International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(22s) 2026 pp. 1836-1852
https://doi.org/10.70917/ijcisim-2026-5782
Article

RIFT-PWLCM: A Reproducible Fixed-Point Image-Encryption Construction with Domain-Separated Bidirectional Feedback and Authenticated Output

Nani Gopal Pal · Abhishek Majumdar

Department of Computer Science and Engineering, Techno India University, West Bengal, India
nanigopalpal81@gmail.com (corresponding author); abhishek.m@technoindiaeducation.com

Abstract: Many chaos-based image ciphers obtain excellent NPCR, UACI and entropy values while retaining floating-point maps, genetic tuning, DNA-labelled operations, global sorting, or unauthenticated ciphertext. This work introduces RIFT-PWLCM (Reversible Integer Feedback Transformation with PWLCM-derived material), a reproducible research construction that removes those elements. A 256-bit key and 128-bit nonce are domain-separated with SHAKE-256 into public-parameter Q0.32 integer PWLCM-derived material for four roles per round (PRE, FWD, BWD and POST). Six reversible rounds combine pre/post whitening with AES S-box forward and backward byte-feedback recurrences. Ciphertext and associated data are protected by an encrypt-then-MAC HMAC-SHA-256 tag truncated to 128 bits. All reported values were regenerated from the stated implementation. Across seven 256×256 grayscale inputs and one 256×256 RGB input, ciphertext entropy is 7.996996–7.999106 bits/byte and the largest absolute adjacent correlation is 0.00927. Over 105 single-bit plaintext trials, NPCR is 99.6080±0.0263% and UACI is 33.4758±0.0885%. Exhaustive single-bit flips of the 256-bit key and 128-bit nonce change 49.9926% and 49.9970% of ciphertext bits on average. Generator autocorrelations through lag 32 remain below 0.00533 in magnitude; bounded differential, linear, NIST SP 800-22, weak-key, nonce-reuse, and authentication screens are also reported. All 10,000 ciphertext-tamper trials were rejected and no false acceptance occurred in 100,000 random-tag trials. The Python/Numba reference implementation reaches roughly 0.42–0.55 MB/s for medium-to-large buffers. These experiments establish reproducibility and feasibility, not a formal security level; RIFT-PWLCM is not proposed as a replacement for standardized AEAD such as Ascon.
Keywords: Authenticated image encryption · Fixed-point chaos · PWLCM · Domain separation · SHAKE-256 · Bidirectional diffusion · Cryptanalysis

1. Introduction

Image-encryption research continues to explore chaotic maps, DNA-inspired coding, optimization procedures, and permutation–diffusion structures [6]–[18]. The resulting ciphertext often has near-uniform histograms, low adjacent-pixel correlation, and NPCR/UACI values close to their random-image expectations. Those properties are useful defect screens, but they do not constitute a cryptographic security argument. NIST likewise cautions that statistical randomness testing cannot substitute for cryptanalysis [1], [2]. Reviews of chaos-based cryptography have repeatedly identified finite-precision degradation, weak key schedules, insufficient chosen-plaintext analysis, and overinterpretation of image statistics as recurring problems [6]–[8].

RIFT-PWLCM is designed as a response to those methodological issues rather than as another feature-stacking cipher. Genetic parameter tuning, DNA coding, floating-point state, and global permutation sorting are absent. The construction instead uses fixed public Q0.32 map parameters, explicit SHAKE-256 domain separation, a byte-wide standard S-box, reversible bidirectional feedback, six evaluation rounds, and authenticated output. The term PWLCM-

Copyright: © 2026 by the authors

derived is used deliberately: the finite-state integer map is inspired by piecewise-linear chaotic maps, but positive Lyapunov exponents of an ideal real-valued map are not treated as a security proof for the digital construction.

The main contributions are: (1) a fully specified counter-indexed integer material generator whose parameters are fixed rather than selected by a genetic algorithm; (2) separate PRE/FWD/BWD/POST derivation domains so that round materials are not overlapping slices of one chaotic word; (3) a six-round reversible bidirectional feedback core without O(n log n) sorting; (4) encrypt-then-MAC authentication with associated data and a 128-bit tag; and (5) a from-scratch evaluation in which every table and figure is generated by the stated implementation. The contribution is a reproducible experimental construction and analysis. No IND-CPA/IND-CCA proof or numerical "security bits" claim is made.

2. Design context and security scope

2.1 Related work and design choices

Classical chaos-image designs use permutation and diffusion driven by discretized chaotic maps [9], [10]. DNA-labelled operations and evolutionary optimization remain common in the literature [17]–[20], while more recent work pursues high-dimensional, lightweight, and real-time variants [12]–[15]. At the same time, dedicated cryptanalysis continues to break statistically attractive image ciphers [16]. RIFT-PWLCM therefore treats "random-looking ciphertext" as a necessary diagnostic rather than the target definition of security.

SHAKE256 is standardized in FIPS 202 [3] and is used here only for fixed-length, domain-separated seed derivation; it is not used as the full message-length keystream. The AES S-box is taken from FIPS 197 [4] as a compact, public, bijective nonlinear byte mapping. Authentication uses HMAC-SHA-256 in an encrypt-then-MAC arrangement. For deployment, standardized AEAD remains preferable; NIST SP 800-232 standardizes Ascon-AEAD128 for constrained environments [5].

2.2 Threat model and supported use

The sender and receiver share a 256-bit secret key K generated uniformly at random or derived by an approved cryptographic KDF from adequate-entropy keying material. For each message, the sender supplies a public 128-bit nonce N that must be unique under K and optional associated data AAD. A monotonic counter encoded into 128 bits is the simplest uniqueness mechanism; randomly generated nonces require collision control. The supported plaintext/ciphertext body length is $0 \le n < 2^{32}$ bytes per nonce so that the 32-bit position counter never wraps. The adversary may observe nonces, ciphertexts and tags and may choose plaintext/AAD inputs. The receiver verifies the authentication tag before decryption. Nonce reuse is outside the supported model. The experiments include a nonce-reuse relation screen only to look for a simple XOR-style failure; it does not make reuse acceptable.

Because each domain seed is 128 bits and the tag is 128 bits, this paper makes no security claim beyond 128 bits and, more importantly, assigns no formal security level at all. A state-recovery, algebraic or structural attack on the custom expansion could invalidate brute-force reasoning based on key length.

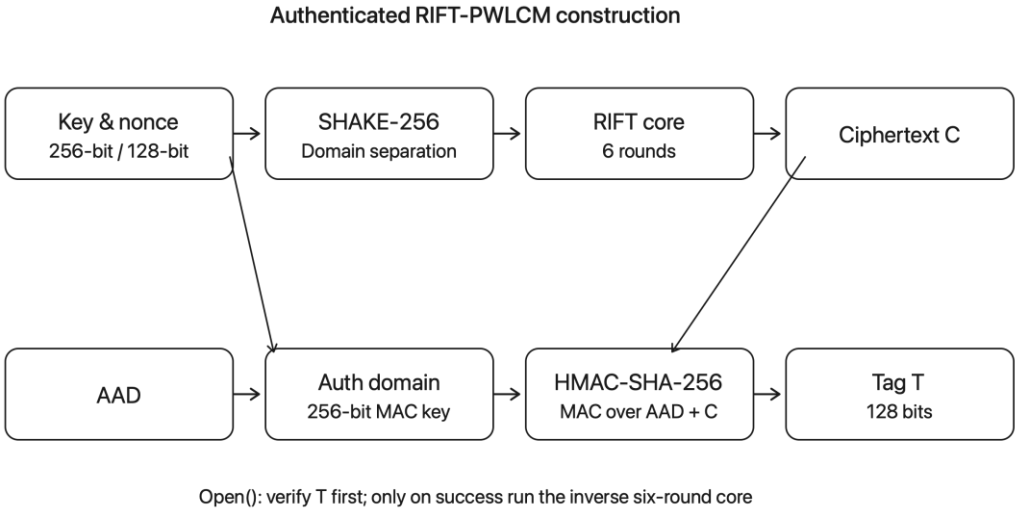


Fig. 1. Overall authenticated RIFT-PWLCM construction. The ciphertext body is produced by the six-round reversible core; the tag authenticates AAD, ciphertext and nonce.

3. RIFT-PWLCM construction

3.1 Integer PWLCM-derived map

Let M=2^32 and H=2^31. State words are unsigned integers in [0,M−1]. For a public integer parameter p with 0<p<H, the implemented reflected integer map G_p is

$$\begin{aligned} G_p(z) &= \left\lfloor \frac{zM}{p} \right\rfloor \bmod M, \quad 0 \le z < p \\ G_p(z) &= \left\lfloor \frac{(z-p)M}{H-p} \right\rfloor \bmod M, \quad p \le z < H \\ G_p(z) &= M - 1 - G_p(M-1-z), \quad H \le z < M \end{aligned} \tag{1}$$

The public parameters are p1=1327217884 (p1/M≈0.3090169942) and p2=1572067138 (p2/M≈0.3660254036). They are fixed constants; there is no per-deployment genetic search. All map arithmetic is integer Q0.32. Consequently, the implementation avoids floating-point reproducibility differences, although it remains a finite-state dynamical system and no ideal-map dynamical property is treated as a cryptographic guarantee.

3.2 Domain-separated material generation

For round r and domain D∈{PRE,FWD,BWD,POST}, the exact derivation input is prefix ‖ D ‖ 0x7C ‖ I2OSP(r,2) ‖ 0x7C ‖ K ‖ 0x7C ‖ N, where prefix is the ASCII byte string “RIFT-PWLCM-v1|”, 0x7C is the ASCII vertical-bar separator, I2OSP(r,2) is the unsigned two-byte big-endian round index, K is exactly 32 bytes, and N is exactly 16 bytes. SHAKE256 returns 16 bytes, parsed as four unsigned big-endian 32-bit words (sx, sy, ta, tb). These fixed lengths and explicit separators make the encoding unambiguous.

$$d(r,D) = \mathrm{SHAKE256}(\mathrm{prefix} \| D \| \text{"|"} \| \mathrm{I2OSP}(r,2) \| \text{"|"} \| K \| \text{"|"} \| N,\ 16\ \text{bytes}) \tag{2}$$

All 32-bit additions, multiplications, XORs, shifts and rotations are performed on unsigned words modulo 2^32 unless a larger intermediate is stated explicitly. ROTL32 and ROTL8 denote circular left rotations of 32-bit and 8-bit

words. In G_p, the products z·2^32 and (z−p)·2^32 are evaluated with an unsigned 64-bit (or wider exact) intermediate before integer division, and the returned state is reduced to the low 32 bits. The counter ii=(i+1) mod 2^32. Four map updates are used at every position.

The public mixer constants are C1=0x9E3779B9, C2=0x85EBCA6B, and C3=0xC2B2AE35. The exact 32-bit avalanche mixer is: z←z⊕(z≫16); z←z·0x7FEB352D; z←z⊕(z≫15); z←z·0x846CA68B; z←z⊕(z≫16), with multiplication reduced modulo 2^32 after each product. The one-byte IV associated with a domain is IV = mix32(ta ⊕ ((tb≪1) mod 2^32)) AND 0xFF. The implementation derives an IV for every domain; only the FWD and BWD IVs are consumed by the feedback recurrences.

Algorithm 1 — Domain-separated byte material (one domain)

```
Input: K (32 B), N (16 B), round r, domain D, length n
d ← SHAKE256(prefix || D || "|" || I2OSP(r,2) || "|" || K || "|" || N, 16 B)
(sx,sy,ta,tb) ← four unsigned big-endian 32-bit words from d
IV ← mix32(ta XOR ((tb << 1) mod 2^32)) AND 0xff
C1=0x9E3779B9; C2=0x85EBCA6B; C3=0xC2B2AE35
for i = 0 ... n−1:
   ii ← (i+1) mod 2^32
   x ← mix32(sx XOR ((ii·C1) mod 2^32) XOR ta)
   y ← mix32(sy XOR ((ii·C2) mod 2^32) XOR ROTL32(tb,13))
   for j = 0 ... 3:
      x ← G_p1((x + ROTL32(y,7) + (ta XOR ((j·C3) mod 2^32))) mod 2^32)
      y ← G_p2((y + ROTL32(x,11) + (((tb+C1) mod 2^32) XOR ((j·C2) mod 2^32))) mod 2^32)
   w ← x XOR ROTL32(y,13) XOR (x>>11) XOR (y>>19)
       XOR mix32(ii XOR ta) XOR ROTL32(tb,(i mod 17)+1)
   stream[i] ← w AND 0xff
Return stream and IV
```

Exact definition of mix32(z)

```
z ← z AND 0xffffffff
z ← z XOR (z >> 16)
z ← (z · 0x7FEB352D) mod 2^32
z ← z XOR (z >> 15)
z ← (z · 0x846CA68B) mod 2^32
z ← z XOR (z >> 16)
Return z AND 0xffffffff
```

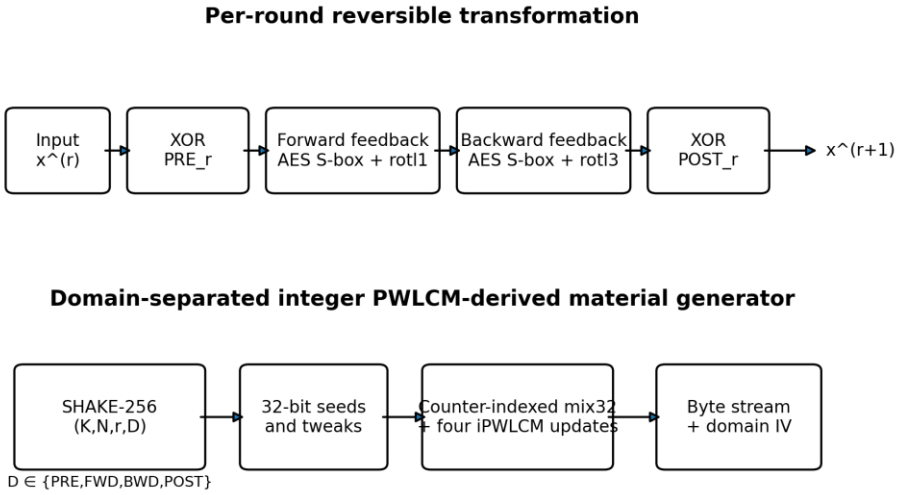


Fig. 2. Per-round reversible data path and the counter-indexed integer PWLCM-derived material generator.

3.3 Six-round reversible feedback core

Let x(r) be the byte vector entering round r. Four separately derived streams PRE_r, KF_r, KB_r and POST_r are generated. First, x is whitened with PRE_r. The forward recurrence scans from left to right, and the backward recurrence scans from right to left. S denotes the AES S-box, ROTL8 is an 8-bit rotation, and all additions are modulo 256:

$$f_i = S[(x_i + \mathrm{ROTL8}(f_{i-1}, 1)) \bmod 256] \oplus KF(r,i), \quad f_{-1} = IV_F \tag{3}$$

$$b_i = S[(f_i + \mathrm{ROTL8}(b_{i+1}, 3)) \bmod 256] \oplus KB(r,i), \quad b_n = IV_B \tag{4}$$

$$x(r+1,i) = b_i \oplus POST(r,i) \tag{5}$$

Decryption is exact because S is bijective and each chaining value needed to invert a recurrence is present in the already available ciphertext-side sequence. For the forward inverse, S^−1(f_i⊕KF_i) is reduced by ROTL8(f_{i−1},1); the backward inverse is analogous while scanning from the end. PRE and POST whitening are self-inverse XORs. Six rounds are used as a conservative evaluation margin: the round experiment shows that image statistics saturate earlier, so six rounds are not justified by a formal reduced-round bound.

Algorithm 2 — Encryption core

```
Input: plaintext bytes P, K, N; R = 6
x ← P
for r = 0 ... R−1:
   (PRE,KF,KB,POST,IVF,IVB) ← domain-separated material
   x ← x XOR PRE
```

```
    x ← ForwardFeedback(x,KF,IVF,AES_SBOX)
    x ← BackwardFeedback(x,KB,IVB,AES_SBOX)
    x ← x XOR POST
Return ciphertext body C ← x
```

Algorithm 3 — Decryption core

```
Input: ciphertext body C, K, N; R = 6
x ← C
for r = R−1 ... 0:
    (PRE,KF,KB,POST,IVF,IVB) ← domain-separated material
    x ← x XOR POST
    next ← IVB
    for i = n−1 ... 0:
        v ← x[i]
        t ← S^−1[v XOR KB[i]]
        x[i] ← (t − ROTL8(next,3)) mod 256
        next ← v
    prev ← IVF
    for i = 0 ... n−1:
        v ← x[i]
        t ← S^−1[v XOR KF[i]]
        x[i] ← (t − ROTL8(prev,1)) mod 256
        prev ← v
    x ← x XOR PRE
Return plaintext P ← x
```

3.4 Authentication

A distinct 256-bit MAC key is derived exactly as K_A=SHAKE256(prefix ‖ “AUTH|” ‖ K ‖ “|” ‖ N, 32 bytes), where prefix already ends in “|”. Let L8(x) denote the unsigned 64-bit big-endian encoding of the byte length of x. The authenticated message is L8(AAD) ‖ AAD ‖ L8(C) ‖ C ‖ N. The tag is the first 16 bytes (128 bits) of HMAC-SHA-256:

$$T = \mathrm{Trunc}_{128}(\text{HMAC-SHA-256}(K_A,\ L_8(AAD) \| AAD \| L_8(C) \| C \| N)) \tag{6}$$

The receiver recomputes T with the same framing, compares the 16-byte tags with a constant-time comparison, and rejects before any decryption on mismatch. AAD may be empty. Ciphertext and AAD lengths are measured in bytes and encoded in exactly eight big-endian bytes. The public message record is (N,C,T), while AAD is supplied alongside the record but is not encrypted. The authentication layer therefore detects ciphertext/AAD modification in the implemented interface; this paper does not claim that the custom encryption core itself is an AEAD primitive with a formal proof.

3.5 Known-answer vector and correctness

A final independent validation reran 32 seal/open round trips on payloads from 1 to 65536 bytes under independently generated keys and nonces; all 32 returned the original plaintext exactly. The following vector was also rechecked against the final implementation.

Table 1 Known-answer test for RIFT-PWLCM

Quantity	Hexadecimal value
Key K	000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
Nonce N	000102030405060708090a0b0c0d0e0f
Plaintext	000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
Ciphertext	ffebbc645d61595b9f9fbc2f6e29b58cb100421ac6bf9fac253ab50d376b9cdc
Tag	f3de5be59b494ee90cec696a58447300

4. Experimental methodology

4.1 Implementation and environment

All reported simulations were executed from the final RIFT-PWLCM-v1 reference implementation. The environment was Linux 6.18.44 (x86-64), Python 3.13.5, NumPy 2.3.5, SciPy 1.17.0, Numba 0.65.1, scikit-image 0.26.0, and Matplotlib 3.10.8 on an Intel Xeon Platinum 8573C allocation (5 online CPU cores; the cipher loops are JIT-compiled with Numba). The main experiment seed was 20260913, with explicitly different seeds for several follow-up screens. Values in the manuscript are read from generated CSV/NumPy outputs rather than manually invented. The SHA-256 digest of the core reference source file rift_pwlcm_experiments.py used for these results is 6e722a9f910ad2e56e506534d1707656a6b811bc7dffd2dcea88446c06504ddd.

4.2 Input set and image metrics

The test set contains five natural 256×256 grayscale images (camera, coins, moon, page and clock) from scikit-image, two constant 256×256 images (all black and all white), and a 256×256 RGB astronaut image. Entropy, 256-bin χ^2 uniformity, and horizontal/vertical/diagonal adjacent correlation are reported. Plaintext sensitivity uses 15 random single-bit flips on each of seven grayscale inputs (105 trials total); NPCR and UACI are computed as in [11].

4.3 Avalanche, rounds, and component ablation

Every one of the 256 key bits and every one of the 128 nonce bits is flipped individually on a 16-KiB buffer. A sampled avalanche-dependency matrix uses 16 independent 4096-byte messages, the first 64 plaintext input bits, and 128 evenly sampled output bits. Because each cell has only 16 Bernoulli observations, the matrix is reported as a sampled screen—not as a definitive Strict Avalanche Criterion proof. Round counts 1–8 and three component variants (full, forward-only, and identity S-box) are evaluated under common plaintext-flip trials.

4.4 Generator and cryptanalytic screens

For each of PRE/FWD/BWD/POST, 262,144 generated bytes are tested for entropy, χ^2 uniformity and autocorrelation through lag 32; pairwise Pearson correlations across domain streams are also measured. The differential screen uses 5,000 random 64-byte plaintext pairs for each of three differences and evaluates the first output-difference byte plus 512-bit output Hamming weight. The linear screen evaluates 1,024 single-bit input/output

mask pairs over 10,000 random 64-byte plaintexts. These sample sizes are screening tools only; they do not provide differential or linear bounds.

4.5 Randomness, authentication, misuse, and performance

Eight SP 800-22 formula-based sanity checks are applied to ten 200,000-bit ciphertext streams produced from structured plaintext under distinct nonces. Authentication is tested by 10,000 random single-bit ciphertext corruptions and 100,000 independent random 128-bit tags. A same-nonce experiment compares plaintext-XOR and ciphertext-XOR bit relations over 100 message pairs but does not relax the nonce-uniqueness rule. A 256-key screen records output entropy and χ^2 behavior. Performance is measured separately for core encryption/decryption and seal/open across 1 KiB–1 MiB; the 1 MiB point uses three median runs and smaller points use seven-run measurements.

5. Experimental results

5.1 Ciphertext appearance and basic image statistics

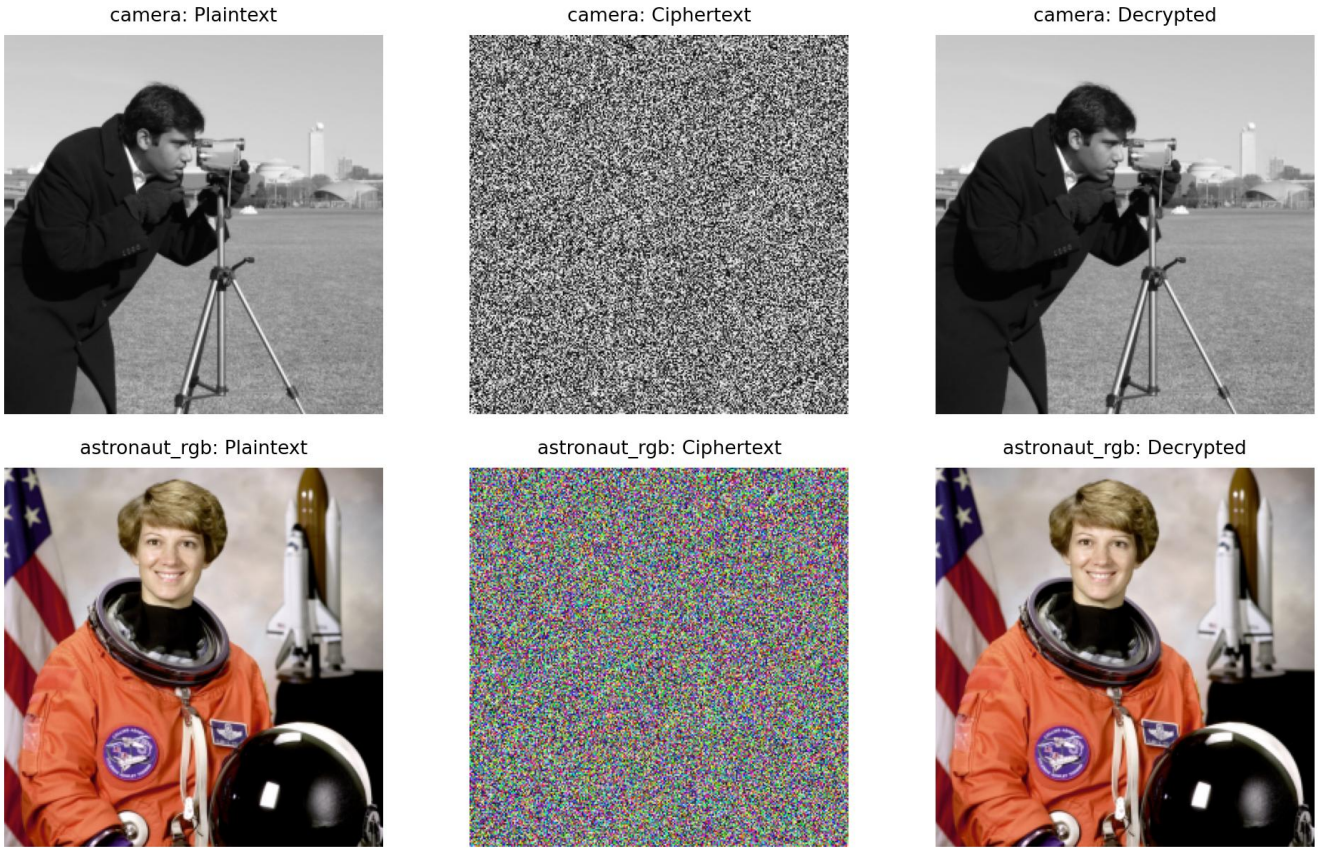


Fig. 3. Plaintext, ciphertext, and exact decryption for representative grayscale and RGB inputs.

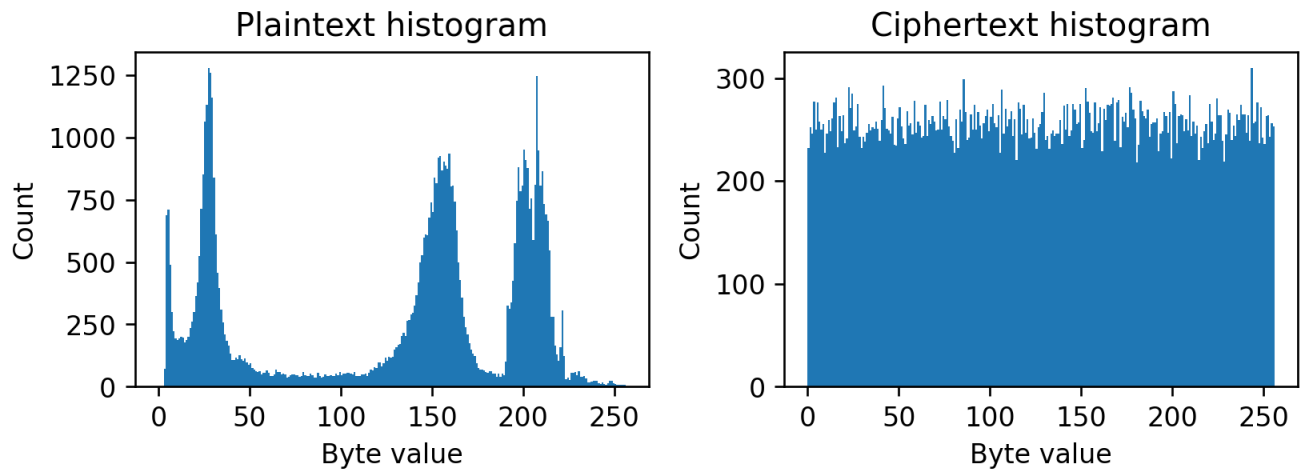


Fig. 4. Camera plaintext and ciphertext byte histograms.

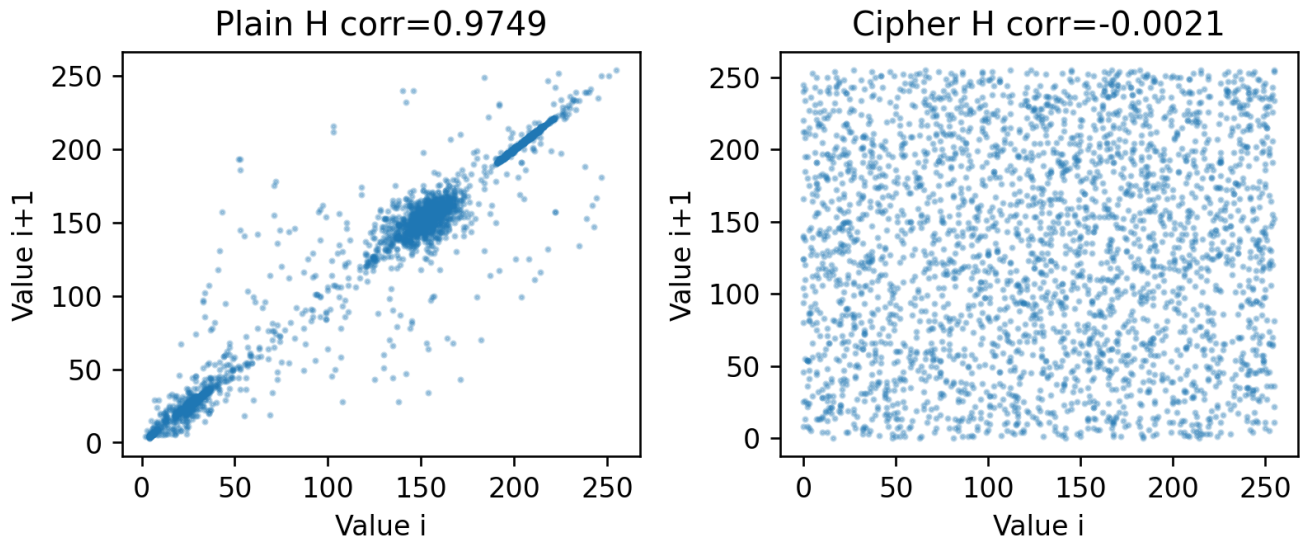


Fig. 5. Horizontal adjacent-value scatter for camera plaintext and ciphertext.

Table 2 Ciphertext statistics on eight test inputs

Input	Bytes	Entropy	χ^2 p	Corr. H	Corr. V	Corr. D
camera	65536	7.997275	0.607	-0.0021	-0.0052	+0.0069
coins	65536	7.997509	0.907	-0.0019	+0.0053	+0.0013
moon	65536	7.997300	0.657	-0.0029	+0.0011	+0.0028
page	65536	7.996996	0.222	-0.0024	-0.0033	-0.0093
clock	65536	7.997608	0.960	-0.0018	-0.0036	+0.0023
black	65536	7.997095	0.325	+0.0048	-0.0026	+0.0002
white	65536	7.997118	0.392	-0.0001	-0.0018	+0.0024
astronaut_rgb	196608	7.999106	0.691	-0.0003	+0.0034	+0.0022

Across all inputs, ciphertext entropy ranges from 7.996996 to 7.999106 bits/byte. The smallest χ^2 p-value is 0.222; the largest absolute adjacent correlation is 0.00927. The constant black and white images do not produce degenerate ciphertext statistics, and the RGB experiment demonstrates that the byte-oriented core is not restricted to grayscale data.

5.2 Plaintext sensitivity

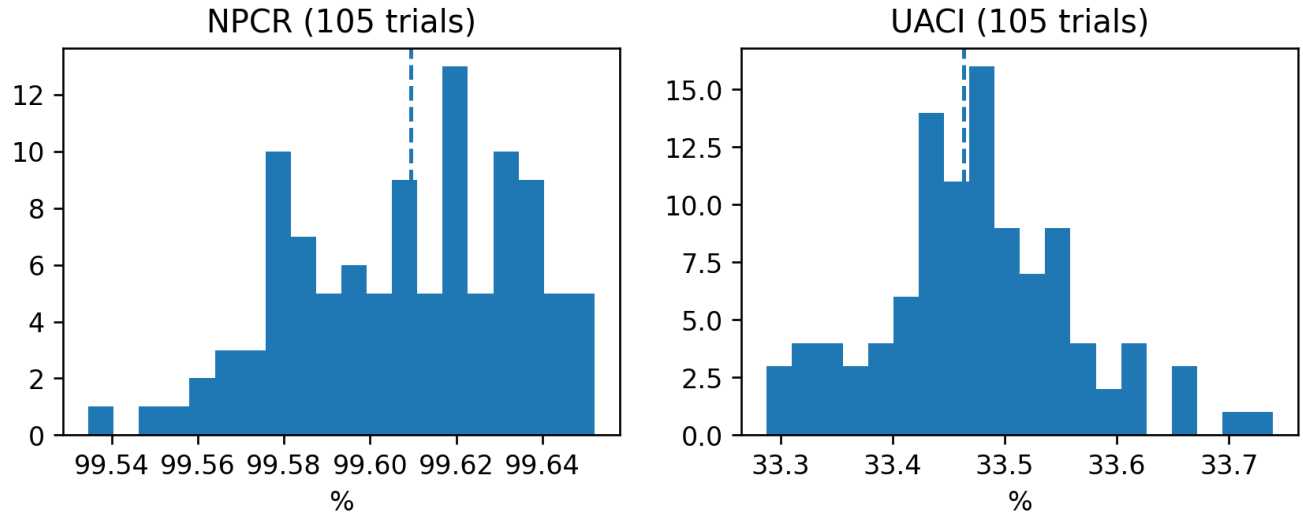


Fig. 6. NPCR and UACI distributions over 105 single-bit plaintext perturbations (15 trials for each of seven grayscale inputs).

The 105-trial mean NPCR is 99.6080% (SD 0.0263%, minimum 99.5346%). Mean UACI is 33.4758% (SD 0.0885%, range 33.2879–33.7393%). These values are close to random-image expectations, but they are diffusion measurements rather than cryptanalytic bounds.

5.3 Key, nonce, and sampled avalanche behavior

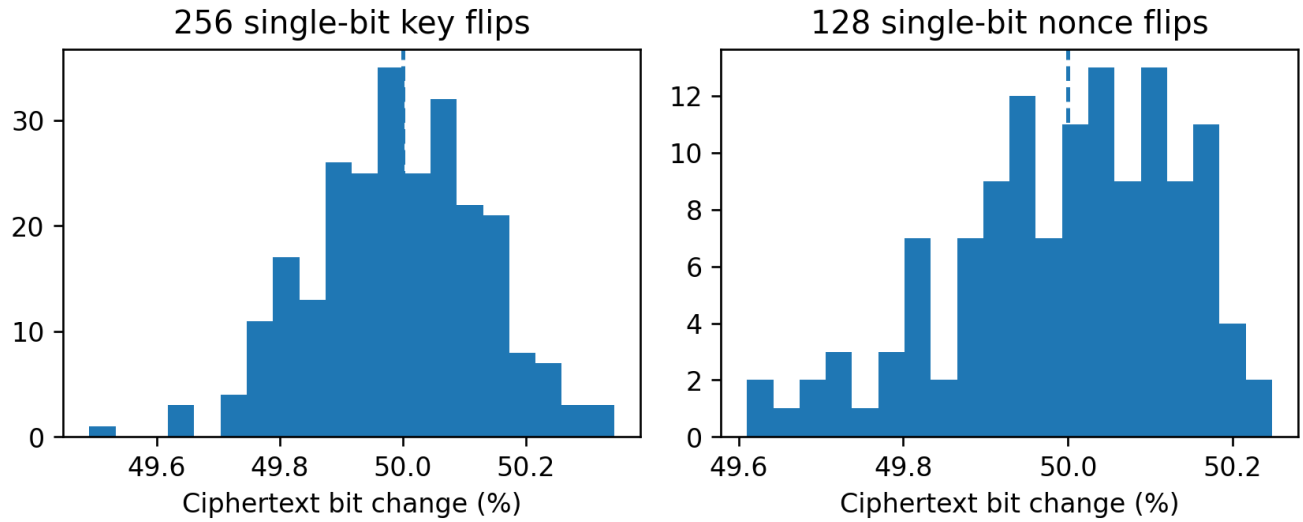


Fig. 7. Ciphertext bit-change distributions for all 256 single-bit key flips and all 128 single-bit nonce flips.

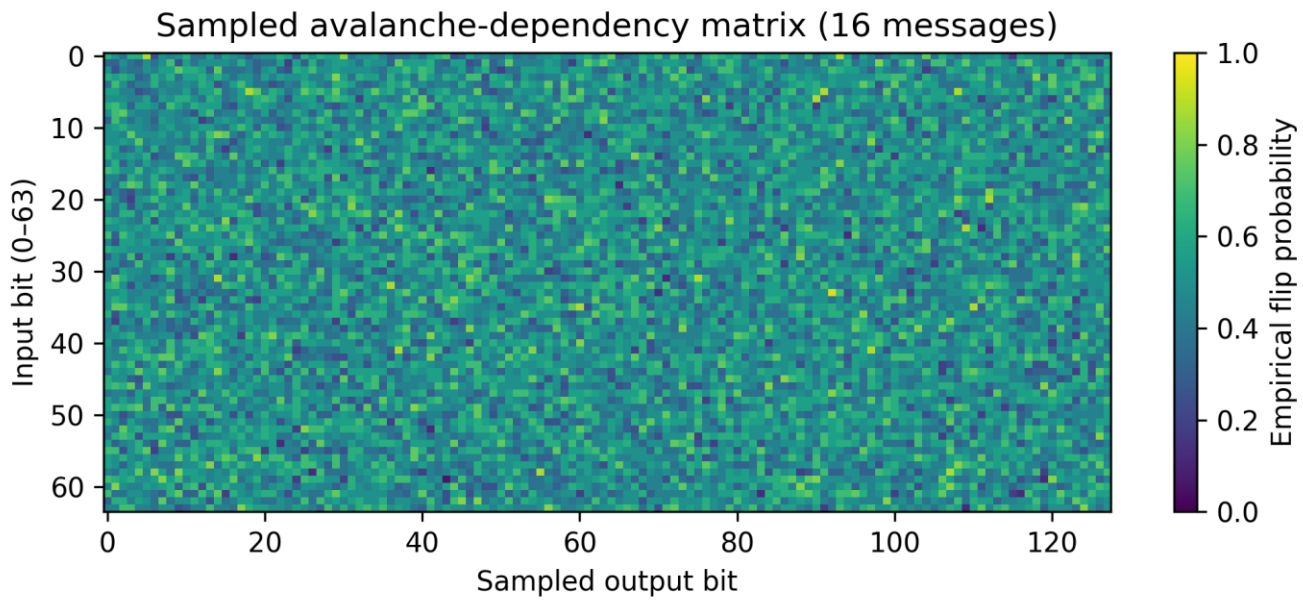


Fig. 8. Sampled avalanche-dependency matrix: 64 input bits × 128 sampled output bits over 16 independent messages.

Single-bit key flips change 49.9926% of ciphertext bits on average (SD 0.1386%), while nonce flips change 49.9970% (SD 0.1393%). The sampled avalanche matrix has overall mean 0.499985; row means lie between 0.4761 and 0.5269. Individual cells range more widely because only 16 messages contribute to each estimate; the heatmap is therefore intentionally not labelled a full SAC certification.

5.4 Round count and ablation

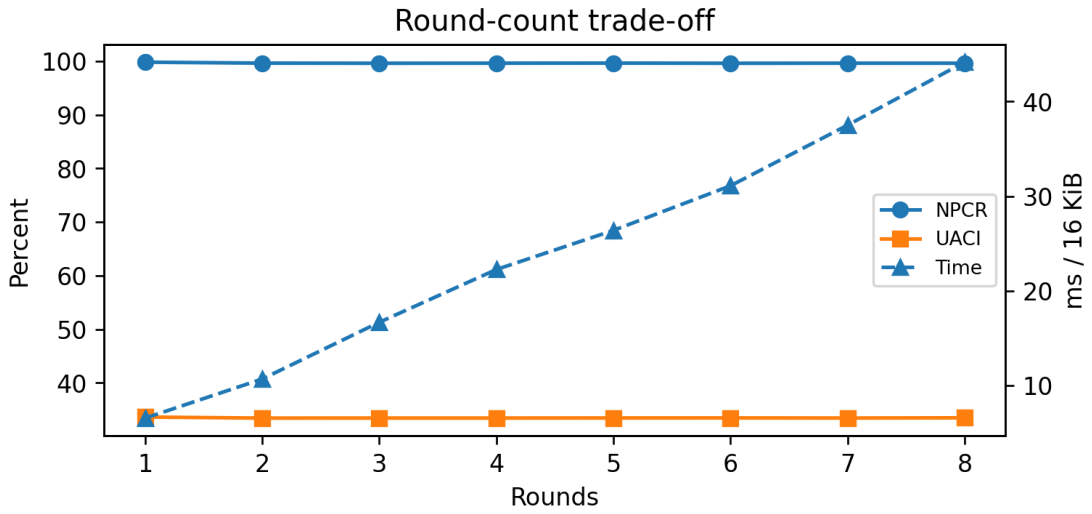


Fig. 9. Measured diffusion/performance trade-off for one through eight rounds on the common 16-KiB protocol.

Table 3 Round-count screen

Rounds	NPCR %	Min NPCR	UACI %	ms/16 KiB
1	99.795	99.591	33.636	6.57

2	99.618	99.524	33.437	10.70
3	99.598	99.536	33.448	16.69
4	99.610	99.524	33.445	22.26
5	99.625	99.542	33.464	26.39
6	99.596	99.524	33.471	31.10
7	99.609	99.469	33.449	37.49
8	99.606	99.487	33.490	44.15

One round already propagates a change across most positions, but its mean UACI is slightly high and the construction would have no round margin. From two rounds onward, the image statistics are essentially saturated. Six rounds are retained as a conservative engineering choice; this is a margin policy, not a formal reduced-round security theorem.

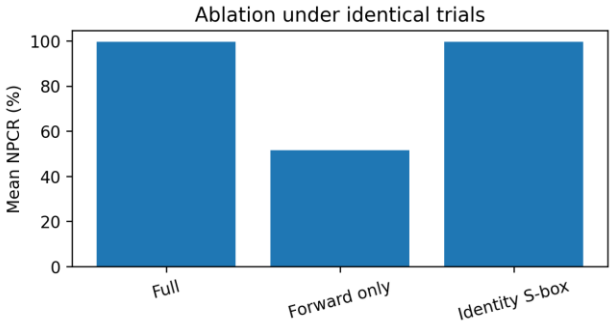


Fig. 10. Component ablation under identical six-round plaintext-flip trials.

Table 4 Ablation results

Variant	NPCR	Min	UACI	Entropy
Full	99.596	99.524	33.471	7.9885
Forward only	51.472	1.770	17.370	7.9893
Identity S-box	99.584	99.500	33.440	7.9897

Removing the backward recurrence is catastrophic for the tested diffusion measure: mean NPCR falls to 51.47% and the worst trial to 1.77%. Bidirectionality is therefore experimentally essential. Replacing the AES S-box by the identity leaves NPCR/UACI nearly unchanged, illustrating an important limitation of image statistics: they cannot establish the security contribution of a nonlinear substitution layer. The S-box is retained for structural nonlinearity, not because NPCR proves its value.

5.5 Generator statistics

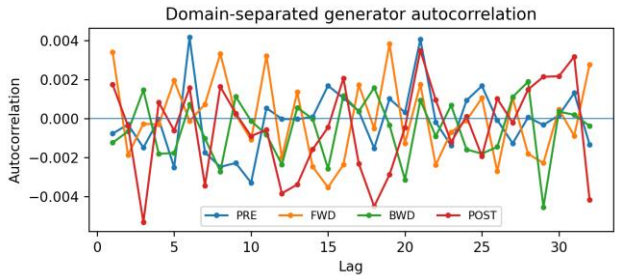


Fig. 11. Autocorrelation through lag 32 for the four independently labelled round-material domains.

Table 5 Generator screen (262,144 bytes/domain)

Domain	Entropy	χ^2 p	Max |ρ|
PRE	7.999450	0.996	0.00416
FWD	7.999240	0.169	0.00383
BWD	7.999365	0.860	0.00455
POST	7.999335	0.710	0.00532

All four streams have entropy above 7.999240 bits/byte, χ^2 p-values between 0.169 and 0.996, and maximum |autocorrelation| no greater than 0.00532. The largest absolute cross-domain Pearson correlation is 0.00745. These observations support implementation sanity and domain separation at the tested scale; they are not a pseudorandom-function proof.

5.6 Differential and linear screening

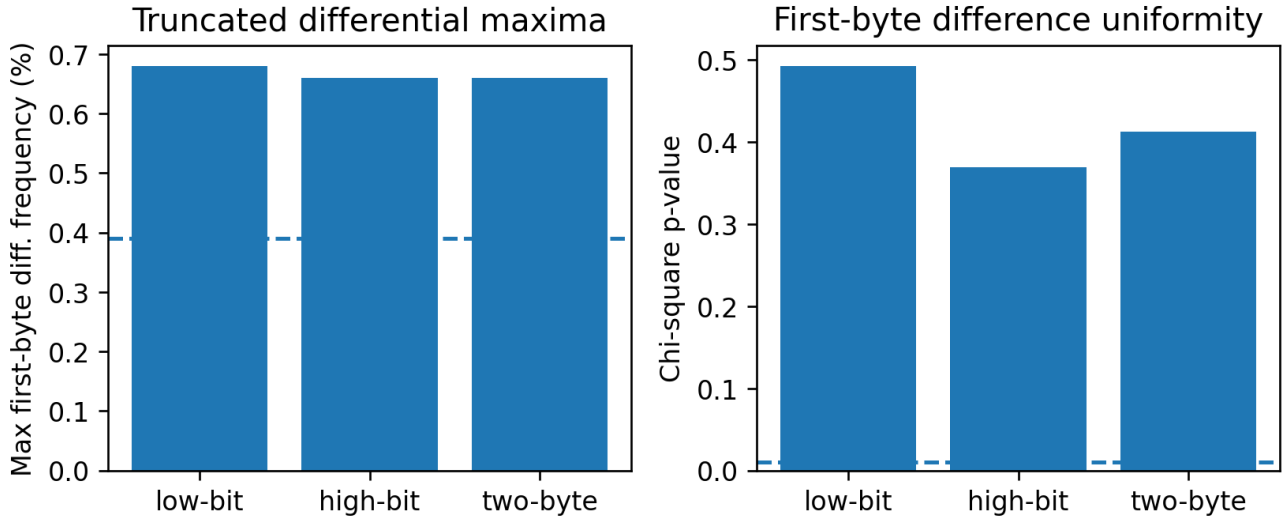


Fig. 12. Truncated differential screen: maximum first-byte difference frequency and χ^2 uniformity p-value.

Table 6 Bounded differential screen

Difference	Pairs	Max first-byte	χ^2 p	HW / 512 bits
low-bit	5000	0.680%	0.492	255.97±11.10
high-bit	5000	0.660%	0.369	256.04±11.49
two-byte	5000	0.660%	0.412	255.80±11.35

For all three tested input differences, the first ciphertext-difference byte is statistically compatible with uniformity at this sample size (p=0.369–0.492), and the mean 512-bit output Hamming weight is approximately 256. This does not bound the probability of the best differential characteristic: it only rules out a gross first-byte bias for these three sampled differences.

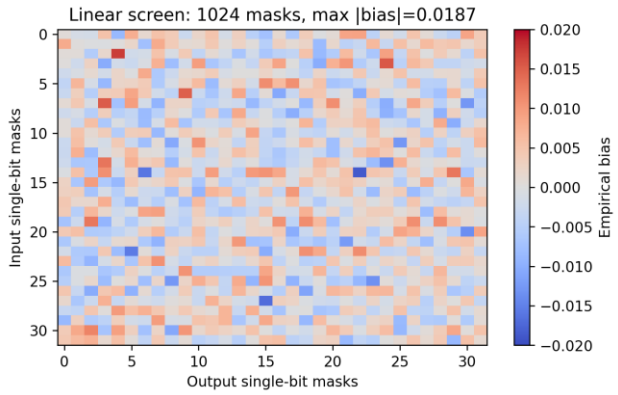


Fig. 13. Empirical bias for 1,024 single-bit input/output mask pairs over 10,000 random 64-byte messages.

The maximum observed absolute linear bias is 0.0187, and the mean absolute bias is 0.00404. For 10,000 unbiased Bernoulli samples, the single-mask standard deviation is approximately 0.0050 and the expected mean absolute sampling bias is about 0.00399, close to the observed average. The maximum corresponds to about 3.74 single-mask standard deviations; with 1,024 inspected masks, an extreme of this scale is not by itself evidence of an exploitable approximation. Larger systematic-mask searches remain necessary.

5.7 SP 800-22 sanity checks

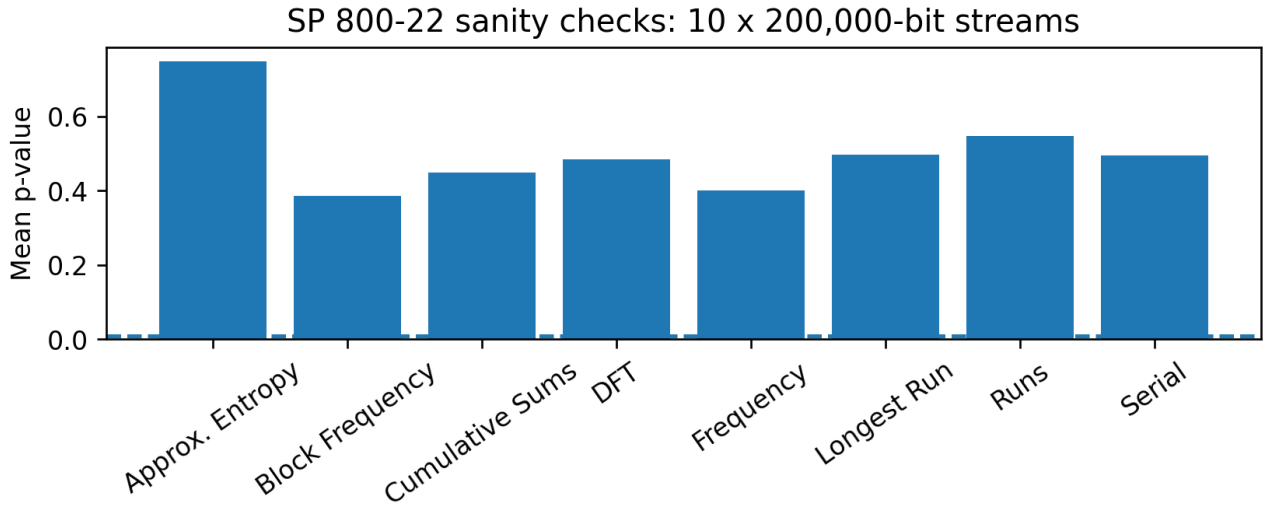


Fig. 14. Mean p-values for eight SP 800-22 formula-based sanity checks over ten 200,000-bit ciphertext streams.

Table 7 NIST SP 800-22 sanity-check summary

Test	Mean p	Min p	Pass
Approx. Entropy	0.750	0.451	100%
Block Frequency	0.386	0.014	100%
Cumulative Sums	0.449	0.162	100%
DFT	0.486	0.134	100%
Frequency	0.401	0.173	100%
Longest Run	0.497	0.102	100%

Runs	0.548	0.028	100%
Serial	0.496	0.070	100%

Every tested stream passed every selected test at α=0.01; the smallest recorded p-value is 0.01395. Following NIST guidance [1], [2], these outcomes are interpreted only as the absence of gross statistical defects in the tested streams. They are not evidence of cryptographic security.

5.8 Authentication, nonce reuse, and key screening

All 10,000 single-bit ciphertext corruptions were rejected by tag verification. None of 100,000 independent random 128-bit tags was accepted. These observations are consistent with the expected behavior of a 128-bit truncated HMAC tag, but the empirical zero count is not itself an estimate of a 2^−128 probability. In the prohibited same-nonce screen, plaintext-XOR and ciphertext-XOR bits were equal with mean probability 0.499209 (SD 0.004015) over 100 pairs, so no direct CTR-like XOR relation was visible. Nonce reuse nevertheless remains unsupported because the entire round-material schedule repeats.

The 256-key generator screen produced no repeated tested seed/tweak tuple. Mean output entropy was 7.954544 bits/byte and the minimum was 7.942089 on the short screening buffers. Two of 256 χ^2 tests rejected uniformity at α=0.01, close to the 2.56 false rejections expected on average under the null. This is a weak-key screen, not a proof that weak or equivalent internal states do not exist.

5.9 Performance

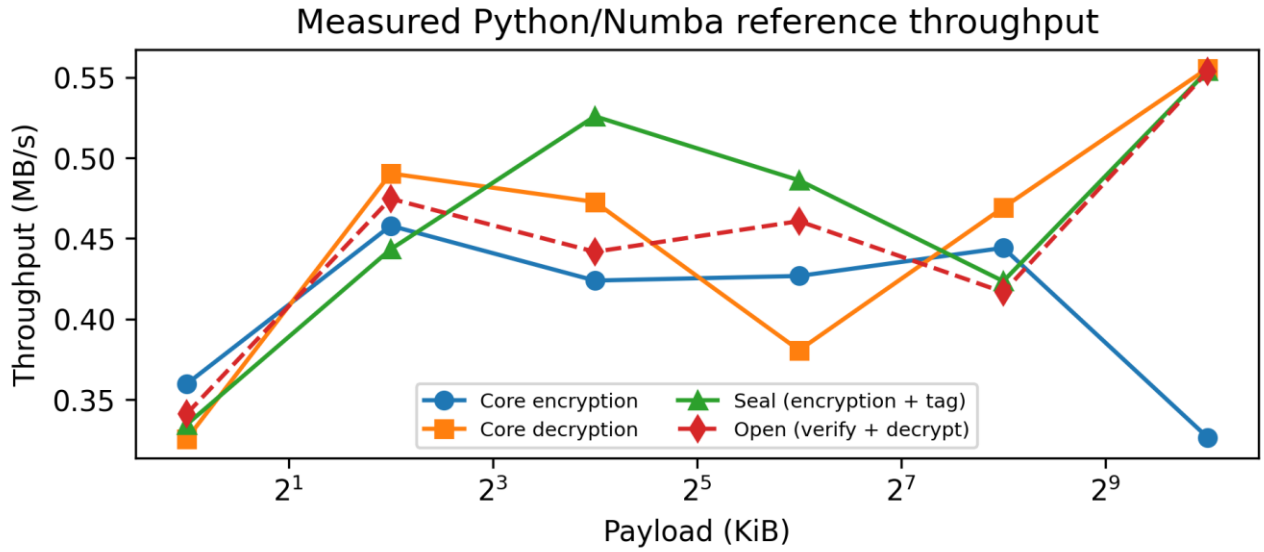


Fig. 15. Measured Python/Numba reference throughput, including the separately measured 1-MiB point.

Table 8 Reference implementation throughput

KiB	Enc MB/s	Dec MB/s	Seal MB/s	Open MB/s
1	0.360	0.325	0.334	0.341
4	0.458	0.490	0.443	0.475
16	0.424	0.473	0.526	0.442
64	0.427	0.381	0.486	0.461
256	0.444	0.469	0.423	0.417

1,024	0.327	0.556	0.554	0.554

For 64–256 KiB payloads, measured core encryption is approximately 0.43–0.44 MB/s and seal throughput approximately 0.42–0.49 MB/s. The separately measured 1-MiB seal/open throughput is about 0.554 MB/s. The implementation is therefore feasible for offline image/file experiments but is not a real-time or lightweight implementation. No energy, gate-count, code-size, side-channel, or embedded-memory measurements are claimed.

6. Discussion

6.1 What the redesign resolves

The final construction removes four weaknesses that commonly complicate interpretation of hybrid image ciphers. First, all chaotic-map arithmetic is fixed-point integer arithmetic, eliminating binary64 seed rounding and platform-dependent floating-point behavior. Second, public parameters replace genetic optimization, so there is no train/validation overfitting argument and no suggestion that evolutionary tuning creates security. Third, DNA notation is eliminated; every operation is expressed directly as conventional byte arithmetic. Fourth, there is no global rank-sort permutation, so the data path is O(nR) rather than O(n log n) per permutation stage.

Domain labels also address a subtler issue: PRE, forward-key, backward-key and POST material are generated from separate SHAKE inputs instead of overlapping bit slices of one state word. The measured cross-domain correlations are near zero at the tested scale. Authentication adds an integrity failure mode that the earlier confidentiality-only class of designs lacks: modified ciphertext is rejected before inverse processing.

6.2 What the experiments do not establish

RIFT-PWLCM remains a custom construction. Its counter-indexed expansion has not been reduced to a standard PRF assumption; no algebraic-degree analysis, SAT/MILP trail search, integral attack, meet-in-the-middle attack, impossible-differential analysis, related-key study, state-recovery bound, or side-channel evaluation is presented. The differential and linear sections are bounded screens. The S-box ablation further demonstrates that NPCR/UACI cannot distinguish an important structural component, reinforcing the paper's central methodological point.

The use of SHAKE-256 for seed derivation also raises a legitimate baseline question: a SHAKE-only message-length expansion would be simpler and rests on a standardized primitive. RIFT-PWLCM retains the integer map because the objective is to study a reproducible chaos-derived expansion under cryptographically separated seeds, not to claim superiority to SHAKE or standardized AEAD. Any deployment decision should prefer established constructions such as Ascon-AEAD128 [5] unless a specific, independently reviewed reason favors this research design.

6.3 Feasibility and novelty scope

The implementation is completely specified, reversible, deterministic for fixed (K,N), and reproducible in integer arithmetic. The novelty claimed here is architectural: the combination of counter-indexed Q0.32 PWLCM-derived expansion, SHAKE-separated PRE/FWD/BWD/POST roles, reversible bidirectional byte feedback, and explicit encrypt-then-MAC output, together with a negative-result-aware evaluation. It is not claimed that PWLCM, SHAKE, HMAC, AES substitution, or feedback diffusion is individually new.

7. Conclusion

This paper presented RIFT-PWLCM, a redesigned authenticated image-encryption research construction intended to be simpler to reproduce and easier to analyze than feature-stacked DNA/GA/floating-point chaos ciphers. The final method uses fixed Q0.32 public parameters, domain-separated SHAKE-256 seed derivation, counter-indexed integer map expansion, six rounds of reversible forward/backward nonlinear feedback, and a 128-bit HMAC-SHA-256 tag. All numerical results and figures were regenerated from that exact design. The implementation round-

trips exactly, produces near-uniform image statistics across natural, constant and RGB inputs, exhibits near-50% key/nonce avalanche, passes the selected randomness sanity checks, and rejects every tested tampering attempt. At the same time, the paper deliberately avoids turning those observations into a formal security claim. The most important future work is independent cryptanalysis of the counter-indexed expansion and feedback composition, formal nonce-respecting security modeling, comparison with a SHAKE-only baseline, and a constant-time compiled implementation benchmarked against standardized AEAD.

Declarations

Code and data availability—The exact Python/Numba reference implementation, experiment drivers, generated CSV/NumPy outputs, and figure-generation scripts used for the reported results are supplied with this manuscript as the supplementary archive RIFT-PWLCM_Supplementary_Reproducibility.zip. The archive includes a README and SHA-256 checksums. The core source hash is also reported in Section IV-A. Conflict of interest—The authors declare no known competing financial interests or personal relationships that could have influenced the work. Funding—This research did not receive a specific grant from public, commercial, or not-for-profit funding agencies

REFERENCES

1. L. Bassham et al., “A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,” NIST SP 800-22 Rev. 1a, 2010.
2. National Institute of Standards and Technology, “Decision to Revise NIST SP 800-22 Rev. 1a,” Apr. 19, 2022.
3. National Institute of Standards and Technology, “SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions,” FIPS PUB 202, Aug. 2015.
4. National Institute of Standards and Technology, “Advanced Encryption Standard (AES),” FIPS PUB 197-upd1, May 2023, doi: 10.6028/NIST.FIPS.197-upd1.
5. M. S. Turan, K. McKay, J. Kang, J. Kelsey, and D. Chang, “Ascon-Based Lightweight Cryptography Standards for Constrained Devices,” NIST SP 800-232, Aug. 2025, doi: 10.6028/NIST.SP.800-232.
6. G. Alvarez and S. Li, “Some basic cryptographic requirements for chaos-based cryptosystems,” Int. J. Bifurcation Chaos, vol. 16, no. 8, pp. 2129–2151, 2006.
7. S. Li, G. Chen, and X. Mou, “On the dynamical degradation of digital piecewise linear chaotic maps,” Int. J. Bifurcation Chaos, vol. 15, no. 10, pp. 3119–3151, 2005.
8. J. S. Teh, M. Alawida, and Y. C. Sii, “Implementation and practical problems of chaos-based cryptography revisited,” J. Inf. Security Appl., vol. 50, Art. no. 102421, 2020.
9. J. Fridrich, “Symmetric ciphers based on two-dimensional chaotic maps,” Int. J. Bifurcation Chaos, vol. 8, no. 6, pp. 1259–1284, 1998.
10. G. Chen, Y. Mao, and C. K. Chui, “A symmetric image encryption scheme based on 3D chaotic cat maps,” Chaos Solitons Fractals, vol. 21, no. 3, pp. 749–761, 2004.
11. Y. Wu, J. P. Noonan, and S. Agaian, “NPCR and UACI randomness tests for image encryption,” J. Sel. Areas Telecommun., pp. 31–38, Apr. 2011.
12. U. Zia et al., “Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains,” Int. J. Inf. Security, vol. 21, no. 4, pp. 917–935, 2022.
13. K. Jain et al., “A lightweight multi-chaos-based image encryption scheme for IoT networks,” IEEE Access, vol. 12, pp. 62118–62148, 2024, doi: 10.1109/ACCESS.2024.3377665.
14. B. Rezaei, M. Mobasseri, and R. Enayatifar, “A secure, efficient and super-fast chaos-based image encryption algorithm for real-time applications,” J. Real-Time Image Process., vol. 20, no. 2, Art. no. 30, 2023, doi: 10.1007/s11554-023-01289-5.
15. H. Zhang and H. Hu, “An image encryption algorithm based on a compound-coupled chaotic system,” Digit. Signal Process., vol. 146, Art. no. 104367, 2024, doi: 10.1016/j.dsp.2023.104367.
16. H. Wen and Y. Lin, “Cryptanalysis of an image encryption algorithm using quantum chaotic map and DNA coding,” Expert Syst. Appl., vol. 237, Art. no. 121514, 2024, doi: 10.1016/j.eswa.2023.121514.
17. X. Chai, Z. Gan, Y. Lu, Y. Chen, and D. Han, “A novel image encryption algorithm based on the chaotic system and DNA computing,” Int. J. Mod. Phys. C, vol. 28, no. 5, 2017.
18. Y. Wang, K. W. Wong, C. Li, and Y. Li, “A novel method to design S-box based on chaotic map and genetic algorithm,” Phys. Lett. A, vol. 376, nos. 6–7, pp. 827–833, 2012.
19. Y. Wang et al., “A genetic algorithm for constructing bijective substitution boxes with high nonlinearity,” Inf. Sci., vol. 523, pp. 152–166, 2020.
20. J. Xu, W. Liu, K. Zhang, and E. Zhu, “DNA coding theory and algorithms,” Artif. Intell. Rev., vol. 58, Art. no. 178, 2025, doi: 10.1007/s10462-025-11132-x.
21. H. Krawczyk, M. Bellare, and R. Canetti, “HMAC: Keyed-Hashing for Message Authentication,” RFC 2104, Feb. 1997..