Round-Key Equivalence and Collisions of the Revised MoFang

Zhang zhihong, Liu fengyaxuan*
Beijing Electronic Science and Technology Institute, Beijing 100070

Abstract

The revised reference implementation of MoFang moves the S-box of the second message half to row 1, which does not eliminate the original per-round key cancellation. Collisions of this type on the original version have been independently reported[2,3]; this paper does not repeat those findings, and instead proves for the revised version that when the nonzero words of that half are confined to three designated positions, the full message expansion preserves the per-round cancellation, so that every input has 16 round-key-equivalent inputs. The property holds for any common chaining value and transfers to the complete behavior of the six reference variants under full padding, fixed initialization, and XOF output interfaces. Furthermore, the sufficient condition is promoted to a complete classification of the round-key map: the common difference of any two equivalent inputs must belong to the original 16 period-two masks, and the equivalence class sizes can only be 1, 2, 4, or 16, with size 16 corresponding exactly to the original domain D_0. For the DM variants, necessary and sufficient conditions for all nontrivial equivalence classes are given, together with an exact integer count of the covered domain ($\log_2 \approx 814.04647$); the domain covered by equivalence classes of size at least four admits a three-state matrix counting formula. These results characterize the equivalence of complete round-key sequences; the domain sizes do not indicate running time or security strength. The object of study is fixed to the 64-bit word model corresponding to the public submission; mathematical proofs, computer-assisted counting, and experimental verification against the reference C code are stated separately.

Keywords: MoFang; message expansion; round-key equivalence classes; collision; necessary and sufficient condition; exact counting

1 Introduction

MoFang[1] is one of the first-round hash candidates in the Next Generation Commercial Cryptography (NGCC) program run by the Institute of Commercial Cryptography Standards of China. This paper studies one concrete question: after the revised MoFang moves the nonlinear layer of the second message branch to a new position, which distinct inputs still produce exactly the same combined round-key sequence? This question arises before the final digest comparison. If all round keys are equal, then for any common input state, the per-round states and the complete compression outputs of the two executions are equal; conversely, equal final outputs do not require equal round-key sequences. Since equal round-key sequences imply equal complete compression outputs, this equivalence relation corresponds directly to a family of colliding messages of the revised MoFang, and is directly relevant to the security evaluation of the NGCC candidates. This paper characterizes only round-key equivalence, a sufficient condition for equal outputs, and does not involve differential attacks based on probability estimates; the domain sizes given from Section 5 onward describe the scope of the equivalence relation and do not indicate running time or security strength.

The work most closely related to ours is the common-word-position orbit analysis of the original MoFang[2]. In the original version, both message branches apply the S-box to row

*Corresponding author: Li xiaodong, E-mail: 1984637692@qq.com

0; that paper analyzes, around this structure, the orbits of common word positions, period-two masks, and per-round injection cancellation, where the relevant differences never enter the nonlinear layer and the conclusions do not depend on the base message. The idea of transferring round-key equivalence to fixed initialization, multi-block messages, and outputs of finite-length extendable output functions (XOFs) also appears there. A public report[3] independently records the per-round key-cancellation collisions of the original version, but only records the revised scheme without evaluating its security. The errata and the reference code then move the S-box of the second branch to row 1[4,5]; from then on, the original difference trail touches the S-box, and whether the cancellation continues depends on the actual input values.

As background theory, Hirose double-block-length compression[6] and the analysis framework based on commutating transformations and invariant subspaces[7] are, respectively, the definitional source of the DBL variants and the origin of our classification tool. To the best of our knowledge, at the time of writing there is no other public, independent security analysis of the revised version.

We keep this difference mechanism and concentrate the new analysis on the base-value conditions of the revised implementation, the domain of applicability, and the complete round-key equivalence classes, with the goal of promoting the sufficient condition to a classification of the complete round-key map $\Phi(A, B) = (K_0, \ldots, K_{16})$. The framework of commutating transformations and invariant subspaces on which the classification relies is due to Leander et al.[7]; the exact counting uses traces of transition matrices and Boolean-constraint compilation with arbitrary-precision integer elimination (Appendices B and C). The object of study is fixed to the 64-bit word model of the six reference implementations in the pinned commit `bfdc6a89ffda8e2d83f6f6354b40c69aa707b8c8` of the public repository[8]; rotations are interpreted according to the standard definition of rotation on 64-bit words. Mathematical proofs, computer-assisted counting, and experimental verification against the reference C code are stated separately.

The revised design document[5] has one assignment target that differs from the reference C code, and the original C's zero-rotation expression contains an undefined shift; for these reasons, this paper does not merge the two inconsistent algorithms into a single specification, nor does it generalize the outcome of a single compilation to all C implementations; see Section 6.2 for the specific differences.

Our contributions are as follows. Sections 2 and 3 build the minimal restricted domain D_0 and the sixteen-fold round-key equivalence on it, and transfer the collision conclusion to the behavior of the six reference variants under full padding, fixed initialization, and the XOF output interfaces. Sections 4 and 5 drop the six-zero-word restriction, characterize the necessary and sufficient conditions for complete round-key equivalence classes, and give all nontrivial equivalence classes of the DM class as well as exact counts of the domains of size at least four for both the DM and DBL classes. Section 6 reports the specification-versus-code differences and the verification against official vectors; Section 7 concludes. Table 1 summarizes the main quantitative results. The detailed finite-round bounds and the combinatorial counting are deferred to the appendices; the main text keeps only the definitions and proof outlines needed to understand the conclusions.

2 Message Expansion and the Restricted Domain

This section gives the exact model of the message expansion and defines the original domain D_0 used in the subsequent analysis.

Table 1: Overview of results in the fixed 16-round model

Result	covered domain size	class size	applies to
original domain D_0 (six-zero-word condition)	2^{768}	16	DM, DBL
complete domain of size at least 4	$\approx 2^{790.38966}$	4 or 16	DM
complete domain for a fixed single direction	$\approx 2^{812.04647}$	at least 2	DM
all nontrivial equivalence classes	$\approx 2^{814.04647}$	2, 4, 16	DM
complete domain of size at least 4	$\approx 2^{769.16886}$	4 or 16	DBL

Note: all inputs are complete 1152-bit original blocks; "covered domain size" counts inputs, and the $\log_2$ values are approximate; the exact integers appear in the main text. 2^{768} is the cardinality u_0 of the original domain D_0 defined in Section 2.2; $\approx 2^{790.38966}$, $\approx 2^{812.04647}$, $\approx 2^{814.04647}$, $\approx 2^{769.16886}$ correspond to Eqs. (36), (10), (12), and (41), respectively, where Eqs. (36) and (41) are in Appendix B.

2.1 Exact definition of the message expansion

A message block consists of 144 bytes, read as 18 little-endian 64-bit words and split into two nine-word halves $A, B \in V = (\mathbb{F}_2^{64})^9$. Word coordinates are numbered $0, \dots, 8$ in row-major order of a 3×3 matrix; coordinates 5 and 7 are the matrix positions $(1, 2)$ and $(2, 1)$. Write $\rho_s(x) = \mathrm{rol64}(x, s)$ for the operation that moves bit k to bit $(k + s) \bmod 64$. The symbols $\oplus$, $\wedge$, $\vee$ denote bitwise XOR, AND, and OR, and $\neg x$ denotes the bitwise complement within a 64-bit word.

The three-word S-box of the reference implementation uses the sequential assignment

$$p = x \oplus (y \wedge z), \qquad q = y \oplus (p \wedge z), \qquad s = z \oplus (p \vee q). \tag{1}$$

Hence $S(x, y, z) = (p, q, s)$. After simplification it can also be written as

$$S(x, y, z) = \big(x \oplus yz,\ y \oplus (x \oplus y)z,\ z \oplus (x \vee y)\big),$$

where the bitwise AND is abbreviated as multiplication only in Boolean polynomials of this kind. Let S_0 and S_1 apply S to rows 0 and 1 of a nine-word vector, respectively.

The two word permutations are defined by "the input position read at each output position":

$$P_0 = (0, 4, 8, 2, 3, 7, 1, 5, 6), \qquad P_1 = (0, 6, 3, 4, 1, 7, 8, 5, 2).$$

Let R_b apply the following rotation amounts at each output position:

$$R_0 = (1, 5, 11, 17, 0, 31, 41, 47, 59), \qquad R_1 = (3, 7, 19, 43, 53, 61, 0, 29, 37).$$

The message expansion of the DM class is

$$A_{r+1} = R_0 P_0 S_0(A_r), \qquad B_{r+1} = R_1 P_1 S_1(B_r) \oplus I(c_r), \qquad 0 \le r < 16, \tag{2}$$

where $I(c)$ places the word value c only at coordinate 0, and the round constants are the same in both executions. The initial value is $(A_0, B_0) = (A, B)$, and the combined round key is $K_r = A_r \oplus B_r$. If F_r denotes the state round function of the reference implementation, then encryption and Davies–Meyer compression satisfy

$$X_0 = H \oplus K_0, \qquad X_{r+1} = F_r(X_r) \oplus K_{r+1}, \qquad C_{\mathrm{DM}}(H, A, B) = X_{16} \oplus H. \tag{3}$$

Together with the initial injection, 16 full rounds use $K_0, \dots, K_{16}$ in total. The state path reads the message through these combined keys, while the two halves are still updated separately in the expansion. Hence $A \oplus B = A' \oplus B'$ alone does not guarantee equal subsequent round keys. The proof only needs determinism of F; its concrete definition is given in Appendix D.

2.2 The original domain D_0

Define

$$U = \{(u, 0, 0, 0, 0, v, 0, w, 0) : u, v, w \in \mathbb{F}_2^{64}\}, \qquad D_0 = V \times U. \tag{4}$$

The six zero words here are actual values of the second half, not merely zero differences between two executions. By Eq. (1), $S(0, 0, z) = (0, 0, z)$; P_1 fixes coordinate 0 and swaps coordinates 5 and 7, so the complete update of the second branch preserves U, and its three free words update as

$$(u, v, w) \mapsto \big(\rho_3(u) \oplus c_r,\ \rho_{61}(w),\ \rho_{29}(v)\big).$$

What the update preserves is the set and the zero-word pattern, not each base value itself. This domain consists of the six zero words and the three free words of the second half; below we call it the original domain D_0 (also the six-zero-word domain).

Write

$$E = \texttt{5555555555555555}, \qquad O = \texttt{aaaaaaaaaaaaaaaa}, \qquad F = \texttt{ffffffffffffffff},$$

$$P = \{0, E, O, F\}.$$

This is exactly the period-two subspace of words satisfying $\rho_2(x) = x$. Let $\sigma = \rho_1$; then σ swaps E and O, fixes 0 and F, and $\sigma^2 = \mathrm{id}$ (here F is a 64-bit mask, unrelated to the round-function notation F_r of Eq. (3)). Define the two-coordinate difference and the common translation

$$D(a, b) = (0, 0, 0, 0, 0, a, 0, b, 0), \tag{5}$$

$$T_{a,b}(A, B) = \big(A \oplus D(a, b),\ B \oplus D(a, b)\big).$$

2.3 Equivalence on the restricted domain

Property 1 (Round-key cancellation)**.** *Suppose that at some snapshot the common difference of the two message branches is $D(a, b)$ with $a, b \in P$, and that this difference is preserved by both S-boxes (for the first branch the difference avoids the input row of S_0; for the second branch the difference in the third input keeps its shape through S_1). Then after one expansion step the common differences of the two branches are both $D(\sigma b, \sigma a)$, and the combined round-key difference stays zero. In particular, when the second half lies in U, the S-box condition holds automatically.*

Theorem 2.1 (Equivalence on the restricted domain)**.** *Let $(A, B) \in D_0$. The S-box condition of Property 1 holds automatically, so the common difference $D(a, b)$ is preserved round by round; hence for all $a, b \in P$, the inputs (A, B) and $T_{a,b}(A, B)$ have identical combined round keys. The common difference of the two message branches is $D(a_r, b_r)$, where*

$$(a_r, b_r) = \begin{cases} (a, b), & r \text{ even}, \\ (\sigma(b), \sigma(a)), & r \text{ odd}. \end{cases}$$

Therefore, for any common H, the per-round states and the complete compression outputs of the two executions are equal. If $(a, b) \neq (0, 0)$, the two inputs are distinct. This conclusion does not depend on the concrete values of the S-box, the permutations, or the round constants; it uses only the determinism of the message expansion.

Proof. Both branches start with common difference $D(a, b)$, so $\Delta K_0 = 0$. Suppose the common difference at the r-th snapshot is $D(a_r, b_r)$. Row 0 of the first half is identical in the two executions, so S_0 does not change its difference. The second half and its translation both lie

in U, so S_1 preserves each of them, again leaving the difference unchanged. Both permutations swap coordinates 5 and 7, so the next differences are

$$D\big(\rho_{31}(b_r),\, \rho_{47}(a_r)\big), \qquad D\big(\rho_{61}(b_r),\, \rho_{29}(a_r)\big).$$

The four rotation amounts are odd and act as σ on P, so the two expressions are equal; the round constants cancel. Induction yields that all 17 round keys coincide; then Eq. (3) gives equal states and feed-forward outputs round by round. A nonzero (a, b) gives a nonzero $D(a, b)$, so the two inputs differ. □

Example 2.1 (A numerical instance of one-round propagation)**.** This example demonstrates word by word one step of the proof of Theorem 2.1: how the difference of the two messages keeps the form $D(a_r, b_r)$ in every round and makes the combined round-key difference $\Delta K_r = 0$. Take $(a, b) = (F, O)$, i.e., the common difference of Eq. (5)

$$D(a, b) = (0, 0, 0, 0, 0, F, 0, O, 0).$$

The two halves differ only in word 5 by F and in word 7 by O; the other 7 words have zero difference (see Figure 1 for this propagation). By Theorem 2.1, in round 0 both branch differences equal $D(a, b)$, so $\Delta K_0 = 0$. After one expansion step, the next difference of the first half (R_0P_0) is $D(\rho_{31}(b), \rho_{47}(a))$ and that of the second half (R_1P_1) is $D(\rho_{61}(b), \rho_{29}(a))$. Substituting the values (64-bit rotation to the left; the four components are word 5 of the first half, word 5 of the second half, word 7 of the first half, and word 7 of the second half):

$$\begin{aligned}
\rho_{31}(O) &= \rho_{31}(\texttt{aaaaaaaaaaaaaaaa}) = \texttt{5555555555555555} = E,\\
\rho_{61}(O) &= \rho_{61}(\texttt{aaaaaaaaaaaaaaaa}) = \texttt{5555555555555555} = E,\\
\rho_{47}(F) &= \rho_{47}(\texttt{ffffffffffffffff}) = \texttt{ffffffffffffffff} = F,\\
\rho_{29}(F) &= \rho_{29}(\texttt{ffffffffffffffff}) = \texttt{ffffffffffffffff} = F.
\end{aligned}$$

The next differences of the two branches are equal component by component, so the combined round-key difference in round 1 is still $\Delta K_1 = 0$. The reason is the rotation property of the period-two subspace P: the four rotation amounts 31, 61, 47, 29 are odd and all act as σ on P, with

$$\sigma(E) = O, \qquad \sigma(O) = E, \qquad \sigma(F) = F, \qquad \sigma(0) = 0, \qquad \sigma^2 = \mathrm{id}.$$

Thus $(a_1, b_1) = (\sigma(b), \sigma(a)) = (E, F)$, matching the parity phases of Theorem 2.1. Induction over rounds gives that all 17 round keys $K_0, \ldots, K_{16}$ coincide and the state difference is identically zero. We checked K_r and the state X_r round by round against the colliding message pair of Section 3.4; the reference implementation of the pinned commit[8] produces word-identical outputs in the two executions (100 pairs per variant, all passing; see Table 8).

Observation 1. *The induction of Theorem 2.1 does not depend on the concrete values of the round constants; as long as the expansion rule is kept and every round receives a common constant, the conclusion holds for any finite number of rounds.*

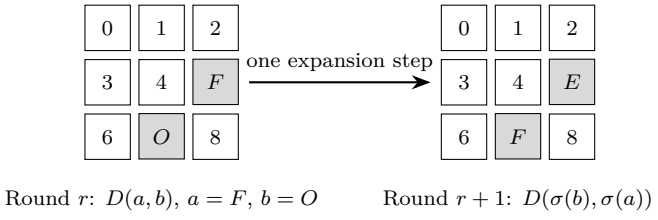


Figure 1: One-round difference propagation of Example 2.1: the masks stay in the period-two subspace P

Observation 2. *The period-two condition is also necessary on the present support: one-step cancellation requires* $\rho_{30}(b) = b$ *and* $\rho_{18}(a) = a$*, and* $\gcd(64, 30) = \gcd(64, 18) = 2$*, so exactly* $a, b \in P$ *work. This necessity is limited to the stated support and one-step propagation; Section 4 drops the restriction.*

3 Complete Message Interface and Basic Counting

This section transfers the equivalence on the restricted domain to the complete hash interface: first the padding and the fixed initialization, then the uniform treatment of the six reference variants, and finally the input counting and one complete collision instance.

3.1 Padding and fixed initialization

For a message of bit length ℓ, append a single 1 bit, then the minimal number of zeros so that the padded length reaches $1152(\lfloor \ell/1152 \rfloor + 1)$. The padding carries no length field; a message already aligned to a whole block still gets one full padding block. In particular, the common padding block of a 144-byte raw message is $Q = 80 \parallel 0^{143}$. Let $J(c)$ place the word value c only at coordinate 8. The initial state (Initialization Vector, IV) of MoFang-256 is $J(5)$; before processing the last padded block, $J(1)$ is XORed in, and the output is the little-endian sequence of the first four words of the final state.

Corollary 3.1 (Complete hash interface)**.** *If the two distinct inputs of Theorem 2.1 are serialized as 144-byte raw messages, then under the fixed initialization, the full padding, and the finalization, the two have the same complete final state and digest.*

Proof. The first raw block starts from the same initial state (IV) and, by Theorem 2.1, yields the same complete state. Afterwards both sides perform the same finalization marker and process the same padding block Q, so the final states coincide, and so do the truncated outputs. The padding block itself need not satisfy the six-zero-word condition. □

3.2 Uniform treatment of the six variants

MoFang-256, MoFang-512, and MoFang-256-XOF share Eq. (2); we call them the DM class (Davies–Meyer). The other three variants use Hirose double-block-length (DBL) compression[6], with a complete chaining value $(g, h) \in V^2$; we call them the DBL class. Their chaining-value branch satisfies $Z_0 = h$, $Z_{r+1} = P_0 Z_r$, and the message expansion is

$$A_{r+1} = R_0 P_1 S_0(A_r), \qquad B_{r+1} = R_1 P_0 S_1(B_r) \oplus I(c_r). \tag{6}$$

Eq. (6) differs from Eq. (2) only in that the two permutations are swapped; the round key is $Z_r \oplus A_r \oplus B_r$. The two executions take the same chaining value, so Z_r is always the same. After the swap the permutations still fix coordinate 0 and swap coordinates 5 and 7, so the induction of Theorem 2.1 is unchanged. If $E_{h,A,B}$ is the corresponding encryption, then

$$C_{\mathrm{DBL}}\big((g, h), A, B\big) = \Big(E_{h,A,B}(g) \oplus g,\ E_{h,A,B}\big(g \oplus J(1)\big) \oplus g \oplus J(1)\Big).$$

Both encryptions re-expand from the initial key, so the two complete output halves coincide.

Both XOF variants first output one block from the post-absorption state, then iterate

$$H_{t+1} = C_{\mathrm{DM}}\big(H_t \oplus J(1),\, m_{\mathrm{last}}\big), \qquad (g_{t+1}, h_{t+1}) = C_{\mathrm{DBL}}\big((g_t,\, h_t \oplus J(2)),\, 0,\, 0\big).$$

The former keeps the last padded message block m_{last}, while the latter uses a zero message block. As long as the raw message changes only within complete raw blocks, the last padding block is the same; starting from the same absorbed state, the two recursions keep equal states step by step, and the outputs agree for any common finite output length. Hence Corollary 3.1 holds for each of the six variants.

The concrete interface parameters of the six variants are collected in Table 2.

Table 2: Initialization, last-block marker, and output of the six reference variants

Variant	initial chaining value	update before the last block	output block
MoFang-256	$J(5)$	$H \leftarrow H \oplus J(1)$	$\mathrm{out}_4(H)$
MoFang-512	$J(6)$	same as above	$\mathrm{out}_8(H)$
MoFang-256-XOF	$J(7)$	same as above	$\mathrm{out}_4(H)$
MoFang-768	$(J(10), 0)$	$g \leftarrow g \oplus J(2)$	$\mathrm{out}_6(g) \parallel \mathrm{out}_6(h)$
MoFang-1024	$(J(11), 0)$	same as above	$\mathrm{out}_8(g) \parallel \mathrm{out}_8(h)$
MoFang-768-XOF	$(J(15), 0)$	same as above	$\mathrm{out}_6(g) \parallel \mathrm{out}_6(h)$

Note: out_j denotes the little-endian output of the first j words, and $\parallel$ denotes concatenation.

3.3 Inputs, equivalence classes, and multi-block composition

The first half has 576 free bits and the second half 192, so $|D_0| = 2^{768}$. The 16 translations $T_{a,b}$ form a free group action: different masks give different inputs; every nonzero translation is fixed-point-free and squares to the identity. The original domain therefore contains 2^{764} orbits of size 16; each fixed nonzero mask gives 2^{767} unordered equivalent pairs, and all nonzero masks together give $15 \cdot 2^{767}$ distinct unordered pairs.

Observation 3. *The counts above are numbers of inputs, orbits, and unordered pairs, not numbers of distinct digests; different orbits may still share a digest.*

If a message contains q non-overlapping complete raw blocks aligned to block boundaries that lie in D_0, each block can independently adopt one of the 16 translations, giving 16^q distinct messages of equal length. The construction requires the changes to stay within complete raw blocks: once a difference leaves a complete raw block, or the two messages have different lengths, the common padding block is no longer the same and the equivalence is lost. Inducting block by block from the same IV shows that a changed block stays compression-equivalent for any common chaining value while unchanged blocks perform the same computation; the final states and outputs therefore coincide. The common prefix, suffix, and padding block need not lie in D_0. The basic construction requires no search; for fixed word size and single-block length its algebraic cost is constant, while writing out many long messages still costs proportionally to the total output length.

3.4 A complete collision instance

Take a base message M in the restricted domain $D_0 = V \times U$: the first half is the fixed increasing sequence of byte values `0x00` through `0x47`; the three free words of the second half are $u =$ `0706050403020100`, $v =$ `4f4e4d4c4b4a4948`, $w =$ `5756555453525150`, and the other six words are zero. Let $(a, b) = (F, O)$ and $M' = T_{F,O}(M)$. Both messages are 144 bytes = 1152 bits, exactly one complete raw block; the common padding block $Q = 80 \parallel 0^{143}$ is appended as the next block by the padding rule, which is precisely the situation of Corollary 3.1. Table 3 gives the message pair (18 little-endian 64-bit words), and Table 4 gives the common digests of the six variants.

All digests above were verified pairwise against the reference implementation of the pinned commit[8] (100 pairs × 6 variants, all passing; see Table 8 of Section 6.1). It is worth noting that the complete internal states of M and M' are also equal (576 bits for the DM class, 2×576 bits for the DBL class), not merely the truncated digests; hence the collision holds for all six variants, and for the XOFs it holds for any common output length as well.

Table 3: A colliding message pair (MoFang, 18 little-endian 64-bit words, 3 words per row)

message	first half A (words 0–8)		
M	0706050403020100	0f0e0d0c0b0a0908	1716151413121110
	1f1e1d1c1b1a1918	2726252423222120	2f2e2d2c2b2a2928
	3736353433323130	3f3e3d3c3b3a3938	4746454443424140
M'	0706050403020100	0f0e0d0c0b0a0908	1716151413121110
	1f1e1d1c1b1a1918	2726252423222120	d0d1d2d3d4d5d6d7
	3736353433323130	9594979691909392	4746454443424140

message	second half B (words 9–17)		
M	0706050403020100	0000000000000000	0000000000000000
	0000000000000000	0000000000000000	4f4e4d4c4b4a4948
	0000000000000000	5756555453525150	0000000000000000
M'	0706050403020100	0000000000000000	0000000000000000
	0000000000000000	0000000000000000	b0b1b2b3b4b5b6b7
	0000000000000000	fdfcfffef9f8fbfa	0000000000000000

Note: $M' = M \oplus \Delta$, where Δ affects only words 5 and 7 of both halves: word 5 (bytes 40--47 of the first half, bytes 112--119 of the second half) is XORed bytewise with 0xff, and word 7 (bytes 56--63 of the first half, bytes 128--135 of the second half) with 0xaa. For example, word 5 of the first half changes from 2f2e2d2c2b2a2928 to d0d1d2d3d4d5d6d7, and word 7 from 3f3e3d3c3b3a3938 to 9594979691909392.

Table 4: Common digests $D(M) = D(M')$ of the example message pair for the six variants

variant	common digest
MoFang-256	37db0203500e4ec4f69738284aaa8f9862bbfdc71385e8a4ac055d9d44df70b3
MoFang-512	1ed999c855544e051fd553389ce5d2b213d895e4e7a36ff89861e419fd3aefa4e d50cba1c117d0eaa31b20e2019df17450b87740df299dc6f6cc9258aeae5769
MoFang-768	6c7f1c47417ebcd4ee8dc84583d593492c5f71c94855c98cb73acb1fcbc67d74 8e73cc7fe201afd20c4206f289370ac1387fb81f6c4bbc4361a142187617d06e 1ca7c2031dcf7c6449387615b86df393d4fd2fecd4104a355d3d64d2d2fb6e08
MoFang-1024	c6871e1c313baf75ed7604bbacd9df0edd074b9996225359d87b78b80b61448b 5a72abfc09204a82d6645ac2c5a29bdb469357b02ae0f48d0bcdd6aa1b706981 40f6f2f948be63a939a8697b1d1f8654751f5c453e1cf5f71e1dd390f4097130 2f72c98459b4db54c0b97001e38c671052d5f57c5d3de70141bfa4a353e43fe6
MoFang-256-XOF (256-bit output)	62ebdc11977e089aa956521e764c30d78e375926caae643faa8f346dc3b811d3
MoFang-768-XOF (768-bit output)	c84425fa4772993671bebd2864c31c43e2e3e23f6d7269d8e45861e002069d37 9582b18974178d9e3741eb8463c724b89a96b11bc3d8ddef05acc6a321284a81 17b617db521dd395b0617c24b90ac5153ad4e2b32e7c8e6e374576f892528edd

4 Characterizing the Complete Round-Key Equivalence Classes

In this section we no longer assume $B \in U$. The analysis framework based on commutating transformations and invariant subspaces is due to[7]. Define the round-key map

$$\Phi(A, B) = (K_0, K_1, \ldots, K_{16}).$$

For the DBL class, after fixing the same chaining-value branch, the same definition applies; that common branch cancels in the comparison. Inputs with the same Φ-value are called round-key equivalent, and their equivalence class is a preimage set of Φ. This relation guarantees equal per-round states, but does not include all cases in which different round-key sequences happen to produce the same final output.

4.1 Global difference restriction

Theorem 4.1 (Global difference restriction)**.** *In the 16-round model of the DM or DBL class, if $\Phi(A, B) = \Phi(A', B')$, then there are unique $a, b \in P$ such that*

$$A \oplus A' = B \oplus B' = D(a, b).$$

Hence any round-key equivalence class contains at most 16 inputs.

The proof is in Appendix A. The main line of argument is: first restrict an arbitrary common difference using coordinates that do not pass through an S-box, then exclude the extra positions with the exact difference formula of the S-box. Three expansion steps already restrict the difference to a 70-dimensional space; five steps exclude two of its full-word directions; finally coordinate 0 is excluded. The argument applies for at least 8 expansion steps in the DM class and at least 15 in the DBL class, hence covers the actual 16 steps. The round counts stated are sufficient; we do not claim their minimality.

4.2 Exact equivalence-class test

A direct computation from Eq. (1) shows that when the third input word carries difference d,

$$S(x, y, z) \oplus S(x, y, z \oplus d) = \big(y \wedge d,\ (x \oplus y) \wedge d,\ d\big). \tag{7}$$

Hence the output difference remains $(0, 0, d)$ if and only if $x \wedge d = y \wedge d = 0$. On the expansion trajectory of the base value B, write $x_r = B_{r,3}$, $y_r = B_{r,4}$, and define

$$L_e = \bigvee_{r=0,2,\ldots,14} (x_r \vee y_r), \qquad L_o = \bigvee_{r=1,3,\ldots,15} (x_r \vee y_r).$$

Theorem 4.2 (Exact equivalence-class test)**.** *Let*

$$\Gamma_B = \{(a, b) \in P^2 : a \wedge L_e = 0,\ \sigma(b) \wedge L_o = 0\}. \tag{8}$$

Then the common differences for which Property 1 holds for (A, B) are exactly $(a, b) \in \Gamma_B$, i.e., the complete round-key equivalence class of (A, B) is exactly $\{T_{a,b}(A, B) : (a, b) \in \Gamma_B\}$. Here A is arbitrary, and Γ_B is a linear subspace of P^2.

Proof. Theorem 4.1 first excludes all other initial differences. For the remaining $D(a, b)$, the difference of the first message branch always avoids S_0 and propagates along the period-two trajectory of Theorem 2.1. The third-word difference of the S-box in the second branch is a at even snapshots and $\sigma(b)$ at odd ones. The per-round matching conditions of Eq. (7) are exactly Eq. (8), so the conditions are necessary and sufficient. They require certain parity bits of a and b to be zero, so the solution set is a linear subspace. □

4.3 Maximal classes and the original domain

Proposition 4.3 (Maximal classes and the original domain)**.** *If Γ_B contains an element with $a = F$ or $b = F$, then $B \in U$. Therefore*

$$|\Gamma_B| = 16 \iff B \in U; \qquad B \notin U \implies |\Gamma_B| \in \{1, 2, 4\}.$$

Proof. Write $(p_r, q_r, s_r) = S(B_{r,3}, B_{r,4}, B_{r,5})$. The two kinds of second branch satisfy the following exact delayed relations:

$$\text{DM:} \quad x_{r+1} = \rho_{43}(q_r), \quad y_{r+5} = \rho_{52}(p_r); \qquad \text{DBL:} \quad y_{r+1} = \rho_{53}(p_r), \quad x_{r+5} = \rho_{42}(q_r).$$

If $a = F$, the test condition forces $x_r = y_r = 0$ at even snapshots, hence $p_r = q_r = 0$. The two delayed relations then give two more zero words at snapshots 5 and 7, so snapshots $4, 5, 6, 7, 8$ all satisfy $x_r = y_r = 0$ consecutively. These five snapshots cover the six non-allowed positions through the word permutations, so $B \in U$. Each expansion step is invertible and maps U bijectively onto itself, hence $B \in U$. If $b = F$, the same argument runs with the odd snapshots $1, 3, 5, 7, 9$.

Γ_B is the direct product of two subspaces of parity-bit coordinates. If one projection contains two independent components, it contains F, which forces $B \in U$. Hence outside the domain each projection has dimension at most one and the total dimension is at most two; inside the domain all four components are allowed. □

Therefore, when all 16 equivalent inputs are to be preserved, the six-zero-word condition is already necessary. Enlarging the domain of applicability requires accepting smaller equivalence classes. The following sections compute this trade-off exactly, always counting the number of allowed inputs and never interpreting its logarithm as a time complexity.

5 Enlarging the Domain of Applicability and Exact Counting

This section first discusses the DM class; the counted objects are complete 1152-bit raw blocks (A, B). Let $D_{a,b}$ denote the set of all initial values for which the common difference $D(a, b)$ satisfies the equivalence relation; the D_0 of Eq. (4) is the original six-zero-word full-message domain, and set $u_0 = |D_0| = 2^{768}$.

5.1 Four-element domains and single-direction domains

Outside U, the four-element direction space of the DM class can only be $\{0, E\} \times \{0, O\}$ or $\{0, O\} \times \{0, E\}$. This restriction is proved in Appendix B, and the complete base-value conditions reduce to a three-state count on one fixed cycle. If

$$Q = \begin{pmatrix} 4 & 4 & 4 \\ 2 & 1 & 2 \\ 1 & 0 & 0 \end{pmatrix}, \qquad T = \operatorname{tr}(Q^{32}) = 506813528802536089100170 25,$$

then $|D_{E,O}| = |D_{O,E}| = 2^{704}T$. The two domains intersect exactly in D_0, so the complete domain of size at least four is $2^{705}T - 2^{768} \approx 2^{790.3896571457}$ (Eq. (36) of Appendix B.1).

When only the direction $(E, 0)$ is required, Theorem 4.2 simplifies to

$$(B_{r,3} \vee B_{r,4}) \wedge E = 0, \qquad r = 0, 2, 4, 6, 8, 10, 12, 14. \tag{9}$$

Odd input snapshots impose no additional base-value conditions, so not all conditions of the four-element domain need to be maintained. Here we count Eq. (9) directly, without prescribing the shape of the other side words.

Computation 5.1 (Complete domain for a single direction)**.** *Compiling Eq.* (9) *together with the exact message expansion into finite Boolean constraints and eliminating variables with arbitrary-precision integer arithmetic gives*

$$N = |D_{E,0}| = 2^{736} \cdot 78031276805304028734527, \qquad \log_2 N \approx 812.0464705942. \tag{10}$$

The value was cross-checked with an independent Boolean compiler and a different integer-elimination implementation.

The count writes Eq. (9) and the message expansion as Boolean constraints and eliminates variables one by one. Each valid initial assignment determines the intermediate states uniquely, so intermediate variables cause no double counting. Including the initial word $B_{0,0}$, the unconstrained remaining bits, and the arbitrary A gives the cardinality above. The full constraint size, the elimination steps, and the independent cross-checks are in Appendix C. The result uses exact integers; only the logarithm is written approximately.

Example 5.1 (A numerical instance of a binary class outside the domain)**.** Take

$$B = (0, 0, 0, 0, \texttt{28aa00aa0a28a80a}, \texttt{a8ab00aa6a69b98b}, 0, \texttt{4d5407d85145c675}, 0).$$

Its words 4, 5, 7 are nonzero, so $B \notin U$; A is arbitrary, here the fixed increasing sequence of bytes 00 through 47. By Eq. (8), the expansion trajectory gives

$$L_e = \texttt{2aaa8aaaaa2aaa0a} \text{ (odd bits only)}, \qquad L_o = \texttt{b5e2cbe54af9e3da} \text{ (both parities)},$$

hence $\Gamma_B = \{(0,0), (E,0)\}$, and the equivalence class is exactly binary: (A, B) and $T_{E,0}(A, B)$ differ by E in word 5. Round-by-round checking of $K_0, \ldots, K_{16}$ and the states $X_0, \ldots, X_{16}$ against the reference implementation of the pinned commit[8] gives full agreement; the common digests of the DM class are in Table 5.

Table 5: Common digests $D(M) = D(M')$ of the Example 5.1 message pair for the DM class

variant	common digest
`MoFang-256`	`e286f1a251b300e987862d6ac1668b1c9bce143f5f4b68e5183db7c7 82de4c9b`
`MoFang-512`	`abd46938860ab76d4d7e0633e65b76acb277a8f04871e4c8a9efdd1e5740899a 1cf455ffff52d3b21640e1bab65970f3c6c0811606d01fca961689420e5cf650`
`MoFang-256-XOF (256-bit output)`	`8e56c31506564899615cd8e3510888148a0756d3b8f935976cdf6adf3f47b1a1`

This instance lies in the single-direction domain $D_{E,0}$ of Computation 5.1: outside the domain the base values no longer have sixteen-element classes, but the single direction $(E, 0)$ still gives a binary pair of equivalent inputs. The second branch of the DBL class has a different permutation (Appendix B.2), so this instance does not apply to the DBL variants.

5.2 Intersections and union of all directions

Define the four basic allowed domains

$$A_1 = D_{E,0}, \qquad A_2 = D_{O,0}, \qquad A_3 = D_{0,E}, \qquad A_4 = D_{0,O}.$$

Since the supports of E and O are disjoint, the condition of any nonzero $D(a, b)$ is exactly the conjunction of the corresponding basic-domain conditions. Thus the allowed domains of all fifteen nonzero directions are the fifteen nonempty intersections of these four domains. All four

basic domains have cardinality N (Eq. (10)); rotating all bits by one position and a one-step bijective expansion also give their equinumerosity.

The two larger pairwise intersections are

$$A_1 \cap A_4 = D_{E,O}, \qquad A_2 \cap A_3 = D_{O,E}.$$

Let their common cardinality be M (this M is a set cardinality, unrelated to the message notation of Section 3.4). Exact elimination and the transition-matrix count of the four-element domain independently give

$$M = 2^{704} \cdot 506813528802536089100170 25, \qquad \log_2 M \approx 789.3896574083. \tag{11}$$

The remaining four pairwise intersections, all four triple intersections, and the unique quadruple intersection all equal D_0. Intersections involving a full-word direction are handled by Proposition 4.3; the two same-phase four-element directions are excluded in Appendix B.1. All intersections were also counted with exact integers, with consistent results. The fifteen nonzero directions and the cardinalities of their complete allowed domains are collected in Table 6.

Table 6: Nonzero directions and the cardinalities of their complete allowed domains

nonzero direction (a, b)	number of directions	cardinality of the complete allowed domain per direction
$(E,0),(O,0),(0,E),(0,O)$	4	N
$(E,O),(O,E)$	2	M
other nonzero directions	9	$u_0 = 2^{768}$

Let Z denote the complete domain having at least one nontrivial equivalence direction; its cardinality is

$$\begin{aligned} Z = |Z| = \Big|\bigcup_{i=1}^{4} A_i\Big| &= 4N - (2M + 4u_0) + 4u_0 - u_0 = 4N - 2M - u_0 \\ &= 2^{705} \cdot 67028351320644622187802969726515 1, \\ \log_2 Z &\approx 814.0464704851. \end{aligned} \tag{12}$$

5.3 Distribution of complete equivalence classes

Combined with the bound on complete round-key equivalence classes given in the main text, the class-size distribution over all initial values is shown in Table 7.

Table 7: Distribution of complete round-key equivalence classes of the DM class

class size	number of inputs	number of classes
1	$2^{1152} - Z$	$2^{1152} - Z$
2	$4(N - M)$	$2(N - M)$
4	$2(M - u_0)$	$(M - u_0)/2$
16	u_0	$u_0/16$

The classes of size 16 cover exactly the original domain; removing the original domain from each of the two four-element direction domains (Eq. (11)) gives $2(M - u_0)$ inputs. Subtracting these two parts from Z gives $4(N - M)$ inputs in binary classes. The number of classes in each row equals the number of inputs divided by the class size.

Observation 4. *By the distribution of Table 7, classes of size 8 do not exist.*

These cardinalities describe the scope of the complete round-key equivalence relation; they should not be read as a running-time improvement of the same magnitude, nor as the linear dimension of these nonlinear allowed sets.

The ratio of the complete domain of Eq. (12) to the original domain is

$$Z/2^{768} \approx 2^{46.04647}.$$

The scope of inputs admitting nontrivial round-key equivalence is therefore enlarged, although in general a newly added input has only two equivalent choices. The sixteen-fold property of the original domain is preserved, but one cannot demand equally many equivalent choices from all newly added inputs.

5.4 Domains of size at least four for the DBL class

The test for the DBL class is still given by Eq. (8), but the second message branch has a different permutation, so the values N, M, Z do not carry over. Appendix B.2 gives the necessary and sufficient conditions from the central snapshot and proves that each of the two four-element direction domains has

$$M_{\mathrm{DBL}} = 2^{736} \cdot 17^8$$

inputs (Eq. (40)), intersecting exactly in the original domain. Hence the complete domain of size at least four is

$$2M_{\mathrm{DBL}} - u_0 = 2^{737} \cdot 17^8 - 2^{768} \approx 2^{769.16886}$$

(Eq. (41)). This is about 2.24834 times the original domain, a much smaller enlargement than for the DM class. The count of all binary equivalence domains of the DBL class is not among the results of this paper.

6 Implementation Verification

This section reports the comparison among the mathematical model, a corrected copy, and the unmodified C code of the pinned commit, and explains the differences between the specification and the code.

6.1 Objects and results of verification

The mathematical model, a C copy with a well-defined rotation semantics, and the unmodified C of the pinned commit are compared pairwise. Except for the official-KAT row, the checks of Table 8 were performed with GCC 13.3.0 on x86-64 Linux, mainly with `-std=c99 -O2 -shared -fPIC`; the full comparison against the official Known Answer Test (KAT) KAT_2_12 was additionally built from pure source on the local machine with GCC 9.4.0 on x86-64 Linux using `-O2 -fwrapv -std=c99`. The round-by-round checks cover the actual values of A_r, B_r, K_r, X_r; the complete-interface checks additionally cover initialization, feed-forward, padding, finalization, and output. Table 8 collects the completed checks that are directly relevant to the conclusions of this paper. Random testing is used to detect model or porting errors; conclusions for arbitrary parameters follow from the proofs and the exact counts stated above.

Verification against official vectors. The Test Vector directory of the pinned commit contains four groups of official vectors: KAT_2_12, KAT_2_23, KAT_2_33, and KAT_Loop. This paper performs a full byte-by-byte comparison for KAT_2_12 (message lengths 0–4096 bits, 4097 entries per variant): the six reference implementation variants (256, 512, 768, 1024, 256-XOF, and 768-XOF) all pass 4097/4097, 24582 entries in total; the four fixed-length optimized implementation variants (256, 512, 768, 1024) likewise pass 4097/4097, 16388 entries in total. The build environment is stated at the beginning of this section; the compiler versions differ

from the environment that generated the official vectors, although the comparison is byte by byte.

KAT_2_23, KAT_2_33, and KAT_Loop generate very long messages with Msg_Seed/ Msg_Exp or loop calls and are not vectors of the single-message interface; they were not replayed this time. The complete padding, finalization, and output path of the single-message interface is covered by the interface checks of Table 8.

Table 8: Independent checks completed

item	quantity or range	what is checked
official KAT 2_12	6 reference + 4 optimized variants, 4097 entries each	byte-by-byte match against the official Test Vectors; 24582 reference and 16388 optimized entries all pass; 2_23/2_33/Loop not replayed
original-domain interfaces, six variants	100 pairs per variant	complete compression output for any common chaining value, and complete output for the prescribed IV; original and corrected rotation versions agree
four-element extended-domain interfaces	180 pairs per variant	complete compression and complete output agree across the six variants
new single-direction domain	250 base values, both parity directions	all 17 stages of MoFang-256 agree with the independent model; 250 pairs of complete output for each of the three DM variants
exact mask test	312 base values, 4992 mask comparisons	the combined round-key sequences of Eq. (8) agree with the actual C, covering class sizes 1, 2, 4, 16
big-integer counting	independent compilation and different eliminators	Boolean constraints, free bits, all 15 nonempty intersections, and integer results cross-checked

Note: unless stated otherwise, all items pass; "pair" means a comparison of two executions.

6.2 Differences between the specification and the code

The assignment target of Algorithm 3 on page 14 of the revised design document[5] writes the S-box output of row 1 into row 0; the reference C updates row 1 in place. The two propagate the relevant differences differently, so our theorems do not apply verbatim to the letter of that document. On the other hand, the document and the C code also differ in the interpretation of the S-box assignments, some rotation notation, and the padding text; this paper follows the fixed-code mathematical model of Section 1 throughout.

The rotation expression of the original C computes a 64-bit right shift when the shift amount is zero, which is undefined behavior under the shift rules of the draft C standard[9]. The mathematical model sets $\rho_0(x) = x$; the copy used for cross-validation reduces the shift amount modulo 64. In the test environment the two C implementations give identical outputs; this does not constitute a guarantee for all compilers.

7 Conclusion

This paper promotes the sufficient condition on the restricted domain to a classification of the complete round-key map Φ and enlarges its known domain of applicability. The completed classification is as follows: all nontrivial equivalence classes of the DM class are counted exactly, the class sizes can only be 1, 2, 4, or 16, and the logarithm of the domain covered by all nontrivial equivalence classes is about 814.04647; for the DBL class the classification of the domains of size at least four is completed, while the count of all binary equivalence domains remains open; the case where final digests are equal but the complete round-key sequences differ is not classified

in this paper. These remaining problems also mark the next steps: combine the round-key equivalence classification with the final digest comparison to characterize the set of inputs with different round-key sequences but equal digests; the count of all binary equivalence domains of the DBL class can also be pursued along the central-snapshot method of Appendix B.2.

A public report also points out that for short messages of at most 1151 bits, the outputs of MoFang-768 and MoFang-1024 are public byte permutations of each other[3]; this property is independent of our round-key equivalence classification and does not affect the conclusions above.

From a design perspective, merely moving the S-box does not eliminate the per-round key cancellation in the message expansion. For the evaluation of NGCC candidates, message expansions should explicitly avoid symmetries such as the closure of the period-two mask subspace under odd rotations; the complete classification of this paper also provides a reusable checking method: first prove a sufficient condition, then give an exact test and counting for the value-dependent part.

A A Finite-Round Proof of the Round-Key Equivalence Class Bound

This appendix studies only the equivalence of complete round-key sequences. Write

$$\Phi_R(A, B) = (K_0, \dots, K_R), \qquad (A, B) \in V^2.$$

Here $1 \le R \le 16$ and there are R message expansions; for the DBL case the same chaining-value branch is fixed. Two inputs belong to the same equivalence class if all their $R + 1$ combined round keys coincide. The notation V, ρ_s, E, O, F, P, σ, $D(a, b)$, U, D_0 of the main text carries over; in particular $P = \{d : \rho_2 d = d\}$ and $\sigma^2 = \mathrm{id}$.

Theorem A.1 (Bound on complete round-key equivalence classes)**.** *Take $R \ge 8$ for the normal message expansion and $R \ge 15$ for the expansion with swapped permutations of the DBL class. If $\Phi_R(A, B) = \Phi_R(\tilde{A}, \tilde{B})$, then necessarily*

$$A \oplus \tilde{A} = B \oplus \tilde{B} = D(a, b), \qquad a, b \in P.$$

Hence each equivalence class contains at most 16 inputs. On $D_0 = V \times U$ this bound is attained. In particular, the conclusion covers all 16 rounds of both expansions.

We now give a proof independent of the linear-elimination program. The equal-key assumption gives each snapshot a common difference

$$D_r = A_r \oplus \tilde{A}_r = B_r \oplus \tilde{B}_r, \qquad 0 \le r \le R. \tag{13}$$

The round constants cancel in the differences, so the argument below applies to any suffix; the notation of Eq. (13) carries over to the lemmas. Saying that n expansion steps remain from snapshot r means precisely $R - r \ge n$. For single-step comparisons we write $\delta_i = D_{r,i}$, $\delta'_i = D_{r+1,i}$.

Lemma A.2 (Linear restriction from three expansion steps)**.** *If $R - r \ge 3$, then*

$$D_r = (d_0, \eta F, \theta F, \eta F, \theta F, d_5, \theta F, d_7, \eta F), \qquad \eta, \theta \in \{0, 1\},\ d_5, d_7 \in P. \tag{14}$$

If two adjacent snapshots both satisfy this form, the tags update as $(\eta_{r+1}, \theta_{r+1}) = (\theta_r, \eta_r)$.

Proof. By Eq. (13), only coordinates that do not pass through the S-box of their row are compared. For the normal arrangement, one-step agreement of the two branches gives

$$\delta_4 = \rho_2 \delta_6, \qquad \delta_3 = \rho_{53} \delta_1, \qquad \delta_2 = \rho_{22} \delta_6, \qquad \delta_7 \in P, \tag{15}$$

where $\delta_7 \in P$ follows from $\rho_{31}\delta_7 = \rho_{61}\delta_7$ and $\gcd(30, 64) = 2$. Furthermore,

$$\delta'_2 = \rho_{11}\delta_8, \qquad \delta'_6 = \delta_8, \qquad \delta'_4 = \delta_3, \qquad \delta'_8 = \rho_{59}\delta_6, \qquad \delta'_7 = \rho_{47}\delta_5. \tag{16}$$

The relation $\delta'_2 = \rho_{22}\delta'_6$ of the next expansion step forces $\rho_{11}\delta_8 = \rho_{22}\delta_8$. By Eqs. (15)–(16), since $\gcd(11, 64) = 1$, we get $\delta_8 \in \{0, F\}$. Then $\delta'_4 = \rho_2\delta'_6$ gives $\delta_1 = \delta_3 = \delta_8$, and $\delta'_7 \in P$ gives $\delta_5 \in P$. These conclusions need only two expansion steps. The third step makes the next snapshot's δ'_8 lie in $\{0, F\}$ as well, hence $\delta_2 = \delta_4 = \delta_6 \in \{0, F\}$.

For the swapped arrangement of the DBL class, the corresponding relations are

$$\delta_4 = \rho_{26}\delta_2, \qquad \delta_3 = \rho_8\delta_8, \qquad \delta_1 = \rho_{41}\delta_8, \qquad \delta_7 \in P, \tag{17}$$

$$\delta'_1 = \rho_5\delta_6, \qquad \delta'_8 = \rho_{37}\delta_6, \qquad \delta'_3 = \rho_{17}\delta_4, \qquad \delta'_6 = \rho_{41}\delta_8, \qquad \delta'_7 = \rho_{47}\delta_5. \tag{18}$$

The relation $\delta'_1 = \rho_{41}\delta'_8$ of the next snapshot gives $\rho_5\delta_6 = \rho_{78}\delta_6$; by Eqs. (17)–(18), since $\gcd(73, 64) = 1$, we get $\delta_6 \in \{0, F\}$. Then $\delta'_3 = \rho_8\delta'_8$ gives $\delta_2 = \delta_4 = \delta_6$, and $\delta'_7 \in P$ gives $\delta_5 \in P$. The third expansion step guarantees $\delta'_6 \in \{0, F\}$, hence $\delta_1 = \delta_3 = \delta_8 \in \{0, F\}$. This proves that both arrangements satisfy Eq. (14). Finally, in both cases $\delta'_1 = \theta F$ and $\delta'_2 = \eta F$, so the tags swap. □

Lemma A.3 (Five expansion steps exclude the two constant directions)**.** *If $R - r \geq 5$, then $\eta = \theta = 0$ in Eq. (14).*

Proof. When $R - r \geq 3$, write the three output differences of S_1 as $(\beta_3, \beta_4, \beta_5)$. Coordinate comparison forces

$$(\beta_3, \beta_4, \beta_5) = (\eta F, \theta F, d_5). \tag{19}$$

Concretely, the three components of Eq. (19) are determined as follows: for the normal arrangement β_3 is determined by δ'_2, and β_4 by $\delta'_3 = \rho_{53}\delta'_1 = \rho_{43}\beta_4$; for the DBL class β_3 is determined by δ'_1, and β_4 by $\delta'_4 = \rho_{26}\delta'_2 = \rho_{53}\beta_3$. In both cases $\beta_5 = \rho_{18}d_5 = d_5$. The snapshot relations used here need only one more expansion step.

The sequential assignment of the S-box can be written equivalently as

$$S(x, y, z) = \big(x \oplus yz,\ y \oplus (x \oplus y)z,\ z \oplus (x \vee y)\big), \tag{20}$$

where products denote bitwise AND. A bitwise expansion shows: if the input direction is (η, θ, c) and the output direction is again (η, θ, c) with $(\eta, \theta) \neq (0, 0)$, then (η, θ) requires

$$(1,0):\ c = 0,\ y = 1,\ z = 0; \qquad (0,1):\ c = 0,\ x = 1,\ z = 0; \qquad (1,1):\ c = 0,\ x \oplus y = 1,\ z = 0. \tag{21}$$

Thus on the current S_1 input of the base value B, we must have $d_5 = 0$, $z = 0$, $x \vee y = F$, and the third output is F. This output is the full word F; through the permutations and rotations of both B-expansions it propagates to coordinate 5 of snapshot $r + 2$ (F is invariant under any rotation), so $B_{r+2,5} = F$. If $R - r \geq 5$, Lemma A.2 applies to $r, r + 1, r + 2$ simultaneously, and the tags remain nonzero after two swaps. Applying Eq. (21) again at $r + 2$ would give $B_{r+2,5} = 0$, a contradiction. □

Lemma A.4 (Excluding a nonzero difference at coordinate 0)**.** *For the normal arrangement with $R \geq 8$, and for the DBL class with $R \geq 15$, we have $D_{0,0} = 0$.*

Proof. Write $d_r = D_{r,0}$, and write the S_0 input of the base value A as (x_r, y_r, z_r) and its output as (p_r, q_r, s_r). By Lemma A.3, for $0 \leq r \leq R - 5$ the S_0 input direction can only be $(d_r, 0, 0)$. Coordinate comparison forces the output direction to be $(\rho_2 d_r, 0, 0)$: the first component comes from $\rho_1\Delta p_r = \rho_3 d_r$; for the normal arrangement the second and third components come from $\delta'_6 = \delta_8 = 0$ and $\delta'_3 = \rho_{53}\delta'_1 = 0$; for the DBL class the third and second components come from

$\delta'_8 = \rho_{37}\delta_6 = 0$ and $\delta'_4 = \rho_{26}\delta'_2 = 0$. On the other hand, the actual output direction of Eq. (20) is $(d_r, d_r \wedge z_r, d_r \wedge \neg y_r)$. Therefore

$$d_r \in P, \qquad y_r \wedge d_r = d_r, \qquad z_r \wedge d_r = 0 \qquad (0 \le r \le R-5). \tag{22}$$

Here $\bar{w} = F \oplus w$ denotes the bitwise complement. Coordinate 0 of B does not pass through S_1; together with $d_0 \in P$, this gives, over all snapshots,

$$d_{r+1} = \rho_3 d_r = \sigma d_r, \qquad 0 \le r < R. \tag{23}$$

Normal arrangement. By Eq. (22), q_r equals 1 on all active bits of d_r. The second output propagates along the path $1 \to 6 \to 8 \to 2$ of A without passing through S_0, so

$$z_{r+3} = \rho_{41+59+11}\, q_r = \rho_{111}\, q_r. \tag{24}$$

Period-two behavior gives $\rho_{111} d_r = \rho_9 d_r = d_{r+3}$, where the last equality follows by iterating Eq. (23). If $R \ge 8$, Eq. (22) applies to both $r = 0, 3$, while Eq. (24) forces z_3 to be 1 on d_3. Hence $d_0 = 0$ is the only possibility.

DBL class. The two corresponding paths are $1 \to 4 \to 3 \to 2$ and $2 \to 8 \to 6 \to 1$, so

$$z_{r+3} = \rho_{28}\, q_r, \qquad y_{r+3} = \rho_{105}\, s_r. \tag{25}$$

If $d_0 = F$, then $q_0 = F$ gives $z_3 = F$, already contradicting Eq. (22) when $R \ge 8$. It remains to exclude $d_0 \in \{E, O\}$. In this case d_r swaps its parity phase round by round; in particular $d_{r+3} = \bar{d}_r$. When $0 \le r \le R-8$, q_r is 1 on d_r, while $z_{r+3} = \rho_{28} q_r$ is 0 on $\bar{d}_r$; since even rotations preserve both parity classes of bit positions,

$$q_r = d_r \ (0 \le r \le R-8), \qquad z_j = \bar{d}_j \ (3 \le j \le R-5). \tag{26}$$

On the inactive bits of $3 \le r \le R-8$ (second relation of Eq. (26)), $z_r = 1$, and Eq. (20) gives $q_r = x_r$, so

$$x_r \wedge \bar{d}_r = 0 \qquad (3 \le r \le R-8). \tag{27}$$

Now use the exact recurrence of A: $x_{r+1} = \rho_1(x_r \oplus y_r z_r)$. For $3 \le r \le R-9$, Eq. (27) holds at both r and $r+1$; combined with $d_{r+1} = \rho_1 d_r$ and $z_r = \bar{d}_r$, this gives $y_r \wedge \bar{d}_r = 0$. By Eq. (22),

$$y_r = d_r, \qquad s_r = z_r \oplus (x_r \vee y_r) = F \qquad (3 \le r \le R-9). \tag{28}$$

When $R \ge 15$, this range contains $r = 3, 6$. Eq. (28) gives $s_3 = F$ and $y_6 = d_6$, but Eq. (25) gives $y_6 = \rho_{105} s_3 = F$, a contradiction. Hence $d_0 = 0$. □

Proof of Theorem A.1. Applying Lemmas A.2, A.3, and A.4 in order, the initial common difference can only be $D(a, b)$ with $a, b \in P$. After fixing one input of the equivalence class, the other inputs are determined uniquely by this initial difference, so the class size does not exceed $|P|^2 = 16$.

For attainability, take any $(A, B) \in D_0$. Both B-updates preserve U, because the input of S_1 has the form $(0, 0, z)$ and its output is again $(0, 0, z)$; the permutations preserve the support $\{0, 5, 7\}$, and round constants are added only to coordinate 0. For all $a, b \in P$, the difference of the two branches updates as

$$D(a, b) \mapsto D(\sigma b, \sigma a).$$

The difference of A avoids S_0, and the difference of B passes through S_1 unchanged on this base value. Hence all 16 distinct inputs $\big(A \oplus D(a, b),\ B \oplus D(a, b)\big)$ lie in the same class as (A, B). □

The numbers 8 and 15 in this proof are sufficient round counts; their minimality is not claimed. The necessary condition of the theorem concerns equal complete round-key sequences; it does not exclude different round-key sequences producing the same final output, so it cannot be interpreted as a bound on the equivalence classes of the compression function or of the complete hash function.

B The Complete Domain with at Least Four Equivalent Choices

The notation of the main text carries over: $x_r = B_{r,3}$, $y_r = B_{r,4}$, $z_r = B_{r,5}$, $(p_r, q_r, s_r) = S(x_r, y_r, z_r)$, and the original domain $D_0 = V \times U$ with cardinality u_0. The main text has proved: when $B \notin U$, a round-key equivalence class has at most four elements; if it has exactly four, the allowed (a, b) must be the product of two one-dimensional coordinate spaces. This appendix gives the complete base-value conditions and exact counts for the two arrangements.

B.1 DM: two complementary four-element direction spaces

When the B-branch uses P_1, the exact recurrence is

$$x_{r+1} = \rho_{43} q_r, \qquad y_{r+5} = \rho_{52} p_r. \tag{29}$$

Outside U the four-element direction space can only be

$$G_E = \{0, E\} \times \{0, O\}, \qquad G_O = \{0, O\} \times \{0, E\}.$$

Why the other two candidates are excluded follows from the same recurrence. For example, for $\{0, E\} \times \{0, E\}$, set $\bar{B}_r = \rho^{-r} B_r$ (wordwise rotation). The exact direction conditions require $\bar{x}_r$ and $\bar{y}_r$ to contain only odd bits, hence $\bar{p}_r$ and $\bar{q}_r$ too. Eq. (29) becomes $\bar{y}_{r+5} = \rho_{47} \bar{p}_r$; 47 is odd, so $\bar{p}_r = 0$ for $0 \le r \le 10$. Then $\bar{y}_r = 0$ $(5 \le r \le 15)$, and in turn $\bar{x}_r = 0$ $(5 \le r \le 10)$. Five consecutive snapshots with $x_r = y_r = 0$ force the six side words to be zero, and by the forward and backward invariance of U we get $B \in U$, a contradiction. $\{0, O\} \times \{0, O\}$ is analogous.

Complete base-value conditions. It suffices to discuss G_E. It requires $x_r \wedge E = y_r \wedge E = 0$ for $0 \le r \le 15$. Then q_r contains only odd bits, and the first relation of Eq. (29) rotates them to even positions, so $q_r = 0$ for $0 \le r \le 14$, and $x_r = 0$ for $1 \le r \le 15$. Checking $q = y \oplus (x \oplus y)z$ bit by bit gives

$$q_r = 0 \iff x_r \wedge z_r = 0, \quad y_r \wedge z_r = 0. \tag{30}$$

Under this condition (Eq. (30)), $S(x, y, z) = (x \oplus y, 0, z \oplus x \oplus y)$ and $x \wedge y = 0$. The four initial delayed inputs $y_1, \ldots, y_4$ further give the following necessary and sufficient conditions:

$$\begin{aligned} &B_{0,1} \wedge O = B_{0,2} \wedge O = 0, \\ &B_{0,3} \wedge E = B_{0,4} \wedge E = B_{0,6} \wedge E = B_{0,8} \wedge E = 0, \\ &x_r \wedge z_r = 0, \qquad y_r \wedge z_r = 0, \qquad 0 \le r \le 14. \end{aligned} \tag{31}$$

Necessity was proved above. Conversely, the first row guarantees the odd-bit support of the first five y_r, and the second row makes $q_r = 0$; by induction on Eq. (29) the later x_r and y_r keep the required support, hence sufficiency. The last input snapshot $r = 15$ only needs the direction condition; $q_{15} = 0$ is no longer required.

Reduce all constraints to a count on one cycle. Write $v = B_{0,5}$ and $w = B_{0,7}$. Only the odd bits of v and the even bits of w matter, 64 bits in total; the other two half-words and $B_{0,0}$ contribute 128 free bits. Below, control bits are labeled by their initial position; when the same position appears again, the previously flipped value is used. Table 9 lists, for one nonzero bit k of each side word, the positions at which it appears in Eq. (31); all subscripts are modulo 64. A nonzero bit of y requires the corresponding bit of z to be 1 and turns it to 0; a nonzero bit of the initial x_0 requires that bit to be 0 and turns it to 1.

The next constraint five rounds later moves as follows:

$$(v, k) \mapsto (w, k + 3) \mapsto (v, k + 38) \tag{32}$$

By Eq. (32), the 32 bits of v and the 32 bits of w form exactly one cycle of length 64: on the set of odd positions, adding 38 is adding 19 modulo 32, and $\gcd(19, 32) = 1$. Each of the first

Table 9: The three constraints corresponding to one nonzero initial side-word bit in the DM four-choice domain

initial word	parity of k	constraint times	first corresponding initial control bit
$B_{0,4}$	odd	$0, 5, 10$	(v, k)
$B_{0,1}$	even	$1, 6, 11$	$(w, k-8)$
$B_{0,6}$	odd	$2, 7, 12$	$(v, k-30)$
$B_{0,8}$	odd	$3, 8, 13$	$(w, k-27)$
$B_{0,2}$	even	$4, 9, 14$	$(v, k-19)$
$B_{0,3}$	odd	$0, 5, 10$, the first one being $0 \to 1$	(v, k)

Note: all subscripts are modulo 64; the first constraint of $B_{0,3}$ requires that bit to turn $0 \to 1$, and all other constraints require $1 \to 0$.

five rows occupies three consecutive positions of the cycle and requires $1 \to 0$ three times; the last row requires $0 \to 1$ at its first position and $1 \to 0$ at the following two. A given position undergoes $1 \to 0$ at most once, because all $0 \to 1$ happen only at time 0. The first position of the initial $B_{0,3}$ may coincide with the last position of the previous block; other overlaps would duplicate a $1 \to 0$ and are not allowed. This rule captures all valid cases; it is not an additional disjointness assumption.

Process the pairs (v, w) one by one in cycle order. States 0, 1, 2 indicate that, in the current pair, zero, only the first, or both positions have already been occupied by earlier three-position constraints. The transition matrix (rows: current state, columns: next state) is

$$Q = \begin{pmatrix} 4 & 4 & 4 \\ 2 & 1 & 2 \\ 1 & 0 & 0 \end{pmatrix}. \tag{33}$$

Taking state 0 as an example (the matrix entries follow Eq. (33)): if neither bit is used there are $2^2 = 4$ assignments; an ordinary three-position constraint starting at v has three initial sources, plus one from $B_{0,3}$, so the weight to state 1 is $3 + 1 = 4$; two sources start at w, and the unused bit of v has two more assignments, so the weight to state 2 is 4. In state 1, the current v is the last position of the previous block: a free w contributes weight 2, the allowed overlap of $B_{0,3}$ contributes 1, and starting at w contributes 2. State 2 returns to state 0 with weight 1. These weights simultaneously count the choice of the side-word source and the assignment of unconstrained control bits.

The cycle has 32 pairs, and the closing boundary condition gives

$$T = \mathrm{tr}(Q^{32}) = 506813528802536089100170 25. \tag{34}$$

The positions of the cycle are fixed, so we do not divide by 32 or 64 here. Let $\mathcal{F}_E = \{(A, B) : G_E \subseteq \Gamma_B\}$; then by Eq. (34),

$$M := |\mathcal{F}_E| = 2^{576+128}\, T = 2^{704}\, T. \tag{35}$$

Rotating positions 1 to 8 of B uniformly by one bit gives the other domain $\mathcal{F}_O$, with the same cardinality M of Eq. (35). The two domains impose opposite parity-support requirements on the six side words and intersect exactly in D_0. Hence the complete message domain covered by DM round-key equivalence classes with at least four elements is

$$|\mathcal{F}_E \cup \mathcal{F}_O| = 2M - u_0 = 2^{705}T - 2^{768}. \tag{36}$$

In this union, points outside D_0 have exactly four equivalent choices, and points inside have sixteen.

B.2 DBL: the complementary classification from the central snapshot

When the B-branch switches to P_0, the recurrence becomes

$$y_{r+1} = \rho_{53} p_r, \qquad x_{r+5} = \rho_{42} q_r. \tag{37}$$

This time the four-element direction spaces outside U can only be

$$H_E = \{0, E\} \times \{0, E\}, \qquad H_O = \{0, O\} \times \{0, O\}.$$

For example, if G_E were allowed, then x_r, y_r, p_r, q_r would all contain only odd bits in every round. The first relation of Eq. (37) forces $p_r = 0$ ($0 \le r \le 14$), giving five consecutive snapshots with $x_r = y_r = 0$, which again can only lie in U. G_O is analogous.

For H_E, set $\bar{B}_r = \rho^{-r} B_r$. The direction condition becomes: $\bar{x}_r$ and $\bar{y}_r$ contain only odd bits, and the second delayed relation becomes

$$\bar{x}_{r+5} = \rho_{37}\, \bar{q}_r.$$

Hence $\bar{q}_r = 0$ for $0 \le r \le 10$. Tracing the five-step path of the q outputs backwards, in the central snapshot $\bar{B}_5$ the positions 1, 2, 3, 6, 8 are all zero, and only positions 0, 4, 5, 7 may be nonzero. Write the three relevant words as

$$y = \bar{B}_{5,4}, \qquad v = \bar{B}_{5,5}, \qquad w = \bar{B}_{5,7}.$$

The complete necessary and sufficient conditions are

$$\begin{aligned} &y \wedge E = 0, \qquad y \wedge \rho_{16}\, y = 0, \qquad y \wedge \rho_{32}\, y = 0, \\ &J = y \vee \rho_{16}\, y \vee \rho_{32}\, y, \qquad J \wedge v = 0, \qquad \rho^{-8} J \wedge w = 0. \end{aligned} \tag{38}$$

(here J is a 64-bit word, unrelated to the vector notation $J(c)$ of Section 3.2). The proof can be completed in the six central steps $r = 5, \ldots, 10$. In these six steps $\bar{x}_r = 0$, and the condition $\bar{q}_r = 0$ requires every nonzero bit of $\bar{y}_r$ to meet a 1 in the corresponding $\bar{z}_r$, which then turns that bit to 0. One initial nonzero bit k requires exactly the six bits

$$v: \ k,\ k+16,\ k+32; \qquad\qquad w: \ k-8,\ k+8,\ k+24 \tag{39}$$

to be 1. By Eq. (39), within the same residue class modulo 16, the three-bit sets of any two bits of y intersect, so at most one can be chosen; the control bits of different residue classes are disjoint. This is exactly Eq. (38).

It remains to check that the central condition covers all 16 rounds. Looking forward, $\bar{q}_5 = \cdots = \bar{q}_{10} = 0$ guarantees $\bar{x}_r = 0$ up to $r = 15$; in between, $\bar{y}_{r+1} = \rho_{52} \bar{p}_r$ keeps its odd-bit support. Looking backward, the five zero side words of the center force $\bar{q}_0 = \cdots = \bar{q}_4 = 0$. When $q = 0$, the relation $q = (x \oplus y)z$ gives $y = p \wedge z$, and then $p = x \oplus yz$ gives $x = p \oplus y$; given z, the inverse S-box recovers (x, y) uniquely from p, so the five backward steps likewise preserve the odd-bit support. Hence Eq. (38) is indeed necessary and sufficient. The five expansion steps and the wordwise rotations are invertible, so counting the central snapshots is the same as counting the initial states.

The 32 selectable bits of y split into eight residue classes modulo 16, each with four candidate bits and eight control bits. Choosing no bit of y leaves $2^8 = 256$ control assignments; choosing one bit fixes four positions, sets six control bits to 1, and leaves the remaining two free, giving $4 \cdot 2^2 = 16$. Each class contributes 272 possibilities, and the eight classes are independent. Together with $\bar{B}_{5,0}$ and the unused half-words of v and w, there are 128 free bits, so

$$M_{\mathrm{DBL}} = 2^{576+128}\, 272^8 = 2^{736}\, 17^8. \tag{40}$$

The domain corresponding to H_O has the same size, and the intersection is again exactly D_0; hence the complete domain of the DBL class with at least four choices is

$$2M_{\mathrm{DBL}} - u_0 = 2^{737}\,17^8 - 2^{768}. \tag{41}$$

This subsection classifies the four-choice domains of the DBL class; the value N for the fixed single-direction complete domain in the main text applies only to the DM class and cannot be transferred here directly.

C Compilation, Elimination, and Independent Cross-Checks of the Exact Counting

This appendix gives the finite Boolean model and the integer-counting procedure of Computation 5.1. All counted objects are initial messages of the DM class.

C.1 Bitwise compilation of the Boolean constraints

By Theorem 4.2, the allowed condition for direction (a, b) is

$$(B_{r,3} \vee B_{r,4}) \wedge a = 0,\ r = 0, 2, \ldots, 14; \qquad (B_{r,3} \vee B_{r,4}) \wedge \sigma(b) = 0,\ r = 1, 3, \ldots, 15. \tag{42}$$

These conditions constrain the real expansion trajectory B; the shapes of the side words are not prescribed in advance. All 576 bits of A are free.

Below is a counting procedure that can be recomputed using only finite Boolean relations and arbitrary-precision integers. Let $B_{r,i}[k]$ denote bit k of word i, with $0 \le i \le 8$ and $0 \le k < 64$. For every bit triple (x, y, z) entering an S-box, introduce three output variables (p, q, s) and add the six-variable Boolean factor of Eq. (43):

$$\varphi(x, y, z, p, q, s) = 1 \iff \begin{cases} p = x \oplus (y \wedge z), \\ q = y \oplus (x \wedge z) \oplus (y \wedge z), \\ s = z \oplus x \oplus y \oplus (x \wedge y). \end{cases} \tag{43}$$

The factor has exactly eight allowed assignments. The permutations and rotations only connect variables; no additional relation table is needed. The concrete connections use $P_1 = (0, 6, 3, 4, 1, 7, 8, 5, 2)$ and $R_1 = (3, 7, 19, 43, 53, 61, 0, 29, 37)$, and (for $i \neq 0$)

$$B_{r+1,i}[k] = \hat{B}_{r,P_1(i)}\big[(k - R_1(i)) \bmod 64\big],$$

where $\hat{B}$ is the state after applying the S-box to words 3, 4, 5. The round constants act only on the independent word 0; word 0 never enters the factors or the conditions above, so it is separated in the count and multiplied by 2^{64} at the end. For $(a, b) = (E, 0)$ the last constrained state is $r = 14$, so it suffices to compile the first 14 expansion steps and these eight groups of even-round conditions. Compiling additionally up to the 16-th step and eliminating the terminal deterministic variables yields the same initial allowed domain.

C.2 Simplification, existential projection, and integer elimination

The computation proceeds in the following order.

1) Fix the initial or intermediate bits specified by Eq. (42) to zero. Delete truth-table rows that are inconsistent factor by factor; if a column is constantly zero, merge it into the zero constant; if two columns agree on all remaining rows, merge the two variables. Repeat until nothing changes.

2) Apply exact existential projection to leaf variables that appear in only one factor and are not initial variables. The set of allowed rows after projection is obtained by deleting that column and deduplicating. Continue deleting new leaf variables of this kind, and delete always-true factors that allow all assignments. Initial variables are always kept, including free initial roots that end up in no nontrivial factor.

3) Perform integer variable elimination on the remaining 0/1 factors. If the factors containing a variable v are F_v, set

$$g(t) = \sum_{v \in \{0,1\}} \prod_{f \in F_v} f(v, t_f), \tag{44}$$

where t is the union of the variables of the relevant factors other than v. Replace F_v by g (Eq. (44)) until all variables are eliminated; the final scalars from different connected components are multiplied. All multiplications and additions are exact over the nonnegative integers.

4) Finally multiply by the number of assignments of the free initial roots outside the graph, by 2^{64} for the independent word 0, and by 2^{576} for the free A, to obtain the cardinality of the complete allowed message domain.

Step 2 uses existential projection rather than keeping multiplicities of internal assignments; its correctness comes from the determinism of the circuit: any complete initial assignment corresponds to at most one full set of intermediate values, hence to at most one set of retained intermediate values. Each leaf-variable projection preserves exactly the remaining assignments that can be extended. Thus the remaining assignments counted by the elimination are in bijection with the valid initial assignments; intermediate variables cause no double counting.

For $(E, 0)$, the eight words of B other than the initial word 0 have 512 bits, of which 128 are fixed to zero. After simplification there are 384 initial variable roots, of which 352 enter the unique connected component and 32 are completely free, namely the even bits of the initial $B_{0,5}$. The connected component contains 864 variables and 448 factors; classified by scope size, there are 64 binary, 256 ternary, 32 quaternary, 32 quinary, and 64 senary factors. Using a minimum-fill-first elimination order, the largest intermediate tensor of this computation has 21 binary axes. The exact integer result is

$$C = 1439422992972231685577404418668365067845632 = 2^{64} \cdot 78031276805304028734527. \tag{45}$$

This gives the cardinality of the single-direction complete domain

$$N = |D_{E,0}| = 2^{576+64+32}\, C = 2^{736} \cdot 78031276805304028734527, \qquad \log_2 N \approx 812.0464705942. \tag{46}$$

The factorization $C = 2^{64} \cdot 78031276805304028734527$ is an integer factorization; it does not additionally claim the existence of 64 independently settable raw bits.

C.3 Independent compilation and elimination cross-checks

This computation was cross-checked at three independent layers. First, another implementation compiled all 16 expansion steps, obtaining 1024 raw S-box factors, and then applied leaf-variable elimination keeping integer weights; the maximum projection weight of the 672 eliminated leaf variables was 1, the scalar factor produced in this stage was 1, and the resulting 448 nontrivial factors and the quotient mapping of initial variables agree one by one with the 14-step compilation above. Second, another integer-elimination implementation reordered the tensor axes and the factor schedule, recomputing once with the original order and once with an order changed by 96 adjacent swaps; the latter changed 165 order positions, and both runs reproduced Eqs. (45) and (46) exactly, with maximum tensor rank 21 in both cases. Third, all fifteen nonempty

intersections (including the four basic domains) were compiled and counted exactly one by one, consistently giving the values N, M, Z stated in the main text. These results are integer counts over finite Boolean relations, not random sampling estimates. The scripts and intermediate data used for the cross-checks are available from the authors upon request.

D The State Round Function and Version Identification

State round function. Write the state as three rows L_0, L_1, L_2, let $t = L_0 \oplus L_1 \oplus L_2$, and $\mu(t_0, t_1, t_2) = (t_0,\ t_1 \oplus t_2,\ t_0 \oplus t_2)$. The linear layer updates each row to $L_i \oplus \mu(t)$. The state round function applies Eq. (1) to the three rows in order, performs this linear layer, and then applies the wordwise rotation table

$$Y_0 = (0, 3, 6, 9, 12, 15, 18, 21, 24), \qquad Y_1 = (0, 24, 48, 8, 32, 56, 16, 40, 7).$$

according to the parity of the round number. The first 16 constants of the message expansion are the hexadecimal word values

```
24 3f 6a 88 85 a3 08 d3 13 19 8a 2e 03 70 73 44.
```

The constants act only on coordinate 0 of the second message half. That word updates independently and does not enter the constraint counts of the side words.

Byte and bit order. If the raw block bytes are $m[0], \ldots, m[143]$, then word j of half h is $\sum_{t=0}^{7} m[72h + 8j + t]\, 2^{8t}$, with $h \in \{0, 1\}$, $j \in \{0, \ldots, 8\}$. Digest words are also output little-endian; the interface counts bits within a byte from the most significant to the least significant. For non-byte-aligned XOF lengths, the high bits needed by the last byte are kept and the remaining low bits are cleared.

Fixed source files. Table 10 lists the `CryptHash_AlgorithmInstance.c` of the corresponding variants in the pinned commit[8]. Each SHA-256 is shown in two segments per line; concatenate them when reading.

Table 10: Fixed source files and their SHA-256

variant	SHA-256
`MoFang-256`	`99074ca5b9dac088a1201ffd1c441319` `a50d92a48bae9243a7d916c834e2888c`
`MoFang-512`	`460417b52d6989b5f4a9007a7cc35e16` `92444003c9418287e193af2aa82a1f1b`
`MoFang-256-XOF`	`460aac73d7e9be57ed4596efb6888ef5` `81f6422648875984b51d28206db76d8e`
`MoFang-768`	`c61eaaf9d9c12dc02608cac8973825fa` `34d4d7139a4a0c28b5e43440f73a0bec`
`MoFang-1024`	`009d3d09dd67f06b99d47bf4073716c5` `cd68e87a1047f26b30a74ff238397865`
`MoFang-768-XOF`	`5f5500c422c6e9d7420802cce9c7f9ec` `11ae4dd0ab6853d167df4d8f889b034a`

The mathematical proofs concern the word model of Section 1; the source-file identifiers pin down the experimental object and do not remove the undefined behavior of the original C.

Fixed version of the cited paper. The original manuscript records that the file used for checking[2] has 47 pages, with SHA-256

```
1570eabbc55b0223b4540ff8f477f27b  93dad58ac72592e3c4f791f51b7b6f49
```

This paper keeps that version record; this revision did not re-fetch the PDF and does not substitute the online revision date for the file identifier.

References

[1] BI W, ZHAO B, DONG X, et al. MoFang hash function[Z]. Submission to the ICCS NGCC program, 2026-06-29.

[2] YUAN Y, WU R, WEI S, et al. Structural analysis of seven hash functions submitted to the NGCC[EB/OL]. (2026)[2026-11-10]. https://eprint.iacr.org/2026/2152.

[3] NGCC.DEV. MoFang hash function—hash report (hash-19)[EB/OL]. (2026)[2026-11-10]. https://ngcc.dev/reports/hash-19.html.

[4] MOFANG SUBMISSION TEAM. Erratum and revisions of the hash function MoFang[Z]. 2026-09-28.

[5] BI W, ZHAO B, DONG X, et al. MoFang hash function (revised)[Z]. Submission to ICCS NGCC, 2026-09-28.

[6] HIROSE S. Some plausible constructions of double-block-length hash functions[C]// ROBSHAW M. Fast Software Encryption. Berlin: Springer, 2006: 210-225.

[7] LEANDER G, MINAUD B, RØNJOM S. A generic approach to invariant subspace attacks: cryptanalysis of Robin, iSCREAM and Zorro[C]//OSWALD E, FISCHLIN M. Advances in Cryptology—EUROCRYPT 2015. Berlin: Springer, 2015: 254-283.

[8] BI W, ZHAO B, DONG X, et al. MoFang hash function: reference implementations[EB/OL]. (2026)[2026-11-10]. https://github.com/biwenquan/MoFang-Hash-function.

[9] ISO/IEC JTC1/SC22/WG14. Programming languages—C: committee draft N1570[S/OL]. (2011-04-12)[2026-11-10]. https://www.open-std.org/jtc1/sc22/wg14/www/docs/n1570.pdf.