
$(W_8, W_9, W_{13}, W_{16}, W_{18})$	SHA-256 SHA-512	28 28	Practical Practical	Eurocrypt 2013 Eurocrypt 2024
$(W_5, \ldots, W_9, W_{16}, W_{18})$	SHA-256 SHA-512	31 31	Practical Theoretic	Asiacrypt 2024 Asiacrypt 2024

Collision Attack framework for 31-step SHA-512

The collision attack framework based on a two-block message consists of three steps, where the first message block is denoted by M_0, which is freely chosen.

1. **Pre-processing Phase.** Find valid solutions of

$$(A_1, \ldots, A_{12}, E_5, \ldots, E_{12}, W_9, ..., W_{12}).$$

Then choose N_{start} solutions with distinct

$$(A_1, \ldots, A_4, E_5, \ldots, E_8).$$

Finally, according to the state update function and each starting point $(A_1, \ldots, A_4, E_5, \ldots, E_8)$, first exhaust all possible (W_8, E_4) to obtain A_0. Then exhaust all possible (W_7, E_3) to obtain A_{-1} from each tuple (W_8, E_4, A_0). Based on such a process, we can obtain all valid tuples $(A_{-1}, \ldots, A_{12}, E_3, \ldots, E_{12}, W_7, \ldots, W_{12})$, and store them in a table denoted by $\mathtt{TAB}_2$.

Collision Attack framework for 31-step SHA-512

2. **Matching Phase.** Try an arbitrary M_0, and get the corresponding chaining input $(A_{-4}, A_{-3}, A_{-2}, A_{-1}, E_{-4}, E_{-3}, E_{-2}, E_{-1})$ to match A_{-1} from $\texttt{TAB}_2$. Once a match is found, perform the on-the-fly detection of the validity of A_{-2} and A_{-3}, which is indeed to test the conditions on (W_5, W_6).
3. **Fulfill the Conditions on $(E_{13}, E_{14}, E_{15}, W_{16}, W_{18})$.** Up to this step, $(W_i)_{0 \leq i \leq 12}$ have been fixed. Use the degrees of freedom in $(W_i)_{13 \leq i \leq 15}$ to fulfill the remaining uncontrolled conditions on $(E_{13}, E_{14}, E_{15}, W_{16}, W_{18})$. If it fails, go to Step 2.

Complexity Evaluation

In the Step 1, suppose that there are n_1 n_2 ,n_3 and n_4 bit conditions on W_8, E_4, W_7 and E_3, respectively. N_{start} is defined as the number of starting points. Denote the time complexity to obtain one starting point by T_{sat}. Denote the number of all conditions on (W_5, W_6) by N_{pro}
The time complexity of Step 1 is estimated as

$$T_{pre} = N_{start} \times (T_{sat} + min(2^{n-n_1}, 2^{n-n_2}) + 2^{n-n_1-n_2} \times min(2^{n-n_3}, 2^{n-n_4})).$$

The time complexity of Step 2-3 is estimated as

$$T_{match} = \frac{2^{N_{pro}+\beta+n_1+n_2+n_3+n_4-n}}{N_{start}} + 2^{N_{pro}+\beta}.$$

The total time complexity of memory-efficient collision attack framework is

$$T_{pre} + T_{match}$$

The memory complexity denoted by M is

$$M = N_{start} \cdot 2^{2n-n_1-n_2-n_3-n_4}$$

.

Collision Attacks on 31-Step SHA-512 in Asiacrypt 2024

i	ΔA_i	ΔE_i	ΔW_i
-4	==	==	
-3	==	==	
-2	==	==	
-1	==	==	
0	==	==	==
1	==	==	==
2	==	==	==
3	==	=11=00000====10=0=====000==101=0===1========0===1==1==011110====	==
4	==	=00011100=11101=1===10111==010=1===0===1=10=01=00==101101011=10=	==
5	=nunuuuuuu======u======u===unnnu===u=n======u=====nn===unnnn====	uuu1nuuuuu101unnn10001ununnnnn1n=1unnu10=010u011nu0u11unnnnn1111	==n======n=====un======u=====u=nunnnnu======u=====nn=======u====
6	========n=======u=================un=======u=====unnnn====nuunn=	00nnnnnnu0unu0u0n111unnn101u010u0011110u0un0nn0uu11uun10n0nn1nu1	n0u011=01u=====0n====10n====un=============n====uu===u=u=nuuuuu=
7	=n======nunnu==u=u=========u==uu=n=u========nnn====n=====u=u====	u1uu11u1n0nuu1000nnu10uu1un1uu0u110un111110nu1uu0n1010011nn=uu1u	=u=============n============u====u==unn==n====u=======n=0=1====u
8	===u	11110011001100u000100u1n00n0011==n=0000n=0nu0un0n1n00010n0111110	n=====nu=====u=====nu=====n======u=====u======u==========0======
9	========================n=======================================	11n1==111=010111011101unn01000u0=011u1u00=0110010101=1==10101101	============u============n============n========================n
10	============u===========nu============n======================nuu	==10==1===111=10u0==101=0n0==11==11010n11=1u=0110=0=0n==1101=nuu	==
11	==	=1u1======un0=n=nn===11001u==un====unnn0n==n=========u==u=10=100	==
12	==	=000======00n=1=11===10==u1==00===1010u11==1=========1==1==0=11u	==
13	==	==1=======110=1=01=======11==11====1111=1==0=========0==1======1	==
14	==	============u============n============n========================n	==
15	==	============0============0============0========================0	==
16	==	============1============1============1========================1	=======unnnnnn====nuuuuuuuu=======nuuuuu==========11====nuuuuuuuu
17	==	==	==
18	==	==	============n============u============u========================u
19	==	==	==
20	==	==	==
21	==	==	==
22	==	==	==
23	==	==	==
24	==	==	==
25	==	==	==
26	==	==	==
27	==	==	==
28	==	==	==
29	==	==	==
30	==	==	==

- $n = 64, n_1 = 20, n_2 = 29, n_3 = 36, n_4 = 16, \mathcal{N}_{\texttt{start}} = 4, \mathcal{N}_{\texttt{pro}} = 65,\ \beta \approx 0.9.$
- Time Complexity: $2^{94.7}$, Memory Complexity: $2^{35.2}$.

Modeling the Two-Bit Conditions of Boolean Functions

■ SHA-512 mainly have three Boolean functions, $\mathrm{XOR}, \mathrm{IF}$ and MAJ are given by

$$\begin{aligned} \mathrm{XOR}(x, y, z) &= x \oplus y \oplus z, \\ \mathrm{IF}(x, y, z) &= (x \wedge y) \oplus (x \wedge z) \oplus z, \\ \mathrm{MAJ}(x, y, z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z). \end{aligned}$$

■ $\nabla w = XOR(\nabla x, \nabla y, \nabla z)$

For $\nabla w = \mathrm{XOR}(\nabla x, \nabla y, \nabla z)$, consider the propagation rule `[n==n]`, where: $\nabla x[i] =$ `[n]`, $\nabla y[i] =$ `[=]` $\nabla w[i] =$ `[=]`, $\nabla w[i] =$ `[n]`.

1. In the fast model: `[n==n]`
2. In the full model: `[n==n0**]`

Both model have the condition

$$x[i] = 0, y[i] \oplus z[i] = 0.$$

Both models do not capture the bit conditions $y[i] \oplus z[i] = 0$!!!

Modeling the Two-Bit Conditions of Boolean Functions

Definition

In our cryptanalysis of SHA-512, a condition controlling difference propagation is called a **2-bit condition** if it takes the form of either $a = b$ or $a \neq b$, where $a, b \in \mathbb{F}_2$.

To capture the 2-bit conditions, we slightly modify the propagation rules of Boolean functions in the full model.

1. In the fast model: `[n==n]`
2. In the full model: `[n==n0**]`
3. In the modified full model: `[n==n0**1]`

Specifically, we consider the possible values of the following tuple by adding an extra flag variable $flag[i]$:

$$(\nabla x[i], \nabla y[i], \nabla z[i], \nabla w[i], x[i], y[i], z[i], flag[i]).$$

If the propagation rule implicitly involves a 2-bit condition, then $flag[i] = 1$; otherwise, $flag[i] = 0$.

Modeling the Two-Bit Conditions of Boolean Functions

The full model for the Boolean functions XOR, IF and MAJ

Rules for XOR
$(\nabla x[i], \nabla y[i], \nabla z[i], \nabla w[i], x[i], y[i], z[i], flag[i])$
[====,***,0],
[n==n,0**,1],[n==u,0**,1],[u==u,1**,1],[u==n,1**,1],
[=n=n,*0*,1],[=n=u,*0*,1],[=u=u,*1*,1],[=u=n,*1*,1],
[==nn,**0,1],[==nu,**0,1],[==uu,**1,1],[==un,**1,1],
[nn==,00*,0],[n=n=,0*0,0],[=nn=,*00,0],[nu==,01*,0],[n=u=,0*1,0],[=nu=,*01,0],
[uu==,11*,0],[u=u=,1*1,0],[=uu=,*11,0],[un==,10*,0],[u=n=,1*0,0],[=un=,*10,0],
[nuun,011,0],[nunu,010,0],[nnuu,001,0],[nnnn,000,0],
[unnu,100,0],[unun,101,0],[uunn,110,0],[uuuu,111,0].

Rules for IF
$(\nabla x[i], \nabla y[i], \nabla z[i], \nabla w[i], x[i], y[i], z[i], flag[i])$
[====,***,0],
[n===,0**,1],[=n==,00*,0],[==n=,1*0,0],[==nn,0*0,0]
[u===,1**,1],[=u==,01*,0],[==u=,1*1,0],[==uu,0*1,0],
[nn==,001,0],[n=n=,000,0],[n==n,010,0],
[nu==,010,0],[n=u=,011,0],[n==u,001,0],
[uu==,111,0],[u=u=,101,0],[u==u,110,0],
[un==,100,0],[u=n=,110,0],[u==n,101,0],
[=n=n,10*,0],[=u=u,11*,0],[=nnn,*00,0],[=uuu,*11,0],
[=nuu,001,0],[=unn,010,0],[=nun,101,0],[=unu,110,0],
[nn=n,00*,0],[nnu=,001,0],[n=uu,001,0],[nun=,010,0],[nu=u,011,0],[n=nn,010,0],
[uu=u,110,0],[uun=,110,0],[u=nn,100,0],[unu=,101,0],[un=n,101,0],[u=uu,111,0],
[nuuu,011,0],[nnnn,000,0],[unnn,100,0],[uuuu,111,0].

Rules for MAJ
$(\nabla x[i], \nabla y[i], \nabla z[i], \nabla w[i], x[i], y[i], z[i], flag[i])$
[====,***,0],
[==u=,**1,1],[=u==,*1*,1],[u===,1**,1],[n===,0**,1],[=n==,*0*,1],[==n=,**0,1],
[n==n,0**,1],[u==u,1**,1],[=n=n,*0*,1],[=u=u,*1*,1],[==nn,**0,1],[==uu,**1,1],
[u=n=,1*0,0],[n=u=,0*1,0],[un==,10*,0],[nu==,01*,0],[=nu=,*01,0],[=un=,*10,0],
[=nnn,*00,0],[=uuu,*11,0],[n=nn,0*0,0],[u=uu,1*1,0],[nn=n,00*,0],[uu=u,11*,0],
[nnun,001,0],[uunu,110,0],[unuu,101,0],[unnn,100,0],[nuuu,011,0],[nunn,010,0],
[nnnn,000,0],[uuuu,111,0].

[*] represents the 2-bit condition.

Search for the new 31-step differential trail

1. **Find a solution of** $(\nabla W_i)_{0\leq i\leq 30}$ **with the minimal** $\mathbf{H}(\nabla W_{16})$ **and the minimal** $\mathbf{H}(\nabla W_{18})$.
2. **Find the minimal differential conditions on** $(E_i)_{14\leq i\leq 16}$.
3. **Find the minimal Hamming weight and 2-bit conditions of** $(A_i)_{0\leq i\leq 30}$.
4. **Find the minimal Hamming weight of** $(E_i)_{0\leq i\leq 30}$.
5. **Detection free bit value of** $(A_i)_{3\leq i\leq 12}$**.**

The Collision Attacks on 31-Step SHA-512

i	∇A_i	∇E_i	∇W_i
-4	==	==	
-3	==	==	
-2	==	==	
-1	==	==	
0	==	==	==
1	==	==	==
2	==	==	==
3	==	==0============1===========110=0=====0======1===11=0============	==
4	==	==1===10====11=0=====1==0=0011=1=====1==10==0===0000=1=1==0===1=	==
5	==n==0==nu0=====u======u======uu====uu======u====n=u=======u===1	01n0=0111n00111un0=1=0un110unn=n00=1uu010000u=10nu0u00001=1u1100	==n======n=====un======u=====u=nunnnnu======u=====nn=======u====
6	=====0====0=====u================u=u======u===n====n=u====u====1	=100n0uu0101nu1u00111u00u0n0nu1u00u11011un00u00001nn1u0n00n000n1	n0u011=01u=====0n====10n====un=============n====uu===u=u=nuuuuu=
7	=n==0====nnnu==u=u=========u====u=nn======n=uuu==un=====u==u===1	1nnu1un0n0=u0001un0001uu011n000nu00=u10100n1111=10000110n000nn1n	=u=============n============u====u==unn==n====u=======n=0=1====u
8	====0=====0==u	01011100100001u11001=u0n11n010111n00100n10nu0un=n0n01000n=0=1110	n=====nu=====u=====nu=====n======u=====u======u==========0======
9	==========0=============n======================================0	00n1=01011=1=10=011101unn01101u10001u1u10==1=11=011==1==11=11111	============u============n============n========================n
10	============u===========nu============n======================nuu	==10==0====01=10u0==101=0n0==01==11010n11=1u=000100=1n==1110=nuu	==
11	==	=0u1======un0=n=nn===01001u==un====unnn0n==n=========u==u=00=100	==
12	==	=000======00n=1=11===10==u1==00===1010u11==1=========1==1==0=11u	==
13	==	==1=======110=1=01=======11==11====1111=1==0=========0==1======1	==
14	==	============u============n============n========================n	==
15	==	============0============0============0========================0	==
16	==	============1============1============1========================1	=======unnnnn====nuuuuuuuu=======nuuuuu==========11====nuuuuuuuu
17	==	==	==
18	==	==	============n============u============u========================u
19	==	==	==
20	==	==	==
21	==	==	==
22	==	==	==
23	==	==	==
24	==	==	==
25	==	==	==
26	==	==	==
27	==	==	==
28	==	==	==
29	==	==	==
30	==	==	==
31	==	==	==

- $n = 64, n_1 = 20, n_2 = 27, n_3 = 36, n_4 = 17, \mathcal{N}_{\texttt{start}} = 2^{10.7}, \mathcal{N}_{\texttt{pro}} = 65,\ \beta \approx 0.9$.
- Time Complexity: $2^{85.5}$, Memory Complexity: $2^{44.4}$.

29-step Collision Attacks on SHA-512

Finding a valid attack requires attackers to finish the following three tasks:

Three tasks

- Task 1: Select the message difference to construct a local collision;
- Task 2: Search for a corresponding differential trail in (W_i, A_i, E_i);
- Task 3: Find a colliding message pair based on the differential trail.

Detailed Analysis of the Message Expansion in SHA-512

According to the SHA-2 message expansion, when $i \geq 16$,

$$W_i = \sigma_1(W_{i-2}) \boxplus W_{i-7} \boxplus \sigma_0(W_{i-15}) \boxplus W_{i-16}.$$

Analysis of this equation reveals that (W_{i-15}, W_{i-16}) are adjacent, W_i and W_{i-2} has distance 2, and W_i and W_{i-7} has distance 7. So, if we introduce difference in two consecutive message words (W_i, W_{i+1}), they will cause differences in $(W_{i+15}, W_{i+16}, W_{i+17})$.

Relationship between (W_i, W_{i+1}, W_{i+5}) and local collisions

$(i, i+1, i+5)$	local collision	relationship	attacked steps
$(9, 10, 14)$	0-28	W_{14} updates W_{29}	29
$(10, 11, 15)$	0-29	W_{15} updates W_{30}	30

The time complexity remains impractical for a 30-step collision attack!!!

Detailed Analysis of the Message Expansion in SHA-512

Based on the above analysis, injecting differences in the expanded message words

$$(W_9, W_{10}, W_{14}, W_{17}, W_{19})$$

can create a local collision spanning 11 steps (steps 9 to 19) in the message expansion, which allows a collision attack on 29-step SHA-512.

											W_0	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}	W_{11}	W_{12}	W_{i-16}
											W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}	W_{11}	W_{12}	W_{13}	$\sigma_0(W_{i-15})$
											W_9	W_{10}	W_{11}	W_{12}	W_{13}	W_{14}	W_{15}	W_{16}	W_{17}	W_{18}	W_{19}	W_{20}	W_{21}	W_{i-7}
											W_{14}	W_{15}	W_{16}	W_{17}	W_{18}	W_{19}	W_{20}	W_{21}	W_{22}	W_{23}	W_{24}	W_{25}	W_{26}	$\sigma_1(W_{i-2})$
W_0	$\cdots$	W_7	W_8	W_9	W_{10}	W_{11}	W_{12}	W_{13}	W_{14}	W_{15}	W_{16}	W_{17}	W_{18}	W_{19}	W_{20}	W_{21}	W_{22}	W_{23}	W_{24}	W_{25}	W_{26}	W_{27}	W_{28}	W_i

Search for the 29-step differential trail

1. **Find a solution of $(\nabla W_i)_{0\leq i\leq 28}$ with the minimal $\mathbf{H}(\nabla W_{17})$ and the minimal $\mathbf{H}(\nabla W_{19})$.**
2. **Find the minimal number of differential conditions on $(E_i)_{15\leq i\leq 17}$.**
3. **Find the minimal Hamming weight of $\mathbf{H}(\nabla A_i)_{9\leq i\leq 17}$.**
4. **Find the minimal Hamming weight of $\mathbf{H}(\nabla E_i)_{9\leq i\leq 17}$.**
5. **Detect free bit value of $(A_i)_{7\leq i\leq 13}$.**

The 29-step differential trail

i	∇A_i	∇E_i	∇W_i
-4	==	==	
-3	==	==	
-2	==	==	
-1	==	==	
0	==	==	==
1	==	==	==
2	==	==	==
3	==	==	==
4	==	==	==
5	==	==	==
6	==	==	==
7	0010001111100100110100010010001111001010010001111=======00100010	======1====00=======11==1=1=100=====1===1===111=0==============0	==
8	0010000111100001111111011010100011101100011100011101010111100001	1=00==01===1101===1000=00=01010===100==00===011111===1==10====01	==
9	010unnu00nu0u01unnnn10n0u0u011110u0100nn1n000uunu0000110100unnnn	10111uu01=0nn1000=00uuu1uun1unnnnn01nu00uu1unnu1n00100=101=10+0u	unnnn=nnuuu=u====unn==n=u=u=unnnnn=nuu==uu==u=n=n=unnnnnnnnnnnnn
10	u0nn00nu0u0n11011nuuuuuuu1uuuuuuuuuuuu1nuuuuuuu0un001001n1110000	u1uu01un110u0nu000unnn1u011n10n101nu100u01=10unnuu100n0=un111nn1	======unnnnnn=====nuuuuuuu=========unnn===nuuuuuuuuuuuuuuuuuuuuu
11	unnnnnnnnnnnn000011110100n00001011010un0110010nuuuuuuuuuuuuuuuuu	1nu100101==1n000=10001010n1nuuu1110011001001000100==0u==1011=0un	==
12	1111101101010010111101001010001011101010000011111100100111110001	10u0==01==1001uu1n100nuuu0u000u==1100=10u=1n101101010n01u0=01001	==
13	010010111001100000==0110110100=======0001=0100011111011010001110	=01=========0=10=0===1011u1011========1=0==1=========0==0=====10	==
14	==	==00========n=0000===111100===1========n=0=01=========1==0======u	========nu=============nnnnnnnnnnnnn================unnnnn=un===
15	==	============u============n============u========================n	==
16	==	============0============0============0========================0	==
17	==	============0============1============0========================0	=======unnnnnn====nuuuuuuuu=======unnnnnn==========11====nuuuuuuuu
18	==	==	==
19	==	==	============n============u============n========================u
20	==	==	==
21	==	==	==
22	==	==	==
23	==	==	==
24	==	==	==
25	==	==	==
26	==	==	==
27	==	==	==
28	==	==	==

Message modification

1. Find a valid solution in the expanded message words $W_9 - W_{14}$, as well as the state words $(A_i)_{1 \le i \le 14}$ and $(E_i)_{5 \le i \le 14}$ by the step update function. We call such a valid solution as **starting points**.
2. After determining the message words and state words in Step 1, the remaining the message words $(W_0, \ldots, W_8, W_{15})$ have not been fixed. At this step, our primary goal is to use (W_0, . . . , W_8) to connect the state words with the initial value IV. Before using $(W_0, \ldots, W_7)$ to connect the state words with the IV, we must first determine W_8.
3. At this point, $(W_0, \ldots, W_{14})$ have already been fixed, and only W_{15} remains unfixed. We can use the degree of freedom of W_{15} to satisfy the conditions on $(E_{15}, E_{16}, E_{17}, W_{17}, W_{19})$. If all conditions are satisfied, the colliding message pair will be found. Otherwise, go back to Step 2 and choose new W_8. If W_8 is used, choose a new starting point.

Details of Step 3.

We propose a novel approach to efficiently exploit the degrees of freedom of W_{15} to fulfill the remaining conditions. Up to this, $(W_0, \ldots, W_{14})$ have already been fixed, and there are 56 conditions in W_{17} in total. And we can obtain all 2^8 possible values for W_{17}, which are stored in table $\mathtt{TAB}_{w_{17}}$. We first verify W_{19} based on the update function

$$W_{19} = \sigma_1(W_{17}) \boxplus W_{12} \boxplus \sigma_0(W_4) \boxplus W_3,$$

where W_{17} can be obtained by exhaustively checking $\mathtt{TAB}_{w_{17}}$, and W_{12}, W_4, W_3 are known. Once the conditions on W_{19} are satisfied, the bit conditions on (E_{15}, E_{16}, E_{17}) can be verified based on the update function

$$W_{17} = \sigma_1(W_{15}) \boxplus W_{10} \boxplus \sigma_0(W_2) \boxplus W_1,$$

where $(W_1,\ W_2,\ W_{10})$ have already been fixed in Step 2.

Complexity Evaluation.

The probability of Step 1 and Step 2 being satisfied is 1. In Step 3, we need to satisfy 88 conditions in $(E_{15}, E_{16}, E_{17}, W_{17}, W_{19})$, of which 56 conditions are in W_{17}. To satisfy these conditions, we first determine all possible values of W_{17}, i.e., there are 2^8 possible values for W_{17}. Once W_{17} is satisfied, there are 32 remaining conditions in $(E_{15}, E_{16}, E_{17}, W_{19})$ that need to be met. Therefore, the overall time complexity is 2^{32} and the memory complexity is 2^8.

The colliding message pair for 29-step SHA-512

IV	6a09e667f3bcc908	bb67ae8584caa73b	3c6ef372fe94f82b	a54ff53a5f1d36f1
	510e527fade682d1	9b05688c2b3e6c1f	1f83d9abfb41bd6b	5be0cd19137e2179
M	36fc57878e6a1478	39356d4e68533f81	11720aae7e5496f3	e25446d46d336ce3
	3fbe62e17052367f	65eb73407a88f8bb	def9586059b730a8	72e21b64757e2d03
	00d43e0b169d0ea7	7b173317d3029fff	85f92000ef600000	70cdb9b71952cc80
	43f82cfa8e26489d	f0c2a87d655d9c26	9a8cfbfff7d847cc	daffd76f9b8e668a
M′	36fc57878e6a1478	39356d4e68533f81	11720aae7e5496f3	e25446d46d336ce3
	3fbe62e17052367f	65eb73407a88f8bb	def9586059b730a8	72e21b64757e2d03
	00d43e0b169d0ea7	80ff41b80fc82000	86011fc0f15fffff	70cdb9b71952cc80
	43f82cfa8e26489d	f0c2a87d655d9c26	9a4cfa0007d84814	daffd76f9b8e668a
hash	fc22023fba9ae4af	87e29a5cfa5346ad	16ba7265981828ca	407a30473e590c97
	9d4ce7c9ce70d936	f621d63828584973	dd1ce5282e8f1f08	780750f1be08fabf

Application to 29-step SHA-256

SHA-256 and SHA-512 share a similar structure, with the primary difference being the size of the state words and the Boolean function σ and Σ. We can also apply similar message word selection methods and message modification techniques to SHA-256, enabling the generation of practical collision message pairs for 29 steps of SHA-256.

The colliding message pair for 29 steps of SHA-256

IV	6a09e667	bb67ae85	3c6ef372	a54ff53a	510e527f	9b05688c	1f83d9ab	5be0cd19
M	02faff1b	d7e755b4	27138a63	b70c6987	8b4ceb5d	64c30d15	5a315ded	b5b3ec6a
	2977f996	6ce55306	5baf40c9	e3b173bc	151b8802	e1991d09	dcbc83e9	55189e54
M′	02faff1b	d7e755b4	27138a63	b70c6987	8b4ceb5d	64c30d15	5a315ded	b5b3ec6a
	2977f996	5fdd4384	5d8f37c8	e3b173bc	151b8802	e1991d09	e21c9b33	55189e54
hash	b6631f1f	071314ff	56cb6d39	6a6f192c	12509316	cc8f897c	0c916a47	e76f6ba1

Summary of (semi-free-start) Collision Attacks on SHA-2

State size	Hash size	Attack type	Steps	Time	Memory	Year
256	All	collision	28	*practical*	\	2013
			29	***practical***	\	**2025**
			31†	$2^{65.5}$	2^{34}	2013
			31†	$2^{49.8}$	2^{48}	2023
			31†	*practical*	$2^{19.8}$	2024
		SFS collision	38	*practical*	\	2013
			39	*practical*	\	2023
512	All	collision	27	*practical*	\	2015
			28	*practical*	\	2023
			29	***practical***	\	**2025**
			31†	$2^{115.6}$	$2^{77.3}$	2023
			31†	$2^{97.3}$	$2^{35.2}$	2024
			31†	$2^{85.5}$	$2^{44.4}$	**2025**
		SFS collision	38	*practical*	\	2014
			39	*practical*	\	2015

† It is a two-block collision.