The First Practical Collision for 31-Step SHA-256

Yingxin Li[1,2], Fukang Liu[2], Gaoli Wang[1], Xiaoyang Dong[3], Siwei Sun[4]

[1]East China Normal University
[2]Tokyo Institute of Technology
[3]Tsinghua University
[4]University of Chinese Academy of Sciences

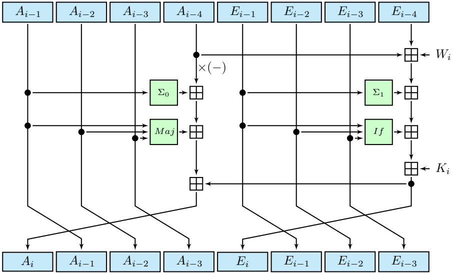
December, 2024

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, e.g. SHA-256 is used in Bitcoin.

Compression Functions of SHA-256



- Step function

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

Compression Functions of SHA-256

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

$$
\begin{aligned}
\mathrm{IF}(x, y, z) &= (x \wedge y) \oplus (x \wedge z) \oplus z, \\
\mathrm{MAJ}(x, y, z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z), \\
\Sigma_0(x) &= (x \ggg 2) \oplus (x \ggg 13) \oplus (x \ggg 22), \\
\Sigma_1(x) &= (x \ggg 6) \oplus (x \ggg 11) \oplus (x \ggg 25).
\end{aligned}
$$

Compression Functions of SHA-256

■ Message expansion

The message expansion of SHA-256 splits the 512-bit message block M_j into 16 words m_i, $i = 0, \cdots, 15$, and expands them into 64 expanded message words W_i

$$W_i = \begin{cases} m_i & 0 \leq i \leq 15, \\ \sigma_1(W_{i-2}) \boxplus W_{i-7} \boxplus \sigma_0(W_{i-15}) \boxplus W_{i-16} & 16 \leq i \leq 63. \end{cases}$$

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

$$\begin{aligned} \sigma_0(x) &= (x \ggg 7) \oplus (x \ggg 18) \oplus (x \gg 3), \\ \sigma_1(x) &= (x \ggg 17) \oplus (x \ggg 19) \oplus (x \gg 10). \end{aligned}$$

Collision Attacks on SHA-2

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

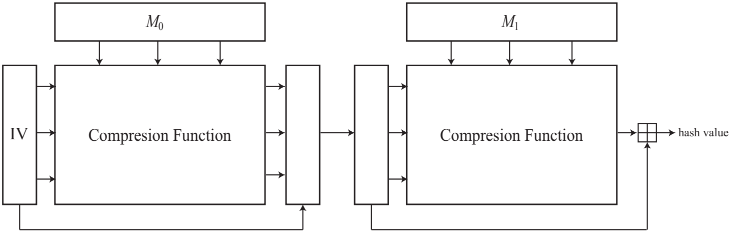
Three tasks

- 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.

Our contribution is Task 3:

- Find a colliding message pair based on the 31-step differential trail.

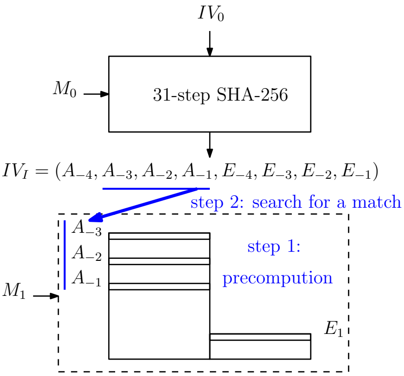
2-Block Attack overview



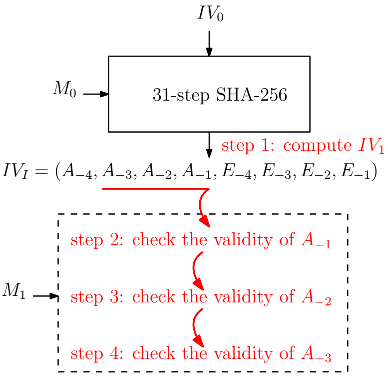
The 2-block collision attack against SHA-2 was first proposed by Mendel at Eurocrypt 2013. In the 2-block method, the difference appears in the second block. The output value of the first block is used as the input for the second block, and the differences in the second block cancel out, leading to a collision.

New Collision Attack framework for 31-step SHA-2

- Our two-phase memory-efficient collision attack framework:
  - Pre-processing phase;
  - Matching phase.



Mendel et al.'s MITM technique



Our new technique

Pre-processing Phase

This phase mainly to find valid solutions of

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

by only considering

$$\begin{aligned} E_i &= A_{i-4} \boxplus E_{i-4} \boxplus \Sigma_1(E_{i-1}) \boxplus \mathrm{IF}(E_{i-1}, E_{i-2}, E_{i-3}) \boxplus K_i \boxplus W_i, \text{ for } 9 \leq i \leq 12. \\ A_i &= E_i \boxminus A_{i-4} \boxplus \Sigma_0(A_{i-1}) \boxplus \mathrm{MAJ}(A_{i-1}, A_{i-2}, A_{i-3}), \text{ for } 5 \leq i \leq 12. \end{aligned}$$

Among them, the distinct

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

can be chosen as ***starting points***, i.e., $(A_1, \ldots, A_4, E_5, \ldots, E_8)$ are distinct in these starting points.

Finding valid A_{-1} from each starting point.

For each obtained starting point, all possible (W_8, E_4) are exhausted to satisfy the following relation:

$$E_8 = A_4 \boxplus E_4 \boxplus \Sigma_1(E_7) \boxplus \mathrm{IF}(E_7, E_6, E_5) \boxplus K_8 \boxplus W_8.$$

For each valid pair (W_8, E_4) satisfying above equation, then the corresponding A_0 can be computed according to the following relation:

$$A_4 \quad = \quad E_4 \boxminus A_0 \boxplus \Sigma_0(A_3) \boxplus \mathrm{MAJ}(A_3, A_2, A_1).$$

For each valid tuple (W_8, E_4, A_0), all possible (E_3, W_7) are similarly exhausted to satisfy

$$E_7 = A_3 \boxplus E_3 \boxplus \Sigma_1(E_6) \boxplus \mathrm{IF}(E_6, E_5, E_4) \boxplus K_7 \boxplus W_7,$$

and the corresponding A_{-1} can be computed according to the following relation:

$$A_3 = E_3 \boxminus A_{-1} \boxplus \Sigma_0(A_2) \boxplus \mathrm{MAJ}(A_2, A_1, A_0),$$

Matching Phase

1. Try an arbitrary M_0, and get the corresponding chaining input

$$IV_1 = (A_{-4}, A_{-3}, A_{-2}, A_{-1}, E_{-4}, E_{-3}, E_{-2}, E_{-1})$$

   for the second message block.

2. **Checking A_{-1}, A_{-2}, A_{-3}:** The A_{-1} obtained from IV_1 is matched with the precomputed phase A_{-1}, and then check the validity of (A_{-2}, A_{-3}).

How to check the validity of (A_{-3}, A_{-2}) ?

Compute (E_0, E_1, E_2) to make the associated (A_0, A_1, A_2) consistent with those computed from this IV_1 according to the following 3 equations:

$$\begin{aligned} A_0 &= E_0 \boxminus A_{-4} \boxplus \Sigma_0(A_{-1}) \boxplus \mathrm{MAJ}(A_{-1}, A_{-2}, A_{-3}), \\ A_1 &= E_1 \boxminus A_{-3} \boxplus \Sigma_0(A_0) \boxplus \mathrm{MAJ}(A_0, A_{-1}, A_{-2}), \\ A_2 &= E_2 \boxminus A_{-2} \boxplus \Sigma_0(A_1) \boxplus \mathrm{MAJ}(A_1, A_0, A_{-1}). \end{aligned}$$

Then, compute (W_4, W_5, W_6) to make the associated (E_4, E_5, E_6) also consistent with those compted from this IV_1:

$$\begin{aligned} E_4 &= A_0 \boxplus E_0 \boxplus \Sigma_1(E_3) \boxplus \mathrm{IF}(E_3, E_2, E_1) \boxplus K_4 \boxplus W_4, \\ E_5 &= A_1 \boxplus E_1 \boxplus \Sigma_1(E_4) \boxplus \mathrm{IF}(E_4, E_3, E_2) \boxplus K_5 \boxplus W_5, \\ E_6 &= A_2 \boxplus E_2 \boxplus \Sigma_1(E_5) \boxplus \mathrm{IF}(E_5, E_4, E_3) \boxplus K_6 \boxplus W_6. \end{aligned}$$

Check whether all the conditions on $(E_0, E_1, E_2, W_4, W_5, W_6)$ hold.

Practical Collisions for 31-step SHA-256

i	∇A_i	∇E_i	∇W_i
−4	================================	================================	
−3	================================	================================	
−2	================================	================================	
−1	================================	================================	
0	================================	================================	================================
1	================================	================================	================================
2	================================	================================	================================
3	================================	==========================10====	================================
4	================================	============0===0=========01===0	================================
5	===================n=unnnnnnn=n=	0001110100011111110nu=11111unnnu1	================nuuu=======0=uu=
6	========n======================u	101011=11==0n0==u11110==1110011n	==========u=====u===u=======n===u
7	===u===n==n========n=========n=u	un0u1100n=01u11111001u1=n110u10n	=u=u=======n=====n=nu=n=====nun=
8	=============================n==	1u01un0u0=1=1=11n=0=u0=001001u0=	=u=nn==========u===u===u==1=====
9	================================	01100001110=0=010===00=11101u0=1	================u==========1=u==
10	================u============u==	=1n1uuuuu0100=1un0=10unnnnnnn010	================================
11	================================	=01u1010uu1==11100===1000001n=0=	================================
12	================================	==110001=11====1n====0011110n=0=	================================
13	================================	===0====01======1===============	================================
14	================================	================u===========0u==	================================
15	================================	================0============1==	================================
16	================================	================1============1==	=============unnnunnnnnnnnnnnn==
17	================================	================================	================================
18	================================	================================	==============1=n=0==========n==
19	================================	================================	================================
20	================================	================================	================================
21	================================	================================	================================
22	================================	================================	================================
23	================================	================================	================================
24	================================	================================	================================
25	================================	================================	================================
26	================================	================================	================================
27	================================	================================	================================
28	================================	================================	================================
29	================================	================================	================================
30	================================	================================	================================

- Pre-compute $2^{19.8}$ valid solutions of (A_{-1}, A_0, A_1, A_2, A_3, A_4, E_5, E_6, E_7, E_8). Store these tuples in a table.
- Try arbitrary M_0 and get corresponding values of $IV_1 = (A_{-4}, A_{-3}, A_{-2}, A_{-1}, E_{-4}, \ldots, E_{-1})$ to match A_{-1} from this table. And then check the validity of (A_{-3}, A_{-2}).
- Use the freedom in (W_{13}, W_{14}, W_{15}) to fulfill the remaining conditions.

■ Practical cost to find a collision: 1.2 hours with 64 threads.

■ Time Complexity: $2^{40.5}$; Memory Complexity: $2^{19.8}$.

Practical Colliding Message Pair for 31-step of SHA-256

Table 1: A colliding message pair for 31-step SHA-256

M_0	8ce3f805	5c401aed	579e5f7f	bc3116cb	ca189b3c	eb75f04c	958f0a0e	7760b082
	dcd5027d	32260ad6	7b12b659	eee66518	ad7f88dd	f8ad20bb	7ae40ffd	21609249
M_1	9abdeb1b	1f195f41	5a7210c1	55614f13	a2269dd1	be888a61	359257d4	adf3737b
	9f0484a6	eb830a58	66add94a	9669232d	45271fa5	b8f69585	428bbce3	0703b904
M_1'	9abdeb1b	1f195f41	5a7210c1	55614f13	a2269dd1	be887a67	35b2dfc5	fde32975
	c70595a6	eb838a5c	66add94a	9669232d	45271fa5	b8f69585	428bbce3	0703b904
hash	ff558659	2977dd01	54638843	35f8de84	a3336841	f4f476f2	7c571548	f7025605

Improve Collision Attacks on 31-Step SHA-512

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	======unnnnn====nuuuuuuuu========nuuuuu==========11====nuuuuuuuu
17	==	==	==
18	==	==	============n============u============u========================u
19	==	==	==
20	==	==	==
21	==	==	==
22	==	==	==
23	==	==	==
24	==	==	==
25	==	==	==
26	==	==	==
27	==	==	==
28	==	==	==
29	==	==	==
30	==	==	==

- Strategy of the improvement:
  - Use a new differential trail sparse in the probabilistically checking part.

Summary of (SFS) Collision Attacks on SHA-2

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