Collision Attacks on SHA-2: 39-Step SHA-256 below the Birthday Bound

Jiaying Gong[1], Yongzhuang Wei[1⋆], Fengrong Zhang[2], and Bohua Wei[1]

[1] Guilin University of Electronic Technology, Guilin, China
jiaying_0517@mails.guet.edu.cn
walker_wyz@guet.edu.cn, weibh@guet.edu.cn
[2] Xidian University, Xian, China
zhangfengrong@xidian.edu.cn

Abstract. SHA-2 is the most widely deployed family of hash functions, and its security margin is measured by the number of steps collision attacks reach. Li et al. gave the first classical collision attack on 38-step SHA-256, with an estimated complexity of $2^{104.3}$. The challenge is to extend the attack to more steps while retaining its advantage over generic collision search. In this paper, we show that collisions for 39-step SHA-256 can be found at an estimated cost below the 2^{128} birthday bound. We keep the published 39-step characteristic and change how its conditions are satisfied. We introduce *joint message-word pinning*, a technique that fixes two expanded message words, W_{16} and W_{17}, to the values prescribed by the characteristic. For each chaining value, we solve for assignments to two state words that carry no direct condition, and every assignment returned gives both message words their prescribed values. To find these assignments, we first fix the value of a function with a small image and then solve the coupled equations for W_{16} and W_{17} in sequence. On 39-step SHA-256, pinning W_{16} alone gives an estimated cost of $2^{123.4}$ scalar compressions per expected collision, and pinning both words reduces it by 14.6 bits to $2^{108.8}$. Under the uniform-chaining-value model, the measured probabilities and the cost of processing one first block give an upper bound of approximately $2^{124.6}$ on the expected cost of finding the first collision, even under worst-case clustering. For 38-step SHA-256, we reduce the estimated cost per expected collision from $2^{104.3}$ to $2^{90.9}$ by using a new differential characteristic and pinning W_{16} alone. On SHA-512, pinning W_{16} alone reduces this cost from $2^{125.4}$ to $2^{100.7}$ for 38 steps and from 2^{178} to $2^{131.3}$ for 39 steps.

Keywords: SHA-256 · SHA-512 · collision attack · message modification

1 Introduction

In 2005, Wang et al. described collision attacks on MD4, RIPEMD, MD5, SHA-0, and SHA-1 using signed differences and message modification [18,20,21,19]. SHA-256 and SHA-512 [16] have withstood these attacks and are now the most widely

⋆ Corresponding author: walker_wyz@guet.edu.cn.

deployed members of the MD-SHA family. Both are used in digital signatures and secure communication protocols, and Bitcoin also uses SHA-256. Their security margin is commonly measured by the number of steps for which collisions can be found faster than by a generic attack.

By 2008, collision attacks on both SHA-256 and SHA-512 had reached 24 steps [7,17]. These attacks adjusted the message words to give the first expanded word a prescribed value. Automated characteristic searches later yielded practical collisions for 28-step SHA-256, a theoretical attack on 31 steps and practical semi-free-start (SFS) collisions for 38 steps [14,15]. On SHA-512, the same approach produced practical collisions for 27 steps and practical SFS collisions for 38 and 39 steps [4,3]. For classical attacks on SHA-256, the 31-step record set in 2013 stood for more than a decade. With search for signed characteristics by MILP and SAT/SMT [13,10] and a memory-efficient framework for satisfying the conditions [9], attacks reached 37 steps of SHA-256 [22] and practical collisions for 35 steps of both functions [11]. For attacks starting from the standard initial value, Li et al. [12] extend the memory-efficient two-block framework of [11] beyond 35 steps and report complexities of $2^{104.3}$ for 38 steps of SHA-256 and $2^{125.4}$ and 2^{178} for 38 and 39 steps of SHA-512.

The low probability of the uncontrolled part of the characteristic prevented effective attacks on 38-step SHA-256 in [11] and on 39-step SHA-256 in [12]. A SAT solver first fixes the dense part through step 13. The first block affects the second only through its chaining value. All conditions from step 16 on are left to chance. For 39-step SHA-256, the characteristic T23 (Tables 21–23 of [12], named after the table of its colliding pair) has $\gamma = 168$ such conditions, which gives 2^{168} against a birthday bound of 2^{128}, and [12] describes this case as ineffective, although practical semi-free-start collisions are known [10]. This count is pessimistic. Given the state before step 16, the 87 conditions at steps 16 and 17 depend on just 64 bits (W_{16} and W_{17}). For uniform words, the values of W_{16} and W_{17} in the semi-free-start pair of T23 are hit with probability 2^{-64}, and a heuristic estimate anchored at these values gives 2^{145} (Sect. 2.3), still 17 bits above the bound. Can W_{16} and W_{17} be controlled in the two-block framework, as W_{16} was in the single-block attacks on 24-step SHA-256 and SHA-512 [7,17]?

We show that both W_{16} and W_{17} can. No condition of T23 involves the state words A_1 and A_2 of the second block directly, and both expanded words depend on them through the same map $f(x) = \sigma_0(x) - x$. For each chaining value, we therefore search for assignments to A_1 and A_2, and every assignment returned fixes the pinned words to their target values. *Single-word pinning* applies the equation of [7,17] per chaining value: a table look-up gives the values of A_1 for which W_{16} takes its target value. Pinning W_{17} as well needs A_2, and the two equations are coupled. However, A_2 enters the equation of W_{16} only through a function with an image of size $2^{20.2}$, because its Boolean functions read A_2 only at positions fixed by the dense part, and once the value of this function is fixed the equations can be solved in turn. Enumerating its largest classes gives *joint pinning*: most chaining values return no solution, and a solution that meets the head conditions on W_9 and W_{10} (the head comprises all conditions before a

chosen cutoff step, and the tail the conditions from that step on) satisfies steps 16 and 17 by construction. The closest technique, the head re-solving of [11, Sect. 4], also re-solves early state words for a given first block, to obtain more second blocks, but sets no target on W_{16} or W_{17}. On 39-step SHA-256, single-word pinning already crosses the birthday bound at $2^{123.4}$ compressions per expected collision, and joint pinning lowers this by 14.6 bits to $2^{108.8}$, 19.2 bits below the bound; it replaces the 2^{64} of steps 16 and 17 by $2^{28.4}$ compressions, including the re-paid head conditions. For uniform chaining values and with the measured probabilities, the expected cost of the first collision is at most $2^{127.2}$ and $2^{124.6}$, respectively, even under worst-case clustering. For joint pinning, we observed no clustering of solutions among the successful entries we sampled. Measured in single-threaded 39-step compressions using the Intel SHA extensions (SHA-NI), the joint-pinning bound is $2^{126.8}$, still below the birthday bound (Sect. 4.3). Pinning changes W_8, W_9 and W_{10}, which the tail reads at later steps, so counting conditions as independent is not reliable at this boundary. We therefore establish every factor of the attacks by analysis, exact enumeration or measurement, and check that the measured tails agree within their intervals when the sampled heads are replaced by heads produced by the real attack pipeline.

Table 1. Summary of (semi-free-start) collision attacks on 38 and 39 steps of SHA-256 and SHA-512.

Version	Attack type	Steps	Time	Memory	Method	References
SHA-256	collision	38	$2^{104.3}$	negligible	T5, framework	[12]
		38	$2^{100.3}$	negligible	C38, framework	Sect. 5.2
		38	$2^{91.7}$	17 GiB	T5, single-word	Sect. 5.1
		38	$\mathbf{2^{90.9}}$	17 GiB	C38, single-word	**Sect. 5.2**
		39	2^{168}	—	T23, framework	[12]
		39	2^{145}	—	T23, anchored	Sect. 2.3
		39	$2^{123.4}$	17 GiB	T23, single-word	Sect. 4.3
		39	$\mathbf{2^{108.8}}$	17 GiB	T23, joint	**Sect. 4.3**
	SFS collision	38	*practical*			[15]
		39	*practical*			[10]
SHA-512	collision	38	$2^{125.4}$	negligible	framework	[12]
		38	$\mathbf{2^{100.7}}$	no table	single-word	**Sect. 5.3**
		39	2^{178}	negligible	framework	[12]
		39	$\mathbf{2^{131.3}}$	no table	single-word	**Sect. 5.3**
	SFS collision	38	*practical*			[4]
		39	*practical*			[3]

T5 and T23 are the characteristics of [12], named after Tables 5 and 23 there; C38 is our new 38-step characteristic. Framework and anchored entries are condition-count estimates. Pinned entries are estimated costs per expected collision, in scalar compressions, combining the measured entry cost with the measured head and tail probabilities. 17 GiB: the table of the preimages of f.

Our contributions. Building on the two-block framework of [12], we extend collision attacks on SHA-256 from 38 to 39 steps, with an estimated cost below the birthday bound. Table 1 compares our results with previous attacks on 38- and 39-step SHA-256 and SHA-512.

1. *A theoretical collision attack on 39-step SHA-256 below the birthday bound.* For each chaining value, we search for assignments to two state words that carry no direct condition; every assignment returned gives W_{16} and W_{17} their target values (Sect. 3). On T23, the framework costs 2^{168} by condition count; joint pinning gives an estimated $2^{108.8}$ per expected collision, and single-word pinning, which adapts the equation from the 24-step attacks [7,17] to the two-block framework, already gives $2^{123.4}$. For uniform chaining values and with the measured probabilities, the expected cost of the first collision is at most $2^{124.6}$, even under worst-case clustering.
2. *Complexities established factor by factor.* The head comes from running the attack pipeline itself, the tail from exact enumeration or sampling of its first step followed by multilevel splitting, and the cost per entry from timing (Sect. 4). For uniform chaining values, head and tail of joint pinning on T23 compose exactly (Proposition 1). With heads produced from real first blocks in place of sampled heads, the measured tails agree within their 95% confidence intervals on SHA-256 and SHA-512 (on 39-step SHA-512, for the part of the tail that reads the head).
3. *Improved collision attacks on 38- and 39-step SHA-512.* Single-word pinning carries over to SHA-512 unchanged, and a bit-sliced algorithm inverts the 64-bit map f at the cost of about three compressions, so the attacks need no precomputed table. They cost $2^{100.7}$ and $2^{131.3}$ on 38 and 39 steps, against $2^{125.4}$ and 2^{178} in [12].
4. *A new characteristic and an improved collision attack for 38-step SHA-256.* Independently of pinning, we run the staged search of [12] with the condition count of [22] as its objective and with value propagation on a window of steps (Sect. 5.2). The resulting characteristic C38 has $\gamma = 98$, against 103 for T5 (Tables 3–5 of [12]) counted in the same way, and comes with a semi-free-start colliding pair. Under the accounting of [12], it gives $2^{100.3}$, below the reported $2^{104.3}$; with single-word pinning, the attack costs $2^{90.9}$.

Organization. Section 2 recalls SHA-2 and the two-block framework. Section 3 introduces single-word and joint pinning and analyzes when each pays off. Section 4 turns joint pinning into the 39-step attack and establishes its cost. Section 5 applies pinning to 38-step SHA-256 and to SHA-512, and presents the condition-count search, the new 38-step characteristic and its combination with single-word pinning. Section 6 concludes. Appendix A gives the conditions of T23 under pinning, the proofs for the cost model and a worked example of joint pinning with a colliding second-block pair. Appendix B gives the measured tail per step and the unit conversion. Appendix C gives the stages of the search, the characteristic C38 with its conditions and semi-free-start colliding pair, the pinning parameters and the measured SHA-512 tails.

2 Preliminaries

2.1 Notations and Differential Characteristics

1. $+$ and $-$ denote addition and subtraction modulo 2^n, where $n = 32$ for SHA-256 and $n = 64$ for SHA-512.
2. $\gg$, $\ggg$, $\oplus$, $\neg$ and $\wedge$ denote right shift, right rotation, bitwise exclusive or (XOR), bitwise not and bitwise and, respectively.
3. Δx denotes the signed difference of a word x of the first message and the corresponding word x' of the second, written bit by bit with the symbols below.
4. $\mathrm{HW}(\Delta x)$ denotes the Hamming weight of Δx, i.e. the number of bits with symbol `u` or `n`.

We describe characteristics with the generalized conditions of [2]: each bit pair (x, x') is constrained by a symbol such as `-` ($x = x'$), `x` ($x \neq x'$), `u` ($x = 1$, $x' = 0$), `n` ($x = 0$, $x' = 1$), `0` ($x = x' = 0$) or `1` ($x = x' = 1$). Besides these single-bit conditions, the propagation of differences through the choice function IF, the majority function MAJ, and Σ_0, Σ_1, σ_0 and σ_1 imposes two-bit conditions, e.g. equalities between bits of consecutive state words. A pair of messages *follows* a characteristic if it satisfies all of these conditions.

2.2 Description of SHA-256 and SHA-512

SHA-256 [16] processes a padded message in 512-bit blocks with the Merkle–Damgård construction. The compression function takes a 256-bit chaining value and a block of sixteen 32-bit words $M_0, \ldots, M_{15}$, and has $R = 64$ steps. SHA-512 has the same structure with 64-bit words, 1024-bit blocks and $R = 80$ steps.

Message expansion. The message is expanded to $W_0, \ldots, W_{R-1}$ by $W_i = M_i$ for $i < 16$ and

$$W_i = \sigma_1(W_{i-2}) + W_{i-7} + \sigma_0(W_{i-15}) + W_{i-16}, \qquad 16 \leq i < R,$$

where

$$\begin{aligned}
\text{SHA-256:} \quad & \sigma_0(x) = (x \ggg 7) \oplus (x \ggg 18) \oplus (x \gg 3), \\
& \sigma_1(x) = (x \ggg 17) \oplus (x \ggg 19) \oplus (x \gg 10), \\
\text{SHA-512:} \quad & \sigma_0(x) = (x \ggg 1) \oplus (x \ggg 8) \oplus (x \gg 7), \\
& \sigma_1(x) = (x \ggg 19) \oplus (x \ggg 61) \oplus (x \gg 6).
\end{aligned}$$

State update. We use the alternative description of the state update in which only the two new words are tracked per step [14]. Step i computes

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

where K_i is a predefined constant, $\mathrm{IF}(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z)$, $\mathrm{MAJ}(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$, and

$$\begin{aligned}
\text{SHA-256:}\quad & \Sigma_0(x) = (x \ggg 2) \oplus (x \ggg 13) \oplus (x \ggg 22),\\
& \Sigma_1(x) = (x \ggg 6) \oplus (x \ggg 11) \oplus (x \ggg 25),\\
\text{SHA-512:}\quad & \Sigma_0(x) = (x \ggg 28) \oplus (x \ggg 34) \oplus (x \ggg 39),\\
& \Sigma_1(x) = (x \ggg 14) \oplus (x \ggg 18) \oplus (x \ggg 41).
\end{aligned}$$

The chaining value is $(A_{-1}, \ldots, A_{-4}, E_{-1}, \ldots, E_{-4})$; after the last step, the output words are added to the chaining value. A step-reduced variant with r steps applies only steps $0, \ldots, r-1$ and keeps the feed-forward. Rearranging the update gives two relations used throughout:

$$E_i = A_i + A_{i-4} - \Sigma_0(A_{i-1}) - \mathrm{MAJ}(A_{i-1}, A_{i-2}, A_{i-3}), \tag{1}$$

$$W_i = E_i - A_{i-4} - E_{i-4} - \Sigma_1(E_{i-1}) - \mathrm{IF}(E_{i-1}, E_{i-2}, E_{i-3}) - K_i. \tag{2}$$

Given the A words, every E_i follows, and given the state words, every message word W_i follows.

2.3 The Two-Block Framework

A *semi-free-start* (SFS) collision is a colliding pair for the compression function C under a chaining value chosen by the attacker. A collision attack starts from the prescribed initial value IV. In the two-block framework considered here, the two messages share a first block B_0, which produces the chaining value used for the second block:

$$CV_1 = C(IV, B_0), \qquad C(CV_1, B_1) = C(CV_1, B_1'), \quad B_1 \neq B_1'.$$

Our attacks build on the memory-efficient two-block framework of [9,11], in the form given by [12, Sect. 4.2] for $r > 35$ steps, where the message differences lie in the second block. For 39 steps the message differences lie in $W_8, \ldots, W_{12}$, W_{16}, W_{17}, W_{24} and W_{26}. This is the local-collision position of the 39-step SFS collisions of [3,10]. The attack has three steps.

Step 1. A SAT solver fixes the dense part of the characteristic: $(W_8, \ldots, W_{13})$, $(A_0, \ldots, A_{13})$ and $(E_4, \ldots, E_{13})$.

Step 2. For a random first block, compute CV_1. The states $A_0, \ldots, A_{13}$ together with CV_1 determine $E_0, \ldots, E_3$ by (1) and then $W_0, \ldots, W_7$ by (2). The first block is *valid* if these words satisfy the conditions of steps 0–7, which happens with a probability $2^{-N_{\mathrm{pro}}}$ that depends on the characteristic. For the 39-step characteristic T23, these steps carry no condition and every first block is valid. For the 38-step characteristics of SHA-256 used in this paper, four conditions on W_7 remain, so $N_{\mathrm{pro}} = 4$.

Step 3. Use the free words (W_{14}, W_{15}) to satisfy the conditions from step 16 on; if none works, return to Step 2.

Hosoyamada and Sasaki [6, Sect. 4.1] explain why the earlier semi-free-start collisions for 38 steps of SHA-256 [15] and 39 steps of SHA-512 [3] do not convert into two-block collisions: their characteristics carry conditions on almost every bit of $E_7, \ldots, E_{20}$ and $E_8, \ldots, E_{22}$, respectively, which fixes the message words of these steps and leaves only 64 free bits to connect the chaining value. The obstacle is specific to those characteristics. In the framework of [12] the dense part ends at step 13, so $W_0, \ldots, W_7$ absorb the chaining value, and the cost is governed by the conditions from step 16 on.

Let γ be the number of conditions from step 16 on, which are paid by probability, and $|(W_{14}, W_{15})|$ the number of admissible values of the two free words. The attack needs about $2^{\gamma - \log_2 |(W_{14}, W_{15})|}$ valid first blocks, so

$$T \approx 2^{\gamma - \log_2 |(W_{14}, W_{15})| + N_{\mathrm{pro}}} + 2^{\gamma}.$$

With the characteristic T23, 39-step SHA-256 has $\gamma = 168$ and $T \approx 2^{168}$, above the birthday bound. 39-step SHA-512 has $\gamma = 178$, below its bound of 2^{256} [12]. For the 38-step characteristic T5, [12] reports $2^{104.3}$. These estimates treat the conditions as independent and count each one as a factor of two. For T23 this is pessimistic at steps 16 and 17. Assigning each condition to the last step it involves, 87 conditions fall on these two steps and 81 on later ones, although, given the state before step 16, the two steps depend only on the 64 bits of W_{16} and W_{17}. For uniform words, the values of W_{16} and W_{17} in the semi-free-start pair of T23 are hit with probability 2^{-64}, which gives a heuristic estimate of $2^{64+81} = 2^{145}$ anchored at these values, still above the birthday bound. These are the values that pinning takes as its targets in Sect. 3, so 2^{145} is the estimate from which pinning starts. Measuring steps 18 and 19 at these values gives about $2^{144.4}$; from step 20 on the estimate remains a condition count.

3 Pinning the First Expanded Message Words

In the two-block framework (Sect. 2.3), every first block yields a chaining value CV_1 from which $W_0, \ldots, W_7$ follow, and all conditions from step 16 on are paid by probability. For the 39-step characteristic T23 of [12] with the (W_{14}, W_{15}) of its semi-free-start pair, steps 16 and 17 admit a single pair of target values $(W_{16}^{\star}, W_{17}^{\star})$. When the Step-1 values and (W_{14}, W_{15}) are fixed and CV_1 is uniform, (W_{16}, W_{17}) takes these values with probability 2^{-64}, because $W_0, \ldots, W_7$ vary with CV_1 (Sect. 4.4). Our main idea is that both words can instead be set by construction. For each chaining value, we search for values of two state words of the second block that no condition of T23 involves directly, and every solution found gives both words their target values. We call the conditions before a fixed step the *head* and those from that step on the *tail* (Sect. 4.2). Figure 1 contrasts the framework with our attack.

3.1 W_{16} and W_{17} as Functions of the Free State Words

In T23, no condition involves A_0, A_1 or A_2 directly. The only early conditions, seven bits of E_7, depend on $E_7 = A_7 + A_3 - \Sigma_0(A_6) - \mathrm{MAJ}(A_6, A_5, A_4)$ and

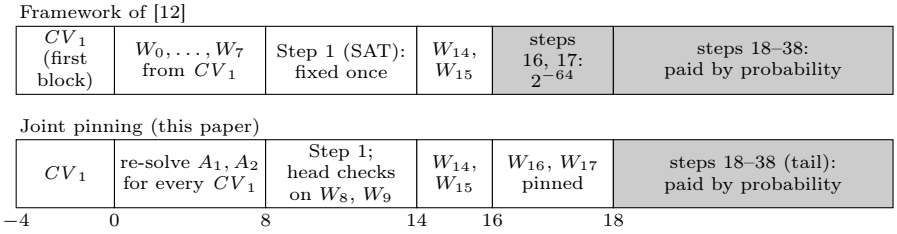

Fig. 1. The second block of the 39-step attack, by step. Top: the framework of [12]. Bottom: joint pinning. Shaded: paid by probability; solving and the head checks yield $2^{-20.47}$ head-passing solutions per first block.

are kept by the Step-1 values. Changing A_0, A_1, A_2 alters only $E_0, \ldots, E_6$ and $W_0, \ldots, W_{10}$; Table 7 in Appendix A.1 lists the conditions this affects. We keep $A_3, \ldots, A_{13}$, $E_7, \ldots, E_{13}$ and W_{11}, W_{12}, W_{13} at their Step-1 values and (W_{14}, W_{15}) at the values of the semi-free-start pair, and let A_1, A_2 (and, in Sect. 3.4, A_0) vary. Substituting (1) and (2) into the message expansion and writing $f(x) = \sigma_0(x) - x$ gives

$$W_{16} = \sigma_1(W_{14}) + W_9 + \sigma_0(W_1) + W_0 = f(W_1) + C_{16}(A_0, CV_1) + h(A_2), \quad (3)$$

$$\begin{aligned} W_{17} &= \sigma_1(W_{15}) + W_{10} + \sigma_0(W_2) + W_1 \\ &= f(W_2) + C_{17} + g(A_1, CV_1) + W_1. \end{aligned} \quad (4)$$

Both words are governed by the same map f, so a single table of f^{-1} serves both. The two equations are coupled, but A_2 enters (3) only through $h(A_2)$. Given $v = h(A_2)$, (3) yields W_1 and A_1, then (4) yields W_2 and A_2, and it remains to check that $h(A_2) = v$ (Sect. 3.3).

The remaining terms follow from (1) and (2):

$$\begin{aligned} W_1 &= A_1 + c_1(A_0, CV_1), & W_2 &= A_2 + g(A_1, CV_1), \\ W_9 &= c_9 - A_1 + h(A_2), & W_{10} &= c_{10} - A_2, \end{aligned}$$

where

$$\begin{aligned} c_1 &= -\Sigma_0(A_0) - \mathrm{MAJ}(A_0, A_{-1}, A_{-2}) - E_{-3} \\ &\quad - \Sigma_1(E_0) - \mathrm{IF}(E_0, E_{-1}, E_{-2}) - K_1, \\ g &= -\Sigma_0(A_1) - \mathrm{MAJ}(A_1, A_0, A_{-1}) - E_{-2} \\ &\quad - \Sigma_1(E_1) - \mathrm{IF}(E_1, E_0, E_{-1}) - K_2, \\ c_9 &= E_9 - 2A_5 + \Sigma_0(A_4) - \Sigma_1(E_8) - K_9, \\ c_{10} &= E_{10} - 2A_6 + \Sigma_0(A_5) + \mathrm{MAJ}(A_5, A_4, A_3) \\ &\quad - \Sigma_1(E_9) - \mathrm{IF}(E_9, E_8, E_7) - K_{10}. \end{aligned}$$

Here c_1, c_9, c_{10} do not depend on A_1, A_2, the function g also depends on the fixed value of A_0, and

$$h(A_2) = \mathrm{MAJ}(A_4, A_3, A_2) - \mathrm{IF}(E_8, E_7, E_6),$$
$$E_6 = A_2 + k_6, \qquad k_6 = A_6 - \Sigma_0(A_5) - \mathrm{MAJ}(A_5, A_4, A_3),$$

depends only on Step-1 constants and A_2. Finally, $C_{16} = \sigma_1(W_{14}) + c_9 + c_1 + W_0$ and $C_{17} = \sigma_1(W_{15}) + c_{10}$.

3.2 Single-Word Pinning

With A_2 fixed, (3) determines W_1 up to the preimages of f:

$$W_1 \in f^{-1}\big(W_{16}^{\star} - C_{16}(A_0, CV_1) - h(A_2)\big), \qquad A_1 = W_1 - c_1.$$

For uniform CV_1, the number of solutions per chaining value has mean exactly one, since $E_{-4} \mapsto W_{16}$ is a bijection for fixed other parameters. It is at most 14, the largest preimage size of f.

Changing A_1 alters $E_1, \ldots, E_5$ and $W_1, \ldots, W_9$. In T23 the only conditions affected are the 10 conditions on W_8 and W_9, which are re-paid by probability. Step 17 then holds with probability exactly 2^{-32} for uniform CV_1 (Lemma 1).

Relation to previous work. Indesteege et al. [7] write W_{16} as $\sigma_0(W_1) - W_1$ plus terms independent of W_1, and Sanadhya and Sarkar [17] solve the resulting equation with a table of f or a guess-and-determine algorithm. Both work on a single block with a free message, and both leave W_{17} and later words to probability; [17] also writes W_{17} in this form, so the equations of Sect. 3.3 are theirs, and solving them together is the step we add. In the two-block framework, the message words of the second block are tied to the fixed states by (2) and to the chaining value of the first block. We therefore apply the equation per chaining value, where it removes the step-16 conditions at the cost of one table look-up per first block. The head re-solving of [11, Sect. 4] re-solves $A_0, \ldots, A_3$, $E_0, \ldots, E_7$ and $W_0, \ldots, W_{11}$ with a SAT solver for each valid first block while keeping the later states. This yields several second blocks per first block and amortizes the cost of finding valid first blocks, but imposes no target on W_{16} or W_{17}.

3.3 Joint Pinning of W_{16} and W_{17}

Enumerating the value of h. Equations (3) and (4) are coupled: A_1 enters (4) through g and W_1, and A_2 enters (3) through h. Fixing either word first leaves a 2^{-32} check on the other, so there is no triangular order. The coupling, however, passes through $h(A_2)$ only, and h has a small image. Bit by bit, $\mathrm{MAJ}(A_4, A_3, A_2)$ reads A_2 only where A_4 and A_3 differ, and $\mathrm{IF}(E_8, E_7, E_6)$ reads $E_6 = A_2 + k_6$ only where E_8 is zero. Both sets of positions are fixed by Step 1; on T23 they have 15 and 13 elements, so h takes at most 2^{28} values. A bit read by both terms can also cancel in the modular difference, up to the carries of k_6. Evaluating h on all 2^{32} inputs gives $|\mathrm{Im}\, h| = 2^{20.2}$ for T23 and $2^{18.0}$ for T5. We therefore enumerate by the value $v = h(A_2)$:

1. look up $W_1 \in f^{-1}(W_{16}^{\star} - C_{16} - v)$ and set $A_1 = W_1 - c_1$;
2. look up $W_2 \in f^{-1}(W_{17}^{\star} - C_{17} - g(A_1, CV_1) - W_1)$ and set $A_2 = W_2 - g(A_1, CV_1)$;
3. accept if $h(A_2) = v$.

Every solution is found in the round $v = h(A_2)$; Appendix A.3 gives a worked example.

Two refinements. The cost of the attack is governed by the expected number of head-passing solutions per unit of work, not by finding every solution of a chaining value, and two refinements raise this number. First, $W_9 = c_9 - W_1 + c_1 + v$ is known after the first look-up, so the six W_9 conditions filter before the second look-up. Second, the five conditions that depend on A_2 alone (four on W_{10} and one at the IF of step 9) are absorbed by enumerating only *good* classes. These are the preimage sets $h^{-1}(v)$ restricted to values of A_2 that satisfy the five conditions. A candidate is therefore accepted only if $h(A_2) = v$ and A_2 satisfies the five conditions: a class value v with good elements does not guarantee that the recovered A_2 is good. Enumerating a set V of the largest classes costs about $(1 + 2^{-6})\,|V|$ look-ups per chaining value, since the second look-up is made only for candidates that pass the W_9 filter, and the expected number of solutions per chaining value is

$$\mathrm{mass}(V) \cdot 2^{-32}, \qquad \mathrm{mass}(V) = \sum_{v \in V} \left| h^{-1}(v) \cap \mathrm{good} \right|$$

before the W_9 filter, and $\mathrm{mass}(V) \cdot 2^{-38}$ after it. Here $\mathrm{mass}(V)$ is the number of good values of A_2 in the selected classes, and V is a design parameter.

Application to T23. For T23 there are $2^{27.97}$ good values of A_2; the largest good classes have 4 096 elements each. With $|V| = 300$ class values per chaining value, all of this size, so that every look-up has the same expected yield, these classes contain $\mathrm{mass}(V) = 2^{20.23}$ good values of A_2, so a chaining value yields $2^{-11.77}$ solutions in expectation before the W_9 filter. Single-word pinning yields one solution in expectation, which passes step 17 with probability 2^{-32}. Every solution that passes the W_9 filter satisfies steps 16 and 17 by construction, since the differences of W_{16} and W_{17} come from those of W_9 and W_{10}.

3.4 Choosing A_0

A_0 carries no condition and enters only $E_0, \ldots, E_4$ and $W_0, \ldots, W_8$; in particular $W_8 = c_8(A_1, A_2) - A_0$ with c_8 independent of A_0, so W_8 is affine in A_0. Instead of the value from Step 1 we fix, for the whole attack, a value of A_0 selected among 4 096 candidate values for a high probability of the four W_8 conditions given the W_9 conditions. For T23 this raises that probability from $2^{-6.09}$ (Step-1 value) and 2^{-4} (uniform A_0) to $2^{-2.62}$. The estimate is re-measured on fresh samples, so it carries no selection bias.

3.5 Complexity and When Joint Pinning Pays Off

We write the attack complexity as

$$\log_2 T = u + x_{\text{item}} + t, \tag{5}$$

where 2^u is the cost of one entry (one first block, the look-ups and the head checks) in compressions, $2^{-x_{\text{item}}}$ is the expected number of solutions per entry that pass all conditions before the tail, and 2^{-t} is the probability that such a solution is accepted as a collision by the tail checks. Section 4.2 defines these quantities precisely and fixes the step at which the tail starts. With both tails starting after step 17, the step-17 factor $2^{-c_{17}}$ of single-word pinning is part of x_{item} and joint pinning replaces it by $2^{-(32-\log_2 \text{mass}(V))}$, which decomposes the gain as

$$G = c_{17} - \big(32 - \log_2 \text{mass}(V)\big) - G_u - G_{x'} - G_t, \tag{6}$$

where G_u is the extra cost per entry, $G_{x'}$ the change in the re-paid head conditions, and G_t the change in the tail. Joint pinning pays off only if the step-17 factor it removes outweighs the sum of the three other terms. The decisive comparison is usually between c_{17} and $32 - \log_2 \text{mass}(V)$. The class mass is large when few conditions involve A_2 alone, since every such condition shrinks the good classes. Joint pinning therefore suits characteristics whose step 17 is expensive and whose W_{10} and step-9 conditions are sparse. Sections 4.3 and 5.1 evaluate (6) once all factors have been measured.

4 The 39-Step Attack and Its Cost

This section turns joint pinning into a collision attack on 39-step SHA-256 and establishes its cost. Collision attacks on step-reduced SHA-2 are usually costed by counting conditions and treating them as independent. This is reliable when conditions are sparse but can be off by several bits where they interact. Pinning acts at such points: at the pinned steps and at the boundary between the adapted head and the probabilistic tail. We therefore obtain every factor of (5) by analysis, exact enumeration or direct measurement, and we check the head distribution with real first blocks.

4.1 Attack

The attack uses the characteristic T23 and its Step-1 solution from [12], with (W_{14}, W_{15}) and the pinning targets $(W^\star_{16}, W^\star_{17})$ taken from the semi-free-start pair of T23 (Appendix C.3).

Precomputation. Build the table of f^{-1} (2^{32} entries grouped by the top 28 bits of the image, 16 GiB plus a 1 GiB index); 83–90 s on 32 threads, about $2^{35.4}$ SHA-NI compressions. Enumerate all A_2, keep those satisfying the five A_2-only conditions, and group them by $h(A_2)$; keep the 300 largest classes ($\text{mass}(V) = 2^{20.23}$). Fix A_0 as in Sect. 3.4.

Online phase.

1. Pick a first block B_0 and compute CV_1.
2. For every class value v, solve as in Sect. 3.3, filtering on W_9 after the first look-up.
3. For every solution, derive $B_1 = (W_0, \ldots, W_{15})$, set B'_1 by the word-wise XOR difference of the published pair, and check the head conditions on W_8, which include the signs of its differences.
4. Evaluate the remaining steps, stopping at the first violated condition.
5. If the two compressions of the second block collide, output $(B_0 \| B_1, B_0 \| B'_1)$; otherwise continue with the next solution, and return to item 1 once the solutions of B_0 are exhausted.

The two messages have equal length, so their padding blocks are identical and the collision carries over to the hash value. A solution is thus *accepted* if it satisfies every condition of the characteristic that the attack checks and the pair collides. All our results use this rule.

Memory and precomputation. The SHA-256 attacks keep the table of f^{-1} resident (17 GiB, with a peak of 18 GiB while it is built); the class census of the joint variant uses an 8 GiB counter array, which is released before the table is built. Building both takes less than 2^{36} SHA-NI compressions, which is negligible against the attack.

4.2 Cost Model

An *entry* is one first block, i.e. one chaining value CV_1, together with the solving step of Sect. 3 and the head checks of all its solutions. Let H be the number of solutions of an entry that pass the head, and Z the number of them that would be accepted if the whole entry were processed; the attack itself stops at the first accepted pair. We define $T = 2^u/\mathbb{E}[Z]$, where 2^u is the cost of an entry, so $\log_2 \mathbb{E}[Z] = -(x_{\text{item}} + t)$ with $2^{-x_{\text{item}}} = \mathbb{E}[H]$ and $2^{-t} = \mathbb{E}[Z]/\mathbb{E}[H]$. Thus 2^{-t} is the acceptance rate of the tail averaged over head-passing solutions, weighted as the attack produces them. This is the cost per expected collision, the quantity that condition counting estimates in previous work. Every accepted pair collides, so the accepted event is contained in the success event of the attack. Charging every entry its full cost, the expected cost of the first collision is $2^u/\Pr[Z \geq 1] \geq T$; stopping early inside an entry only lowers it. If an entry has at most $s_{\max}$ solutions, then $Z \leq s_{\max}$ and $\Pr[Z \geq 1] \geq \mathbb{E}[Z]/s_{\max}$, so the first-collision cost is at most $s_{\max} \cdot T$. For SHA-256 with a fixed A_0, the largest preimage size of the 32-bit map f is 14, found by evaluating f on all 2^{32} inputs. Single-word pinning performs one inversion per entry, so $s_{\max} = 14$. Joint pinning inverts in two levels for each of the 300 classes, each level with at most 14 branches, so $s_{\max} \leq 300 \cdot 14^2 = 58\,800$. The quantities are measured under the standard assumption that CV_1 is uniformly distributed. As in [12], the one-off cost of Step 1 is not included.

Head and tail. The tail starts at a fixed step L_0, and the head comprises all conditions before it. For joint pinning on T23, $L_0 = 18$, and the head factor is the yield measured on the attack pipeline. For single-word pinning on T23 we also take $L_0 = 18$, so that both variants share the same tail; its head factor is the head rate measured on the pipeline times the step-17 factor 2^{-32} of Lemma 1. For single-word pinning on 38 steps, $L_0 = 17$, and the head factor is the head rate measured on the pipeline.

Unit. We express T in single-thread scalar compressions of the attacked step count, the primitive closest to the per-trial accounting of [12]; one compression comprises the message expansion, all steps and the feed-forward. The entry cost u is timed on the target machine for the complete entry: one first-block compression, the table look-ups, and the step-by-step checks of every solution until its first violated condition. A random look-up into the 2^{32}-entry table of f^{-1} costs 3.5–4.0 scalar compressions, against about 350 for the guess-and-determine inversion of [17] in our implementation (18 guessed bits, all preimages, 512 candidates in parallel by bit-slicing in a 512-bit vector register; single thread, at a machine load of about one half). Appendix B.2 lists the SHA-256 results also in SHA-NI units.

4.3 Complexity

Table 2 decomposes (5) for both variants on T23; Sect. 4.4 establishes the factors. An entry of joint pinning costs 243.5 scalar compressions ($2^{7.93}$), dominated by about 305 table look-ups; a single-word entry costs 5.5. Such a look-up costs about one fifth of the random look-up of Sect. 4.2. This is consistent with locality in the table accesses: on the largest classes the values of h are close, and a look-up takes about 136 ns, against 158–333 ns on classes whose values of h are spread out. The entry cost above is timed as a whole and does not depend on this explanation. An entry yields $2^{-17.77}$ solutions in expectation, and each passes the remaining head conditions (on W_8) with probability $2^{-2.62}$ by the head sampler and $2^{-2.67}$ in the pipeline run (1 482 of 9 439 solutions). The pipeline measures $x_{\mathrm{item}} = 20.47$ directly (1 482 head passes in 2^{31} entries). Every such solution satisfies steps 16 and 17 by construction, and the tail from step 18 is $2^{-80.41}$, obtained by exact enumeration and multilevel splitting (Sect. 4.4), so

$$\log_2 T = u + x_{\mathrm{item}} + t \approx 7.93 + 20.47 + 80.41 \approx 108.80.$$

The total is $2^{108.8}$ per expected collision, a factor of $2^{19.2}$ below the birthday bound. Combining the sampling intervals of the head and the tail gives the approximate range $[108.59, 109.03]$ for $\log_2 T$, conditional on the recorded head pool and the measured entry cost; the entry cost is a timing and varies by a few tenths of a bit with the machine load (Appendix B.2). The first two factors, $2^{7.93+20.47} = 2^{28.4}$ compressions including the re-paid head conditions, replace the 2^{64} of steps 16 and 17. With the same probability estimates, the expected cost of the first collision is at most $2^{108.80+15.84} = 2^{124.6}$ (Sect. 4.2).

Table 2. Decomposition of the attack complexity on 39-step T23, in scalar units.

Pinning	u	x_{item}	t	$\log_2 T$
joint	7.93	20.47	80.41	108.80
single-word	2.46	$8.49 + 32$	80.42	123.37

Table 3. Cost of the 39-step attacks on T23 ($\log_2$, single thread): per expected collision, and the bound on the expected cost of the first collision under worst-case clustering.

	Per expected collision		First collision, at most	
Pinning	Scalar	SHA-NI	Scalar	SHA-NI
joint	108.80	111.00	124.6	126.8
single-word	123.37	125.11	127.2	128.9

Single-word pinning on the same characteristic costs $2^{123.4}$, already below the birthday bound. It removes the step-16 conditions; joint pinning removes the 2^{-32} at step 17 as well. The first-collision cost of single-word pinning is at most $2^{123.37+3.81} = 2^{127.2}$, given the same probability estimates. Both bounds are below the birthday bound in scalar units; in single-thread SHA-NI units the bound of joint pinning remains below it, at $2^{126.8}$, while that of single-word pinning rises above it, to $2^{125.11+3.81} = 2^{128.9}$ (Table 11); Table 3 collects these figures. These worst cases are far from what we observe. We recover the chaining values of the 3 102 final survivors of eight splitting runs of joint pinning with different seeds (351 to 419 survivors per run) and re-solve each with all 300 classes: every one has exactly one solution. These survivors descend from 2 417 distinct roots, i.e. values of W_{18} that pass step 18, and no clustering of solutions appears among them. The bound $2^{124.6}$ holds without this measurement. Joint pinning gains 14.6 bits over single-word pinning (Table 4 in Sect. 5.1), because the step-17 factor 2^{-32} is replaced by $2^{-11.77}$ at an extra cost of $G_u = 5.47$ bits per entry. Of this gain, 11.28 bits hold with the same A_0 for both variants, its Step-1 value, where the joint attack costs $2^{112.09}$ (Table 11); selecting A_0 for joint pinning saves the other 3.29 bits. The selected A_0 lowers x_{item} by 3.52 bits, from 23.99 to 20.47, while the entry cost rises by 0.20 bits, from 212.2 to 243.5 scalar compressions; the two tails agree within 0.04 bits.

4.4 Establishing the Factors

All measurements were taken on an Intel Xeon Platinum 8582C (Appendix B.2).

Head. The head factor is measured by running the attack pipeline itself: counter-mode first blocks, the solving step, and step-by-step checking of every solution until the first failure. The 64-bit counter serves the measurement only; the attack draws first blocks from the whole 512-bit space. For each SHA-256 variant we process 2^{31} to 2^{36} entries. The implementation stores at most 64 solutions of an

entry for checking; solutions beyond this are counted but not checked. In the measured runs, the largest number of solutions observed for one entry is 14 for single-word pinning, equal to the maximum preimage size of f, and 1 for joint pinning. The number of solutions per entry matches its exact expectation (1 for single-word pinning, $\text{mass}(V) \cdot 2^{-38}$ for joint pinning after the W_9 filter). A separate head sampler draws solutions uniformly from the solution set. It agrees with the pipeline to within 0.01 bits for single-word pinning and 0.08 bits for joint pinning with the selected A_0 (20.39 against 20.47 on T23). We use the pipeline values for the SHA-256 attacks in Table 1; Sect. 5.3 gives the head factors of SHA-512.

Step 17. For single-word pinning, the step-17 factor is exact (Lemma 1 in Appendix A.2). The proof shows that, for fixed A_1, the two words E_{-3} and E_{-4} of CV_1 map bijectively to (W_{16}, W_{17}), while the head conditions of T23 depend on A_0, A_1, A_2 and Step-1 constants only. Hence every fixed A_1 satisfies $W_{16} = W_{16}^{\star}$ with probability 2^{-32} and both equations with probability 2^{-64}, which is also the cost of steps 16 and 17 without pinning. The pipeline agrees with the lemma (Appendix A.2).

Tail. The tail probability 2^{-t} starts at the step L_0 fixed in Sect. 4.2 ($L_0 = 18$ for T23 with either variant, $L_0 = 17$ for single-word pinning on 38 steps). We fix the pinned words and the head (for single-word pinning on T23, conditioned on step 17), enumerate all 2^{32} values of the free word at step L_0 exactly, and then use multilevel splitting [8,5,1] over steps $L_0 + 1, \ldots, 23$. Survivors of each step are replicated and extended with fresh uniform message words. From step 23 on, each trial carries a head drawn from the attack distribution, since $W_8, \ldots, W_{10}$ change with A_0, A_1, A_2 and enter the expansion of $W_{23}, \ldots, W_{26}$; the remaining steps are deterministic. For 38 steps, the four conditions on W_7 are conditions on the same message words, so step 23 is measured conditionally on them. For uniform CV_1, this samples from the exact conditional distribution of the tail inputs (Remark 1 in Appendix A.2). Every survivor of the last step collides in the measurement program, and 64 per characteristic were recomputed with an independent reference implementation of SHA-256, which confirms each of them as a colliding second-block pair. Intervals are cluster intervals over the roots of the splitting tree (Appendix B.1).

With every step checked against all conditions of the characteristic, the measured tail of T23 is $2^{-80.41}$ ($[80.27, 80.56]$), against 81 counted conditions. Deviations from counting occur only at the first one or two steps after the pinned words, where conditions are fixed by the pinned values or linearly dependent. From there on every step matches its count to within 0.07 bits (Table 10 in Appendix B.1).

Composition. Equation (5) multiplies a head factor measured on the pipeline with a tail factor measured from sampled heads. For joint pinning on T23 and uniform CV_1, this product is exact: the attack produces heads uniformly on the set of head-passing solutions, which is the distribution the sampler draws

from (Proposition 1 in Appendix A.2). We also check the implementation, and the head distribution under real first blocks: the pipeline writes out every head-passing solution it produces, and these real heads replace the sampled heads in the tail measurement. The real heads are combined with fresh uniform tail words, so the check concerns the distribution of the heads; the independence of head and tail rests on the uniformity of CV_1 (Remark 1). For T23 with joint pinning, 6 085 real heads give a tail of $2^{-80.36}$ against $2^{-80.45}$ from sampled heads. The difference is -0.086 bits (real minus sampled; paired 95% interval $[-0.301, +0.102]$), and against a sampled-head subset of the same size it is -0.004. At this resolution we detect no difference between real and sampled heads. In the pipeline run itself, none of the 6 085 real heads passes step 18, where 0.017 are expected at $2^{-18.42}$; a direct test of the tail under real first blocks is therefore out of reach, and the tail rests on Proposition 1 and the uniformity of CV_1. The sampled-head tails $2^{-80.45}$ here ($[80.31, 80.60]$) and $2^{-80.41}$ above ($[80.27, 80.56]$) come from separate runs and agree within their intervals. These intervals quantify the tail-sampling uncertainty conditional on the recorded head pools and do not include the uncertainty from generating those pools. The same check for C38 and for SHA-512 is reported with those attacks (Sects. 5.2 and 5.3).

5 38-Step SHA-256 and SHA-512

Table 1 summarizes all our attacks. The cost model and the measurements of Sect. 4 carry over to other targets; this section shows where pinning changes. On the 38-step characteristic T5, step 17 is cheap and the good classes are small, so the cost model selects single-word pinning. A condition-count search then gives the new characteristic C38, which lowers the attack further (Sect. 5.2). On SHA-512, single-word pinning carries over unchanged, and the 64-bit map f is inverted without a table.

5.1 38-Step SHA-256: Choosing the Pinning Variant

Table 4 evaluates (6) for T23 (Sect. 4.3) and for the 38-step characteristic T5 of [12]. On 39-step T23, step 17 costs the full $c_{17} = 32$ bits, few conditions depend on A_2, and good classes are large. Here joint pinning gains 14.6 bits. On 38-step T5, step 17 costs $c_{17} = 16$ bits while 23 conditions depend on A_2, which leaves 768 good values. Here joint pinning, with $A_0 =$ `b5da7d58` selected as in Sect. 3.4, costs $2^{103.34}$, so $G = -11.6$ and single-word pinning is the variant of choice. The terms $G_{x'}$ and G_t are small in both cases; the sign of G is decided by c_{17} against $32 - \log_2 \text{mass}(V)$. The last column of the table equals the difference of the two attack complexities. Its head estimates come from the pipeline, and from the sampler for joint pinning on T5, where the step-17 factor is counted in the head; this regroups the factors and leaves the total unchanged. With single-word pinning, T5 costs $2^{91.73}$ (Table 6 in Sect. 5.2).

Table 4. Gain of joint over single-word pinning, Eq. (6), in scalar units.

Steps	Characteristic	c_{17}	$\log_2 \text{mass}(V)$	G_u	$G_{x'}$	G_t	G
39	T23	32	20.23	5.47	0.21	−0.01	+14.56
38	T5	16	9.58	5.08	0.14	−0.02	−11.61

5.2 A Condition-Count Search for the Framework

A better characteristic lowers the 38-step attack under both accountings: by about 4 bits under the framework, from the reported $2^{104.3}$ to $2^{100.3}$, and by about 0.9 bits with single-word pinning, which already removes step 16, where four of the five saved conditions lie. Independently of pinning, the framework of [12] leaves a mismatch between the search for characteristics and the cost of the attack. The search proceeds in stages and minimizes Hamming weights at every stage. The number γ of conditions from step 16 on, which determines the complexity, is counted only afterwards. As [22] does for its own search, we make the condition count the objective, here within the stages of [12]; the reported γ is recounted afterwards with each condition counted once and linear dependencies among the two-bit conditions removed. We first evaluate the result under the accounting of the framework and then combine it with single-word pinning.

Search procedure. We encode the signed characteristic, the propagation through all Boolean functions and the message expansion as a SAT model. Conditions are counted with the per-variable model of [22]: an IF or MAJ bit contributes the number of independent conditions that its uncontrolled inputs must satisfy, and every active σ_0 or σ_1 output bit contributes its two-bit conditions on W. The search keeps the stages of [12] (Appendix C.1) and differs in three decisions. First, after minimizing $\sum_i \text{HW}(\Delta W_i)$, it minimizes the condition count from step 16 on instead of a Hamming weight. Second, allowing two conditions of slack in this count, it reduces the count from step 14 on; we stopped this stage at the value 217 without a proof of optimality. This count is a proxy for $|(W_{14}, W_{15})|$ and the validity of the first block, so that the reduction in γ is not paid back there. The actual $|(W_{14}, W_{15})|$ and N_{pro} are evaluated afterwards on the Step-1 solution. Third, it adds value propagation on a window of steps, which excludes some contradictions early; the final value model then confirms that the characteristic has a conforming pair.

Value windows. At 38 steps, characteristics that are optimal in the difference model can be contradictory in their values, with minimal contradictory windows inside steps 9–16. We add value propagation on the window of steps 10–15 to the stage that minimizes the controlled ΔE_i. This excludes contradictions of that span at a cost of about 4 hours per query. A contradiction outside the window is excluded by a local clause. With this, the final stage finds a conforming pair for C38 in 16 seconds.

Table 5. 38-step characteristics of SHA-256 under the framework of [12]. The last two rows apply one accounting to the Step-1 solution of each published pair.

Characteristic	γ	$\log_2 \lvert(W_{14}, W_{15})\rvert$	N_{pro}	$\log_2 T$
T5, as reported in [12]	102	2	4	104.3
T5, same accounting	103	0	4.00	107.09
C38, same accounting	98	2.00	4.00	**100.32**

Results. Table 5 compares C38 with T5 of [12] under one accounting: the same condition count, $|(W_{14}, W_{15})|$ counted exactly, and N_{pro} estimated from 2^{32} chaining values, both on the Step-1 solution of each colliding pair. These are condition-count complexities, as in [12]. The row marked "framework" in Table 1 uses the same accounting. Under this accounting C38 has $\gamma = 98$ against 103 for T5, with $|(W_{14}, W_{15})| = 2^2$ and $N_{\text{pro}} = 4$, and its framework complexity is $2^{100.32}$, against $2^{107.09}$ for T5 and the $2^{104.3}$ reported in [12]. The gain holds beyond the first Step-1 solution: ten further Step-1 solutions of C38, enumerated with the SAT model, all give $|(W_{14}, W_{15})| = 2^2$ and $N_{\text{pro}} = 4$. Both characteristics come with semi-free-start colliding pairs (Appendix C.2 for C38), verified with a reference implementation.

Combination with single-word pinning. C38 and T5 have the same number of conditions on every step from 18 on; the counts differ in steps 16 and 17 (25 and 15 conditions for C38, 29 and 16 for T5). Single-word pinning removes step 16 for both, so four of the five conditions that C38 saves disappear, and the comparison moves to step 17 and the tail. With C38, an entry costs 5.8 scalar compressions, the head yields $2^{-16.29}$ solutions per entry in expectation, and the tail from step 17 is $2^{-72.02}$, for a total of $2^{90.85}$ (Table 6). With T5, the same procedure gives $2^{91.73}$. The two figures for C38, $2^{100.3}$ and $2^{90.9}$, are a condition-count estimate under the framework and a measured complexity of the pinned attack, respectively. The gap between them is accounted for as follows: pinning removes the 25 conditions of step 16, and pays 12.29 bits more in the head and 2 bits for fixing (W_{14}, W_{15}); the measured complexity also charges the entry cost, 2.54 bits, and its tail lies 0.98 bits below the count of the same conditions ($2^{100.3}$ includes 0.32 bits from the sum of the two terms of the framework formula). For C38, both the composition of head and tail and the first tail step are checked directly. The tail measured from 854 987 real heads is $2^{-72.01}$ against $2^{-72.02}$ from sampled heads. The difference is -0.004 bits (real minus sampled; paired 95% interval $[-0.068, +0.057]$), conditional on the recorded head pools. In addition, the pipeline reaches step 17 of C38 directly: 23 of 854 987 real heads pass it, against 26.1 predicted by the product of the separately measured factors.

Table 6. Decomposition of the attack complexity on 38 steps with single-word pinning, in scalar units, with the tail from step 17.

Characteristic	u	x_{item}	t	$\log_2 T$
C38	2.54	16.29	72.02	90.85
T5	2.48	16.27	72.98	91.73

5.3 SHA-512

Inversion and pinning targets. Single-word pinning applies to SHA-512 unchanged, with $f(x) = \sigma_0(x) - x$ on 64-bit words. We evaluate only single-word pinning here: on the published 39-step Step-1 solution, a class of h yields $2^{-36.8}$ solutions per chaining value on average, against the step-17 factor $2^{-33.61}$ (Sect. 3.5); the gap is larger at 38 steps, and joint pinning on the largest good classes is not evaluated. We invert f without a table. For $\sigma_0(Y) = Y + t$, we guess the low 8 bits of Y; the bit equations at positions $0, \ldots, 55$ then determine the remaining bits one by one along the carries of the addition, and the last 8 equations act as a filter. Guessing the 8 least significant bits to determine a message word of SHA-512 goes back to the 24-step attack of [7, Sect. 5], and the full version of [17] (Appendix B) solves the same equation by guessing the low 8 bits and 7 carries; propagating the carries bit by bit, we guess only the 8 bits. The small offsets 1, 7 and 8 of σ_0 in SHA-512 keep the number of guessed bits low, against 18 for SHA-256, and the 256 branches run bit-sliced in a 256-bit register at a cost of 2.5–3.2 scalar compressions. We also choose the pinning target so that step 17 is cheap. We sample values from the set admissible at step 16, rank them by the number of admissible W_{17}, and screen the 32 best (plus the value of the published pair) with a short tail measurement up to step 20. The tail of each selected target is then measured in a full run with about 128 times the samples of the screening. The selected targets lower the attack complexity by 2.7 and 5.2 bits compared with the published pair's value at 38 and 39 steps.

Complexity. Since A_0 carries no condition and has 64 free bits, an entry here is one first block together with $B = 1\,024$ values of A_0. Each sub-attempt (CV_1, A_0) inverts f, solves for A_1 and checks the head. This costs 3.16 (38 steps) and 2.88 (39 steps) scalar compressions single-threaded, and the first block adds $1/B$ per sub-attempt. We state costs and yields per sub-attempt. An entry costs B times as much and yields B times as many collisions in expectation, regardless of correlations between sub-attempts, so the cost per expected collision is the same in either unit. On the characteristics of [12], with the tail starting at step 17 ($L_0 = 17$) in both cases, the costs per expected collision are

$$2^{1.66+13.01+86.01} = 2^{100.7} \quad (38 \text{ steps}), \quad 2^{1.526+9.00+120.82} = 2^{131.3} \quad (39 \text{ steps}),$$

against $2^{125.4}$ and 2^{178} in [12]. The head factors 13.01 and 9.00 are measured on 2^{30} sub-attempts with random CV_1 and A_0; the real-first-block runs of the composition check give 13.002 and 9.001. A sub-attempt that passes the head then costs about two more compressions, which adds less than 0.01 bits.

Tail measurement. The tails follow the procedure of Sect. 4.4 with $L_0 = 17$ and 65 536 roots that pass step 17. At 39 steps, step 17 is enumerated exactly by a depth-first search over the bits of W_{17} from the least significant one, which tracks the carries and prunes on the conditions of the step: 1 404 469 248 of the 2^{64} values of W_{17} pass, a rate of $2^{-33.61}$, and the roots are drawn from them by reservoir sampling. At 38 steps the passing set is too large to enumerate, and step 17 is estimated by sampling, with 1 048 986 passes in 2^{39} uniform trials ($2^{-19.00}$). Steps 18–23 use multilevel splitting, where the number of children per parent is adapted in powers of two towards a target number of survivors within a budget of 2^{40} trials per step, and the later steps are evaluated with heads drawn from the attack distribution. The runs end with 261 941 and 113 final survivors, all of which collide, and give tails of $2^{-86.01}$ and $2^{-120.82}$ with standard errors of 0.011 and 0.136 bits (Table 15); the standard errors add the variances of the steps, neglecting covariance across steps, and are conditional on the recorded head pools. Three later runs at 39 steps with the same target and about 1 000 final survivors each, taken for the composition check below, give $2^{-120.61}$, $2^{-120.65}$ and $2^{-120.70}$ (standard error 0.05 each); the attack complexity uses $2^{-120.82}$.

First collision. The first-collision bound of Sect. 4.2 applies per entry with $s_{\max} \leq B \cdot m$, where m is the largest preimage size of the 64-bit map f. Each guess of the low 8 bits gives at most one preimage, so $m \leq 2^8$ and $s_{\max} \leq 2^{18}$. With the measured probabilities, the expected costs of the first collision are therefore at most $2^{118.7}$ and $2^{149.3}$, far below the birthday bound of 2^{256}. In practice preimages are small: over 2^{12} random targets, the number of preimages is consistent with a Poisson distribution with mean 1 and never exceeds 5.

Step checks. The tails are measured against the condition tables of [12] with the corrections of Appendix C.4. Every final survivor of the tail measurement is a collision; 64 per step count were expanded to (CV_1, M, M'), with CV_1 recovered from the words of the survivor, and verified to collide under the r-step compression function. The step checks are conservative for differences in the most significant bit. Such a difference is the same modular difference for either sign, so the checks accept both signs. Every enforced two-bit condition, however, is checked against the relation it has in the published pair, with the conventions listed in Appendix C.4. A solution that flips the sign of a difference in the most significant bit may need the opposite relations later, and it is then rejected. The measured tails therefore estimate the probability of an event contained in the success event, and the complexities are conservative in this respect. The pinning targets are listed in Appendix C.4.

Composition. The composition check of Sect. 4.4 computes each chaining value with a real first-block compression and uses the attack's entry structure ($B =$ 1 024 values of A_0 per chaining value). At 38 steps, 65 536 real heads give a tail from step 17 that differs from the sampled-head tail by $+0.001$ bits (approximate 95% interval ± 0.032). At 39 steps, we compare the part of the tail that reads the

head, steps 24–38; 1 048 576 real heads give a difference of +0.036 bits (±0.127). The earlier steps do not read the head. Heads with a fresh chaining value for every value of A_0 agree within the same intervals. Here the steps before the compared part are measured on different survivors, so the intervals are unpaired. They add the variances of the steps and of the two runs, neglecting covariance across steps; as for SHA-256, they are conditional on the recorded head pools.

6 Conclusion

The probability of the uncontrolled part limits the two-block framework, which pays every condition from step 16 on by probability. For 39-step SHA-256, its condition-count estimate for the characteristic T23 is 2^{168}, and the heuristic estimate anchored at the target values of W_{16} and W_{17} is 2^{145}, both above the birthday bound. We showed that the first two expanded message words of the second block can be controlled. For each chaining value, joint pinning searches for assignments to two state words that carry no direct condition, and every assignment returned that meets the head conditions on W_9 and W_{10} satisfies steps 16 and 17 by construction; the two coupled equations can be solved in turn once the value of a function with a small image is fixed. On T23, this gives a theoretical collision attack: single-word pinning already crosses the birthday bound at $2^{123.4}$ compressions per expected collision, and joint pinning lowers this by 14.6 bits to $2^{108.8}$. For uniform chaining values and with the measured probabilities, the expected cost of the first collision is at most $2^{124.6}$ even under worst-case clustering, and the successful entries we sampled show no clustering of solutions. The attack covers 39 steps, as many as the best semi-free-start collisions on SHA-256. Combined with the characteristic C38 from our condition-count search, single-word pinning gives $2^{90.9}$ on 38-step SHA-256, and on 38- and 39-step SHA-512 it gives $2^{100.7}$ and $2^{131.3}$.

Every factor of these complexities is established by analysis, exact enumeration or measurement, for a uniform chaining value. Replacing the sampled heads by heads produced by the real attack pipeline leaves the measured tails unchanged within their intervals on SHA-256 and SHA-512, on 39-step SHA-512 for the part of the tail that reads the head. Joint pinning gains 14.6 bits over single-word pinning on T23 and costs 11.6 bits more on T5: the outcome is decided by the step-17 factor it removes against the class mass, the extra look-ups and the re-paid head conditions. This contrast makes the class mass attainable at a fixed look-up budget a natural target for characteristic search, together with the cost of the remaining tail.

A natural next question is whether the two-block framework can reach 40 steps of SHA-256 below the birthday bound. The message-difference pattern of the 39-step attack cannot simply be shifted by one step: with differences in $W_9, \ldots, W_{13}$, the word $W_{16} = \sigma_1(W_{14}) + W_9 + \sigma_0(W_1) + W_0$ receives the difference of W_9, and none of its other inputs carries a difference that could cancel it, so the local collision does not close. A 40-step attack along this line therefore needs a different message-difference pattern. We leave open whether a different message-

difference pattern can support a 40-step characteristic for which joint pinning remains effective.

References

1. Cérou, F., Guyader, A.: Adaptive multilevel splitting for rare event analysis. Stochastic Analysis and Applications **25**(2), 417–443 (2007). https://doi.org/10.1080/07362990601139628
2. De Cannière, C., Rechberger, C.: Finding SHA-1 characteristics: General results and applications. In: Advances in Cryptology – ASIACRYPT 2006. Lecture Notes in Computer Science, vol. 4284, pp. 1–20. Springer (2006). https://doi.org/10.1007/11935230_1
3. Dobraunig, C., Eichlseder, M., Mendel, F.: Analysis of SHA-512/224 and SHA-512/256. In: Advances in Cryptology – ASIACRYPT 2015, Part II. Lecture Notes in Computer Science, vol. 9453, pp. 612–630. Springer (2015). https://doi.org/10.1007/978-3-662-48800-3_25
4. Eichlseder, M., Mendel, F., Schläffer, M.: Branching heuristics in differential collision search with applications to SHA-512. In: Fast Software Encryption – FSE 2014. Lecture Notes in Computer Science, vol. 8540, pp. 473–488. Springer (2015). https://doi.org/10.1007/978-3-662-46706-0_24
5. Glasserman, P., Heidelberger, P., Shahabuddin, P., Zajic, T.: Multilevel splitting for estimating rare event probabilities. Operations Research **47**(4), 585–600 (1999). https://doi.org/10.1287/opre.47.4.585
6. Hosoyamada, A., Sasaki, Y.: Quantum collision attacks on reduced SHA-256 and SHA-512. In: Advances in Cryptology – CRYPTO 2021, Part I. Lecture Notes in Computer Science, vol. 12825, pp. 616–646. Springer (2021). https://doi.org/10.1007/978-3-030-84242-0_22, cryptology ePrint Archive, Paper 2021/292
7. Indesteege, S., Mendel, F., Preneel, B., Rechberger, C.: Collisions and other non-random properties for step-reduced SHA-256. In: Selected Areas in Cryptography – SAC 2008. Lecture Notes in Computer Science, vol. 5381, pp. 276–293. Springer (2009). https://doi.org/10.1007/978-3-642-04159-4_18
8. Kahn, H., Harris, T.E.: Estimation of particle transmission by random sampling. In: Monte Carlo Method. pp. 27–30. No. 12 in National Bureau of Standards Applied Mathematics Series, U.S. Government Printing Office (1951)
9. Li, Y., Liu, F., Wang, G., Dong, X., Sun, S.: The first practical collision for 31-step SHA-256. In: Advances in Cryptology – ASIACRYPT 2024, Part VII. Lecture Notes in Computer Science, vol. 15490, pp. 237–266. Springer (2025). https://doi.org/10.1007/978-981-96-0941-3_8
10. Li, Y., Liu, F., Wang, G., Qian, H., Dong, X., Sun, S., Shi, D.: New records in collision attacks on SHA-2. Journal of Cryptology **39**(4), 27 (2026). https://doi.org/10.1007/s00145-026-09582-9
11. Li, Y., Liu, F., Wang, G., Shi, J.: Pushing the limit of memory-efficient collision attack framework for SHA-2. In: Advances in Cryptology – CRYPTO 2026, Part VI. Lecture Notes in Computer Science, vol. 16805, pp. 289–320. Springer (2026). https://doi.org/10.1007/978-3-032-35412-9_10, cryptology ePrint Archive, Paper 2026/1080
12. Li, Y., Zhang, Z., Li, M., Liu, F., Qian, H., Zhu, J.: Pushing collision attacks on SHA-2 to 39 steps. Cryptology ePrint Archive, Paper 2026/1120 (2026), https://eprint.iacr.org/2026/1120

13. Liu, F., Wang, G., Sarkar, S., Anand, R., Meier, W., Li, Y., Isobe, T.: Analysis of RIPEMD-160: New collision attacks and finding characteristics with MILP. In: Advances in Cryptology – EUROCRYPT 2023, Part IV. Lecture Notes in Computer Science, vol. 14007, pp. 189–219. Springer (2023). https://doi.org/10.1007/978-3-031-30634-1_7
14. Mendel, F., Nad, T., Schläffer, M.: Finding SHA-2 characteristics: Searching through a minefield of contradictions. In: Advances in Cryptology – ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 288–307. Springer (2011). https://doi.org/10.1007/978-3-642-25385-0_16
15. Mendel, F., Nad, T., Schläffer, M.: Improving local collisions: New attacks on reduced SHA-256. In: Advances in Cryptology – EUROCRYPT 2013. Lecture Notes in Computer Science, vol. 7881, pp. 262–278. Springer (2013). https://doi.org/10.1007/978-3-642-38348-9_16
16. National Institute of Standards and Technology: Secure hash standard (SHS). Tech. Rep. FIPS PUB 180-4, NIST (2015). https://doi.org/10.6028/NIST.FIPS.180-4
17. Sanadhya, S.K., Sarkar, P.: New collision attacks against up to 24-step SHA-2. In: Progress in Cryptology – INDOCRYPT 2008. Lecture Notes in Computer Science, vol. 5365, pp. 91–103. Springer (2008). https://doi.org/10.1007/978-3-540-89754-5_8, full version: Cryptology ePrint Archive, Paper 2008/270
18. Wang, X., Lai, X., Feng, D., Chen, H., Yu, X.: Cryptanalysis of the hash functions MD4 and RIPEMD. In: Advances in Cryptology – EUROCRYPT 2005. Lecture Notes in Computer Science, vol. 3494, pp. 1–18. Springer (2005). https://doi.org/10.1007/11426639_1
19. Wang, X., Yin, Y.L., Yu, H.: Finding collisions in the full SHA-1. In: Advances in Cryptology – CRYPTO 2005. Lecture Notes in Computer Science, vol. 3621, pp. 17–36. Springer (2005). https://doi.org/10.1007/11535218_2
20. Wang, X., Yu, H.: How to break MD5 and other hash functions. In: Advances in Cryptology – EUROCRYPT 2005. Lecture Notes in Computer Science, vol. 3494, pp. 19–35. Springer (2005). https://doi.org/10.1007/11426639_2
21. Wang, X., Yu, H., Yin, Y.L.: Efficient collision search attacks on SHA-0. In: Advances in Cryptology – CRYPTO 2005. Lecture Notes in Computer Science, vol. 3621, pp. 1–16. Springer (2005). https://doi.org/10.1007/11535218_1
22. Zhang, Z., Li, M., Gao, L., Wang, M.: Collision attacks on SHA-256 up to 37 steps with improved trail search. In: Advances in Cryptology – EUROCRYPT 2026, Part VI. Lecture Notes in Computer Science, vol. 16546, pp. 91–120. Springer (2026). https://doi.org/10.1007/978-3-032-25333-0_4

Supplementary Material

A Pinning: Conditions and Proofs

A.1 Conditions of T23 under Pinning

Table 7 lists, for the 39-step characteristic T23, the conditions up to step 17 that the two pinning variants change, and how each variant pays them. Changing A_1 alters $W_1, \ldots, W_9$, and changing A_2 in addition alters W_{10} and E_6; no other condition up to step 17 is affected (Sect. 3.1). The last row gives the measured head rate per solution: for single-word pinning it covers the W_8 and W_9 conditions; for joint pinning it covers the W_8 conditions after the W_9 filter, with the selected A_0 of Sect. 3.4.

Table 7. Conditions of T23 up to step 17 affected by pinning.

Conditions	Number	Single-word pinning	Joint pinning
step 16	all	by construction	by construction
step 17	all	2^{-32} (Lemma 1)	by construction
W_8	4	re-paid	re-paid
W_9	6	re-paid	re-paid, filtered early
W_{10}	4	unaffected	absorbed by good classes
IF of step 9	1	unaffected	absorbed by good classes
head, per solution		$2^{-8.49}$	$2^{-2.62}$ (given W_9)

A.2 Proofs for the Cost Model

Lemma 1. *Fix the Step-1 values, (W_{14}, W_{15}), A_0 and A_2, and let CV_1 be uniform. Then the expected number of solutions A_1 that pass the pipeline head (the conditions on W_8 and W_9) and satisfy $W_{16} = W_{16}^\star$ and $W_{17} = W_{17}^\star$ is exactly 2^{-32} times the expected number of those that satisfy $W_{16} = W_{16}^\star$.*

Proof of Lemma 1. Fix all words of CV_1 except E_{-3} and E_{-4}, and fix A_1. By (1) and (2), E_0 does not depend on E_{-3}, E_{-4}. Hence $W_1 = A_1 - \Sigma_0(A_0) - \mathrm{MAJ}(A_0, A_{-1}, A_{-2}) - E_{-3} - \Sigma_1(E_0) - \mathrm{IF}(E_0, E_{-1}, E_{-2}) - K_1$ is $-E_{-3}$ plus a constant, while W_2, W_{10} and W_{15} do not involve E_{-3} or E_{-4}. Hence $W_{17} = \sigma_1(W_{15}) + W_{10} + \sigma_0(W_2) + W_1$ is a bijective function of E_{-3} alone. For fixed E_{-3}, $W_0 = A_0 - E_{-4} + \kappa$ with κ independent of E_{-4}, and W_1, W_9, W_{14} do not involve E_{-4}, so W_{16} is a bijective function of E_{-4}. Thus $(E_{-3}, E_{-4}) \mapsto (W_{16}, W_{17})$ is a bijection. For uniform CV_1, every fixed A_1 then satisfies $W_{16} = W_{16}^\star$ with probability 2^{-32} and both equations with probability 2^{-64}. The head conditions in T23 (W_8 and W_9) depend on A_0, A_1, A_2 and Step-1 constants only, not on

CV_1. Let $\mathcal{H}$ be the set of values of A_1 that pass the head conditions on W_8 and W_9. Summing over $\mathcal{H}$ gives the two expectations $|\mathcal{H}| \cdot 2^{-32}$ and $|\mathcal{H}| \cdot 2^{-64}$. □

The pipeline agrees: over 2^{33} entries, the 16 low bits of W_{17} on solutions passing the pipeline head are consistent with uniform ($\chi^2 = 65\,310.6$ on $65\,535$ degrees of freedom).

Remark 1. Fix the Step-1 values, (W_{14}, W_{15}) and a head (A_0, A_1, A_2), hence E_4, E_5, E_6 and W_8, W_9, W_{10}. For uniform CV_1, the words $(W_{16}, \ldots, W_{23})$ are uniform. Given the pinned words, the remaining ones, $(W_{18}, \ldots, W_{23})$ or $(W_{17}, \ldots, W_{23})$, are uniform and independent of the head; the derivation below obtains this from two bijections. The tail from step L_0 on depends on the entry only through these words and W_8, W_9, W_{10}. Sampling $(W_{L_0}, \ldots, W_{23})$ uniformly and the head from the attack distribution is therefore exact, and it assumes only that CV_1 is uniform. For 38 steps the four conditions on W_7 are conditions on the same tuple, which is why step 23 is measured conditionally on them.

Derivation of Remark 1. Step 2 of [12] maps CV_1 to $(W_0, \ldots, W_7)$. Run backwards, W_7 gives E_3 by step 7 and A_{-1} by step 3; likewise W_6, W_5, W_4 give E_2, E_1, E_0 and A_{-2}, A_{-3}, A_{-4}, and then $W_3, \ldots, W_0$ give $E_{-1}, \ldots, E_{-4}$. So both maps

$$CV_1 \mapsto (W_0, \ldots, W_7) \qquad \text{and} \qquad (W_0, \ldots, W_7) \mapsto (W_{16}, \ldots, W_{23})$$

are bijections of 2^{256} values; the inverse of the second solves the expansion for $W_7, W_6, \ldots, W_0$ in turn. Hence for uniform CV_1 and every fixed head, $(W_{16}, \ldots, W_{23})$ is uniform, and given the pinned words the remaining ones are uniform and independent of the head.

Proposition 1. *Consider joint pinning on T23, fix the Step-1 values, A_0 and (W_{14}, W_{15}), and let $\mathcal{S}$ be the set of pairs (A_1, A_2) with A_2 good and in a selected class that satisfy the head conditions on W_8 and W_9. For uniform CV_1, every element of $\mathcal{S}$ is a solution of an entry with probability exactly 2^{-64}, and*

$$\mathbb{E}[H] = 2^{-64}\,|\mathcal{S}|, \qquad \mathbb{E}[Z] = 2^{-64} \sum_{(A_1,A_2)\in\mathcal{S}} \Pr[\mathit{tail} \mid A_1, A_2],$$

where the probability is over uniform $(W_{18}, \ldots, W_{23})$. Hence 2^{-t} is the average of the tail success rate over the uniform distribution on $\mathcal{S}$.

Proof of Proposition 1. Membership in $\mathcal{S}$ does not depend on CV_1. By the proof of Lemma 1, an element of $\mathcal{S}$ gives $(W_{16}, W_{17}) = (W_{16}^\star, W_{17}^\star)$ with probability 2^{-64}. Given this event, $(W_{18}, \ldots, W_{23})$ is uniform and independent of the head by Remark 1, and the tail depends on the entry only through these words and W_8, W_9, W_{10}. Linearity of expectation over $\mathcal{S}$ gives both formulas, and dividing them gives the average. □

With $|\mathcal{S}| = 2^{32} \cdot \mathrm{mass}(V) \cdot 2^{-6} \cdot 2^{-2.62}$, the first formula gives $x_{\mathrm{item}} = 20.39$, the value of the head sampler; the attack uses the pipeline value 20.47.

Several solutions per chaining value. A chaining value can have up to 14 single-word solutions, so a priori the gap between $\mathbb{E}[Z]$ and the success probability of an entry (Sect. 4.2) is at most $\log_2 14 = 3.8$ bits. We measure it at the level of the pipeline head by aggregating head passes per chaining value. Let H_{pre} be the number of solutions of an entry that pass this pipeline head. For joint pinning and for the 38-step attacks, $H_{\text{pre}} = H$. For single-word pinning on T23, the pipeline head stops before the step-17 factor, so $\mathbb{E}[H] = 2^{-32}\,\mathbb{E}[H_{\text{pre}}]$. With $\rho = \mathbb{E}[H_{\text{pre}}(H_{\text{pre}} - 1)]/\mathbb{E}[H_{\text{pre}}]$, $\Pr[H_{\text{pre}} \geq 1] \geq \mathbb{E}[H_{\text{pre}}](1 - \rho/2)$. Over 2^{33} to 2^{36} entries per characteristic, the gap between $\mathbb{E}[H_{\text{pre}}]$ and $\Pr[H_{\text{pre}} \geq 1]$ is at most 0.002 bits (T23: $\rho = 2.78 \cdot 10^{-3}$; T5: $5.8 \cdot 10^{-5}$). For joint pinning the gap is 0 in the measured run, since no chaining value had two solutions passing the pipeline head. Pipeline-head solutions of one chaining value are rare, so clustering at the level of collisions would have to come from strongly correlated tails; for joint pinning, Sect. 4.3 samples this directly. The worst case remains the bound $s_{\max} \cdot T$ of Sect. 4.2.

A.3 A Worked Example of Joint Pinning

Table 8 gives one head-passing solution of the attack pipeline on T23, entry 3 529 047 of the run of Sect. 4.4. All compressions have 39 steps and the feed-forward, and $CV_1 = C(IV, B_0)$. Together with the Step-1 values from A_3 on and the (W_{14}, W_{15}) of the semi-free-start pair, the listed A_0, A_1, A_2 give B_1 by (1) and (2). Running the step function forward on (CV_1, B_1) reproduces these state words, and W_{16} and W_{17} take their target values $W_{16}^{\star}$ and $W_{17}^{\star}$. The solution pins both words but does not collide: it leaves the characteristic at step 18, which holds with probability $2^{-18.42}$.

Table 8. A solution of joint pinning on T23. Words are in hexadecimal.

B_0	0035d957	e2e00000	243f6a88	85a308d3	13198a2e	03707344	a4093822	299f31d0
	082efa98	ec4e6c89	452821e6	38d01377	be5466cf	34e90c6c	c0ac29b7	c97c50dd
CV_1	8632517c	8e447e07	211a5786	76e738e7	73faee4c	5c24e09a	f5bb0791	26b69a60
(A_0, A_1, A_2)	aaa3d38c	9f756786	4ad6d4b7					
$v = h(A_2)$	939588e9							
B_1	7ecac643	062560f6	5ab97f1a	a39fc519	24d5a0af	3ddbf6a2	77e9423d	112b6cad
	ceb93a0c	c028b529	362124e8	07021c8f	e8c6e622	b06304c1	43660040	4891771f
(W_{16}, W_{17})	b421d321	31e9ba98						

Table 9 gives a final survivor of the tail measurement, with the head values $A_0 =$ `aaa3d38c`, $A_1 =$ `7e441ea6`, $A_2 =$ `4cb5d436`. Its W_{16} and W_{17} take the target values and its two messages collide under the 39-step compression function. The chaining value is recovered from the words of the survivor and is not the output of a first block, so this is a colliding second-block pair and not an instance of the two-block attack.

Table 9. A colliding second-block pair with pinned W_{16} and W_{17} on T23. Words are in hexadecimal.

CV_1	a26f6534	26635e31	64d22349	40d3a3ce	eb611783	a81c4f8c	4ec5a634	3f7e40ca
M	49ae11fd	c85009e9	f5f78230	1f1dcfdd	b55184f6	fe9a1056	54a8e340	4050f7f1
	cea7e20d	e159fd8b	34422569	07021c8f	e8c6e622	b06304c1	43660040	4891771f
M'	49ae11fd	c85009e9	f5f78230	1f1dcfdd	b55184f6	fe9a1056	54a8e340	4050f7f1
	8ea7e20d	e9d9ed8b	74422569	01003fbb	e8d6fe22	b06304c1	43660040	4891771f
output	12c9303c	dbb907eb	08037066	9a191396	a06888c0	04a9b58e	af4f2a8c	4aec8a4e

B Measurements

B.1 Measured Tail per Step

Table 10 gives the measured pass rate of every step of the tail (Sect. 4.4) next to its condition count. Steps that are not listed carry no condition. For 38 steps, step 23 is measured conditionally on the four W_7 conditions (Remark 1); the steps that read the head (24 and 26 for T23, 23 and 25 for 38 steps) use heads drawn from the distribution of the respective attack. The total is the product of the measured rates; the per-step values are rounded to two decimals.

Table 10. Measured tail per step: $-\log_2$ of the pass rate (Rate) and the condition count (Count).

	T23, joint (39 steps)		C38, single-word (38)		T5, single-word (38)	
Step	Rate	Count	Rate	Count	Rate	Count
17	pinned		15.00	15	16.00	16
18	18.42	19	13.00	14	13.00	14
19	18.01	18	15.00	15	15.00	15
20	15.00	15	6.00	6	6.00	6
21	6.00	6	7.00	7	7.00	7
22	7.00	7	1.00	1	1.00	1
23	1.00	1	11.00	11	11.00	11
24	11.01	11		—		—
25		—	4.01	4	3.98	4
26	3.97	4		—		—
Tail	80.41	81	72.02	73	72.98	74
95% interval	[80.27, 80.56]		[71.97, 72.06]		[72.94, 73.03]	

Splitting parameters. The survivors of the exact enumeration at step L_0 are the roots of the splitting tree: 12 288 for T23, 131 072 for C38 and 65 536 for T5. At each step j from $L_0 + 1$ to 23, every parent spawns 2^{s_j} children with fresh message words. If processing all parents at step j would exceed 2^{38} trials,

$m = 2^{38}/2^{s_j}$ parents are drawn uniformly without replacement by a partial shuffle; each of them produces 2^{s_j} children, and the pass rate of the step is the number of passing children divided by $m\,2^{s_j}$. This occurs only at step 19 of the 38-step runs, where 2 097 152 of 4 195 971 parents are kept for C38 and 2 097 152 of 2 097 373 for T5; the intervals of these two tails are approximate. We use $(s_{19}, \ldots, s_{23}) = (20, 22, 8, 7, 0)$ for T23 and $(s_{18}, \ldots, s_{23}) = (18, 17, 8, 7, 1, 6)$ for C38 and T5. The heaviest step takes $2^{37.6}$ trials for T23 and 2^{38} for T5, and 386, 4 043 and 4 152 final survivors remain for T23, C38 and T5. Step L_0 is exact, step $L_0 + 1$ uses a Kullback–Leibler bound, and later steps use cluster intervals with the root as the unit. The interval of the whole tail is a normal interval computed from the final survivors per root, with the set of roots fixed.

B.2 Unit Conversion

Table 11 gives the SHA-256 attacks in three units: single-thread scalar compressions (used throughout the paper), single-thread SHA-NI compressions, and SHA-NI compressions with 32 threads running in parallel. SHA-NI accelerates the compression but not the table look-up, which costs 3.5–4.0 scalar compressions and 14–18 SHA-NI compressions. The single-thread SHA-NI figures are therefore 1.5–2.2 bits higher. Each column divides the entry time by the compression time measured in the same setting, so a parallel figure can lie below the single-thread one: in the parallel runs, a compression takes longer relative to an entry. SHA-512 has no instruction-set support on our machine, and its results are given in scalar units only. All timings were taken on a shared machine (Intel Xeon Platinum 8582C, gcc `-O3 -march=native`). The cost per entry varies with the load: between runs of the same configuration it differed by 0.1 to 0.6 bits.

Table 11. SHA-256 attack complexities ($\log_2$) in three units.

			1 thread		32 threads
Steps	Char.	Pinning	Scalar	SHA-NI	SHA-NI
39	T23	joint, selected A_0	108.80	111.00	110.58
39	T23	joint, uniform A_0	110.18	112.12	111.70
39	T23	joint, Step-1 A_0	112.09	114.15	114.04
39	T23	single-word	123.37	125.11	124.65
38	C38	single-word	90.85	≈92.36	92.31
38	T5	single-word	91.73	93.75	93.19

C Characteristics and Colliding Pairs

C.1 Stages of the Characteristic Search

The search of Sect. 5.2 has seven stages. It keeps the five steps of [12], replaces the objective of the second, inserts a new stage 3, and adds the search for a conforming pair as stage 7:

1. minimize $\sum_i \mathrm{HW}(\Delta W_i)$;
2. minimize the condition count from step 16 on;
3. allowing two conditions of slack in this count, reduce the count from step 14 on, which governs $|(W_{14}, W_{15})|$ and the validity of the first block; we stopped this stage at the value 217 without a proof of optimality;
4. minimize the Hamming weight of $\Delta E_{14}, \Delta E_{15}$ (stages 4–6 as stages 3–5 of [12]);
5. minimize the Hamming weight of the controlled ΔA_i;
6. minimize the Hamming weight of the controlled ΔE_i, with value propagation on steps 10–15;
7. fix all signed differences and solve the full value model for a conforming pair.

Each stage only bounds the next, as in [12]. For C38, differences are allowed in $A_7, \ldots, A_{17}$ and $E_7, \ldots, E_{21}$; stages 2–5 use value propagation on steps 12–15 and stage 6 on steps 10–15.

C.2 The Characteristic C38

Table 12 gives the signed differences of C38 in the notation of Sect. 2.1, with - for bits without difference; the most significant bit is on the left. All words of steps -4 to 6 and 26 to 37 have no difference. Table 13 lists the bit conditions from step 16 on, by the largest step index among the inputs of the function, which the pair below satisfies; $X_i[j]$ is bit j of X_i in the first message, and bit 0 is the least significant. Each entry is a condition imposed by one output bit of a function, and conditions implied across different bits are not removed. Of the 89 entries, the 82 that involve a word of step 16 or later have rank 81; with the 17 difference bits of A, E and W from step 16 on (Table 12), they give $\gamma = 98$ and, with each entry assigned to the latest step among its own variables, the condition counts of Table 10. For example, step 17 counts 15: twelve entries of its row, the entries $E_{16}[3] = E_{17}[3]$ and $E_{16}[17] = E_{17}[17]$ of the row of step 18, and the difference bit of A_{17}; the entry $A_{15}[28] = A_{16}[28]$ of its row belongs to step 16. The other 7 entries are conditions on E_{14} and E_{15} alone.

Semi-free-start colliding pair for C38. The pair in Table 14 follows C38 and collides after 38 steps of the compression function, verified with a reference implementation; words are in hexadecimal.
The words of CV_1 are listed in the standard order of the SHA-256 initial value, i.e. $(A_{-1}, \ldots, A_{-4}, E_{-1}, \ldots, E_{-4})$. The message words differ in $M_7, \ldots, M_{11}$ and M_{15}, as in Table 12. Single-word pinning on C38 uses the value of W_{16} in this pair, $W_{16}^{\star} =$ `457f88fd`; on T5 it uses the value in the pair of [12], $W_{16}^{\star} =$ `15fab414`.

Table 12. The differential characteristic C38 for 38-step SHA-256.

i	ΔA_i	ΔE_i	ΔW_i
7	--un----------	nnnn----------	---u----------
8	----------n----u------n---	----uu-------nu-----un---n--n---	------n---u----------u----------
9	----------	------n-un-n----u------n-----nn-	---n----------
10	----------u-	----------n-------un-	-------nn-------n---n---nn--nn-u
11	-----u----------u---------n--	n--u-u-----n---------u-------u--	----------u------u-u---------
12	---n---un--u---n------uu-------n	n--n-------n--n--u---u-n-nu-u---	----------
13	---u----n----------	----u-nnn-n-------nnn-nn----u-uu	----------
14	-----u--nu-----n-n-------n--n---	-----n----u------u---u----------	----------
15	--------nn-------u----------	----------nu---u------	------u---u----------n----------
16	----------	--n-------u----n----n--u--------	---n----------
17	---n----------	----------	----------
18	----------	----------n----------n---	----------
19	----------	--------nn-------u----------	----------
20	----------	----------	----------
21	----------	---n----------	----------
22	----------	----------	----------
23	----------	----------	--------uu-------n----------
24	----------	----------	----------
25	----------	----------	---u----------

Table 13. Bit conditions of C38 from step 16 on.

Step	Conditions
16	$E_{16}[6] \neq E_{16}[25]$, $E_{16}[13] = E_{16}[27]$, $E_{16}[10] = E_{16}[15]$, $E_{16}[2] \neq E_{16}[7]$, $E_{16}[3] \neq E_{16}[22]$, $E_{16}[6] = 0$, $E_{14}[8] = 1$, $E_{15}[8] = 0$, $E_{14}[11] = 0$, $E_{16}[14] = 0$, $E_{14}[16] = E_{15}[16]$, $E_{15}[21] = 1$, $E_{16}[26] = 0$, $E_{14}[29] = 1$, $E_{15}[29] = 0$, $A_{15}[3] = A_{16}[3]$, $A_{15}[6] = A_{16}[6]$, $A_{15}[16] = A_{16}[16]$, $A_{15}[26] = A_{16}[26]$, $W_{16}[19] = W_{16}[26]$, $W_{16}[21] = W_{16}[30]$, $W_{16}[3] = W_{16}[5]$, $W_{16}[13] \neq W_{16}[17]$, $W_{16}[7] = W_{16}[24]$, $W_{16}[0] = W_{16}[11]$
17	$A_{17}[8] = A_{17}[19]$, $A_{17}[5] \neq A_{17}[17]$, $A_{17}[7] \neq A_{17}[16]$, $E_{17}[6] = 0$, $E_{17}[8] = 0$, $E_{17}[10] = 0$, $E_{17}[16] = 1$, $E_{17}[21] = 1$, $E_{17}[29] = 0$, $A_{16}[14] \neq A_{17}[14]$, $A_{16}[22] \neq A_{17}[22]$, $A_{16}[23] = A_{17}[23]$, $A_{15}[28] = A_{16}[28]$
18	$E_{18}[12] = E_{18}[31]$, $E_{18}[16] = E_{18}[21]$, $E_{18}[4] \neq E_{18}[22]$, $E_{18}[8] \neq E_{18}[22]$, $E_{16}[3] = E_{17}[3]$, $E_{18}[8] = 1$, $E_{18}[11] = 1$, $E_{18}[16] = 1$, $E_{16}[17] = E_{17}[17]$, $E_{18}[21] = 1$, $E_{18}[29] = 0$, $A_{16}[28] = A_{18}[28]$
19	$E_{19}[9] = E_{19}[28]$, $E_{19}[1] \neq E_{19}[19]$, $E_{19}[4] = E_{19}[17]$, $E_{19}[5] \neq E_{19}[18]$, $E_{19}[9] \neq E_{19}[27]$, $E_{19}[10] \neq E_{19}[28]$, $E_{19}[0] \neq E_{19}[27]$, $E_{19}[3] = E_{19}[8]$, $E_{19}[4] \neq E_{19}[9]$, $E_{19}[3] = 1$, $E_{17}[14] = E_{18}[14]$, $E_{19}[17] = 1$, $E_{17}[22] = E_{18}[22]$, $E_{17}[23] = E_{18}[23]$, $A_{18}[28] = A_{19}[28]$
20	$E_{20}[3] = 1$, $E_{20}[14] = 0$, $E_{20}[17] = 1$, $E_{20}[22] = 0$, $E_{20}[23] = 0$
21	$E_{21}[9] \neq E_{21}[14]$, $E_{21}[10] \neq E_{21}[23]$, $E_{21}[1] \neq E_{21}[15]$, $E_{21}[14] = 1$, $E_{21}[22] = 0$, $E_{21}[23] = 1$, $E_{19}[28] = E_{20}[28]$
22	$E_{22}[28] = 0$
23	$E_{23}[28] = 1$, $W_{23}[13] \neq W_{23}[20]$, $W_{23}[15] = W_{23}[24]$, $W_{23}[16] = W_{23}[25]$, $W_{23}[29] = W_{23}[31]$, $W_{23}[0] = W_{23}[30]$, $W_{23}[12] = 0$, $W_{23}[16] = 0$
25	$W_{25}[19] \neq W_{25}[26]$, $W_{25}[21] \neq W_{25}[30]$, $W_{25}[3] \neq W_{25}[5]$

Table 14. The SFS colliding message pair for 38 steps of SHA-256 (characteristic C38).

CV_1	511625c5	79e2f0ea	8f3b507a	cf567053	f8abb95a	51058334	e34aa619	a8b1abed
M	5539892d	5d347e17	57810b9b	42160779	61fbf450	3a0fc048	49dc03bc	b6937cfe
	11628e20	e3f27efc	b40b5503	53ee0ff3	a30c5263	97f5aa88	db9b7c1e	06a0d0ea
M'	5539892d	5d347e17	57810b9b	42160779	61fbf450	3a0fc048	49dc03bc	a6937cfe
	13428a20	f3f27efc	b58bddce	53ea05f3	a30c5263	97f5aa88	db9b7c1e	0480d4ea
hash	752dc7bb	33d721a6	62f61ea9	6db751d1	e6ffc333	629f3f38	bda53e54	800e9c2e

C.3 Pinning Parameters for T23

The 39-step attacks use the characteristic T23, its Step-1 solution and its semi-free-start pair from [12]. The pinning targets are the values in that pair, $W_{16}^{\star} =$ `b421d321` and $W_{17}^{\star} =$ `31e9ba98`. Joint pinning fixes $A_0 =$ `aaa3d38c` (Sect. 3.4) instead of the Step-1 value `ea3ff0b4`; single-word pinning keeps the Step-1 values of A_0 and A_2. The conditions labelled W_{23} and W_{25} in Table 22 of [12] are read as conditions on W_{24} and W_{26}, the words of the characteristic in Table 21 that carry the message differences.

C.4 Pinning Targets and Corrections for SHA-512

The SHA-512 attacks use the characteristics, Step-1 solutions and pairs of [12], with the pinning targets selected as in Sect. 5.3: $W_{16}^{\star} =$ `34b5e5a683c48aab` on 38 steps and $W_{16}^{\star} =$ `e90725beef8b6d13` on 39 steps. In the 39-step pair (Table 26 of [12]), the words M_{14} and M_{15} are printed in interchanged order in both messages. We use $M_{14} =$ `bf9503d51a5963de` and $M_{15} =$ `fba13633dc5703e6`; with these words the pair collides, gives the published hash value and agrees bit by bit with the characteristic in Table 24 of [12]. For both step counts, the signs of the characteristics are those of the published pairs with the message printed as M' taken as the first message. The step checks follow the condition tables of [12], writing $X[j]$ for bit j of X, with three corrections, none of which changes the signed differences of the characteristics. First, the conditions labelled W_{23} and W_{25} in Table 25 are read as conditions on W_{24} and W_{26}, the words of the 39-step characteristic that carry the message differences. Second, the two-bit conditions $E_{19}[63] = E_{20}[63]$ (38 steps), $W_{24}[6] \neq W_{24}[19]$ and $W_{24}[38] = W_{24}[60]$ (39 steps) are added; the tables omit them, and they hold in the published pairs. Third, the relations between $A_{18}[57]$ and $A_{17}[4]$ and between $E_{20}[18]$ and $E_{21}[59]$ listed in Table 25 are not enforced. Among the solutions that pass the respective step they hold with frequency 0.499 each (40 850 of 81 870 and 269 216 of 539 142), and the resulting pairs collide regardless; enforcing either of them would cost about one more bit. At the steps of the added conditions, the measured rates are $2^{-7.00}$ at step 20 of the 38-step tail, against six conditions counted in [12], and $2^{-13.06}$ at step 24 of the 39-step tail, against eleven.

Table 15 gives the measured tails per step for the selected targets.

Table 15. Measured SHA-512 tails per step ($-\log_2$), with the tail starting at step 17. Steps not listed hold with probability 1.

Steps	17	18	19	20	21	22	23	24	25	26	27	28	Total
38	19.00	19.01	16.00	7.00	6.00	2.00	13.00	—	3.00	—	1.00	—	86.01
39	33.61	19.02	20.00	15.00	8.01	6.00	2.00	13.06	—	2.99	—	1.13	120.82