A Collision for 38-Step SHA-256

Omer Goldzweig
omer.goldzweig@gmail.com

15 November 2026

Abstract

We give a collision for 38-step SHA-256 from the standard initial value, to our knowledge the first. The attack is a two-block differential attack whose message differences are in the same words as in the 38-step semi-free-start collision of Mendel, Nad and Schläffer. Most state words of the second block are fixed in advance. For each first block, a table indexed by the first word of the chaining value supplies the first state words, and one lookup on a 32-bit key matches them with the last free state words so that the characteristic holds through step 17; the remaining steps hold by chance. The expected time is $2^{54.8}$ calls of the 38-step compression function, with $2^{57.5}$ table accesses that this count does not charge. The tables take 7.4 GB, and finding the fixed state words took 52,300 vCPU-hours. A search of 53 GPU-hours found the collision after about 10% of the expected work.

Keywords: SHA-256, collision attack, differential cryptanalysis, step-reduced hash functions.

AI-use designation: Predominantly AI-generated work. This work was developed through iterative research and writing by Anthropic Claude agents, mainly Claude Opus 5.5, under the author's direction. The author takes responsibility for the results and presentation.

1 Introduction

Collision attacks on step-reduced SHA-256 have advanced quickly in the last three years. After the 31-step attack of Mendel, Nad and Schläffer [MNS13], the SAT/SMT-based search for signed differential characteristics of Li, Liu and Wang [LLW24a] led to a practical 31-step collision [LLW+24b], to attacks on 36 and 37 steps [ZLGW26, LLWS26], to practical collisions for 35 and 36 steps [LLWS26, LZL+26], and to the first classical 38-step collision attack, with time $2^{104.3}$ [LZL+26]. In an earlier unpublished manuscript [Gol26], we lowered the estimate for 38 steps to $2^{83.6}$, and a recent preprint [GWZW26] estimates $2^{90.9}$ for 38 steps and $2^{108.8}$ for 39 steps. No collision for 38 steps from the standard initial value was known; semi-free-start collisions, in which the attacker chooses the chaining value, are known for 38 and 39 steps [MNS13, LLW24a, ANB24].

The collision attacks above are two-block attacks. A first block M_0 produces a chaining value $H = F_r(IV, M_0)$, and a pair of second blocks (M_1, M_1') that follows a differential characteristic from H gives the collision $F_r(H, M_1) = F_r(H, M_1')$. The cost is dominated by the conditions of the characteristic that have to be met by trial and error once H is known.

This paper gives a collision for 38 steps. Its message differences are in the same words as in the 38-step semi-free-start collision of [MNS13]: W_7, W_8, W_{10}, W_{15} and the expanded words W_{23}, W_{24}. In [MNS13], a characteristic gives a collision only when enough message words precede its local collision to connect it to the chaining value; this local collision begins at step 7, so they give only a semi-free-start collision. We connect to the chaining value differently. We fix the state words $A_3, \ldots, A_{13}$ of both executions, the *core*, and compute $W_0, \ldots, W_7$ from the chaining value and the state (Section 2.1). The state differences through step 13 then hold for every chaining value, and the chaining value enters the attack only through the message words that it determines.

One of the resulting conditions involves only the first word H_0 of the chaining value. A table indexed by H_0 returns, for each first block, the choices of A_0, A_1, A_2 that satisfy it, about 108 *lower solutions*. Up to step 17, a lower solution and the last free words (A_{14}, A_{15}), the *upper pair*, interact only through one 32-bit word, the *key*, so a precomputed table of 8.5×10^8 entries (key, upper pair) meets all conditions up to step 17 with one lookup. A lower solution matches 0.20 table entries on average, and these *matches* satisfy the remaining conditions, of steps 18 to 24 and of the message expansion, with an average probability of $2^{-54.7}$. Since most lower solutions match no entry, a lower solution costs 0.21 calls of the compression function on average.

Contributions.

1. A modular characteristic for 38 steps with its message differences in the same words as [MNS13] and with different state differences in steps 7 to 17. Its digits at steps 14 to 17 are chosen so that carries merge neighbouring digits, which lowers the number of conditions on the upper pair (Section 3).
2. An attack that fixes most of the second block in advance and, in addition to the usual table over the chaining value, uses a second table indexed by a 32-bit key, so that the conditions up to step 17 are met by lookups. Its success probability is counted up to step 19 and estimated beyond (Sections 4 to 7).
3. A GPU implementation of the attack and a search of 53 GPU-hours that found the collision (Sections 8 and 9).

Table 1. Collision attacks on 38-step SHA-256 from the standard IV. Time is in calls of the 38-step compression function (Section 2.3), except for [Gol26], which converts arithmetic to evaluations and charges every table access as one evaluation, and [GWZW26], which times the attack, table lookups included, in units of one scalar compression; the bracketed figure in the last row uses the convention of [Gol26] (Section 7.1). For this work, memory is that of the tables (Section 7.2).

Time	Memory	Status	Reference
$2^{104.3}$	negligible	estimate	[LZL+26]
$2^{83.6}$	17 GiB	estimate	[Gol26]
$2^{90.9}$	17 GiB	estimate	[GWZW26]
$2^{54.8}$ ($2^{57.6}$)	$2^{29.7}$ table entries, 7.4 GB (6.8 GiB)	collision found	this work

Prior work. The 38-step characteristic of [LZL+26] has message differences in $W_7, \ldots, W_{11}$, W_{15}, W_{16}, W_{23} and W_{25}, and its attack has negligible memory. Ours has differences in four message words and two expanded words (Section 3). Precomputed solutions of the middle steps, stored in a table indexed by words of the chaining value and matched against first blocks, go back to [MNS13] and are used in [LLW24a, ZLGW26, LLWS26]. In these attacks the conditions after the matched part are met by a search over the remaining free message words or by chance. We add a second table, indexed by a 32-bit key, that matches the last free state words to each lower solution, so that the conditions up to step 17 are met by lookups. The attacks of [Gol26] already used a fixed core, a table of the preimages of $\sigma_0(z) - z$ to force W_{16}, and a join of two lists on E_{16}. The present paper is self-contained and can be read independently of [Gol26].

2 Preliminaries

2.1 The alternative description of SHA-256

All words have 32 bits, all additions are modulo 2^{32}, and $x[i]$ is bit i of x with bit 0 the least significant. We use the functions of FIPS 180-4 [Nat15]:

$$\Sigma_0(x) = x^{\ggg 2} \oplus x^{\ggg 13} \oplus x^{\ggg 22}, \qquad \Sigma_1(x) = x^{\ggg 6} \oplus x^{\ggg 11} \oplus x^{\ggg 25},$$

$$\sigma_0(x) = x^{\ggg 7} \oplus x^{\ggg 18} \oplus (x \gg 3), \qquad \sigma_1(x) = x^{\ggg 17} \oplus x^{\ggg 19} \oplus (x \gg 10),$$
$$\mathrm{Ch}(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z), \qquad \mathrm{Maj}(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z).$$

A block consists of 16 words $W_0, \ldots, W_{15}$ and is expanded by

$$W_t = \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16} \qquad (t \geq 16). \tag{1}$$

The state is described by two sequences A_t, E_t [MNS11, LLW24a]. A chaining value $H = (H_0, \ldots, H_7)$ initialises $(A_{-1}, A_{-2}, A_{-3}, A_{-4}, E_{-1}, E_{-2}, E_{-3}, E_{-4}) = (H_0, \ldots, H_7)$, and step $t = 0, 1, \ldots$ computes

$$E_t = A_{t-4} + E_{t-4} + \Sigma_1(E_{t-1}) + \mathrm{Ch}(E_{t-1}, E_{t-2}, E_{t-3}) + K_t + W_t, \tag{2}$$
$$A_t = E_t - A_{t-4} + \Sigma_0(A_{t-1}) + \mathrm{Maj}(A_{t-1}, A_{t-2}, A_{t-3}). \tag{3}$$

After r steps the compression function returns $F_r(H, M) = (A_{r-1} + H_0, \ldots, A_{r-4} + H_3, E_{r-1} + H_4, \ldots, E_{r-4} + H_7)$. Both equations can be inverted:

$$E_t = A_t + A_{t-4} - \Sigma_0(A_{t-1}) - \mathrm{Maj}(A_{t-1}, A_{t-2}, A_{t-3}), \tag{4}$$
$$W_t = E_t - A_{t-4} - E_{t-4} - \Sigma_1(E_{t-1}) - \mathrm{Ch}(E_{t-1}, E_{t-2}, E_{t-3}) - K_t. \tag{5}$$

A chaining value H and the words $A_0, \ldots, A_{15}$ therefore determine the block $W_0, \ldots, W_{15}$. For $t \geq 8$, W_t depends only on $A_{t-8}, \ldots, A_t$, so $W_8, \ldots, W_{15}$ do not depend on H. The chaining value enters through $W_0, \ldots, W_7$ only, and triangularly: for fixed $A_0, \ldots, A_7$,

$$W_{7-i} = \kappa_i(H_0, \ldots, H_{i-1}) - H_i \qquad (i = 0, \ldots, 7),$$

where the functions κ_i depend on $A_0, \ldots, A_7$: in (5) for $t = 7 - i$, the word H_i enters only through the term $-E_{3-i}$, as the summand A_{-1-i} of (4) if $i \leq 3$ and as E_{3-i} itself if $i \geq 4$, and the other terms depend only on $H_0, \ldots, H_{i-1}$. Hence the map $H \mapsto (W_0, \ldots, W_7)$ is a bijection, and for a uniform H the words $W_0, \ldots, W_7$ are independent and uniform. We write $W_7 = \theta - H_0$, with $\theta = \kappa_0$ a constant that depends on $A_0, \ldots, A_7$.

2.2 Differences

For a word pair (x, x') we write $\delta x = x' - x \bmod 2^{32}$, and for the message words of the two blocks also $\Delta W_t = \delta W_t$. For a function f we write $\Delta f(x; d) = f(x + d) - f(x)$; this is the modular difference of $f(x)$ when x changes by $+d$, and it depends on x. We write a modular difference as a sum of *digits* $\pm 2^i$ (signed bit differences); for example $-\texttt{0x21000} = -2^{17} - 2^{12}$. A *modular characteristic* for r steps prescribes δA_t, δE_t and δW_t for every step, where the prescription for a state word may be a set of values. A pair of executions (a *block pair*) *follows* it up to step t if these differences are as prescribed for all steps up to t. The *depth* of a block pair is the first step in which its differences are not as prescribed, or r if there is none; a block pair of depth at least t follows the characteristic in all steps before t. The *first execution* uses the block M_1 and the *second* the block M_1', which differs from M_1 by ΔW word by word modulo 2^{32}. If $\delta A_t = \delta E_t = 0$ for the last four steps, the block pair collides, since $F_r(H, \cdot)$ is H plus these eight words.

The functions σ_0, σ_1 are linear over $\mathbb{F}_2$ and bijective. With the XOR difference $D = (x + d) \oplus x$ and $Y = \sigma(D)$,

$$\Delta\sigma(x; d) = Y - 2\,(\sigma(x) \wedge Y), \tag{6}$$

because $\sigma(x + d) = \sigma(x) \oplus Y$ and $v \oplus Y = v + Y - 2(v \wedge Y)$.

2.3 Attack model and cost convention

A collision of r-step SHA-256 is a pair of distinct messages with equal r-step hash from the standard IV. Our attack chooses a first block M_0, computes $H = F_r(IV, M_0)$, finds $M_1 \neq M_1'$ with $F_r(H, M_1) = F_r(H, M_1')$, and outputs $M_0\|M_1$ and $M_0\|M_1'$.

Time. As in the recent attacks on reduced-step SHA-256 [ZLGW26, LLWS26, LZL+26], time is counted in calls of the r-step compression function F_r, and one call is charged for every first block M_0 and for every match (a lower solution together with a table entry for its key, Section 4.4). Table accesses are not charged, as in [ZLGW26]; we report their number beside the time. The precomputation and the memory are stated separately.

3 The characteristic

3.1 Message difference and expansion conditions

The message difference is nonzero exactly in W_7, W_8, W_{10} and W_{15}:

$$\Delta W_7 = w, \qquad \Delta W_8 = -4, \qquad \Delta W_{10} = d, \qquad \Delta W_{15} = 4,$$

and exactly W_{23} and W_{24} among the expanded words have a difference, $\Delta W_{23} = -a$ and $\Delta W_{24} = -4$. The constants w, d and a belong to the core (Section 4). For the core used here $w = \texttt{0x31de0240}$, $d = \texttt{0x00028000}$ and $a = \texttt{0xf8010000} = -\texttt{0x07ff0000}$; the constant a is δA_{15} (Section 3.2). With these differences, equation (1) gives the conditions of Table 2. They are the complete list: every other expanded word up to W_{37} has no differing term, or its differing terms cancel by one of the conditions.

Table 2. Conditions on the message expansion.

t	Condition	Paid by
17	$\Delta\sigma_1(W_{15}; 4) = -d$	the upper pair
22	$\Delta\sigma_0(W_7; w) = -4$ (condition V)	the first word of the chaining value ($W_7 = \theta - H_0$)
23	$\Delta\sigma_0(W_8; -4) = -a - w$	the lower part (W_8 takes one of 4 values)
24	$\Delta W_{24} = \Delta W_8 = -4$	automatic
25	$\Delta\sigma_1(W_{23}; -a) + \Delta\sigma_0(W_{10}; d) = 0$	probability $2^{-10.34}$
26	$\Delta\sigma_1(W_{24}; -4) + d = 0$	probability $2^{-2.75}$
30	$\Delta\sigma_0(W_{15}; 4) = a$	the upper pair
31	$\Delta W_{31} = \Delta W_{24} + \Delta W_{15} = 0$	automatic

The probabilities in rows 25 and 26 are exact: counted over all 2^{32} values of W_{23}, resp. W_{24}, the first holds for 3,276,800 or 3,309,568 of them according to $\tau = -\Delta\sigma_0(W_{10}; d)$ (the two best of the possible values of τ, which depends on the variant of the core, Section 4), and the second for 637,534,208. The set $V_7 = \{x : \Delta\sigma_0(x; w) = -4\}$ has 2^{14} elements and is a union of 512 aligned blocks of 32 consecutive values. Exactly four values of W_8 satisfy row 23. About one value of W_{15} in 2^3 satisfies row 30 and one in $2^{2.75}$ satisfies row 17.

As in [MNS13], the difference in W_{15} enters W_{17}, W_{22}, W_{30} and W_{31} and is cancelled there by ΔW_{10}, by $\Delta\sigma_0(W_7; w)$, by ΔW_{23} and by ΔW_{24} (rows 17, 22, 30 and 31 of Table 2).

This characteristic does not extend to 39 steps: $W_{38} = \sigma_1(W_{36}) + W_{31} + \sigma_0(W_{23}) + W_{22}$ has the term $\sigma_0(W_{23})$ with the nonzero difference $\Delta\sigma_0(W_{23}; -a)$, so $\delta E_{38} \neq 0$ and the block pair does not collide after 39 steps.

3.2 State differences

The state differences of steps 7 to 13 depend on the core (Section 4): they are those of the chosen values. From step 14 on the characteristic prescribes the differences of Table 3, with $s = \pm 1$ a sign that is fixed by the core.

Table 3. The end of the characteristic. $\epsilon_i = \pm 1$. The set of δE_{16} has 8 elements and the set of δE_{17} has 4.

t	δA_t	δE_t	ΔW_t
14	0	e_{14} (a constant of the core; $-$`0x21000` here)	0
15	a	$-s\,2^9$	4
16	4	$\epsilon_1 2^{21} + s\,2^9 + \epsilon_2 2^5 + \epsilon_3 2^2$	0
17	0	$\epsilon_4 2^{27} + \epsilon_5 2^{23} + s\,2^9$	0
18	0	0	0
19	0	a	0
20	0	4	0
21, 22	0	0	0
23	0	0	$-a$
24	0	0	-4
≥ 25	0	0	0

The expanded differences cancel the state differences of steps 19 and 20. By (2), $\delta E_{23} = \delta E_{19} + \Delta W_{23} = 0$ provided that $\mathrm{Ch}(E_{22}, E_{21}, E_{20})$ has no difference, that is, that E_{22} is 1 in the bits in which E_{20} and E'_{20} differ. Since E_{21}, E_{22} and E_{23} then have no difference, $\delta E_{24} = \delta E_{20} + \Delta W_{24} = 0$ holds without further conditions: a block pair that follows the characteristic through step 23 also follows it at step 24, and no block pair has depth 24. From step 21 on there is no difference in A_t and E_t, so a block pair that follows the characteristic collides. The expansion conditions of rows 25 and 26 of Table 2 are the last two conditions: the other rows are met before step 18, and once steps up to 24 hold, every later word difference is zero if $\Delta W_{25} = \Delta W_{26} = 0$.

The characteristic of [MNS13] has the same differences, up to sign, in A_{15}, A_{16}, E_{19} and E_{20}, but different differences in $A_7, \ldots, A_{13}$ and in $E_7, \ldots, E_{17}$; for example, it has $\delta E_{14} = 0$. We chose the digits of Table 3 as follows. In step 18 the function Ch passes a digit of δE_{16} to the next step when the corresponding bit of E_{17} is 1, and passing a digit costs the same number of bit conditions as blocking it. Choices of this kind let δE_{14} absorb a digit next to a digit that it must contain anyway, so that two digits merge into one by a carry; the same is done at step 21 for δE_{17} (the digit at bit 27 of δE_{17} comes from merging two digits), and the digits at bit 9 of δE_{15}, δE_{16} and δE_{17} come from a further merge. Here δE_{14} has two digits, at bits 12 and 17, and E_{14} and E'_{14} differ exactly in these two bits (Section 4.3). Under the three rotations of Σ_1 their images fall on bits 1, 6, 19 and 6, 11, 24, and the two at bit 6 cancel, so the XOR difference of $\Sigma_1(E_{14})$, which the conditions on A_{14} must cancel, has four bits. An earlier choice of digits for the same message difference left nine bits; with it, the probability that a block pair of depth at least 19 is a collision was $2^{-45.09}$ on the best core we had, against $2^{-41.11}$ here (rows 3 and 4 of Table 4). The new choice also gives more upper pairs per core.

4 The second block

4.1 Variables

The second block is described by the words $A_0, \ldots, A_{15}$ of the first execution and $A'_7, \ldots, A'_{15}$ of the second; $A_t = A'_t$ for $t \leq 6$ because the characteristic has no difference before step 7. We divide these words into four groups.

- The *core* is $A_3, \ldots, A_{13}$ of both executions. It is fixed for a whole attack, and it determines the state differences of steps 7 to 13 and the constants w, d, a.
- The *variant* is A_2, one of 56 values found for the core (Section 6.1). By (4), A_2 is equivalent to E_6; the variants differ in E_6 and, through the prescription of Section 4.2, in the values of E_5 on F_7.
- The *lower part* is (A_0, A_1), common to both executions.
- The *upper pair* is (A_{14}, A_{15}) with $A'_{14} = A_{14}$ and $A'_{15} = A_{15} + a$.

By (4) the words E_t and E'_t follow from the A words, and by (5) the words W_t, W'_t for $t \geq 8$ follow from these words without the chaining value. The chaining value H and these words give M_1 and M'_1.

4.2 Lower solutions

Fix a variant. Write $E_5 = A_1 + c_5$ and $E_4 = A_0 + c_4(A_1)$ for a constant c_5 and a function c_4 given by $A_2, \ldots, A_5$ (equation (4)). The conditions on the lower part that do not depend on H are:

- $\Delta W_8 = -4$. By (5), $W_8 = c_8(E_5) - E_4$ with c_8 a function of E_5 and the core, and the difference ΔW_8 depends on E_5 only through the term $\mathrm{Ch}(E_7, E_6, E_5)$, which selects the bits of E_5 at the positions where E_7 and E'_7 differ. Writing F_7 for the set of these positions, $\Delta W_8 = -4$ prescribes E_5 on F_7, and the prescription is unique. The other $32 - |F_7|$ bits of E_5 are free. For the core used here $|F_7| = 13$.
- Row 23 of Table 2. W_8 takes one of four values, $W_8^{(1)}, \ldots, W_8^{(4)}$, which depend only on a and w. Together with E_5 this fixes $E_4 = c_8(E_5) - W_8^{(j)}$ and hence A_0.

A *lower candidate* is a triple (v, E_5, j) of a variant, an admissible value of E_5 and an index j of the four values of W_8. A variant has $4 \cdot 2^{32-|F_7|} = 2^{21}$ lower candidates. The only remaining condition on a lower part involves H: by Section 2.1, $W_7 = \theta - H_0$ with $\theta = \theta(v, E_5, j)$ a constant, and condition V ($W_7 \in V_7$) is a condition on H_0 alone. A lower candidate is a *lower solution* for the chaining value H if $\theta - H_0 \in V_7$; it satisfies this for a uniform H_0 with probability $2^{14}/2^{32} = 2^{-18}$. A variant therefore has $2^{21} \cdot 2^{-18} = 8$ lower solutions per first block on average, and the 32 variants that are used have 256.

Since V_7 is a union of 512 aligned blocks of 32 values, a lower candidate with $\theta = 32\,\theta_h + \theta_l$ is a lower solution for $H_0 = 32h + l$ if and only if $\theta_h - h - [\theta_l < l]$ is one of 512 block indices. The values θ are strongly concentrated, so we group the lower candidates of each variant by θ_h and keep a table indexed by $h = H_0 \gg 5$ (2^{27} buckets) that lists, for each bucket, the groups for which $\theta - H_0$ can lie in V_7. One lookup in this table per first block returns all lower solutions of all variants. Groups with fewer than 320 lower candidates are dropped, since each costs a table access whenever its bucket is reached and yields few lower solutions. The remaining 28,414,080 lower candidates in 45,832 groups give $28{,}414{,}080 \cdot 2^{-18} = 108.39$ lower solutions per first block.

4.3 The key and the upper pairs

By (1), $W_{16} = \sigma_1(W_{14}) + C_{16}$ with $C_{16} = W_9 + \sigma_0(W_1) + W_0$. The word W_{14} depends on the upper pair and the core only, and C_{16} depends on the lower solution and on H, because W_9 depends on E_5 and W_0, W_1 depend on H by Section 2.1. By (2),

$$E_{16} = P_{16}(\pi) + C_{16}, \qquad P_{16}(\pi) = A_{12} + E_{12} + \Sigma_1(E_{15}) + \mathrm{Ch}(E_{15}, E_{14}, E_{13}) + K_{16} + \sigma_1(W_{14}), \quad (7)$$

where $\pi = (A_{14}, A_{15})$ is an upper pair and the words $E_{12}, \ldots, E_{15}$ depend on the core and π. We call C_{16} the *key* of the lower solution. The differences of steps 14 to 17 depend on the lower solution and on H only through the key. The words A_t, E_t and W_t of steps 14 and 15 of both executions depend only on the core and π, by (4) and (5). Since W_0, W_1 and W_9 have no difference, both executions have the same key c; by (7) and (3), E_{16} and A_{16} are, in each execution, a function of π and the core plus c, so δE_{16} and δA_{16} do not depend on c. By (2) and (3), the differences of step 17 depend only on the words of steps 13 to 16 of both executions, because W_{17} has no difference and cancels. Hence for an upper pair π the set $K(\pi)$ of keys for which step 17 follows the characteristic is a function of π and the core. Call an upper pair *valid* if steps 14 to 16 follow the characteristic, the XOR differences in E_{14}, A_{15}, E_{15} and W_{15} are those of their digits (`0x21000`, `0x08010000`, `0x200` and 4), and rows 17 and 30 of Table 2 hold. For the core used here, 1.68×10^7 upper pairs are valid. The number of entries (c, π) with $c \in K(\pi)$ is $4.64 \times 10^9 = 2^{32.11}$. For a uniform key c the expected number of upper pairs with $c \in K(\pi)$ is the number of entries divided by 2^{32}.

Not every entry can lead to a collision. We keep only the entries with $E_{16}[27] = E_{15}[27]$, half of them (2.32×10^9, in 2.58×10^6 upper pairs). This condition is necessary when E_{16} and E_{17} differ from E'_{16} and E'_{17} exactly in the bits of their digits: otherwise $\mathrm{Ch}(E_{17}, E_{16}, E_{15})$ passes the digit of δE_{17} at bit 27, a term $\pm 2^{27}$ that the other terms of step 18, smaller than 2^{24} in absolute value together, cannot cancel. In general it is a pruning rule, and the rates of Table 4 are counted with the pruned table. A second condition is that step 18 can follow the characteristic for some value of the free word W_{17}, given the digits of $\delta E_{15}, \delta E_{16}, \delta E_{17}$. For every entry we count such values by Monte Carlo sampling, and we call the entry *live* if the count is nonzero. This leaves $N = 853{,}938{,}951$ live entries, 37% of the entries that satisfy the condition at bit 27, in 2,345,041 upper pairs. The sampling rarely misses an entry for which such a value exists: an exact search on 6,659 rejected entries found such a value for only one of them. In a test on 5×10^7 first blocks with all 56 variants and no groups dropped (448 lower solutions per first block), the attack found the same 5,535 block pairs of depth 20 or more with the live entries as with all entries that satisfy the condition at bit 27. The table of the attack is the set of live entries, stored by key.

4.4 The tail

For a lower solution and a live entry (c, π) with the key c of the lower solution (a *match*), the block pair is determined, and the remaining steps hold by chance. The words $W_8, \ldots, W_{15}$ are fixed by $A_0, \ldots, A_{15}$, W_7 by H_0, and W_0, W_1, W_2 are constrained by the selection (the key depends on W_0, W_1, W_9, and the test of step 18, which computes step 18 of both executions and checks $\delta A_{18} = \delta E_{18} = 0$, on W_1, W_2). As usual for two-block attacks, we treat the chaining values of first blocks as uniform. For a fixed lower candidate and upper pair, the words $W_0, \ldots, W_7$ are then independent and uniform (Section 2.1). The selection depends on H only through W_7 and $W_0, \ldots, W_2$ (the words W_4, W_5, W_6 do not enter W_{16}, W_{17}, W_{18}), so W_4, W_5, W_6 are uniform and independent of it. They enter the expanded words $W_{19}, \ldots, W_{22}$, and the conditions of steps 20 to 24 and of rows 25 and 26 of Table 2 are conditions on them. Table 4 lists the success probability stage by stage.

Table 4. Success per lower solution, stage by stage. The first row is the expected number of matches. The second and third rows are counted on real chaining values, the fourth is an estimate with its standard error, and the fifth is the product of rows 2 to 4.

Stage	Value	Basis
matches (expected number): $N/2^{32}$	0.198823	counted
step 18 passes, per lower solution	1.5925×10^{-5}	counted on 2.1×10^{15} lower solutions; 1.5929×10^{-5} over the whole run
step 19 passes, given step 18	0.015367	counted; 0.015368 over the whole run
collision, given step 19	$2^{-35.082}$ ($\pm 0.5\%$)	estimated from 43,371 states of depth 20
success per lower solution p	$6.729 \times 10^{-18} = 2^{-57.04}$	product

The expected number of lower solutions per collision is $1/p = 1.486 \times 10^{17} = 2^{57.04}$ and, with 108.39 lower solutions per first block, the expected number of first blocks is $2^{50.28}$. The fourth row is a sequential Monte Carlo estimate. Starting from the state through step 18 of a real depth-20 pair, it draws the free words W_4, W_5, W_6 stage by stage, derives $W_{19}, \ldots, W_{22}$ from them as in the attack, and estimates the probability that steps 20 to 24 and rows 25 and 26 hold. For each of 43,371 depth-20 pairs it uses 128 particles and draws 2^{14}, 2^{12} and 2^{16} values of W_4, W_5 and W_6 per particle; the value is the mean of the 43,371 per-pair estimates, and the error is their sample standard deviation divided by $\sqrt{43{,}371}$. These pairs passed step 19 and failed step 20, so a state is sampled with weight $1 - q$ relative to all pairs that pass step 19, where q is the probability that step 20 passes from it. The first stage of the estimator measures q, which is at most 0.0011 for every state, and reweighting each state by $1/(1 - q)$ changes the estimate by 0.05%. Treating $W_{19}, \ldots, W_{22}$ as independent uniform words gives nearly the same value, $2^{-35.079}$.

5 The attack

Algorithm 1 summarises the attack.

Algorithm 1. The attack, with the table of Section 4.3. M_0 is a first block.

```
for each first block M0 of a stream:
    H <- F38(IV, M0)
    for each lower solution of H, found by one lookup of h = H0 >> 5:   # 108.39 on average
        c <- C16 of the lower solution
        for each upper pair pi with (c, pi) in the table:               # 0.1988 on average
            compute steps 16..18 of both executions                     # step 18 passes
            ↪ with 8.0e-5
            if deltaA18 = deltaE18 = 0:
                compute steps 14..37 of both executions, stop at the first step
                whose differences differ from the characteristic        # 1.5e-2 have depth
                ↪ >= 20
                if all 38 steps follow it: output M0||M1 and M0||M1'
```

6 The core and the table

6.1 Finding the core

The attack needs a core with many live entries, several variants and many lower solutions per first block. We produce cores by a search in three stages.

1. A *skeleton* fixes the digits of δA_t and δE_t for steps 11 to 20, with the end of Table 3 for steps 14 to 20. Skeletons are generated by a SAT model of both executions and screened by the number of step-17 entries of the cores obtained from them.
2. For a skeleton, the late states are completed to values by a SAT solver (the CDCL solver CaDiCaL [BFF+24] with an external propagator, as in [ANB24], that propagates the modular additions exactly through carry states). The completion starts from chosen values of A_{14} and E_{14}, which are varied together with further late words; they serve only to construct the core, since in the attack A_{14} ranges over the upper pairs.
3. For each completion, the words $E_{12}, E_{11}, \ldots, E_5$ are chosen one at a time in eight levels. By (5), W_{s+8} depends on A_s only through the term $-A_s$, and every condition on the difference of W_{s+7} is a T-function of the word being chosen (bit i of the condition depends only on bits $\leq i$ of that word). Each level is therefore solved exactly by a depth-first search from the least significant bit. The last level constrains E_5 on the positions of F_7 (the condition $\Delta W_8 = -4$ of Section 4.2); the level before it chooses E_6 and hence the variants.

Few partial assignments complete to a core; the search that produced the core used here took 52,300 vCPU-hours on Arm machines (about US$290).

The search found 56 variants for this core. They are not equivalent: they differ in W_{10}, which enters row 25 of Table 2, so that the success probability depends on the variant. For 32 of the 56 variants the exact count in row 25 is 3,309,568 or 3,276,800 out of 2^{32} ($2^{-10.34}$, $2^{-10.36}$), and for the other 24 it is $2^{-10.84}$ to $2^{-10.87}$. We use the 32 better variants. This raises the average pass rate of row 25 by 15%, while the number of matches per lower solution, which dominates the cost, does not depend on the variant.

6.2 The table

For the core, the table of Section 4.3 was built in two steps. First all valid upper pairs were enumerated (a bit-serial search over A_{14} and A_{15}), and for each of them the keys of $K(\pi)$ that satisfy the condition at bit 27 were computed. Then the step-18 test of Section 4.3 was applied to every entry, and the entries that are not live were removed. The two steps took 35 minutes on one machine with 72 vCPUs (42 vCPU-hours) and gave a file of 3.44 GB. For the search the keys are reduced modulo 2^{30}; each pair of a reduced key and an upper pair carries a 4-bit mask of the values of the two top bits for which the full key is in the table, so the reduction neither loses nor adds entries. The table is a hash table of 5.44 GB over the 193,853,061 distinct reduced keys, read one 32-byte bucket per probe, and a filter over the reduced keys fits in the cache of the GPU (Section 8.1).

7 Complexity

7.1 Time

A first block gives 108.39 lower solutions on average, and a lower solution has 0.1988 matches. With $1/p = 2^{57.04}$ lower solutions per expected collision there are $2^{50.28}$ first blocks and $2^{54.71}$ matches. Charging one call per first block and one per match (Section 2.3) gives

$$2^{50.28} + 2^{54.71} = 2^{54.78} \text{ calls of } F_{38},$$

about $2^{54.8}$; 96% of the calls are matches. The attack makes 1.395 table accesses per lower solution: one probe of the filter, 0.183 probes of the exact table, at most 0.199 reads of the record of an upper pair, and 0.013 accesses for the first block and the bucket table. This gives $2^{57.52}$ accesses in all. In the convention of [Gol26], which converts the arithmetic of the attack

to evaluations of F_{38} (0.092 per lower solution) and charges every table access as one evaluation, the time is $2^{57.6}$.

On the GPU used (Section 8.1), a lower solution took 10.67 ps of kernel time on average during the run and one evaluation of F_{38} takes 33.75 ps, so a lower solution costs the time of 0.32 evaluations, and an expected collision $2^{55.4}$ evaluations' worth of GPU time, 0.6 bit above the call count.

7.2 Memory and precomputation

The table holds $N = 853{,}938{,}951$ live entries ($2^{29.7}$). The tables of the search take 7.35 GB: the hash table of the live entries 5.44 GB, the table of the lower solutions 1.09 GB (the lower candidates 341 MB, the bucket offsets 537 MB, the bucket entries 188 MB, the bucket bitmap 17 MB, the group tables 6 MB), the filter 89 MB, and 0.74 GB for the records of the upper pairs and the rest. The memory we report, 7.4 GB, is that of these tables; the GPU program also allocates 18 GB of working buffers for batching. The stored table is 3.44 GB.

The search for the core took 52,300 vCPU-hours on Arm machines, and the table 42 vCPU-hours. The search for the characteristic and for the skeletons is not included.

8 Implementation and the run

8.1 The GPU program

The search ran on NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs (128 MB of L2 cache), rented in the cloud. Each GPU works on its own range of first blocks, so no communication between GPUs is needed, and processes the first blocks in batches of 2^{25}.

1. *First blocks (CPU).* Twelve threads of the host compute $H = F_{38}(IV, M_0)$ for the first blocks of the batch with vectorised (AVX-512) code.
2. *Bucket lookup (GPU).* The GPU probes, for each H, a bitmap of the non-empty buckets of the bucket table (17 MB, in the L2 cache); 6.7% of the first blocks have a non-empty bucket, and 5.2% of all first blocks give at least one lower solution. For these, the groups of the bucket are queued.
3. *Lower solutions and keys (GPU).* The groups are processed group by group: the lower candidates of a group are staged in shared memory, and for every queued first block the key of each lower solution is computed.
4. *Filter and exact table (GPU).* The key is tested against a filter over the reduced keys of the table that fits in the L2 cache (89 MB; 18.3% of the keys pass), then against the exact table in device memory (one 32-byte bucket), and for a hit the record of the upper pair π is read and steps 16 to 18 are computed. Matches that pass step 18 are written out with the index of the first block, the lower candidate and the upper pair.
5. *Classification (CPU).* The host rebuilds both executions of each of these matches from the first block, the lower candidate and the upper pair, computes the depth, and writes the block pairs of depth 25 and more with their messages. A block pair of depth 38 is a collision.

On four ranges of 2^{28} first blocks, the matches written out by the GPU program and their depths are identical to those of a CPU implementation of Algorithm 1.

At the rate of 10.67 ps per lower solution measured over the run (Section 7.1), one expected collision takes about 440 GPU-hours of kernel time.

8.2 The run

The run started at 12:17 UTC on 31 October 2026 and found the collision after 1 h 22 min, using 53.3 rented GPU-hours (about US$95), 45.2 of them kernel time. By then it had processed $2^{47.0}$ first blocks and 1.52×10^{16} lower solutions, about 10% of the expected work. From the number of block pairs of depth at least 20, the tail estimate of Section 4.4 predicts the numbers of depth at least 21, 22, 23 and 25, and these predictions agree with the run within 2%.

9 The collision

Table 5 gives the collision: the two messages are $M_0\|M_1$ and $M_0\|M_1'$, each of 128 bytes with words in big-endian order as in FIPS 180-4; "hash" is the output of the second compression, and "digest" is the 38-step hash of the padded messages (a third compression with the same number of steps over the padding block).

Table 5. The collision for 38-step SHA-256.

M_0	`33ee28ac 5f8ef994 c787777c 079d17e1`
	`1198d1be b527c855 4bfeff01 4ab53195`
	`1cf31bd2 bbfc637e 122da4f9 48e6e447`
	`d5a6b8b6 ea9e7af5 aac1aeab eeaceb0d`
$H = F_{38}(IV, M_0)$	`d00b8f69 1466ed03 49c1a713 7c64bf45`
	`bd8254a1 c00b0cc6 a1382798 5f21292d`
M_1	`bd07668b 833fe88e 689ffe8d a52259da`
	`14b9855e ee5138ba 88355f48 06d5cf5d`
	`161c5001 4c6b7d7e 0e7eb888 747559fc`
	`990a7af6 9dd8e59f 57d7f6a3 c0fbd03b`
M_1'	`bd07668b 833fe88e 689ffe8d a52259da`
	`14b9855e ee5138ba 88355f48 38b3d19d`
	`161c4ffd 4c6b7d7e 0e813888 747559fc`
	`990a7af6 9dd8e59f 57d7f6a3 c0fbd03f`
hash $F_{38}(H, M_1) = F_{38}(H, M_1')$	`9f5936ca 5239e115 c1db7dd9 11f2d688`
	`10171e5f e37fa366 79b07ba4 2ba4433c`
digest (padded)	`5b61bad37673bb3323d64bc4fad45b44`
	`a7e6bf048d787d6127ec259b12d0d63f`

The two messages differ in the second block only, in the words W_7, W_8, W_{10}, W_{15}: $\Delta W_7 =$ `31de0240`, $\Delta W_8 = -4$, $\Delta W_{10} =$`00028000`, $\Delta W_{15} = 4$. This block pair uses the variant with $E_6 =$`99ccff09` (the 20th of the 32 variants used; $A_2 =$`42747b1a`), the lower part $A_0 =$`248e4477`, $A_1 =$`166a78d5` (so $E_4 =$`cd0d33ea` and $E_5 =$`8426bd20`) and the upper pair $A_{14} =$`5bde5962`, $A_{15} =$`9b3a6d8a`.

The path. Table 6 shows the differences of the two executions. Steps 0–6 and 25–37 have no difference.

Table 6. Modular differences of the two executions (hexadecimal).

t	δA_t	δE_t	ΔW_t
7	31de0240	31de0240	31de0240
8	4f5565d8	39f440a0	fffffffc
9	a1b48496	a8dcc1ef	0
10	4396fb5f	6774e337	00028000
11	01201041	39a879c6	0
12	f9810e38	06627ff8	0
13	f7a10e01	c43d71fb	0
14	0	fffdf000	0
15	f8010000	00000200	00000004
16	00000004	ffdffe24	0
17	0	f77ffe00	0
18	0	0	0
19	0	f8010000	0
20	0	00000004	0
21, 22	0	0	0
23	0	0	07ff0000
24	0	0	fffffffc

In the notation of Table 3 the block pair has $s = -1$: $\delta E_{15} = +2^9$, $\delta E_{16} = -2^{21} - 2^9 + 2^5 + 2^2$ and $\delta E_{17} = -2^{27} - 2^{23} - 2^9$.

The conditions of Table 2 hold for this block pair. W_{15} =c0fbd03b has $\Delta\sigma_1(W_{15}; 4)$ = fffd8000= $-d$ and $\Delta\sigma_0(W_{15}; 4) = a$; W_7 =06d5cf5d has $\Delta\sigma_0(W_7; w) = -4$; W_8 =161c5001 has $\Delta\sigma_0(W_8; -4)$ =d620fdc0= $-a - w$; W_{23} =16420b4b and W_{24} =7a948b7f give $\Delta W_{25} = \Delta W_{26} = 0$. For this variant $\tau = -\Delta\sigma_0(W_{10}; d)$ =6002033f, for which row 25 holds for 3,276,800 of the 2^{32} values of W_{23}.

Verification. The collision was checked with five independent implementations of the reduced compression function. With both blocks reduced to the same number of steps s, the s-step hashes of the two messages are equal for $s = 38$ and for $s \leq 7$ only (for $s \leq 7$ equality is trivial, since the blocks agree in $W_0, \ldots, W_6$).

10 Conclusion

We have given a collision for 38-step SHA-256 from the standard initial value. The attack fixes most of the second block in advance and meets the conditions up to step 17 with one table lookup per lower solution, so that the search pays mainly for the conditions from step 18 on, $2^{-54.7}$ per match. The expected time is $2^{54.8}$ calls of the compression function, with $2^{57.5}$ table accesses; on the GPU used, an expected collision takes about 440 GPU-hours of kernel time. The characteristic does not extend beyond 38 steps (Section 3.1), so a collision for more steps needs a different characteristic.

References

[ANB24] Nahiyan Alamgir, Saeed Nejati, and Curtis Bright. SHA-256 collision attack with programmatic SAT. In *Proceedings of the 2024 International Workshop on Satisfiability Checking and Symbolic Computation (SC² 2024)*, 2024. arXiv:2406.20072.

[BFF+24] Armin Biere, Tobias Faller, Katalin Fazekas, Mathias Fleury, Nils Froleyks, and Florian Pollitt. CaDiCaL 2.0. In *Computer Aided Verification – CAV 2024, Part I*, volume 14681 of LNCS, pages 133–152. Springer, 2024.

[Gol26] Omer Goldzweig. Improved collision attacks on 36- to 39-step SHA-256. Unpublished manuscript, version 1.2, 30 September 2026. Zenodo, doi:10.5281/zenodo.23066177.

[GWZW26] Jiaying Gong, Yongzhuang Wei, Fengrong Zhang, and Bohua Wei. Collision attacks on SHA-2: 39-step SHA-256 below the birthday bound. Cryptology ePrint Archive, Paper 2026/2867, 2026.

[LLW24a] Yingxin Li, Fukang Liu, and Gaoli Wang. New records in collision attacks on SHA-2. In *Advances in Cryptology – EUROCRYPT 2024, Part I*, volume 14651 of LNCS, pages 158–186. Springer, 2024.

[LLW+24b] Yingxin Li, Fukang Liu, Gaoli Wang, Xiaoyang Dong, and Siwei Sun. The first practical collision for 31-step SHA-256. In *Advances in Cryptology – ASIACRYPT 2024, Part VII*, volume 15490 of LNCS, pages 237–266. Springer, 2024.

[LLWS26] Yingxin Li, Fukang Liu, Gaoli Wang, and Jiali Shi. Pushing the limit of memory-efficient collision attack framework for SHA-2. In *Advances in Cryptology – CRYPTO 2026, Part VI*, volume 16805 of LNCS, pages 289–320. Springer, 2026. Cryptology ePrint Archive, Paper 2026/1080.

[LZL+26] Yingxin Li, Zhuolong Zhang, Muzhou Li, Fukang Liu, Haifeng Qian, and Jinwei Zhu. Pushing collision attacks on SHA-2 to 39 steps. Cryptology ePrint Archive, Paper 2026/1120, 2026. Version of 9 September 2026.

[MNS11] Florian Mendel, Tomislav Nad, and Martin Schläffer. Finding SHA-2 characteristics: Searching through a minefield of contradictions. In *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of LNCS, pages 288–307. Springer, 2011.

[MNS13] Florian Mendel, Tomislav Nad, and Martin Schläffer. Improving local collisions: New attacks on reduced SHA-256. In *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of LNCS, pages 262–278. Springer, 2013.

[Nat15] National Institute of Standards and Technology. Secure hash standard (SHS). Federal Information Processing Standards Publication 180-4, 2015. https://doi.org/10.6028/NIST.FIPS.180-4.

[ZLGW26] Zhuolong Zhang, Muzhou Li, Lei Gao, and Meiqin Wang. Collision attacks on SHA-256 up to 37 steps with improved trail search. In *Advances in Cryptology – EUROCRYPT 2026, Part VI*, volume 16546 of LNCS, pages 91–120. Springer, 2026. Cryptology ePrint Archive, Paper 2026/232.