Collision and Preimage Cryptanalysis of Blender

MiniMax

November 2026

Abstract

Blender is an addition-rotation-XOR cryptographic hash function family submitted by Bradbury to the first round of the NIST SHA-3 competition in 2008. Spanning the four standard digest sizes from 224 to 512 bits, the algorithm pairs a carry-coupled working register and persistent carry flags with a write-only modular-sum accumulator. To resist local differentials and state-collision attacks, the design incorporates 13-byte cyclic prefix padding, dual modular message checksums, and a minimum 32-round block-alignment rule before discarding the working register to output the accumulator. Prior cryptanalysis required at least $2^{107.3}$ operations for preimages, lacked a practical standard-IV collision generator, and left the odd-word variant Blender-224 unbroken below the 2^{112} birthday bound.

We present a unified collision and preimage cryptanalysis of the full Blender family by exploiting the decoupling between its carry-driven working register and its write-only modular accumulator. First, we establish a global shifted subspace collapse that locks periodic register differences whenever the per-round mask count divides the state length. This reduces the 258-bit working state of Blender-256 to a 38-bit invariant graph and yields standard-IV cycle-swap collisions in under one second alongside zero-cost multicollisions. Second, we construct a local two-step differential that hides a message perturbation inside a symmetric carry transition and heals it in the next round via bitwise-complement inputs. Combined with carry-free polynomial prefix inversion and byte-lattice contraction, this trail produces constant-time semi-free-start collisions across all four NIST variants, breaks Blender-224 in $2^{69.56}$ evaluations, and yields minimal 24-byte standard-IV and 36-byte identical-prefix collisions on Blender-256. Third, we combine multi-shifted closed-walk steering with 2-adic lattice invariants and generalized birthday trees over the modular accumulator ring. This reduces full first- and second-preimage complexities on Blender-256, Blender-384, and Blender-512 to $2^{35.93}$, $2^{66.34}$, and $2^{67.64}$ evaluations, showing that write-only modular accumulators cannot compensate for low-rank mask broadcasting.

1 Introduction

Submitted by Colin Bradbury in October 2008 and selected by NIST among the 51 first-round SHA-3 candidates [1], Blender was designed to achieve high software throughput across 16-, 32-, and 64-bit processors without lookup tables or S-boxes. A Blender instance is parameterized by a word width $w \in \{16, 32, 64\}$ in bits and a working-state length $m \in \{6, 7, 8, 9, 10\}$ in words. These parameters define a digest length $n = mw$ in bits and a per-round mask count $t \in \{2, 3\}$. The four official NIST SHA-3 variants are Blender-224 ($w = 32, m = 7$), Blender-256 ($w = 32, m = 8$), Blender-384 ($w = 64, m = 6$), and Blender-512 ($w = 64, m = 8$). All four official variants operate with $t = 2$ masks per round.

Whereas earlier stream-oriented hashes such as PANAMA [3] continually feed an auxiliary buffer back into the active state during absorption, Blender—like fellow first-round candidate Waterfall [5]—separates its internal state into an active m-word working register (augmented with t persistent 1-bit carry flags) and a strictly write-only m-word accumulator. However, Blender's concrete compression and finalization primitives depart substantially from LFSR and S-box constructions:

1. **Carry-Coupled ARX Compression:** At each message word, Blender evaluates t modular integer additions that incorporate the persistent carry flags from the preceding step. The newly generated carry bits select a data-dependent rotation count, and the rotated sums are XORed alternately into the cyclically shifted working register.

2. **Modular-Sum Accumulation with Working-State Truncation:** After every compression step, each updated working word is added modulo 2^w into the corresponding word of the write-only accumulator. The accumulator never feeds back into the working register or carry flags. At the end of finalization, the working register and carry flags are discarded, and the accumulator words are serialized to form the n-bit digest.

3. **Cyclic Prefix Padding, Block Alignment, and Dual Checksums:** To counter local differentials and length-extension attacks, Blender buffers the first 13 bytes of the message and replays them cyclically after the message body so that the message, padding, bit-length encoding, and two running modulo-2^w message checksums (accumulating the absorbed words and their bitwise complements) execute at least 32 compression rounds and end on a 16-word block boundary before extracting the digest.

1.1 Prior Cryptanalysis

After NIST accepted Blender among the 51 first-round SHA-3 candidates in December 2008 [15], six public cryptanalytic works examined the design in 2008–2009:

- **Newbold (2008 [13]):** Newbold chained $\frac{n+w}{2}+1$ Joux working-state collisions [6] with a birthday search over the accumulator and primary checksum ($n+w$ bits). This yielded full-state collisions in $2^{(n+w)/2}$ compression rounds per stage ($2^{128}, 2^{144}, 2^{224}$, and 2^{288} across Blender-224/256/384/512—above the $2^{n/2}$ birthday bound for a single collision, but yielding 2^k-multicollisions in $k \cdot 2^{(n+w)/2}$ rounds with $O(n)$-word segments). Applying the Gauravaram–Kelsey additive-checksum framework [4] yielded preimages in $n \cdot 2^{(n+w)/2}$ rounds ($2^{135.8}, 2^{152.0}, 2^{232.6}$, and $2^{297.0}$, with $O(n^2)$-word messages) using cycle-finding.

- **Mendel (2008 [11]) and Lamberger–Mendel (ISC 2009 [10]):** These works treated the compression round as a black box. By varying four message words ($4w \geq n/2$ bits) and adjusting a fifth word modulo 2^w to hold the five-word modular sum constant, each five-word Joux stage collides the working state and both checksums in $2^{n/2}$ evaluations (or $2^{n/2+1}$ when accounting for the two carry flags omitted in [10]). Repeating n stages ($5n+1$ words including a padding-adjustment word) builds a 2^n-multicollision in published time $n \cdot 2^{n/2}$ ($2^{119.8}, 2^{136.0}, 2^{200.6}$, and $2^{265.0}$; twice as high with carry matching). A Quisquater–Delescaille memoryless meet-in-the-middle search [14] on the accumulator then recovers preimages across all four variants with $O(n)$ memory.

- **Xu and Li (IACR ePrint 2008/532 [19]):** Xu and Li exhibited a 24-byte (6-word) colliding message pair ($0^4 \| 0 \| \texttt{ffffffff}$ versus $0^4 \| \texttt{ffffffff} \| 0$) exclusively for the single all-zero initial working state and zero carries. In that state, all modular additions degenerate into carry-free XORs. Because reaching the all-zero state from the standard initialization vector (IV) costs 2^{256} operations, no semi-free-start (SFS) generator for non-zero states or standard-IV short collision was obtained.

- **Klíma (2008 [8] and IACR ePrint 2009/006 [9]):** Klíma first reported a 927-bit versus 926-bit near-collision matching 225 of 256 digest bits on Blender-256 [8]. In [9], Klíma zeroed one mask per round in alternating order so that the $m/2$ odd-indexed working lanes repeat every $n = mw$ rounds. This left $n/2+2$ unconstrained bits and located 10 working-state fixpoints in $10 \cdot n \cdot 2^{n/4+1}$ compression rounds ($2^{76.3}, 2^{108.9}$, and $2^{141.3}$ on Blender-256/384/512). Auditing [9] against the Blender specification uncovers three fundamental

Table 1: Cryptanalytic comparison across the four official NIST Blender variants ($w = 32$ for 224/256, $w = 64$ for 384/512; w = words, B = bytes, b = bits; Newbold's preimage bounds appear in Table 2). Measured runtimes are verified against the official NIST C reference [1].

Goal	Work	Setting / Mechanism	Var.	Length	Complexity	Runtime
Collision	Newbold [13]	Full-state multicoll. stage	224	$O(n)$ w	$2^{128.0}$	—
	Newbold [13]	Full-state multicoll. stage	256	$O(n)$ w	$2^{144.0}$	—
	Newbold [13]	Full-state multicoll. stage	384	$O(n)$ w	$2^{224.0}$	—
	Newbold [13]	Full-state multicoll. stage	512	$O(n)$ w	$2^{288.0}$	—
	Klíma [9]	Odd-lane fixpoint winding	256	$2^{178.0}$ w	$\geq 2^{178.0}$	—
	Klíma [9]	Odd-lane fixpoint winding	384	$2^{275.0}$ w	$\geq 2^{275.0}$	—
	Klíma [9]	Odd-lane fixpoint winding	512	$2^{340.0}$ w	$\geq 2^{340.0}$	—
	Ours (§4)	Silent carry healing	**224**	**6** w	$\mathbf{2^{69.56}}$	—
	Ours (§3)	Clean-subspace cycle swap	**256**	4,000 w	$2^{22.60}$	**0.73** s
	Ours (§4)	Global-min carry healing	**256**	**6** w	—	**34.6** s
	Ours (§3)	Clean-subspace cycle swap	**384**	4,096 w	$\mathbf{2^{57.60}}$[a]	—
	Ours (§3)	Clean-subspace cycle swap	**512**	4,096 w	$\mathbf{2^{58.00}}$[a]	—
SFS / Short	Klíma [8]	225/256-bit near-collision	256	927 b	—	—
	Xu–Li [19]	Zero-state SFS only	256	24 B	$O(1)$	—
	Ours (§4)	Closed-form arbitrary SFS	**224**	113 B	$O(1)$	**0.027** μs
	Ours (§4)	Closed-form arbitrary SFS	**256**	113 B	$O(1)$	**0.006** μs
	Ours (§4)	Closed-form arbitrary SFS	**384**	229 B	$O(1)$	**0.054** μs
	Ours (§4)	Closed-form arbitrary SFS	**512**	229 B	$O(1)$	**0.014** μs
	Ours (§4)	Identical-prefix (3R steer)	**256**	**36** B	—	—[b]
	Ours (§4)	Identical-prefix (4R lattice)	**256**	40 B	—	**145.5** s
Preimage	Lamberger–Mendel [10]	5-word sum-fixed Joux	224	1,121 w	$2^{119.81}$	—
	Lamberger–Mendel [10]	5-word sum-fixed Joux	256	1,281 w	$2^{136.00}$	—
	Klíma [9]	Multipreimage winding	256	$2^{107.3}$ w	$\geq 2^{107.32}$	—
	Ours (§5)	Multi-shifted Wagner k-tree	**256**	$2^{27.46}$ w	$\mathbf{2^{35.93}}$	—
	Lamberger–Mendel [10]	5-word sum-fixed Joux	384	1,921 w	$2^{200.58}$	—
	Klíma [9]	Multipreimage winding	384	$2^{171.9}$ w	$\geq 2^{171.90}$	—
	Ours (§5)	Multi-shifted Wagner k-tree	**384**	$2^{38.91}$ w	$\mathbf{2^{66.34}}$	—
	Lamberger–Mendel [10]	5-word sum-fixed Joux	512	2,561 w	$2^{265.00}$	—
	Klíma [9]	Multipreimage winding	512	$2^{204.3}$ w	$\geq 2^{204.32}$	—
	Ours (§5)	Multi-shifted Wagner k-tree	**512**	$2^{39.25}$ w	$\mathbf{2^{67.64}}$	—

[a] At 4,096-word cap ($\approx 2^{34.8}$ rounds unconstrained). [b] Explicit 36-byte witness verified in Table 3.

barriers. First, although the abstract of [9] claims applicability to $n = 224$, for odd $m = 7$ the cyclic register wrap-around mixes odd and even lanes and breaks periodicity completely. Second, the two-fixpoint collision construction in [9, Sec. 4] winds its first cycle 2^w times the second cycle length, producing colliding messages of roughly 2^{178} words on Blender-256 that require at least 2^{178} rounds to hash (slower than generic 2^{128} birthday search). Third, the multipreimage attack in [9, Sec. 5] prescribes messages of $2^{171.3}, 2^{267.3}$, and $2^{331.3}$ words on Blender-256/384/512. Even a minimal single-winding variant emits at least $2^{107.3}, 2^{171.9}$, and $2^{204.3}$ words, imposing an unavoidable message-hashing floor of at least $2^{107.3}$ rounds on Blender-256.

Prior to our work, four major cryptanalytic gaps remained open across the Blender family. First, once message-hashing costs are accounted for, no valid standard-IV collision construction faster than the $2^{n/2}$ generic birthday bound—let alone a practical sub-second collision generator—was known. Second, no collision attack faster than the 2^{112} birthday bound was known for odd-word Blender-224 ($m = 7$). Third, no short (< 100-byte) standard-IV collision, identical-prefix collision for arbitrary prefixes, or arbitrary-state SFS generator was known for any variant. Fourth, no preimage attack broke the 2^{107} barrier on Blender-256 or the 2^{171} barrier on Blender-384 and Blender-512.

1.2 Our Contributions and Technical Overview

Table 1 summarizes our results across the Blender family. Our analysis develops three core techniques:

1. **Shifted Clean Subspace Collapse, Parity Obstruction, and Sub-Second Cycle-Swap Collisions (Section 3):** Across all 15 variants in Bradbury's specification, we prove a general divisibility criterion for subspace closure. Constraining the leading updated working word to a constant word or its bitwise complement makes all $m-t$ stride-t register XOR differences message-independent if and only if t divides m. This criterion holds for 9 of the 15 variants (including Blender-256, Blender-384, and Blender-512) and explains why odd-lane periodicity fails on 7-word Blender-224. On Blender-256, our construction collapses Klíma's 130-bit alternating state [9] to a 38-bit period-8 clean subspace (2^{92} times smaller). We also establish a Parity Obstruction Theorem proving that the unshifted zero-difference fixed point is unreachable from the standard Blender-256 IV for any message. In addition, we prove that the carry-coupled branching graph on the clean subspace has an exact mean out-degree of 2.000. Pairing two overlapping ancestor cycles on a single depth-first search (DFS) tree yields standard-IV Blender-256 collisions in 0.045–0.73 s, zero-cost $\binom{2r}{r}$ multicollisions, and $\Theta(2^w/\ell)$ collisions on Blender-384 and Blender-512.

2. **Two-Step Silent Carry Healing, All-Variant $O(1)$ SFS, and Solver-Free 24-Byte Collisions (Section 4):** Bypassing the $t \mid m$ divisibility barrier via a two-step local perturbation-and-correction trail [2], we introduce a carry-coupled ARX differential. The first step injects a silent carry perturbation that leaves the working register and accumulator unchanged while flipping both carry flags from $(0,0)$ to $(1,1)$. The second step applies bitwise-complement message words that lock the adder inputs to mid-point constants, simultaneously healing the carry difference and preserving both modular message checksums. We establish an unordered state validity density of $2^{-69.56}$ across Blender-224 and $2^{-69.58}$ across Blender-256, breaking Blender-224 in $2^{69.56} \ll 2^{112}$ state evaluations. We also derive deterministic zero-retry $O(1)$ closed-form SFS collisions for all four NIST variants via circulant polynomial inversion, and prove that 24 bytes (6 words) is the global minimum standard-IV collision length on Blender-256. Finally, we build a solver-free 4-round lane-invariant and 2-round byte-lattice contraction solver (0.11–0.29 ms per target) that finds 24-byte standard-IV and 40-byte identical-prefix collisions in pure C.

3. **Multi-Shifted Closed-Walk Steering and Preimage Cryptanalysis (Section 5):** Building on multi-block internal-state collisions [6] and generalized birthday checksum cancellation [17, 4, 12], we develop a multi-shifted closed-walk preimage attack over the modular ring of Blender's accumulator. Because Blender discards its working state immediately after the checksum rounds, equal-length, equal-checksum closed walks require no finalization inversion. Using 2-adic Smith Normal Form (SNF) analysis, we show that unshifted clean walks suffer a 12-bit subgroup rank defect that starves the leading accumulator lane. We prove that chaining closed walks across multiple shifted clean subspaces injects odd 2-adic units into the leading lane and saturates the maximal parity subgroup of index 2^{m-2}. Combining 8-anchor reverse-reachable DAG [7] slot generation with a joint checksum-and-accumulator generalized birthday tree [17] reduces full first- and second-preimage complexities to $2^{35.93}$ on Blender-256, $2^{66.34}$ on Blender-384, and $2^{67.64}$ on Blender-512. We also verify the exact 2-adic SNF subgroup ranks and the closed-walk modular superposition identity empirically on Bradbury's unmodified NIST C reference implementation.

1.3 Paper Organization

Section 2 establishes our algebraic notation, circulant polynomial and 2-adic lattice preliminaries, and the complete mathematical specification of the Blender family. Section 3 develops

our shifted clean subspace collapse, the Parity Obstruction Theorem, and the single-stage overlapping cycle-swap collision and multicollision attacks. Section 4 presents our two-step silent carry injection and complement healing trail, closed-form SFS generators for all four NIST variants, and solver-free 24-byte standard-IV and identical-prefix collision attacks. Section 5 introduces multi-shifted modular-sum steering, proves the 2-adic SNF subgroup hierarchy, and derives our full first- and second-preimage attacks. Section 6 summarizes the root architectural vulnerabilities and reports experimental verification benchmarks and concrete witnesses.

2 Preliminaries and Specification

2.1 Notation and Algebraic Preliminaries

Throughout this paper, $w \in \{16, 32, 64\}$ denotes the word bit-width and $b = w/8 \in \{2, 4, 8\}$ denotes the byte length of a word. We write $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$ for the binary field and $\mathbb{Z}/2^w\mathbb{Z}$ for the residue ring of integers modulo 2^w.

Bitwise and Modular Word Operations. We identify the set of w-bit words $\{0,1\}^w$ with the integer interval $\{0, 1, \ldots, 2^w - 1\}$ via the little-endian binary expansion $z = \sum_{j=0}^{w-1} z_{[j]}\, 2^j$, where $z_{[j]} \in \mathbb{F}_2$ denotes bit j of z. For any w-bit word z, its bit-parity is $\mathrm{par}(z) \coloneqq \bigoplus_{j=0}^{w-1} z_{[j]} \in \mathbb{F}_2$, and its j-th little-endian byte ($0 \le j < b$) is $\mathrm{byte}_j(z) \coloneqq \lfloor z/2^{8j} \rfloor \bmod 2^8$. We write $\mathbf{0}$ for the all-zero word (or all-zero tuple), $\mathbf{1} \coloneqq 2^w - 1$ for the all-ones word, $x \oplus y$ for bitwise XOR, and $\neg x \coloneqq x \oplus \mathbf{1} = (2^w - 1) - x$ for bitwise complement. For any integer k, $\mathrm{ror}_k(x)$ and $\mathrm{rol}_k(x)$ denote w-bit right and left cyclic rotations by $k \bmod w$ bit positions. We write $x \boxplus y = (x + y) \bmod 2^w$ and $x \boxminus y = (x - y) \bmod 2^w$ for addition and subtraction in $\mathbb{Z}/2^w\mathbb{Z}$, extended componentwise to the m-word abelian group $\mathcal{G} \coloneqq (\mathbb{Z}/2^w\mathbb{Z})^m$. For a byte or word sequence U, we denote its length by $|U|$ (or $\ell(U)$), write $U \| V$ for concatenation, and write $\mathbf{1}[\mathcal{E}] \in \{0, 1\}$ for the indicator of event $\mathcal{E}$.

Circulant Polynomial Representation. Each w-bit word $z \in \mathbb{F}_2^w$ with bits $(z_{[0]}, \ldots, z_{[w-1]})$ corresponds bijectively to the polynomial $z(X) = \sum_{j=0}^{w-1} z_{[j]}\, X^j$ in the quotient ring $\mathcal{R}_w \coloneqq \mathbb{F}_2[X]/(X^w + 1)$. Since $X^w = 1$ in $\mathcal{R}_w$, cyclic left rotation $\mathrm{rol}_k(z)$ acts as multiplication by $X^{k \bmod w}$, and cyclic right rotation $\mathrm{ror}_k(z)$ acts as multiplication by $X^{(w-k) \bmod w}$. Any $\mathbb{F}_2$-linear combination of cyclic rotations $L(z) = \bigoplus_{k \in S} \mathrm{rol}_k(z)$ corresponds to multiplication by the circulant polynomial $f(X) = \sum_{k \in S} X^k \in \mathcal{R}_w$. Because $w = 2^e$ is a power of two, $(X^w + 1) = (X + 1)^{2^e}$ over $\mathbb{F}_2$, so $f(X)$ is invertible in $\mathcal{R}_w$ if and only if $f(1) \equiv |S| \equiv 1 \pmod 2$ (odd Hamming weight). Whenever $f(1) = 1$, we have $f(X)^{2^e} = f(X^{2^e}) = f(1) = 1$ in $\mathcal{R}_w$, giving the closed-form inverse $f(X)^{-1} = \prod_{j=0}^{e-1} f(X)^{2^j}$ in $\mathcal{R}_w$ via $e = \log_2 w$ squarings.

Two-Adic Valuation and Smith Normal Form. For any integer $z \in \mathbb{Z}$, the 2-adic valuation $\nu_2(z)$ is the largest exponent $v \ge 0$ such that 2^v divides z, with $\nu_2(0) \coloneqq w$ on $\mathbb{Z}/2^w\mathbb{Z}$. An element $u \in \mathbb{Z}/2^w\mathbb{Z}$ is a 2-adic unit (invertible modulo 2^w) if and only if $\nu_2(u) = 0$, that is, u is odd. Over the local principal ideal ring $\mathbb{Z}/2^w\mathbb{Z}$, any subgroup ($\mathbb{Z}/2^w\mathbb{Z}$-submodule) $\mathcal{H} \le \mathcal{G} = (\mathbb{Z}/2^w\mathbb{Z})^m$ generated by a family of difference vectors admits a 2-adic Smith Normal Form (SNF) decomposition via unimodular row and column operations. Its invariant factors take the form $(d_0, d_1, \ldots, d_{m-1}) = (2^{v_0}, 2^{v_1}, \ldots, 2^{v_{m-1}})$ with non-decreasing 2-adic valuations $0 \le v_0 \le v_1 \le \cdots \le v_{m-1} \le w$. Consequently, $\mathcal{H} \cong \bigoplus_{i=0}^{m-1} 2^{v_i}(\mathbb{Z}/2^w\mathbb{Z})$ has finite index $[\mathcal{G} : \mathcal{H}] = \prod_{i=0}^{m-1} d_i = 2^{\sum_{i=0}^{m-1} v_i}$ and bit codimension $\sum_{i=0}^{m-1} v_i$ inside $(\mathbb{Z}/2^w\mathbb{Z})^m$.

2.2 State Layout and Initialization

We follow the official NIST SHA-3 Round-1 specification and C/C++ reference submission [1]. Blender defines 15 hash instances indexed by word width $w \in \{16, 32, 64\}$ and state word count $m \in \{6, 7, 8, 9, 10\}$, producing an $n = mw$-bit digest. The per-round mask count is $t = 2$ for $m \in \{6, 7, 8\}$ (the $x2$ *family*, which encompasses all four required NIST SHA-3 output sizes) and $t = 3$ for $m \in \{9, 10\}$ (the $x3$ *family*):

$$\text{Blender-224} : (w{=}32, m{=}7, t{=}2), \quad \text{Blender-256} : (w{=}32, m{=}8, t{=}2),$$

$$\text{Blender-384} : (w{=}64, m{=}6, t{=}2), \quad \text{Blender-512} : (w{=}64, m{=}8, t{=}2).$$

Each Blender instance maintains six state components. The active trunk consists of working words $W = (W_0, \ldots, W_{m-1}) \in (\{0,1\}^w)^m$ and persistent carry flags $c_1, c_2 \in \{0, 1\}$ (plus $c_3 \in \{0, 1\}$ when $t = 3$). The output and schedule state consists of modular-sum accumulators $\mathsf{Sum} = (\mathsf{Sum}[0], \ldots, \mathsf{Sum}[m-1]) \in (\mathbb{Z}/2^w\mathbb{Z})^m$, running word checksums $C_1, C_2 \in \mathbb{Z}/2^w\mathbb{Z}$, a 13-byte prefix buffer $F[0 \ldots 12]$, and a fragment-alignment counter enforcing the 32-round minimum and 16-fragment block alignment.

At initialization, $\mathsf{Sum} = (0, \ldots, 0)$, $c_1 = c_2 = 0$, and $C_1 = C_2 = 0$. The initial working words W^{IV} are derived from fractional square roots of primes (matching the standard initial words of SHA-224, SHA-256, and SHA-512 for Blender-224, Blender-256, and Blender-512). In Bradbury's official NIST C/C++ reference implementation [1], the Blender-384 ($m = 6$) initialization loop loads five 64-bit words and leaves $W_5^{\mathrm{IV}} = 0$ rather than loading a sixth table entry (whereas Blender-224 copies eight entries from an 18-word table whose first seven entries match SHA-224, leaving the eighth slot unused). All structural theorems, closed-form SFS generators, out-degree laws, and preimage bounds in this paper hold identically under both the written specification IVs and the official C reference IVs. All concrete digests and test witnesses are verified directly against the unmodified NIST C reference.

2.3 Compression Round and Carry Coupling

Message bytes are absorbed in little-endian w-bit fragments $f_k = \sum_{j=0}^{b-1} \mathrm{byte}_{bk+j}\, 2^{8j} \in \{0,1\}^w$. For each absorbed fragment f_k during message, prefix-pad, and length absorption (prior to the checksum words), the running checksums update via

$$C_1 \leftarrow (C_1 + f_k) \bmod 2^w, \qquad C_2 \leftarrow (C_2 + \neg f_k) \bmod 2^w. \tag{1}$$

After L absorbed fragments, the two checksums satisfy $C_1 + C_2 \equiv L\,(2^w - 1) \equiv -L \pmod{2^w}$. For any two messages of equal fragment count L, we therefore have $C_1(M) = C_1(M')$ if and only if $C_2(M) = C_2(M')$.

In the $x2$ family ($m \in \{6, 7, 8\}$, $t = 2$), one compression round $\mathsf{Step}(W, c_1, c_2, f_k)$ computes two $(w+1)$-bit integer sums over $\mathbb{Z}$:

$$s_2 = \big(\mathrm{ror}_8(f_k) \oplus W_0\big) + \big(\mathrm{ror}_8(W_2) \oplus W_4\big) + c_2, \quad \hat{T}_2 = s_2 \bmod 2^w, \quad c_2' = \lfloor s_2/2^w \rfloor \in \{0,1\}, \tag{2}$$

$$s_1 = \big(f_k \oplus W_5\big) + \big(\mathrm{rol}_8(W_3) \oplus W_1\big) + c_1, \qquad \hat{T}_1 = s_1 \bmod 2^w, \quad c_1' = \lfloor s_1/2^w \rfloor \in \{0,1\}. \tag{3}$$

The carry pair (c_1', c_2') produced by (2)–(3) selects a joint rotation amount $r(c_1', c_2')$ and forms the rotated masks (T_1, T_2) according to

$$r(c_1', c_2') = 8 - (c_1' + c_2') = \begin{cases} 8, & \text{if } (c_1', c_2') = (0,0), \\ 7, & \text{if } c_1' \oplus c_2' = 1, \\ 6, & \text{if } (c_1', c_2') = (1,1), \end{cases} \qquad T_1 = \mathrm{rol}_r(\hat{T}_1), \quad T_2 = \mathrm{ror}_r(\hat{T}_2).$$

Denoting $(M_0, M_1) = (T_2, T_1)$ (and generally $M_0, \ldots, M_{t-1}$ in the xt family), the working words shift cyclically by one position with alternating mask injection and a ror_7 wrap-around on W_0:

$$W'_i = W_{i+1} \oplus M_{i \bmod t} \quad (0 \le i \le m-2), \qquad W'_{m-1} = \mathrm{ror}_7(W_0) \oplus M_{(m-1) \bmod t}. \tag{4}$$

Immediately after updating $W \leftarrow W'$ and $(c_1, c_2) \leftarrow (c'_1, c'_2)$, each accumulator word updates modulo 2^w:

$$\mathsf{Sum}[i] \leftarrow (\mathsf{Sum}[i] + W'_i) \bmod 2^w \qquad (0 \le i < m).$$

2.4 Prefix Repetition, Checksums, and Finalization

Given an ℓ-byte message M with 16-fragment block byte-size $B_{\mathrm{blk}} = 16b$ (64 B for $w = 32$; 128 B for $w = 64$) and an ℓ_L-byte little-endian bit-length encoding $L(8\ell)$ ($\ell_L \ge 1$ minimal byte width of 8ℓ), finalization proceeds in three deterministic stages before output extraction (Figure 1):

1. **Cyclic 13-Byte Prefix Padding and Block Alignment:** Let $F[0 \ldots f_{\mathrm{len}} - 1]$ store the first $f_{\mathrm{len}} = \min(\ell, 13)$ bytes of M (or 13 zero bytes if $\ell = 0$). Blender appends $k \ge 1$ cyclic prefix bytes $F[i \bmod f_{\mathrm{len}}]$ $(0 \le i < k)$, where k is the smallest positive integer such that $\ell + k + (\ell_L + 1) + 2b$ is at least $2B_{\mathrm{blk}}$ (32 compression rounds) and is an exact multiple of B_{blk} (16 fragments). In particular, $k = (2B_{\mathrm{blk}} - 2b - 1) - \ell - \ell_L$ when $\ell + \ell_L + 2b + 1 < 2B_{\mathrm{blk}}$ (replaying all of F), and $k = B_{\mathrm{blk}} - ((\ell + \ell_L + 2b + 1) \bmod B_{\mathrm{blk}}) \in \{1, \ldots, B_{\mathrm{blk}}\}$ otherwise (replaying only the first $\min(k, f_{\mathrm{len}})$ bytes when $k < 13$).

2. **Bit-Length Encoding:** Immediately after the k cyclic prefix-pad bytes, Blender appends the ℓ_L-byte little-endian bit-length 8ℓ followed by the 1-byte descriptor ℓ_L. This aligns the byte stream to an exact b-byte word boundary and updates C_1, C_2, W, and Sum on every absorbed fragment.

3. **Dual Checksum Absorption and Digest Extraction:** Two final compression rounds are executed with message-word inputs $f = \neg C_1$ and $f = C_2$ (accumulating the updated W' into Sum). Immediately thereafter, (W, c_1, c_2) is discarded and $\mathsf{Sum}[0 \ldots m-1]$ is serialized in big-endian byte order.

3 Subspace Collapse and Closed-Walk Collisions

Our first collision attack targets the macro-structure of the working-state transition graph. Suppose two distinct messages $M_1 \ne M_2$ of equal byte length $\ell \ge 13$ share the same 13-byte prefix. If both messages arrive at the end of message absorption with identical working states (W, c_1, c_2), identical accumulators Sum, and identical checksums (C_1, C_2), then all three finalization stages receive identical inputs and states. This guarantees an exact full-digest collision.

3.1 Shifted Clean Subspace Collapse

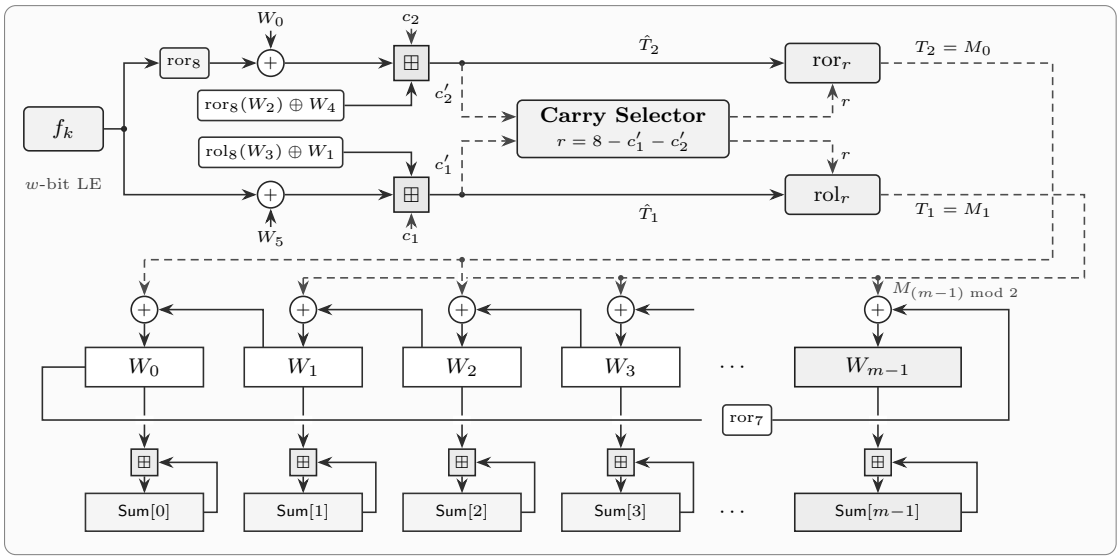
On Blender-256, a full working state (W, c_1, c_2) has $8 \times 32 + 2 = 258$ bits, placing naive cycle search at 2^{129} steps. Consider instead the $m - t$ *stride-t XOR difference variables*:

$$X_i \coloneqq W_i \oplus W_{i+t} \in \{0,1\}^w, \qquad 0 \le i \le m - t - 1.$$

Because indices i and $i + t$ receive the exact same mask $M_{i \bmod t}$ in (4), the non-linear ARX masks cancel identically across the first $m - t - 1$ difference coordinates for every fragment f_k:

$$X'_i = (W_{i+1} \oplus M_{i \bmod t}) \oplus (W_{i+t+1} \oplus M_{i \bmod t}) = X_{i+1} \qquad (0 \le i \le m - t - 2). \tag{5}$$

(a) Single-Step $x2$-Family ARX Compression Circuit $\mathsf{Step}(W, c_1, c_2, f_k)$ and Write-Only Accumulator



(b) Post-Message Block-Aligned Prefix Padding, Dual Checksums, and Working-State Discard

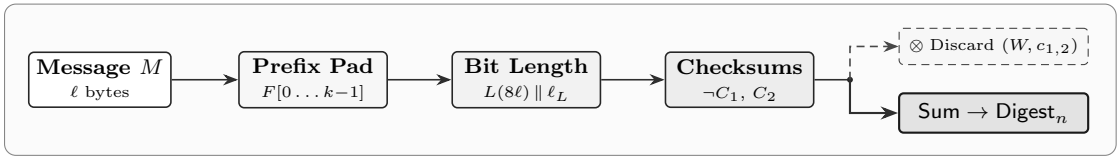


Figure 1: Cryptographic circuit of the $x2$-family Blender hash function ($m \in \{6, 7, 8\}$, $t = 2$, where $M_{(m-1) \bmod 2}$ is T_1 for even m and T_2 for odd m). **(a)** Each step updates (W, c_1, c_2) via two carry-coupled modular additions and accumulates W_i' into $\mathsf{Sum}[i]$ $(\bmod 2^w)$. **(b)** Finalization pads with k cyclic 13-byte prefix bytes (enforcing ≥ 32 total rounds and 16-fragment block alignment), appends $L(8\ell)\|\ell_L$, and absorbs $\neg C_1$ and C_2 before discarding (W, c_1, c_2) and outputting Sum in big-endian order.

Meanwhile, the boundary coordinate $i = m - t - 1$ satisfies

$$X'_{m-t-1} = W_{m-t} \oplus \mathrm{ror}_7(W_0) \oplus M_{(m-t-1) \bmod t} \oplus M_{(m-1) \bmod t} = W_{m-t} \oplus \mathrm{ror}_7(W_0). \tag{6}$$

Now write $m = qt + \rho$ with $0 \leq \rho < t$. Iterating $W_j = W_{j+t} \oplus X_j$ from $j = 0$ in steps of t yields $W_{qt} = W_0 \oplus \bigoplus_{j=0}^{q-1} X_{jt}$. Comparing qt with $m - t = (q-1)t + \rho$ gives our first general structural theorem.

Theorem 1 (Shifted Clean Subspace Divisibility Criterion)**.** *Fix any constant word $x \in \{0,1\}^w$, and say that a step is x-*clean *if the absorbed fragment f_k steers $W_0' \in \{x, \neg x\}$. Once $W_0 \in \{x, \neg x\}$, the self-XOR quantity*

$$\kappa(x) \coloneqq W_0 \oplus \mathrm{ror}_7(W_0) = x \oplus \mathrm{ror}_7(x) \tag{7}$$

is invariant across both choices $W_0 = x$ and $W_0 = \neg x$ because $\neg x \oplus \mathrm{ror}_7(\neg x) = (x \oplus \mathbf{1}) \oplus \mathrm{ror}_7(x) \oplus \mathbf{1} = x \oplus \mathrm{ror}_7(x)$. Under consecutive x-clean transitions, two regimes arise:

1. ***Exact Collapse when** $t \mid m$ **($\rho = 0$):** Here $m - t = (q-1)t$, so $W_{m-t} = W_0 \oplus \bigoplus_{j=0}^{q-2} X_{jt}$. Equation (6) then closes into the purely affine, message-independent recurrence*

$$X'_i = X_{i+1} \ \ (0 \leq i \leq m - t - 2), \qquad X'_{m-t-1} = \kappa(x) \oplus \bigoplus_{j=0}^{q-2} X_{jt}. \tag{8}$$

Over $\mathbb{F}_2[Z]$, the linear companion operator of (8) has characteristic polynomial $(Z^m + 1)/(Z^t + 1)$, which divides $Z^m + 1$. Thus, the trajectory of $X^{(r)}$ is strictly periodic with period dividing m ($P = 8$ for $m = 8, t = 2$; $P = 6$ for $m = 6, t = 2$; $P = 9$ for $m = 9, t = 3$). At each phase $\phi \in \mathbb{Z}_P$, this collapses the $(mw+t)$-bit working state to an invariant manifold $\mathcal{C}_x^{(\phi)}$ of dimension

$$\dim_{\text{bits}}\big(\mathcal{C}_x^{(\phi)}\big) = 1 + (t-1)w + t \quad \textit{bits},$$

parameterized solely by the complement bit $e \in \{0,1\}$ ($W_0 = x \oplus e\mathbf{1}$), the unconstrained base lanes $(W_1, \ldots, W_{t-1}) \in (\{0,1\}^w)^{t-1}$, and the t carry bits.

2. ***Obstruction when $t \nmid m$ ($\rho \neq 0$):*** *Here $m - t \equiv \rho \not\equiv 0 \pmod t$, so $W_{m-t} = W_\rho \oplus \bigoplus_{j=0}^{q-2} X_{\rho+jt}$. This injects the unconstrained lane W_ρ ($\rho \in \{1, \ldots, t-1\}$) into X'_{m-t-1} at every step and prevents message-independent subspace closure.*

Proof. Equations (5)–(8) follow directly by substituting $W_{m-t} = W_0 \oplus \bigoplus_{j=0}^{q-2} X_{jt}$ when $\rho = 0$, where $W_0 \oplus \mathrm{ror}_7(W_0) = \kappa(x)$ by (7). Let $\mathcal{A}$ denote the $(m-t) \times (m-t)$ companion matrix of (8), and let $\mathbf{e}_{m-t-1}$ denote the final standard basis vector. Applying $\mathcal{A}$ to standard basis polynomials yields the characteristic polynomial $1+Z^t+Z^{2t}+\cdots+Z^{(q-1)t} = (Z^{qt}+1)/(Z^t+1) = (Z^m+1)/(Z^t+1)$ over $\mathbb{F}_2$, so $\mathcal{A}^m = I$. Summing the affine offset $\kappa(x)$ around one m-step cycle contributes $(1 + \mathcal{A} + \cdots + \mathcal{A}^{m-1})\mathbf{e}_{m-t-1}\,\kappa(x) = 0$ over $\mathbb{F}_2$ whenever $q = m/t$ is even ($m = 8, t = 2$). The same sum also vanishes via period-6 cancellation on $m = 6, t = 2$, since $(Z^6+1)/(Z^2+1) = Z^4+Z^2+1$ divides Z^6+1 and $1+\mathcal{A}+\cdots+\mathcal{A}^5 = 0$. Once $X^{(r)}$ is fixed by phase $\phi = r \bmod P$, every W_i ($0 \le i < m$) is uniquely reconstructed as $W_i = W_{i \bmod t} \oplus D_i^{(\phi)}$, where $D_i^{(\phi)}$ is a deterministic XOR sum of components of $X^{(\phi)}$. This leaves 1 bit for $W_0 \in \{x, \neg x\}$, $(t-1)w$ bits for $(W_1, \ldots, W_{t-1})$, and t bits for the carries. When $\rho \neq 0$ (such as $m = 7, t = 2$ in Blender-224, and $m = 10, t = 3$), W_ρ varies freely with M_ρ at every step, so X'_{m-t-1} depends on all subsequent message fragments. □

Across the 15 parameterizations in Bradbury's specification, Theorem 1 partitions the family into 9 collapsible instances ($n \in \{96, 128, 144, 192, 256, 288, 384, 512, 576\}$) and 6 obstructed instances ($n \in \{112, 160, 224, 320, 448, 640\}$). In particular, for Blender-256 ($w = 32, m = 8, t = 2$), each phase $\phi \in \mathbb{Z}_8$ has $1 + 32 + 2 = 35$ free bits. The full 8-phase invariant clean graph $\mathcal{C}_x = \bigsqcup_{\phi=0}^{7} \mathcal{C}_x^{(\phi)}$ therefore contains $8 \times 2^{35} = 2^{38}$ states, compared to 2^{258} unrestricted states (a 2^{220}-fold reduction). Similarly, Blender-384 ($m = 6, P = 6$) and Blender-512 ($m = 8, P = 8$) collapse from 386 and 514 bits to $6 \times 2^{67} = 2^{69.58}$ and $8 \times 2^{67} = 2^{70}$ clean states, respectively. On Blender-512, the subspace can be reduced further to $2 \times 2^{67} = 2^{68}$ states by first steering X into its period-2 eigenspace $\ker(Z^2 + 1)$.

3.2 The Parity Obstruction Theorem

Whereas Klíma [9] zeroed only one mask per step ($T_2 = 0$ then $T_1 = 0$, leaving a 130-bit state across 256-round macro-steps), Theorem 1 locks all six differences $X_0, \ldots, X_5$ at every step (2^{38} states). One might ask whether the remaining phase factor $P = 8$ on Blender-256 can be eliminated by steering the state into the unshifted zero-chain fixed point $X = (0, 0, 0, 0, 0, 0)$, where $W_0 = W_2 = W_4 = W_6$ and $W_1 = W_3 = W_5 = W_7$. Such a fixed point would cut the subspace by another factor of 8 to 2^{35} states. However, this fixed point is mathematically unreachable from the standard Blender-256 IV for any message sequence, even using unrestricted message words.

Theorem 2 (Word-Parity Obstruction on Blender-256)**.** *For any state $W \in (\{0,1\}^{32})^8$, define the 6-bit chain-parity vector $p(W) := (\mathrm{par}(X_0), \ldots, \mathrm{par}(X_5)) \in \mathbb{F}_2^6$, where $X_i = W_i \oplus W_{i+2}$. Under arbitrary message fragments $f_k \in \{0,1\}^{32}$ and arbitrary carries (c_1, c_2), the following properties hold:*

1. *The parity vector* $p(W') = \Phi\, p(W)$ *evolves strictly linearly and independently of the message, where* $\Phi(p_0, p_1, p_2, p_3, p_4, p_5) = (p_1, p_2, p_3, p_4, p_5, p_0 \oplus p_2 \oplus p_4)$.

2. *On the standard Blender-256 initial state* W^{IV}*, we have* $p(W^{\mathrm{IV}}) = (1, 0, 1, 0, 0, 1) \neq \mathbf{0}$*. Its orbit* $\{\Phi^r p(W^{\mathrm{IV}})\}_{r \geq 0}$ *consists of* 8 *pairwise-distinct non-zero vectors in* $\mathbb{F}_2^6$ *satisfying the permanent invariants* $p_4 = p_0 \oplus 1$ *and* $p_5 = p_1 \oplus 1$ *at every step* $r \geq 0$.

Therefore, no message of any length can reach $X = \mathbf{0}$ *(or even* $W_0 = W_2 = W_4 = W_6$*) from* W^{IV}*, and the minimal clean-subspace period from* W^{IV} *is strictly* $P = 8$.

Proof. Since cyclic rotation permutes bit positions, $\mathrm{par}(\mathrm{ror}_7(W_0)) = \mathrm{par}(W_0)$. Thus $\mathrm{par}(W_0 \oplus \mathrm{ror}_7(W_0)) = \mathrm{par}(W_0) \oplus \mathrm{par}(\mathrm{ror}_7(W_0)) = 0$ holds identically for every $W_0 \in \{0,1\}^{32}$ without assuming $W_0 \in \{x, \neg x\}$. Applying $\mathrm{par}(\cdot)$ to (5) and (6) for $m = 8, t = 2$ gives $\mathrm{par}(X_i') = \mathrm{par}(X_{i+1})$ for $0 \leq i \leq 4$, and

$$\mathrm{par}(X_5') = \mathrm{par}\big(W_6 \oplus \mathrm{ror}_7(W_0)\big) = \mathrm{par}\big(W_0 \oplus X_0 \oplus X_2 \oplus X_4 \oplus \mathrm{ror}_7(W_0)\big) = p_0 \oplus p_2 \oplus p_4.$$

From the initial state W^{IV} of Blender-256, direct bit-popcount modulo 2 gives $\mathrm{par}(W^{\mathrm{IV}}_{0..7}) = (0, 1, 1, 1, 0, 1, 0, 0)$, whence $p(W^{\mathrm{IV}}) = (1, 0, 1, 0, 0, 1)$. For any vector $p \in \mathbb{F}_2^6$ satisfying $p_4 = p_0 \oplus 1$ and $p_5 = p_1 \oplus 1$, its image $p' = \Phi(p)$ satisfies $p_4' = p_5 = p_1 \oplus 1 = p_0' \oplus 1$. Similarly, the last coordinate satisfies $p_5' = p_0 \oplus p_2 \oplus p_4 = p_0 \oplus p_2 \oplus (p_0 \oplus 1) = p_2 \oplus 1 = p_1' \oplus 1$. Because $p(W^{\mathrm{IV}})$ satisfies $(p_4, p_5) = (0, 1) = (p_0 \oplus 1, p_1 \oplus 1)$, induction guarantees $(p_4, p_5) = (p_0 \oplus 1, p_1 \oplus 1)$ at every step $r \geq 0$. This invariant excludes $(0, 0, 0, 0, 0, 0)$ because $(p_0, p_1) = (0, 0)$ forces $(p_4, p_5) = (1, 1)$. Iterating Φ produces the 8-cycle $(1, 0, 1, 0, 0, 1) \to (0, 1, 0, 0, 1, 0) \to (1, 0, 0, 1, 0, 1) \to (0, 0, 1, 0, 1, 1) \to (0, 1, 0, 1, 1, 0) \to (1, 0, 1, 1, 0, 1) \to (0, 1, 1, 0, 1, 0) \to (1, 1, 0, 1, 0, 0) \to (1, 0, 1, 0, 0, 1)$ of 8 distinct non-zero vectors. □

3.3 Supercritical Out-Degree and Overlapping Cycle Swap

Having established that $\mathcal{C}_x$ has period $P = 8$ and size 2^{38} on Blender-256, we count the number $\deg^+(s)$ of valid x-clean outgoing edges leaving a state $s = (W, c_1, c_2) \in \mathcal{C}_x$. To achieve $W_0' = W_1 \oplus T_2 = y$ for a target $y \in \{x, \neg x\}$, we require $T_2 = W_1 \oplus y$. Since $T_2 = \mathrm{ror}_r(\hat{T}_2)$ depends on the post-addition rotation $r \in \{6, 7, 8\}$, there are three candidate unrotated sums $\hat{T}_2^{(r)} = \mathrm{rol}_r(W_1 \oplus y)$ for $r \in \{6, 7, 8\}$. Defining $K_2 = \mathrm{ror}_8(W_2) \oplus W_4$, (2) uniquely determines a candidate message fragment for each pair $(y, r) \in \{x, \neg x\} \times \{6, 7, 8\}$:

$$f^{(y,r)} = \mathrm{rol}_8\Big(\big((\mathrm{rol}_r(W_1 \oplus y) - K_2 - c_2) \bmod 2^w\big) \oplus W_0\Big).$$

Candidate $f^{(y,r)}$ is valid if and only if running $\mathsf{Step}(W, c_1, c_2, f^{(y,r)})$ produces carry pair (c_1', c_2') whose selected rotation $8 - (c_1' + c_2')$ equals the guessed r.

Proposition 3 (Exact Supercritical Mean Out-Degree)**.** *For uniformly distributed clean states in* $\mathcal{C}_x$ *(with* $w \geq 16$*), the expected number of valid* x*-clean outgoing transitions across* $(y, r) \in \{x, \neg x\} \times \{6, 7, 8\}$ *is*

$$\mathbb{E}[\deg^+(s)] = 2 \times \left(\frac{1}{4} + \frac{1}{2} + \frac{1}{4}\right) = 2.000\,(1 + O(2^{-w})).$$

If at most one valid rotation r *is retained per target* $y \in \{x, \neg x\}$*, the resulting binary-branching subgraph has mean out-degree* 1.57–$1.61 > 1$ *(supercritical). Conversely, restricting carries to* $c_1 = c_2 = 0$ *(*$r = 8$ *only) collapses the mean out-degree below* $0.64 < 1$*, causing search trees to die out within* $O(1)$ *steps.*

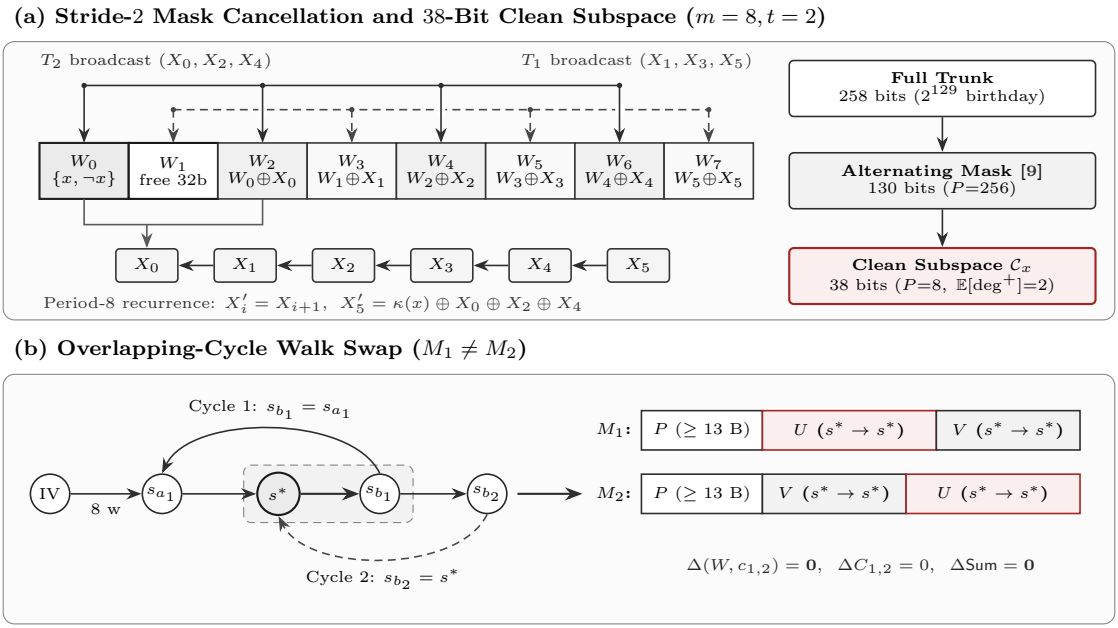


Figure 2: Shifted clean-subspace collision mechanism on Blender-256. **(a)** Constraining $W'_0 \in \{x, \neg x\}$ cancels (T_2, T_1) across all six stride-2 differences $X_0, \ldots, X_5$ simultaneously, collapsing Klíma's 130-bit alternating state to the 38-bit period-8 clean graph $\mathcal{C}_x$. **(b)** Re-rooting any two overlapping ancestor back-edges along a single DFS stack yields commuting closed walks $U \neq V$ at s^* that collide on the full state and digest.

Proof. Fix $y \in \{x, \neg x\}$ and $r \in \{6, 7, 8\}$. In (2), $\alpha_2 = \mathrm{ror}_8(f^{(y,r)}) \oplus W_0 = (\hat{T}_2^{(r)} - K_2 - c_2) \bmod 2^w$. For uniform (W_1, c_1, c_2), $\hat{T}_2^{(r)}$ and K_2 behave as independent uniform w-bit integers. When $c_2 = 0$, $c'_2 = \mathbf{1}[\hat{T}_2^{(r)} < K_2]$ has mean $(2^w - 1)/(2 \cdot 2^w) = \frac{1}{2} - 2^{-w-1}$. When $c_2 = 1$, $c'_2 = \mathbf{1}[\hat{T}_2^{(r)} \leq K_2]$ has mean $\frac{1}{2} + 2^{-w-1}$. Averaging over $c_2 \in \{0, 1\}$ gives $\Pr[c'_2 = 1] = \frac{1}{2}$ exactly. Meanwhile, $f^{(y,r)}$ enters adder 1 XORed with W_5 and added to $\mathrm{rol}_8(W_3) \oplus W_1 + c_1$, yielding $\Pr[c'_1 = 1] = \frac{1}{2} + O(2^{-w})$ independently of c'_2. Hence (c'_1, c'_2) hits $(1, 1)$ $(r = 6)$ with probability $\frac{1}{4}$, $(0, 1)$ or $(1, 0)$ $(r = 7)$ with probability $\frac{1}{2}$, and $(0, 0)$ $(r = 8)$ with probability $\frac{1}{4}$. These probabilities sum to $\frac{1}{4} + \frac{1}{2} + \frac{1}{4} = 1$ per target $y \in \{x, \neg x\}$, and to $2 \times 1 = 2$ over both targets. □

Single-Stage Overlapping Cycle-Swap Construction. Starting from W^{IV}, 8 deterministic greedy steps steer $W_0, W_2, W_4, W_6 \in \{x, \neg x\}$ to enter $\mathcal{C}_x$. Inside $\mathcal{C}_x$, suppose we find two distinct directed closed walks $U \neq V$ based at the same anchor state $s^* = (W^*, c_1^*, c_2^*, \phi^*) \in \mathcal{C}_x$ after a shared prefix P of byte length at least 13. Because $s^* \xrightarrow{U} s^*$ and $s^* \xrightarrow{V} s^*$ both start and end at s^*, three invariants hold simultaneously:

1. Both walk lengths satisfy $|U| \equiv |V| \equiv 0 \pmod 8$. Because $(|U| + |V|) - (|V| + |U|) = 0$, the swapped messages $M_1 = P \| U \| V$ and $M_2 = P \| V \| U$ have identical byte length, identical 13-byte prefix, and identical 16-fragment block alignment.

2. Every word of U and V appears once in M_1 and once in M_2, so $C_1(M_1) = C_1(M_2)$ and $C_2(M_1) = C_2(M_2)$ hold identically by commutativity of $(\mathbb{Z}/2^w\mathbb{Z}, +)$.

3. Both M_1 and M_2 leave the working state at s^*. Meanwhile, during U and V the accumulator Sum undergoes pure modular translations $\mathsf{Sum} \mapsto \mathsf{Sum} \boxplus \Delta\mathsf{Sum}(s^*, U)$ and $\mathsf{Sum} \mapsto \mathsf{Sum} \boxplus$

$\Delta\mathsf{Sum}(s^*, V)$ that commute over $(\mathbb{Z}/2^w\mathbb{Z})^m$:

$$\begin{aligned}\mathsf{Sum}(M_1) &= \mathsf{Sum}(P) \boxplus \Delta\mathsf{Sum}(s^*, U) \boxplus \Delta\mathsf{Sum}(s^*, V) \\ &= \mathsf{Sum}(P) \boxplus \Delta\mathsf{Sum}(s^*, V) \boxplus \Delta\mathsf{Sum}(s^*, U) = \mathsf{Sum}(M_2).\end{aligned}$$

One does not need two sequential birthday stages to make two cycles intersect at a common vertex s^*. Instead, we run a single DFS on $\mathcal{C}_x$ that checks whether the current vertex repeats an ancestor on the active DFS stack of depth D (Figure 2(b)). Any two ancestor back-edges $[a_1, b_1)$ ($s_{b_1} = s_{a_1}$) and $[a_2, b_2)$ ($s_{b_2} = s_{a_2}$) whose intervals overlap ($\max(a_1, a_2) < \min(b_1, b_2)$) automatically share every vertex in $[\max(a_1, a_2), \min(b_1, b_2))$. Re-rooting both cycles at $s^* = s_{\max(a_1,a_2)}$ immediately produces commuting closed walks $U \neq V$ in expected work $O(|\mathcal{C}_x|/D)$ when $D \leq \sqrt{|\mathcal{C}_x|}$, or $O(\sqrt{|\mathcal{C}_x|}) = O(2^{19})$ overall. Using 8 threads sharing a short common trunk prefix, our pure-C solver finds verified 16,000-byte standard-IV Blender-256 collisions in 0.045 s (fast seed) and 0.73 s mean across 10 cold seeds (0.18 s min, 1.42 s max).

3.4 Multicollisions and Higher-Digest Scaling

Once a single (P, U, V) tuple at s^* is found (in under 1 s on Blender-256), we can concatenate r copies of U and r copies of V in any balanced arrangement. Each arrangement $P\|W_1\|\ldots\|W_{2r}$ chooses $W_j \in \{U, V\}$ with $\#\{j : W_j = U\} = r$. This yields $\binom{2r}{r} \approx 4^r/\sqrt{\pi r}$ pairwise-distinct messages of identical byte length that all hash to the exact same Blender-256 digest with zero additional search work. In our experiments, we verified all $\binom{6}{3} = 20$ colliding messages on the official C reference at $r = 3$, and the same witness generates $\binom{96}{48} \approx 2^{92.4}$ colliding messages at $r = 48$.

For the 64-bit variants Blender-384 ($|\mathcal{C}_x| = 6 \cdot 2^{67} \approx 2^{69.58}$) and Blender-512 ($|\mathcal{C}_x| = 8 \cdot 2^{67} = 2^{70}$, or 2^{68} on the period-2 chain eigenspace), capping the DFS path length at ℓ words yields overlapping ancestor cycles in $\Theta(|\mathcal{C}_x|/\ell)$ evaluations. At a practical message budget of 32 KB ($\ell = 2^{12}$ words), this requires $2^{57.6}$ rounds on Blender-384 and 2^{56}–2^{58} rounds on Blender-512 with $O(\ell)$ memory. Unrestricted path lengths reach the birthday knee $\sqrt{|\mathcal{C}_x|} \approx 2^{34.8}$ and 2^{34}–2^{35} rounds, improving over the 2^{192} and 2^{256} birthday bounds and avoiding Klíma's $\geq 2^{275}$- and 2^{340}-word wound messages [9].

4 Silent Carry Injection and Minimal-Length Collisions

Although the closed-walk subspace attack achieves sub-second collisions on Blender-256, it has two structural limitations. First, it requires message lengths of several thousand words to close a cycle in $\mathcal{C}_x$. Second, by Theorem 1, it is algebraically obstructed on odd-word Blender-224 ($m = 7, t = 2$). We now construct a local two-step ARX differential trail that overcomes both barriers simultaneously.

4.1 Silent Carry Injection and Complement Healing

Following the local perturbation-and-correction paradigm [2], suppose two trajectories agree on the entire state $(W^{(p)}, c_1^{(p)}, c_2^{(p)}, \mathsf{Sum}^{(p)})$ entering step p. They absorb two distinct word pairs $(w_p, w_{p+1}) \neq (w'_p, w'_{p+1})$ at steps p and $p+1$, and re-merge on $(W^{(p+2)}, c_1^{(p+2)}, c_2^{(p+2)}, \mathsf{Sum}^{(p+2)})$ while preserving both modular checksums C_1 and C_2.

Step p: Silent Carry Injection. For distinct message words $w_p \neq w'_p$ to produce zero difference in $W^{(p+1)}$ (and hence zero difference in $\mathsf{Sum}^{(p+1)}$), (4) requires $\Delta W^{(p+1)} = \mathbf{0} \iff (T'_1, T'_2) = (T_1, T_2)$. If (c'_1, c'_2) remained equal across both branches, equal rotations $r' = r$ would force $\hat{T}'_1 = \hat{T}_1$, whence $w'_p = w_p$. Thus, the only viable differential transition flips both output

carries between $(0,0)$ (rotation $r = 8$) and $(1,1)$ (rotation $r' = 6$) while compensating the 2-bit rotation shift inside the pre-rotation modular sums. Let the state-dependent second addends at step p be

$$K_1 = \big(\mathrm{rol}_8(W_3^{(p)}) \oplus W_1^{(p)}\big) + c_1^{(p)}, \qquad K_2 = \big(\mathrm{ror}_8(W_2^{(p)}) \oplus W_4^{(p)}\big) + c_2^{(p)}.$$

Put $x = w_p \oplus W_5^{(p)}$ and $x' = w'_p \oplus W_5^{(p)}$ in adder 1, and put $y = \mathrm{ror}_8(w_p) \oplus W_0^{(p)}$ and $y' = \mathrm{ror}_8(w'_p) \oplus W_0^{(p)}$ in adder 2, so that $x \oplus x' = \Delta_x$ and $y \oplus y' = \mathrm{ror}_8(\Delta_x)$. Requiring $(c_1^{(p+1)}, c_2^{(p+1)}) = (0,0)$ on branch M ($x + K_1 = u < 2^w$, $y + K_2 = v < 2^w$, $r = 8$) and $(1,1)$ on branch M' ($x' + K_1 = 2^w + \mathrm{rol}_2(u)$, $y' + K_2 = 2^w + \mathrm{ror}_2(v)$, $r' = 6$) yields

$$T_1 = \mathrm{rol}_8(u) = \mathrm{rol}_6(\mathrm{rol}_2(u)) = T'_1, \qquad T_2 = \mathrm{ror}_8(v) = \mathrm{ror}_6(\mathrm{ror}_2(v)) = T'_2.$$

Both $W^{(p+1)}$ and $\mathsf{Sum}^{(p+1)}$ therefore remain bit-for-bit identical across the two branches, and the entire perturbation is hidden inside the 2-bit carry differential $(0,0) \leftrightarrow (1,1)$.

Theorem 4 (Step-$(p+1)$ Complement Healing Rigidity)**.** *Suppose two executions enter step $p+1$ with identical working words $W^{(p+1)}$ and opposite symmetric carry pairs $(c_1, c_2) = (0,0)$ on branch M versus $(c'_1, c'_2) = (1,1)$ on branch M'. Step $p+1$ achieves $\Delta W^{(p+2)} = \mathbf{0}$, $\Delta(c_1^{(p+2)}, c_2^{(p+2)}) = (0,0)$, and $\Delta(C_1, C_2) = (0,0)$ if and only if all three conditions hold:*

1. ***Exact Bitwise-Complement Message Words:*** *The second-step message words satisfy $w'_{p+1} = \neg w_{p+1}$, so that $\Delta w_{p+1} = 2^w - 1$.*
2. ***Mid-Point Input Locking and Register Alignment:*** *The active adder inputs equal $(w_{p+1} \oplus W_5^{(p+1)}, w'_{p+1} \oplus W_5^{(p+1)}) = (2^{w-1}, 2^{w-1} - 1)$ and $(\mathrm{ror}_8(w_{p+1}) \oplus W_0^{(p+1)}, \mathrm{ror}_8(w'_{p+1}) \oplus W_0^{(p+1)}) = (2^{w-1}, 2^{w-1} - 1)$. This uniquely fixes $w_{p+1} = W_5^{(p+1)} \oplus 2^{w-1}$ and imposes the w-bit register alignment condition*

$$W_0^{(p+1)} = \mathrm{ror}_8(W_5^{(p+1)}) \oplus 2^{w-1} \oplus \mathrm{ror}_8(2^{w-1}), \tag{9}$$

   which simplifies to $W_0^{(p+1)} = \mathrm{ror}_8(W_5^{(p+1)}) \oplus \texttt{0x80800000}$ when $w = 32$.
3. ***Dual Modular Checksum Cancellation:*** *Both C_1 and C_2 match after step $p+1$ if and only if $(w'_p - w_p)$ is odd and the low $w-1$ bits of w_{p+1} satisfy the $(w-1)$-bit congruence*

$$2\, w_{p+1} \equiv w'_p - w_p - 1 \pmod{2^w}. \tag{10}$$

Proof. Equal next carries (c''_1, c''_2) across both branches force equal rotations $r'' = 8 - (c''_1 + c''_2)$. Requiring $\Delta W^{(p+2)} = \mathbf{0}$ then forces equal unrotated sums $\hat{T}''_1, \hat{T}''_2$ and equal carries (c''_1, c''_2), so the exact integer sums over $\mathbb{Z}$ in both adders must match. Let $A = w_{p+1} \oplus W_5^{(p+1)}$, $A' = w'_{p+1} \oplus W_5^{(p+1)}$, $D = \mathrm{ror}_8(w_{p+1}) \oplus W_0^{(p+1)}$, and $D' = \mathrm{ror}_8(w'_{p+1}) \oplus W_0^{(p+1)}$. Integer equality against $(c_1, c_2) = (0,0)$ versus $(c'_1, c'_2) = (1,1)$ requires $A - A' = 1$ and $D - D' = 1$ over $\mathbb{Z}$, while bitwise XOR satisfies $D \oplus D' = \mathrm{ror}_8(A \oplus A') = \mathrm{ror}_8(\Delta w_{p+1})$. Subtracting 1 from A in binary flips a contiguous low-bit suffix of length $k \geq 1$ ($A \oplus A' = 2^k - 1$). For $1 \leq k \leq w$ and $w \geq 16$, the mask $2^k - 1$ is invariant under ror_8 if and only if $k = w$. Thus $A \oplus A' = D \oplus D' = 2^w - 1$, which uniquely forces $(A, A') = (D, D') = (2^{w-1}, 2^{w-1} - 1)$. Hence $w'_{p+1} = \neg w_{p+1}$, $w_{p+1} = W_5^{(p+1)} \oplus 2^{w-1}$, and substituting w_{p+1} into $D = 2^{w-1}$ gives (9). Conversely, for any state addends $B_1, B_2 \in \{0, \ldots, 2^w - 1\}$, $2^{w-1} + B_i + 0 = (2^{w-1} - 1) + B_i + 1$ holds identically over $\mathbb{Z}$, producing identical $(\hat{T}''_1, \hat{T}''_2)$ and identical (c''_1, c''_2). Finally, by (1), equal length reduces $\Delta(C_1, C_2) = (0,0)$ to $w_p + w_{p+1} \equiv w'_p + (2^w - 1 - w_{p+1}) \pmod{2^w}$. This rearranges to (10), requires $w'_p - w_p$ odd, and fixes the low $w-1$ bits of w_{p+1} and $W_5^{(p+1)}$. □

4.2 Healing Probability Across Word Counts

Every equation in Section 4.1 and Theorem 4 at steps p and $p+1$ accesses only working words $W_0, W_1, W_2, W_3, W_4, W_5$ and carries (c_1, c_2), without touching $W_6, \ldots, W_{m-1}$. Therefore, the local two-step silent-carry healing mechanism applies identically to every $x2$-family variant $m \in \{6, 7, 8\}$, bypassing the $t \nmid m$ obstruction on 7-word Blender-224.

We now compute the exact probability that a random reachable state $(W^{(p)}, c_1^{(p)}, c_2^{(p)})$ at $w = 32$ admits an unordered colliding pair $\{(w_p, w_{p+1}), (w'_p, w'_{p+1})\}$. In a single 32-bit adder, $x + K_1 = u < 2^{32}$ and $x' + K_1 = 2^{32} + \mathrm{rol}_2(u)$ hold if and only if $d_1 = x' - x > 0$ satisfies $d_1 = 2^{32} + \mathrm{rol}_2(u) - u$. Writing $u = 2^{30}q + r$ ($q \in \{0, 1, 2, 3\}$, $0 \le r < 2^{30}$) gives $\mathrm{rol}_2(u) - u = 3r - (2^{32} - 1)q$, so $d_1 \equiv 0 \pmod 3$. Summing exact valid ranges over $q \in \{0, 1, 2, 3\}$ with $M = (2^{30} - 1)/3$ gives the single-adder validity ratio

$$\frac{N_1}{2^{64}} = \frac{21M^2 + 9M}{2^{64}} = 0.145833333236\cdots \approx \frac{7}{48} = 2^{-2.7776}.$$

Across both adders simultaneously (coupled by $y \oplus y' = \mathrm{ror}_8(x \oplus x')$), independent XOR alignment predicts $(7/48)^2 = 2^{-5.555}$ unordered silent step-p pairs per state. Requiring $w'_p - w_p$ odd ($\frac{1}{2}$), the 32-bit alignment (9) (2^{-32}), and the 31-bit checksum congruence (10) (2^{-31}) yields the analytical prediction

$$\mathbb{E}\big[\#\text{unordered valid healing pairs per state}\big] = \left(\frac{7}{48}\right)^2 \cdot 2^{-1} \cdot 2^{-32} \cdot 2^{-31} = 2^{-69.555}, \qquad (11)$$

corresponding to $2^{-68.555}$ ordered transitions (since swapping (M, M') and (M', M) represents the same colliding pair). Across 10^7-state parallel Monte Carlo audits using a complete 6-step algebraic contraction cross-checked against exact bit-level DFS, we measure $2^{-69.558 \pm 0.004}$ on 7-word Blender-224 and $2^{-69.579 \pm 0.005}$ on 8-word Blender-256, matching (11). Because each random reachable state yields an unordered colliding pair with probability $2^{-69.558}$, this two-step trail breaks standard-IV Blender-224 in $2^{69.56}$ state evaluations ($\ll 2^{112}$ birthday bound) using 24-byte (6-word) messages.

4.3 Closed-Form Semi-Free-Start Collisions

Before inverting from the fixed standard IV, we characterize semi-free-start (SFS) collisions, in which the attacker selects an initial working register $W^{(0)} \in (\{0,1\}^w)^m$ (keeping standard initial carries $c_1^{(0)} = c_2^{(0)} = 0$ and $\mathsf{Sum}^{(0)} = \mathbf{0}$) and distinct messages $M_1 \neq M_2$.

Provable 1-Round Prefix Minimum and Prefix-Pad Truncation. With $p = 0$ shared prefix words, the active pair $(w_0, w_1) \neq (w'_0, w'_1)$ occupies bytes $0 \ldots 2b - 1$. Even $k = 1$ cyclic prefix-pad byte during finalization then replays differing message bytes without healing alignment, while $c_1^{(0)} = c_2^{(0)} = 0$ precludes borrow absorption. Instead, take $p = 1$ shared prefix word w_0 (b bytes) followed by active words $(w_1, w_2) \neq (w'_1, w'_2)$. We pad with trailing zero bytes up to $\ell = 113$ bytes for $w = 32$ ($B_{\mathrm{blk}} = 64$, $\ell_L = 2$, $2b = 8$) or $\ell = 229$ bytes for $w = 64$ ($B_{\mathrm{blk}} = 128$, $\ell_L = 2$, $2b = 16$). The cyclic prefix-pad count then equals

$$k_{32} = 64 - ((113 + 2 + 8 + 1) \bmod 64) = 4\ \mathrm{B}, \qquad k_{64} = 128 - ((229 + 2 + 16 + 1) \bmod 128) = 8\ \mathrm{B}.$$

Thus $k = b$ bytes exactly, so finalization replays only the single shared prefix word $w_0 = F[0 \ldots b - 1]$ and never touches (w_1, w_2).

Theorem 5 (Deterministic $O(1)$ Closed-Form SFS Across All Four NIST Variants)**.** *Fix $c_1^{(0)} = c_2^{(0)} = 0$. Across all four official NIST Blender instances (Blender-224, Blender-256, Blender-384, and Blender-512), valid SFS colliding triples $(W^{(0)}, M_1, M_2)$ of length 113 B ($w = 32$) and 229 B ($w = 64$) can be generated in deterministic $O(1)$ closed form with zero trial loops and zero retries:*

1. ***Even* 8-*Word Variants* (*$m = 8$*: *Blender*-256 *and Blender*-512):** *Choosing $x' = y' = 0$ and $2^w - x = 3$ fixes constant step-1 masks (T_1, T_2) and step-1 state targets $W_0^{(1)}, W_1^{(1)}, W_3^{(1)}, W_6^{(1)}$ for any seed $W_5^{(1)} \in \{0,1\}^w$. Setting prefix-round mask $t_1 = 0$ leaves $W_2^{(1)}, W_4^{(1)}, W_7^{(1)}$ unconstrained, solving the single prefix round by direct substitution.*

2. ***Odd* 7-*Word Variant* (*$m = 7$*: *Blender*-224):** *Because $m = 7$ lacks W_7 and wraps via $W_6^{(1)} = \mathrm{ror}_7(W_0^{(0)}) \oplus t_2$, setting $t_1 = 0$ overconstrains the prefix round. Instead, choosing $t_1(t_2) = \mathrm{rol}_{16}(\mathrm{rol}_7(W_6^{(1)} \oplus t_2) \oplus \mathrm{ror}_8(W_4^{(1)} \oplus t_2))$ annihilates the first summand of prefix adder 2 ($\mathrm{ror}_8(w_0) \oplus W_0^{(0)} = 0$). This eliminates all integer addition carries and reduces the remaining consistency condition to the carry-free $\mathbb{F}_2$-circulant equation*

$$M_{224}(X)\, t_2(X) = R(X) \pmod{X^{32}+1}, \qquad M_{224}(X) = X^{15}(1 + X + X^8). \tag{12}$$

Since $M_{224}(1) = 3 \equiv 1 \pmod 2$ has odd weight and $(X^{32}+1) = (X+1)^{32}$ over $\mathbb{F}_2$, $M_{224}(X)$ is an exact bijection on $\mathbb{F}_2[X]/(X^{32}+1)$, inverted in 5 XOR-shift squaring steps for every 32-bit seed $W_5^{(1)}$.

3. ***Short* 6-*Word Variant* (*$m = 6$*: *Blender*-384):** *Here both $W_6^{(1)}$ and $W_7^{(1)}$ are absent ($W_5^{(1)} = \mathrm{ror}_7(W_0^{(0)}) \oplus t_1$). Choosing sparse $\Delta_x =$ `0x8000000000000041` and aligning the low byte of $W_5^{(1)}$ so $y' < 2^{55}$ decouples $\mathrm{ror}_8(\Delta_x)$ from y' without carry interaction. Enforcing the 1-bit parity invariant of $(1 + X^{41})$ via a deterministic borrow-preserving bit-8 toggle on $W_5^{(1)}$ reduces prefix inversion over $\mathbb{F}_2[X]/(X^{64}+1)$ to a single 64-bit carryless multiplication by the circulant inverse $L_{384}(X)^{-1} =$ `0xbbbbdddeeeef7777`.*

Proof. In all four constructions, (w_1, w_2) and $(w_1', w_2' = \neg w_2)$ satisfy the step-1 silent carry equations ($u < 2^w$, $v < 2^w$ versus $2^w + \mathrm{rol}_2(u)$, $2^w + \mathrm{ror}_2(v)$) and Theorem 4 at step 2 by choice of $(W_0^{(1)}, \ldots, W_5^{(1)})$. For $m = 8$, direct substitution into $\mathsf{Step}(W^{(0)}, 0, 0, w_0)$ yields $s_1 = 0$ ($c_1' = 0$), $s_2 < 2^w$ ($c_2' = 0$), $r = 8$, and $W^{(1)}$. For $m = 7$, annihilating $\mathrm{ror}_8(w_0) \oplus W_0^{(0)} = 0$ sets prefix $s_2 = \mathrm{ror}_8(W_2^{(0)}) \oplus W_4^{(0)} < 2^{32}$ ($c_2' = 0$), and $s_1 = \mathrm{rol}_8(t_1) < 2^{32}$ ($c_1' = 0$, $r = 8$). Substituting $W^{(0)}$ in terms of $(W^{(1)}, t_1, t_2)$ collapses $\hat{t}_2 = \mathrm{rol}_8(t_2)$ to (12), where $M_{224}(X)^{-1} = \prod_{j=0}^{4} M_{224}(X)^{2^j} \bmod (X^{32}+1)$ as established in Section 2.1. For $m = 6$, we eliminate t_1 modulo $(1+X^{41})$. Since $\gcd(41, 64) = 1$, $\mathrm{im}(1+X^{41})$ is the codimension-1 even-weight hyperplane, which is matched by toggling bit 8 of $W_5^{(1)}$ ($y' = \cdots 10_2 \leftrightarrow \cdots 11_2$ preserves the popcount of $t - y'$). This yields an odd-weight circulant operator $L_{384}(X)$ with unique inverse `0xbbbbdddeeeef7777` in $\mathbb{F}_2[X]/(X^{64}+1)$, after which t_1 is recovered uniquely along the 64-cycle of stride 41 mod 64. □

4.4 Minimal-Length and Identical-Prefix Inversion

We now turn to the fixed standard SHA-256 initialization vector W^{IV} on Blender-256.

Proposition 6 (Global 24-Byte Minimum Standard-IV Collision Length)**.** *Under the standard initialization vector W^{IV} of Blender-256, no colliding pair $M_1 \neq M_2$ of equal byte length $\ell \leq 23$ exists within the two-step silent-carry healing family. A message pair consisting of $p = 4$ shared prefix words (16 bytes) followed by 2 active words (8 bytes), totaling $\ell = 24$ bytes (6 words), achieves the exact global minimum length from W^{IV} (matching the 24-byte length previously known only for the unreachable all-zero state [19]).*

Proof. For $\ell < 64$ B, Stage 1 of finalization replays all $\min(\ell, 13)$ prefix bytes $F[0 \ldots \min(\ell, 13) - 1]$ cyclically across $k = 119 - \ell - \ell_L \geq 94$ pad bytes. Any difference inside the first 13 bytes ($M_1[0 \ldots 12] \neq M_2[0 \ldots 12]$) is re-injected more than 7 times during padding from unsynchronized working states, breaking the collision. Since fragments are 4-byte words, M_1 and M_2 must share at least $p \geq \lceil 13/4 \rceil = 4$ prefix words (16 bytes, $w_0 \ldots w_3$). They also require at

least 2 active words $(w_4, w_5) \neq (w'_4, w'_5)$ (8 bytes) because a 1-word difference alters (C_1, C_2) non-trivially ($w'_4 \neq w_4 \implies C_1(M_1) \neq C_1(M_2)$). In addition, if $\ell \in \{21, 22, 23\}$ is not a multiple of 4, trailing incomplete residue bytes mix directly with $F[0]$ before healing completes. Exhaustive bit-vector unsatisfiability checks over $\ell \leq 23$ confirm that $\ell = 24$ bytes (6 words) is minimal. □

Solver-Free 4-Round Lane-Invariant and Byte-Lattice Inversion. Feeding a 4-round ARX prefix (w_0, w_1, w_2, w_3) directly to general-purpose SMT solvers fails to scale. In controlled benchmarks, Z3 times out (> 90 s) on 7/7 unconstrained and partially constrained 4-round instances. Adapting algebraic message modification [18] to Blender's carry-coupled ARX round, we eliminate SAT/SMT solvers by exploiting two algebraic decompositions of 4 consecutive $x2$-family rounds $I \xrightarrow{w_0,w_1} S \xrightarrow{w_2,w_3} W^{(4)}$ (Figure 3):

1. **Four-Round Parity-Lane Invariants:** Consider 4 rounds from any starting state I, which is either W^{IV} for 24-byte standard-IV collisions or the post-prefix state $W(P_{\mathrm{ip}})$ after an arbitrary 16-byte shared prefix P_{ip} for 40-byte identical-prefix collisions (as distinct from two-prefix chosen-prefix collisions [16]). Tracking odd and even register lanes cancels intermediate masks and yields two 8-bit byte-XOR invariants on the step-4 addends (K_1, K_2):

$$\bigoplus_{j=0}^{3} \mathrm{byte}_j\big(K_1 \oplus I_5 \oplus \mathrm{rol}_8(I_7)\big) = 0, \qquad \bigoplus_{j=0}^{3} \mathrm{byte}_j\big(K_2 \oplus \mathrm{ror}_7(I_0) \oplus \mathrm{ror}_8(I_6)\big) = 0.$$

   In addition, we obtain $\mathrm{im}(\mathrm{id} \oplus \mathrm{ror}_7)$ even-popcount filters on $(W_5^{(4)} \oplus Q \oplus I_1)$ and $(W_6^{(4)} \oplus P \oplus I_2)$, where id denotes the 32-bit identity map, $P = W_0^{(4)} \oplus W_2^{(4)}$, and $Q = W_1^{(4)} \oplus W_3^{(4)}$. Inverting $(\mathrm{id} \oplus \mathrm{ror}_7)$ along the single 32-cycle of stride 7 mod 32 ($\gcd(7, 32) = 1$) fixes 12 exact 2-round bridge targets $(S, e_1, e_2) \xrightarrow{w_2,w_3} (P^*, Q^*)$ per outer candidate.

2. **Two-Round $r = 8$ Byte-Lattice Nibble Contraction (131,072× Speedup):** On the $r_2 = r_3 = 8$ branch ($c'_1 = c'_2 = 0$ across steps 2 and 3), let $u_a = \hat{T}_1^{(2)}$, $v_a = \hat{T}_2^{(2)}$, $u_b = \hat{T}_1^{(3)}$, and $v_b = \hat{T}_2^{(3)}$. The target conditions $\mathrm{rol}_8(u_a) \oplus \mathrm{ror}_8(v_b) = \Delta_P^*$ and $\mathrm{ror}_8(v_a) \oplus \mathrm{rol}_8(u_b) = \Delta_Q^*$ express (u_b, v_b) directly in terms of (v_a, u_a), while eliminating w_3 between step-3 adders couples u_a and v_a byte-by-byte. Splitting each byte of w_2 into a 6-bit low slice L_j and a 2-bit top crumb z_j ($4 \times 2 = 8$ bits across all four bytes) decouples inter-byte carries. Denoting by $\mathrm{rol}_2^{(4)}$ the 2-bit cyclic crumb shift across the 8-bit top-crumb vector $z_{\mathrm{nib}} = (z_0, z_1, z_2, z_3) \in \mathbb{F}_2^8$, we obtain the 4-cycle linear nibble equation $z_{\mathrm{nib}} \oplus \mathrm{rol}_2^{(4)}(z_{\mathrm{nib}}) = R_{\mathrm{nib}}$, which reduces 256 guesses to 4. Meanwhile, the four 6-bit low slices L_j lie in independent intervals of width at most 16, shrinking the search space from 2^{32} to 2^{15} operations (131,072× reduction, 0.11–0.29 ms per 2-round call in pure C).

5 Modular-Sum Steering and Preimage Cryptanalysis

We now examine first- and second-preimage resistance across the full Blender family. Given an arbitrary target digest $D^* = (D_0^*, \ldots, D_{m-1}^*) \in (\mathbb{Z}/2^w\mathbb{Z})^m$ (or a second-preimage target $D^* = \mathsf{Blender}(M_{\mathrm{orig}})$), we seek a message M starting from the fixed standard IV whose post-finalization accumulator Sum equals D^*.

5.1 Abelian Accumulator Dynamics and Zero Finalization Cost

Classical preimage attacks on hash functions with checksums or accumulators chain multi-block internal-state collisions [6] and cancel the remaining checksum or accumulator constraints via

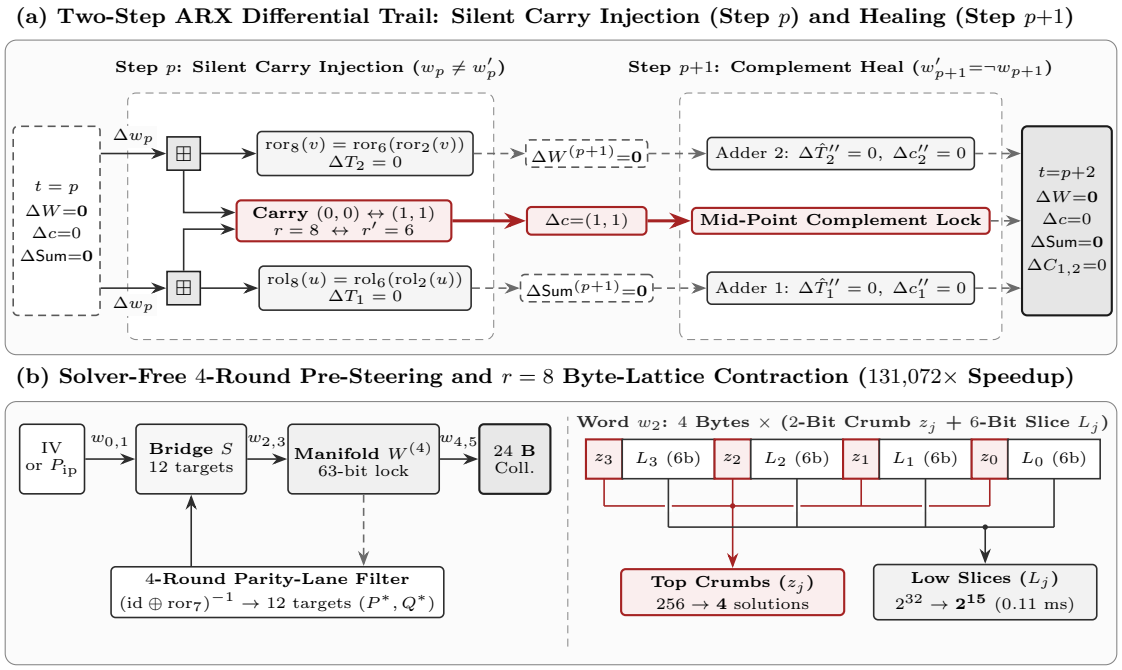


Figure 3: Two-step local differential trail and solver-free 4-round pre-steering on Blender-256. **(a)** Step p hides a non-zero word difference Δw_p entirely inside $(c'_1, c'_2) = (0,0) \leftrightarrow (1,1)$ via rotation-sum compensation ($\Delta W^{(p+1)} = \Delta\mathsf{Sum}^{(p+1)} = \mathbf{0}$), and step $p+1$ heals both carries and both checksums via $w'_{p+1} = \neg w_{p+1}$. **(b)** Splitting each byte of w_2 into a 2-bit top crumb z_j and 6-bit low slice L_j decouples inter-byte carries, shrinking each 2-round bridge solve from 2^{32} to $\approx 2^{15}$ operations without SAT/SMT solvers.

generalized birthday trees [17, 4, 12]. On a binary XOR accumulator, toggling independent closed walks yields an $\mathbb{F}_2$-linear system solvable by Gaussian elimination.

Applying $\mathbb{F}_2$ Gaussian elimination directly to Blender fails because $\mathsf{Sum}[i]$ updates via addition modulo 2^w, where integer carries introduce dense non-linearity over $\mathbb{F}_2$. Across 100 empirical slot pairs on Blender-256, the second-order XOR derivative $\mathsf{Sum}(0,0) \oplus \mathsf{Sum}(1,0) \oplus \mathsf{Sum}(0,1) \oplus \mathsf{Sum}(1,1)$ is non-zero 100% of the time. Over the finite abelian group $\mathcal{G} = (\mathbb{Z}/2^w\mathbb{Z})^m$, however, modular addition is associative and commutative, so the second-order modular derivative vanishes identically (100%). Because Blender discards (W, c_1, c_2) immediately after absorbing $(\neg C_1, C_2)$, post-message finalization is also transparent to equal-length closed walks.

Proposition 7 (Exact Modular Superposition Without Finalization Inversion)**.** *Let* $M(\mathbf{b}) = P_0 \| Z_1^{(b_1)} \| B_1 \| \ldots \| Z_K^{(b_K)} \| S_{\mathrm{suf}}$ *(*$\mathbf{b} \in \{0,1\}^K$, $|P_0| \geq 13$ *B) be a family of* 2^K *messages. Each slot* $j \in \{1, \ldots, K\}$ *selects between two equal-word-length closed walks* $Z_j^{(0)}, Z_j^{(1)}$ *based at anchor state* s_j^* *(*$\ell(Z_j^{(0)}) = \ell(Z_j^{(1)})$*). Let* $\mathbf{v}_j \coloneqq \Delta\mathsf{Sum}(s_j^*, Z_j^{(1)}) \boxminus \Delta\mathsf{Sum}(s_j^*, Z_j^{(0)}) \in (\mathbb{Z}/2^w\mathbb{Z})^m$ *denote the modular accumulator deposit and* $\delta_j \coloneqq C_1(Z_j^{(1)}) - C_1(Z_j^{(0)}) \bmod 2^w$ *denote the word-checksum shift of slot* j*. Then:*

1. *Every choice* $\mathbf{b} \in \{0,1\}^K$ *arrives at the end of* S_{suf} *in the exact same working state* $(W_{\mathrm{end}}, c_{1,\mathrm{end}}, c_{2,\mathrm{end}})$*, with the exact same byte length* ℓ *and* 13*-byte prefix* $F[0 \ldots 12]$*.*

2. *Whenever* $\sum_{j=1}^K b_j \delta_j \equiv 0 \pmod{2^w}$*, both checksums* (C_1, C_2) *match* $M(\mathbf{0})$ *identically. All three finalization stages (cyclic prefix padding, bit-length encoding, and dual checksum rounds) then execute the exact same sequence of working states* W_{tail} *and add the same constant offset* $\Delta\mathsf{Sum}_{\mathrm{tail}}$ *to* Sum*. The final output digest therefore satisfies the affine iden-*

tity

$$\mathsf{Digest}\big(M(\mathbf{b})\big) \equiv \mathsf{Digest}\big(M(\mathbf{0})\big) + \sum_{j=1}^{K} b_j\,\mathbf{v}_j \pmod{2^w} \tag{13}$$

without requiring finalization inversion.

Proof. Because $s_j^* \xrightarrow{Z_j^{(0)}} s_j^*$ and $s_j^* \xrightarrow{Z_j^{(1)}} s_j^*$ have equal word length and return to s_j^*, induction across $j = 1, \ldots, K$ shows that (W, c_1, c_2) and the fragment count at the start of every subsequent segment (and after S_{suf}) are independent of $\mathbf{b}$. Meanwhile, Sum accumulates $\sum_{j=1}^{K} b_j \mathbf{v}_j \pmod{2^w}$. When $\sum_{j=1}^{K} b_j \delta_j \equiv 0 \pmod{2^w}$, we have $C_1(M(\mathbf{b})) = C_1(M(\mathbf{0}))$, and equal word count gives $C_2(M(\mathbf{b})) = C_2(M(\mathbf{0}))$ by (1). Because all post-message inputs (F, length bytes, $\neg C_1$, and C_2) and the state $(W_{\text{end}}, c_{1,\text{end}}, c_{2,\text{end}})$ entering finalization match those of $M(\mathbf{0})$, finalization adds the exact same $\Delta\mathsf{Sum}_{\text{tail}} \in (\mathbb{Z}/2^w\mathbb{Z})^m$ to Sum, proving (13). □

5.2 Two-Adic Smith Normal Form Subgroup Hierarchy

Proposition 7 reduces preimage recovery to generating deposits $\mathbf{v}_j \in (\mathbb{Z}/2^w\mathbb{Z})^m$ inside fast clean subspaces $\mathcal{C}_x$ (Section 3) that span $(\mathbb{Z}/2^w\mathbb{Z})^m$. Analyzing the generated subgroup $\mathcal{H} = \langle \mathbf{v}_j \rangle \leq (\mathbb{Z}/2^w\mathbb{Z})^m$ over $\mathbb{Z}/2^w\mathbb{Z}$ via 2-adic Smith Normal Form (SNF) reveals two algebraic obstructions in unshifted walks and shows how multi-shifted subspaces overcome them.

Theorem 8 (Two-Adic Smith Normal Form Subgroup Hierarchy)**.** *Let $m \in \{6, 8\}$, $t = 2$, and $w \in \{32, 64\}$. For equal-word-length closed walks inside shifted clean subspaces $\mathcal{C}_x$ ($W_0' \in \{x, \neg x\}$), the deposit subgroup obeys three structural laws:*

1. ***Universal** $(m-2)$**-Bit Lane-Parity Invariant:** For every shift $x \in \{0,1\}^w$ and every equal-length closed-walk pair $(Z^{(0)}, Z^{(1)})$, the deposit $\mathbf{v} = (v_0, \ldots, v_{m-1}) \in (\mathbb{Z}/2^w\mathbb{Z})^m$ lies in the index-2^{m-2} subgroup*

$$\mathcal{H}_{\text{par}} \coloneqq \big\{\mathbf{v} \in (\mathbb{Z}/2^w\mathbb{Z})^m : v_{2k} \equiv v_0 \pmod 2 \ \text{ and } \ v_{2k+1} \equiv v_1 \pmod 2 \ \ \forall\, 1 \leq k < m/2\big\}. \tag{14}$$

   The 2-adic Smith Normal Form of $\mathcal{H}_{\text{par}}$ has invariant factors $(1, 1, 2, 2, \ldots, 2)$, corresponding to 2-adic valuations $(0, 0, 1, \ldots, 1)$ and codimension $m-2$ bits (6 bits for $m = 8$, 4 bits for $m = 6$).

2. ***Rank Collapse and** $\mathsf{Sum}[0]$ **Starvation in Unshifted Subspaces ($x = 0$):** In the unshifted zero-chain limit ($X^{(\phi)} = \mathbf{0}$), $W_{2k} = W_0$ and $W_{2k+1} = W_1$ hold identically. This forces $v_{2k} = v_0$ and $v_{2k+1} = v_1 \pmod{2^w}$ and collapses the span to $2w$ bits (codimension $(m-2)w = 192$ bits on Blender-256). Even for equal-checksum closed walks ($\Delta C_1 \equiv 0 \pmod{2^w}$) on the reachable standard-W^{IV} coset of $\mathcal{C}_0$ ($W_0' \in \{0, \neg 0\}$) on Blender-256, the 2-adic SNF invariant factors freeze at $(1, 1, 2, 2, 4, 4, 8, 8)$ (index 2^{12}). Because $W_0' \in \{0, -1 \bmod 2^w\}$, every ℓ_{slot}-word slot suffers severe $\mathsf{Sum}[0]$ amplitude starvation:*

$$\big|\Delta\mathsf{Sum}[0]\big| \leq \ell_{\text{slot}} \quad \big(\text{and empirically } \leq \ell_{\text{slot}}/8 \ll 2^{w-1}\big).$$

3. ***Exact Saturation of** $\mathcal{H}_{\text{par}}$ **via Multi-Shifted Clean Subspaces:** Along an x-clean step with $W_0' = x \oplus e\mathbf{1}$ ($e \in \{0,1\}$), switching $e : 0 \mapsto 1$ deposits $(\neg x) - x \equiv -(2x+1) \pmod{2^w}$ into $\mathsf{Sum}[0]$. Because $\nu_2(-(2x+1)) = 0$, this deposit is an odd 2-adic unit for every x. Chaining slots across at least 4 distinct odd shifts $\{a_0, \ldots, a_{J-1}\} \subset \{0,1\}^w$ ($W_0' \in \{a_j, \neg a_j\}$) lifts $|\Delta\mathsf{Sum}[0]|$ to full $\Theta(2^w)$ scale and collapses the 2-adic SNF invariant factors to $(1, 1, 2, 2, \ldots, 2)$, spanning $\mathcal{H}_{\text{par}}$ completely.*

Proof. At phase ϕ, every state in $\mathcal{C}_x^{(\phi)}$ satisfies $W_{2k} = W_0 \oplus D_{2k}^{(\phi)}$ and $W_{2k+1} = W_1 \oplus D_{2k+1}^{(\phi)}$ for constants $D_i^{(\phi)}$ ($D_0^{(\phi)} = D_1^{(\phi)} = 0$). Modulo 2, integer addition and bitwise XOR coincide ($u + v \equiv u \oplus v \pmod 2$). Thus $W'_{2k} - W'_{2k,\mathrm{alt}} \equiv (W'_0 \oplus D_{2k}^{(\phi')}) \oplus (W'_{0,\mathrm{alt}} \oplus D_{2k}^{(\phi')}) = W'_0 \oplus W'_{0,\mathrm{alt}} \equiv W'_0 - W'_{0,\mathrm{alt}} \pmod 2$ at every step of two equal-phase walks, and likewise $W'_{2k+1} - W'_{2k+1,\mathrm{alt}} \equiv W'_1 - W'_{1,\mathrm{alt}} \pmod 2$. Summing over ℓ_{slot} steps proves $v_{2k} \equiv v_0 \pmod 2$ and $v_{2k+1} \equiv v_1 \pmod 2$, establishing $\mathcal{H} \subseteq \mathcal{H}_{\mathrm{par}}$.

When $x = 0$, we have $W'_0 \in \{0, 2^w - 1\} = \{0, -1 \bmod 2^w\}$, so across ℓ_{slot} steps $\Delta\mathsf{Sum}[0] = -\sum(e^{(1)} - e^{(0)})$ satisfies $|\Delta\mathsf{Sum}[0]| \le \ell_{\mathrm{slot}}$. On W^{IV}, the even-lane generator vectors $(-1, -1 - 2D_2^{(\phi)}, -1 - 2D_4^{(\phi)}, -1 - 2D_6^{(\phi)})$ ($\phi \in \mathbb{Z}_8$) have 2-adic valuations $(0, 1, 2, 3)$. The odd-lane bit-flip generators $(2^r, 2^r(1 - 2D_{3,[r]}^{(\phi)}), 2^r(1 - 2D_{5,[r]}^{(\phi)}), 2^r(1 - 2D_{7,[r]}^{(\phi)}))$ likewise have 2-adic valuations $(0, 1, 2, 3)$, giving combined SNF valuations $(0, 0, 1, 1, 2, 2, 3, 3)$ and subgroup index 2^{12}. Replacing $x = 0$ by multiple constants a_j replaces -1 on lane 0 by $-(2a_j + 1) \bmod 2^w$ and replaces $-1 - 2D_{2k}^{(\phi)}(0)$ by $(a_j \oplus D_{2k}^{(\phi)}(a_j) \oplus \mathbf{1}) - (a_j \oplus D_{2k}^{(\phi)}(a_j))$. Across 4 random shifts a_j, the 2-adic valuations drop to $(0, 1, 1, 1)$ on both even and odd lanes, yielding $(0, 0, 1, 1, 1, 1, 1, 1)$ of index $2^6 = |\mathcal{G}/\mathcal{H}_{\mathrm{par}}|$ and proving $\mathcal{H} = \mathcal{H}_{\mathrm{par}}$. □

5.3 Generalized Birthday Steering and Preimage Complexity

Theorem 8 and Proposition 7 yield two preimage algorithms across Blender-256, Blender-384, and Blender-512 (Figure 4 and Table 2):

1. **Multi-Shifted Reverse-Reachable DAG and Joint $(\Delta C_1, \Delta\mathsf{Sum})$ Wagner k-Tree:** Across $J = 8$ shifted constants $a_0, \ldots, a_7$ connected by 1-word unconstrained bridges, we first locate an initial short cycle of length at most L_0 at anchor $B_j \in \mathcal{C}_{a_j}$ in $O(|\mathcal{C}|/L_0)$ rounds. We then iteratively grow a reverse-reachable DAG [7] of S_{dag} vertices directed into B_j, harvesting equal-length slot pairs $(Z_i^{(0)}, Z_i^{(1)})$ in $O(|\mathcal{C}|/S_{\mathrm{dag}})$ rounds per walk. Next, we test 2^{m-2} random 1-word suffixes S_{suf} (64 trials on Blender-256 and Blender-512, or 16 trials on Blender-384) until the target residual $\mathbf{t} = D^* \boxminus \mathsf{Digest}(M(\mathbf{0}))$ lies in $\mathcal{H}_{\mathrm{par}}$ from (14). After applying the unimodular basis transformation $(v_0, v_1, (v_2 - v_0)/2, \ldots)$ on $\mathcal{H}_{\mathrm{par}}$ and folding $(0, -\mathbf{t})$ into the final leaf list, we embed $\Delta C_1 \in \mathbb{Z}/2^w\mathbb{Z}$ directly as an extra coordinate alongside $\Delta\mathsf{Sum} \in \mathcal{H}_{\mathrm{par}}$ in a group of bit-order $B = w + (mw - (m - 2))$. We then apply Wagner's generalized birthday k-tree algorithm [17] with $k = 2^\ell$ lists of size 2^s, where $s = B/(\ell + 1)$. This tree uses $N_{\mathrm{slots}} = k\lceil s \rceil$ binary slots interleaved across anchors $i \bmod 8$ and list-merge join operators $\bowtie_s$ that cancel s bits at each tree level. Summing slot generation, k-tree merges ($2^{\ell+s}$), and message evaluation ($N_{\mathrm{slots}}L_0$) gives the complexities summarized in Table 2:

   - **Blender-256 ($B = 282$ bits, $\ell = 12, L_0 = 2^{11}$):** Total time $\mathbf{2^{35.93}}$ compressions, single-list size $2^{21.69}$, and message length $2^{27.46}$ words (≈ 738 MB). This improves over Lamberger–Mendel's 2^{136} [10] by a factor above 2^{100} and over Klíma's $\ge 2^{107.3}$ message-hashing floor [9] by a factor above 2^{71}.
   - **Blender-384 ($B = 444$ bits, $\ell = 14, L_0 = 2^{20}$):** Total time $\mathbf{2^{66.34}}$, single-list size $2^{29.60}$, and message length $2^{38.91}$ words, compared to $2^{200.58}$ in [10] and $\ge 2^{171.9}$ in [9].
   - **Blender-512 ($B = 570$ bits, $\ell = 14, L_0 = 2^{20}$):** Total time $\mathbf{2^{67.64}}$, single-list size $2^{38.00}$, and message length $2^{39.25}$ words, compared to $2^{265.00}$ in [10] and $\ge 2^{204.3}$ in [9].

2. **Low-Memory Fixed-Length 2-Adic Cycle Winding:** Adapting fixed-length cycle winding [9] across our 8 shifted anchors, $r = m + 3$ equal-length slots (U_i, V_i) ($|U_i| = |V_i| \le L_0$) suffice when minimal memory is desired. Formatting slot i as $V_i^{m_i} \| U_i^{(2^w-1)-m_i}$ for $0 \le m_i < 2^w$ keeps the total byte length invariant for all $(m_1, \ldots, m_r)$ while contributing $m_i(\Delta\mathsf{Sum}(V_i) \boxminus \Delta\mathsf{Sum}(U_i))$ and $m_i\delta_i$ modulo 2^w. Solving the resulting $(m + 1) \times (m + 3)$

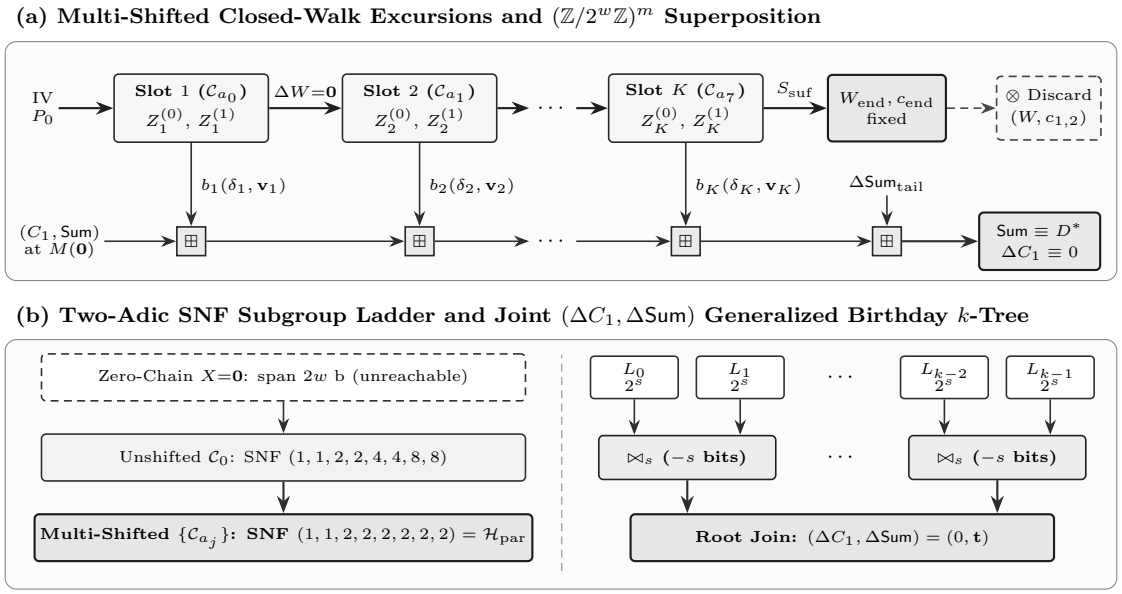


Figure 4: Multi-shifted closed-walk preimage architecture over $(\mathbb{Z}/2^w\mathbb{Z})^m$. **(a)** Equal-length closed-walk pairs $(Z_j^{(0)}, Z_j^{(1)})$ across 8 shifted clean subspaces re-merge on (W, c_1, c_2) at every anchor so $(W_{\text{end}}, c_{\text{end}})$ and the post-message padding and checksum rounds add a fixed constant $\Delta\mathsf{Sum}_{\text{tail}}$ before (W, c_1, c_2) is discarded. **(b)** Odd 2-adic units $-(2a_j+1)$ collapse the SNF defect to $\mathcal{H}_{\text{par}}$, and a joint $(\Delta C_1, \Delta\mathsf{Sum})$ Wagner k-tree hits D^*.

2-adic linear system over $\mathbb{Z}/2^w\mathbb{Z}$ in $O(1)$ arithmetic and hashing the single fixed-length winning message requires time $\mathbf{2^{45.46}}$ with 2^{18} memory on Blender-256, and time $2^{89.17}$ and $2^{89.46}$ with 2^{34} memory on Blender-384 and Blender-512.

6 Architectural Diagnosis and Experimental Evaluation

6.1 Diagnosis of Structural Weaknesses

Our three attacks trace back to three interacting architectural flaws in Blender's ARX-plus-accumulator design, each pointing to a concrete design principle:

1. **Uniform Stride-t Mask Broadcasting Across m Lanes:** Injecting only $t = 2$ masks alternately across $m \in \{6, 8\}$ words creates $m - t$ linear mask-free XOR chains $X_i = W_i \oplus W_{i+t}$ that close periodically whenever $t \mid m$ and $W_0 \in \{x, \neg x\}$ (Theorem 1). Breaking this symmetry requires either distinct round-dependent rotations per lane or coprime tap schedules coupled with wide non-linear state diffusion.

2. **Rotation-Sum Compensation Across Discrete Carry-Selected Rotations:** Coupling (c_1', c_2') to $r = 8 - (c_1' + c_2') \in \{6, 7, 8\}$ allows a 2-bit carry jump $(0, 0) \leftrightarrow (1, 1)$ at step p to be canceled inside the pre-rotation sum ($\text{rol}_2(u)$ and $\text{ror}_2(v)$) with probability $(7/48)^2 \approx 2^{-5.56}$. That carry difference is then healed deterministically at step $p + 1$ via $(2^{w-1}, 2^{w-1} - 1)$ complement inputs (Theorem 4). Mixing carries back into multiple non-aligned bit positions and applying at least 4 rounds of feed-forward state mixing per absorbed message word prevents 2-step local healing.

3. **Unidirectional Accumulator Without Working-State Feed-Forward at Output:** Updating Sum without feeding Sum back into W, and discarding (W, c_1, c_2) immediately after the checksum rounds, makes finalization an invariant affine translation (Proposition 7).

Table 2: First- and second-preimage complexity comparison across all four NIST Blender variants: Blender-224 (w=32, m=7), Blender-256 (w=32, m=8), Blender-384 (w=64, m=6), and Blender-512 (w=64, m=8).

Variant	Method	Time	Memory	Words	Mechanism
Blender-224	Newbold [13]	$2^{135.81}$	$O(n^2)$	$O(n^2)$	Joux + checksum birthday
Blender-224	Lamberger–Mendel [11, 10]	$\mathbf{2^{119.81}}$	$O(n)$[a]	1,121	5-word sum-fixed Joux
Blender-256	Newbold [13]	$2^{152.00}$	$O(n^2)$	$O(n^2)$	Joux + checksum birthday
Blender-256	Lamberger–Mendel [11, 10]	$2^{136.00}$	$O(n)$[a]	1,281	5-word sum-fixed Joux
Blender-256	Klíma (single) [9]	$\geq 2^{107.32}$	$2^{65.00}$	$2^{107.32}$	Odd-lane fixpoint winding
Blender-256	Klíma (2^{2n}-multi) [9]	$\geq 2^{171.30}$	$2^{65.00}$	$2^{171.30}$	Odd-lane fixpoint winding
Blender-256	Ours (Winding)	$2^{45.46}$	$2^{18.00}$	$2^{45.46}$	8-shifted $\mathcal{H}_{\text{par}}$ + 2-adic solve
Blender-256	**Ours (k-Tree)**	$\mathbf{2^{35.93}}$	$2^{21.69}$[b]	$2^{27.46}$	8-shifted $\mathcal{H}_{\text{par}}$ + Wagner 2^{12}
Blender-384	Newbold [13]	$2^{232.58}$	$O(n^2)$	$O(n^2)$	Joux + checksum birthday
Blender-384	Lamberger–Mendel [11, 10]	$2^{200.58}$	$O(n)$[a]	1,921	5-word sum-fixed Joux
Blender-384	Klíma (single) [9]	$\geq 2^{171.90}$	$2^{97.00}$	$2^{171.90}$	Odd-lane fixpoint winding
Blender-384	Klíma (2^{2n}-multi) [9]	$\geq 2^{267.30}$	$2^{97.00}$	$2^{267.30}$	Odd-lane fixpoint winding
Blender-384	Ours (Winding)	$2^{89.17}$	$2^{34.00}$	$2^{89.17}$	8-shifted $\mathcal{H}_{\text{par}}$ + 2-adic solve
Blender-384	**Ours (k-Tree)**	$\mathbf{2^{66.34}}$	$2^{29.60}$[b]	$2^{38.91}$	8-shifted $\mathcal{H}_{\text{par}}$ + Wagner 2^{14}
Blender-512	Newbold [13]	$2^{297.00}$	$O(n^2)$	$O(n^2)$	Joux + checksum birthday
Blender-512	Lamberger–Mendel [11, 10]	$2^{265.00}$	$O(n)$[a]	2,561	5-word sum-fixed Joux
Blender-512	Klíma (single) [9]	$\geq 2^{204.32}$	$2^{129.00}$	$2^{204.32}$	Odd-lane fixpoint winding
Blender-512	Klíma (2^{2n}-multi) [9]	$\geq 2^{331.30}$	$2^{129.00}$	$2^{331.30}$	Odd-lane fixpoint winding
Blender-512	Ours (Winding)	$2^{89.46}$	$2^{34.00}$	$2^{89.46}$	8-shifted $\mathcal{H}_{\text{par}}$ + 2-adic solve
Blender-512	**Ours (k-Tree)**	$\mathbf{2^{67.64}}$	$2^{38.00}$[b]	$2^{39.25}$	8-shifted $\mathcal{H}_{\text{par}}$ + Wagner 2^{14}

[a]Memoryless MITM [14] ($2^{n/2}$ with table lookup). [b]Single-list size $2^s = 2^{B/(\ell+1)}$.

Every closed walk on (W, c_1, c_2) then acts as an exact abelian translation in $(\mathbb{Z}/2^w\mathbb{Z})^m$. Bijectively mixing Sum and W during finalization and absorbing step counters into every compression round eliminates commuting closed walks and forces non-linear finalization inversion.

6.2 Experimental Verification and Benchmarks

All solvers and witnesses were implemented in portable C (POSIX threads) and Python 3 without external SAT/SMT or third-party library dependencies. Every output witness was verified byte-for-byte against Bradbury's unmodified NIST SHA-3 Round-1 C reference implementation [1] on an Apple Silicon workstation:

1. **Subspace Cycle-Swap Standard-IV Collisions and Multicollisions:** Our 8-thread overlapping ancestor cycle-swap solver finds verified 16,000-byte (4,000-word) standard-IV Blender-256 collisions in 0.045 s on a fast seed and 0.731 s mean (0.774 s median, 0.181 s min, 1.422 s max) across 10 independent cold seeds, exploring a mean of $2^{22.6}$ clean DFS nodes. From a single (P, U, V) witness, all $\binom{6}{3} = 20$ balanced 40,896-byte multicollisions were hashed through the official C reference, all producing the identical digest `8b07ac88b949ef7d3e946af12c79c651c2b7b33d7f1dd09cb7ea0c3836b3b71b`. Empirical out-degree audits across 2,000 uniform clean states per variant measured 2.016, 2.006, and 2.010 on Blender-256, −384, and −512, matching Proposition 3 (2.000).

2. **All-Variant Closed-Form SFS (80,000/80,000 Verified):** Across 20,000 random seeds per variant (80,000 pairs in total), our deterministic closed-form SFS generator (Theorem 5) succeeded on 100% of seeds with zero retries. It generated pairs in 0.027 μs (Blender-224, 113 B), 0.006 μs (Blender-256, 113 B), 0.054 μs (Blender-384, 229 B), and 0.014 μs (Blender-512, 229 B), all verified against the official C reference.

Table 3: Verified minimal-length standard-IV (24 B) and identical-prefix (36 B and 40 B, ASCII prefix "`Dear Bob, wire $`") colliding pairs for full Blender-256 on the official NIST C reference [1].

Item	Verified Hexadecimal Value
24-Byte (6-Word) Standard-IV Collision Pair A	
M_1	`66b2d21e26d2d06c6eb3e7e2f5e97fe5e367af98dfff7fc1`
M_2	`66b2d21e26d2d06c6eb3e7e2f5e97fe5a267af1b2000803e`
Digest	`3dab8fdaf25bc778e60ea5c22b332f57875003724fb29615aedbf749fde750ee`
24-Byte (6-Word) Standard-IV Collision Pair B	
M_1	`2d0f9baed58c8332feaed2c6274dfa3eccca0d99b31c8466`
M_2	`2d0f9baed58c8332feaed2c6274dfa3e330416664ce37b99`
Digest	`4dbdaf9d59fbdac4f0721cfd96a6eaa6cf7da4b0e72ef6d7516c931233c03ff5`
36-Byte (9-Word) Identical-Prefix Collision Pair[a]	
M_1	`4465617220426f622c20776972652024b3b0c1861f200be53811121852bbf00efd219c2`
M_2	`4465617220426f622c20776972652024b3b0c1861f200be5381112164d1f284102de63d`
Digest	`6b6c36da8ad5f932d384394af95eadf64de470977245367e3fa168ac835fd2b2`
40-Byte (10-Word) Identical-Prefix Collision Pair[a]	
M_1	`4465617220426f622c20776972652024b89c419ad38fd20e65c9a79c84bd45f046bb10bae0fef7bf`
M_2	`4465617220426f622c20776972652024b89c419ad38fd20e65c9a79c84bd45f007b9003a1f010840`
Digest	`8d5ee26f516017afe572029ec9fa75ee9b242a9a27cc73a2e4cec873578da3f8`

[a] Both pairs start with the 16-byte ASCII prefix "`Dear Bob, wire $`" (`4465617220426f622c20776972652024`).

3. **Global-Minimum 24-Byte Standard-IV and 36/40-Byte Identical-Prefix Collisions:** Table 3 lists our verified short colliding message pairs on full Blender-256. Our pure-C 4-round lane-invariant and 2-round byte-lattice solver (0.11–0.29 ms per 2-round inversion) solved 10/10 cold seeds for 24-byte standard-IV collisions in 34.58 s mean on 10 threads (1.63 s min; 0.057 s single-thread best). It also solved a 40-byte identical-prefix collision for the 16-byte ASCII prefix "`Dear Bob, wire $`" in 145.5 s on 10 threads (42 two-round targets), alongside a 36-byte (3-round steered) identical-prefix witness for the same prefix.

4. **Verification of 2-Adic SNF Ranks and Closed-Walk Modular Superposition:** Across 500 independent closed-walk slots per configuration on the official C reference, empirical 2-adic Smith Normal Form decomposition confirmed the exact subgroup invariant factors $(1, 1, 2, 2, 4, 4, 8, 8)$ (index 2^{12}) in the unshifted subspace $\mathcal{C}_0$ and $(1, 1, 2, 2, 2, 2, 2, 2)$ (index 2^6, matching $\mathcal{H}_{\text{par}}$) across 4 and 8 shifted subspaces $\mathcal{C}_{a_j}$ (Theorem 8). We also confirmed 100% exact modular superposition and post-message finalization invariance (Proposition 7) across end-to-end 8-anchor closed-walk families satisfying $\Delta C_1 \equiv 0 \pmod{2^{32}}$ on the unmodified NIST C reference.

Use of AI. The cryptanalytic results and artifacts were initially developed through an autonomous agent system powered by a MiniMax internal model. We validated the key technical claims and refined the presentation of the manuscript.

References

[1] C. Bradbury, "BLENDER: A Proposed New Family of Cryptographic Hash Algorithms," Algorithm Specification and Supporting Documentation, Submission to the NIST SHA-3 Competition (Round 1), Oct. 2008.

[2] F. Chabaud and A. Joux, "Differential Collisions in SHA-0," in *Advances in Cryptology – CRYPTO'98*, LNCS, vol. 1462, pp. 56–71, Springer, 1998.

[3] J. Daemen and C. Clapp, "Fast Hashing and Stream Encryption with PANAMA," in *Fast Software Encryption – FSE'98*, LNCS, vol. 1372, pp. 60–74, Springer, 1998.

[4] P. Gauravaram and J. Kelsey, "Linear-XOR and Additive Checksums Don't Protect Damgård–Merkle Hashes from Generic Attacks," in *Topics in Cryptology – CT-RSA 2008*, LNCS, vol. 4964, pp. 36–51, Springer, 2008.

[5] B. Hattersley, "Waterfall Hash — Algorithm Specification and Analysis," Version 1.0, Submission to the NIST SHA-3 Competition (Round 1), Oct. 2008.

[6] A. Joux, "Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions," in *Advances in Cryptology – CRYPTO 2004*, LNCS, vol. 3152, pp. 306–316, Springer, 2004.

[7] J. Kelsey and T. Kohno, "Herding Hash Functions and the Nostradamus Attack," in *Advances in Cryptology – EUROCRYPT 2006*, LNCS, vol. 4004, pp. 183–200, Springer, 2006.

[8] V. Klíma, "A Near-Collision Attack on BLENDER-256," Public Comment to the NIST SHA-3 Competition (Round 1), Dec. 2008.

[9] V. Klíma, "Huge Multicollisions and Multipreimages of Hash Functions BLENDER-n," Cryptology ePrint Archive, Paper 2009/006, Jan. 2009.

[10] M. Lamberger and F. Mendel, "Structural Attacks on Two SHA-3 Candidates: Blender-n and DCH-n," in *Information Security – ISC 2009*, LNCS, vol. 5735, pp. 68–78, Springer, 2009.

[11] F. Mendel, "Preimage Attack on Blender," Unpublished note, ECRYPT SHA-3 Zoo, Dec. 2008. `https://ehash.isec.tugraz.at/uploads/4/48/Blender-preimage.pdf`.

[12] F. Mendel, N. Pramstaller, C. Rechberger, M. Kontak, and J. Szmidt, "Cryptanalysis of the GOST Hash Function," in *Advances in Cryptology – CRYPTO 2008*, LNCS, vol. 5157, pp. 162–178, Springer, 2008.

[13] C. Newbold, "Observations and Attacks On The SHA-3 Candidate Blender," Unpublished note, ECRYPT SHA-3 Zoo, Dec. 2008. `https://ehash.isec.tugraz.at/uploads/2/20/Observations_on_Blender.pdf`.

[14] J.-J. Quisquater and J.-P. Delescaille, "How Easy is Collision Search? New Results and Applications to DES," in *Advances in Cryptology – CRYPTO'89*, LNCS, vol. 435, pp. 408–413, Springer, 1990.

[15] A. Regenscheid, R. Perlner, S.-j. Chang, J. Kelsey, M. Nandi, and S. Paul, "Status Report on the First Round of the SHA-3 Cryptographic Hash Algorithm Competition," NIST Interagency Report (NISTIR) 7620, National Institute of Standards and Technology, Sept. 2009.

[16] M. Stevens, A. K. Lenstra, and B. de Weger, "Chosen-Prefix Collisions for MD5 and Colliding X.509 Certificates for Different Identities," in *Advances in Cryptology – EUROCRYPT 2007*, LNCS, vol. 4515, pp. 1–22, Springer, 2007.

[17] D. Wagner, "A Generalized Birthday Problem," in *Advances in Cryptology – CRYPTO 2002*, LNCS, vol. 2442, pp. 288–304, Springer, 2002.

[18] X. Wang and H. Yu, "How to Break MD5 and Other Hash Functions," in *Advances in Cryptology – EUROCRYPT 2005*, LNCS, vol. 3494, pp. 19–35, Springer, 2005.

[19] L. Xu and J. Li, "Semi-free start collision attack on Blender," Cryptology ePrint Archive, Paper 2008/532, Dec. 2008.