MDPI

Article

Supplementary Materials: A Finite-State Audit of Chaotic Maps for Image Encryption — Image Fraction, Cyclic Fraction and Seed Diversity

Mua'ad Alfarajat [1,*], Abdullah Alhaj [2] and Abeer Al-Hyari [3]

[1] Department of Computer Information Systems, School of Information Technology and Systems, The University of Jordan, Aqaba 77110, Jordan; m.abufaraj@ju.edu.jo
[2] Department of Information Technology, The University of Jordan, Aqaba 77110, Jordan; aa.alhaj@ju.edu.jo
[3] Electrical Engineering Department, Al-Balqa Applied University, As Salt 19117, Jordan; abeer.hyari@bau.edu.jo
* Correspondence: m.abufaraj@ju.edu.jo

S1. Scope of this document

This file holds the complete tables and the heuristic model that the main paper summarises. Nothing here is needed to follow the argument; everything here is needed to check it. Every number is generated from a record in the artifact, `https://doi.org/10.5281/zenodo.22669885`, by the script named in its caption, and `code/verify.py` in that artifact recomputes the relations they assert.

Section and table numbers in the main paper are cited without the S prefix; tables in this document carry it. Bracketed citations refer to the main manuscript's numbered reference list; this document has no reference list of its own.

S2. Cross-validation against direct enumeration

Proposition 4 is the load-bearing step of the measurement: it is what lets us report the cycle structure of a map on $2^{(d+1)w}$ states from a table on 2^{dw}. At small widths the joint map can be built outright, so the identity can be checked against ground truth rather than assumed.

We enumerate Φ directly for $w = 2, 3, 4, 5$, for every odd increment in $\{1, 3, 2^w - 1\}$, and for seven structurally distinct base maps: the constant map, the identity, a random permutation, the affine maps $2x + 1$ and $6x + 1$, the quantized logistic map, and a uniformly random self-map. That is 84 runs over 77 distinct cells, since at $w = 2$ the set $\{1, 3, 2^w - 1\}$ lists $\gamma = 3$ twice. In each we decompose the functional graph of Φ and check four things against the section route: that every joint cycle length is divisible by 2^w; that the joint cycle multiset equals $\{2^w L\}$ over the section cycles; that $\beta_\Phi = \beta_{\Psi_0}$; and that $\alpha_\Phi = \alpha_T$ with $n_j(\Phi) = 2^w n_j(T)$ for every j.

All 77 cells agree on all four checks. The families are chosen to span the structural cases rather than to be realistic: the constant map gives maximal contraction, the permutation none, and the random map the generic case. In particular the permutation rows confirm that the section identity holds where the contraction heuristic of Section 4.8 does not, which is the case that matters, since Theorem 1 and Propositions 2 and 4 are supposed to hold for every map.

S3. Increment sensitivity, per map

Table S1 gives the full increment sweep summarised in Section 4.9 of the main paper: the range of the longest section cycle, the collapse fraction and the distribution of cycle

counts over the sampled increments, per map, at $b = 12$ and $b = 16$ under both quantizers. The wider 500-increment sample at $w = 12$ that revises the ceiling from two to three is in the record `robustness_C.json`.

Table S1. Sensitivity of the section-cycle length to the increment, per map, generated from the records. Sampling design: twenty-four distinct odd γ per cell, drawn without replacement from a seed fixed in the analysis plan. The exception is the $(3, 4)$ cell, where only eight odd increments exist and the sweep is therefore exhaustive. Every cell is run under both quantizers. L is the range of the longest section cycle over the sampled increments; "coll." is the fraction of increments giving a single cycle of length one; "cycles" is the distribution of the number of section cycles over the sampled increments, read as *count:frequency*. The rule written into the analysis plan required $L \leq 4$ on every $d = 1$ row and $L \geq 10$ on at least one $d \geq 2$ row. No one-dimensional cell exceeded $L = 2$ under either quantizer. The doubling and affine rows are the closed-form controls of Propositions 5 and 6 of the main paper.

		truncation			round-to-nearest		
map	b	L	coll.	cycles	L	coll.	cycles
doubling	12	1	1.00	1:24	1	1.00	1:24
affine, a=6	12	1	1.00	1:24	1	1.00	1:24
logistic	12	1–2	0.96	1:24	1–2	0.88	1:24
+ 1D-ECE	12	1–2	0.92	1:23, 2:1	1–2	0.96	1:24
sine	12	1	0.92	1:22, 2:2	1	0.96	1:23, 2:1
+ 1D-ECE	12	1–2	0.92	1:24	1–2	0.92	1:23, 2:1
tent	12	1–2	0.54	1:19, 2:5	1–2	0.62	1:18, 2:5, 3:1
+ 1D-ECE	12	1–2	0.92	1:23, 2:1	1–2	0.83	1:22, 2:2
Singer	12	1	0.96	1:23, 2:1	1–2	0.96	1:24
+ 1D-ECE	12	1–2	0.62	1:22, 2:2	1–2	0.62	1:18, 2:6
LPWLCM [9]	12	1–2	0.75	1:21, 2:3	1–2	0.83	1:21, 2:3
Chebyshev	12	1	1.00	1:24	1–2	0.96	1:24
cubic	12	1	1.00	1:24	1–2	0.96	1:24
2D-CCC [11]	12	1–11	0.00	1:2, 2:11, 3:8, 4:3	1–18	0.08	1:7, 2:7, 3:7, 4:2, 6:1
3D logistic [10]	12	5–8	0.00	2:2, 3:3, 4:2, 5:1	3	0.00	11:6, 13:1, 18:1
doubling	16	1	1.00	1:24	1	1.00	1:24
affine, a=6	16	1	1.00	1:24	1	1.00	1:24
logistic	16	1	1.00	1:24	1	1.00	1:24
+ 1D-ECE	16	1–2	0.96	1:24	1–2	0.96	1:24
sine	16	1	1.00	1:24	1	1.00	1:24
+ 1D-ECE	16	1	0.92	1:22, 2:2	1	0.96	1:23, 2:1
tent	16	1–2	0.67	1:18, 2:5, 3:1	1–2	0.79	1:21, 2:3
+ 1D-ECE	16	1	1.00	1:24	1	1.00	1:24
Singer	16	1	1.00	1:24	1	1.00	1:24
+ 1D-ECE	16	1–2	0.88	1:23, 2:1	1–2	0.88	1:23, 2:1
LPWLCM [9]	16	1–2	0.88	1:22, 2:2	1–2	0.83	1:23, 2:1
Chebyshev	16	1	1.00	1:24	1	1.00	1:24
cubic	16	1	1.00	1:24	1	1.00	1:24
2D-CCC [11]	16	1–12	0.04	1:1, 2:6, 3:9, 4:4, 5:3, 7:1	3–21	0.00	1:3, 2:7, 3:5, 4:5, 5:3, 6:1

S4. The random-mapping heuristic [H]

The section map is a composition of 2^w maps on 2^{dw} points. For a composition of independent uniform random maps the surviving image fraction follows [27] $\tau_{j+1} = 1 - e^{-\tau_j}$, decaying like $2/j$. At $d = 1$ the composition length equals the state space and the model gives a surviving image of $m = 2.0$ states. The cycle formulas are asymptotic in m and say nothing at that size; what the figure does suggest is that only one or two states survive, which is consistent with the 25 rows at $L = 1$ and one at $L = 2$ but is not a prediction of them. At $d \geq 2$ the ratio $2^{-(d-1)w}$ is small, the image stays large, and the composition behaves like a random map on $m = \tau_k N$ points with $\frac{1}{2} \ln m$ expected cycles and a longest cycle of order $0.7826\sqrt{m}$ [27]. The heuristic gives 2.4, 3.1, 3.8 and 4.5 cycles at $(2, w)$ for $w = 6, 8, 10, 12$ against 2, 2, 3 and 5 observed, and 3.0 and 5.9 at $(3, 4)$ and $(3, 8)$ against 3 and 5: the right order of magnitude, not a fit. Replacing the composition by a uniform random self-map on m points is a further assumption beyond the mean-field image size, not a consequence of knowing m. The heuristic is also consistent with the absence of

separation at $b = 12$: at $(2, 6)$ the surviving image is 122 states and the composition is still contracting hard.

Table S2 sets the model against the enumeration.

Table S2. Regenerated directly from the recurrence below. Mean-field prediction against enumeration. The section map is a composition of $k = 2^w$ maps on $N = 2^{dw}$ points. Replacing the surviving image size by its mean at each step gives $\tau_{j+1} = 1 - e^{-\tau_j}$, a mean-field approximation to the exact recurrence $\mathbb{E}[R_{j+1} \mid R_j = r] = N(1 - (1 - 1/N)^r)$, and $m = \tau_k N$. The cycle-count and longest-cycle columns then apply the random-mapping asymptotics $\frac{1}{2} \ln m$ and $0.7826\sqrt{m}$ [27]. Both are asymptotic in m and are not meaningful at the $m \approx 2$ of the one-dimensional rows, which are marked accordingly.

		cycles		longest cycle	
(d, w)	m	$\frac{1}{2} \ln m$	observed	$0.7826\sqrt{m}$	observed
$(1, 12)$	2.0	n/a	1	n/a	1 or 2
$(1, 16)$	2.0	n/a	1	n/a	1
$(2, 6)$	121.9	2.4	2	8.6	2
$(2, 8)$	504.8	3.1	2	17.6	15
$(2, 10)$	2039.8	3.8	3	35.3	37
$(2, 12)$	8182.9	4.5	5	70.8	25
$(3, 4)$	436.4	3.0	3	16.3	5
$(3, 8)$	129228.9	5.9	5	281.3	32

Single-increment observations; the increment sweep of Table S1 gives the spread. No prediction interval is quoted because the model is heuristic, not a fitted distribution.

The longest-cycle column is matched much less well. Single-increment observations fall below it at the widest cells, 25 against 71 and 32 against 281; the increment sweep brackets the prediction at $(2, 8)$, where the model gives 17.6 and the sweep observes 1 to 21, but not at $(3, 8)$. We report the model as a qualitative account of how many cycles form, not of how long the longest becomes, and not as a quantitative prediction: the composed maps are translated copies of one map and are strongly dependent, a composition of image size m is not a uniform random self-map on m states, and the τ recurrence is itself a mean-field approximation. This is a heuristic null model rather than a theorem, and is labelled [H] accordingly: the composed maps are specific and translated copies of one map, not independent draws, and Proposition 6 gives a class for which the $d = 1$ answer is exact for reasons unrelated to randomness.

Table S3. Monte Carlo validation of the random-mapping heuristic, $R = 200$ draws per cell. "Independent" composes 2^w independent uniform random maps; "translated" composes 2^w counter-translated copies of *one* uniform random map, which is the dependence structure the section map actually has. Each entry is the mean with a $[5\%, 95\%]$ interval. The last columns are the measured 2D-CCC at $(2, 6)$ and the typical one-dimensional row; dashes mark cells with no measured subject. No large difference between the two ensembles is apparent in these cells; 200 draws do not establish equivalence.

	independent		translated		observed	
(d, w)	cycles	longest	cycles	longest	cycles	longest
$(1, 8)$	1.21 [1, 2]	1.2 [1, 2]	1.20 [1, 2]	1.2 [1, 2]	1	1
$(1, 10)$	1.22 [1, 2]	1.3 [1, 2]	1.23 [1, 2]	1.2 [1, 2]	1	1
$(2, 4)$	2.15 [1, 4]	3.5 [1, 7]	2.10 [1, 4]	3.5 [1, 7]	—	—
$(2, 5)$	2.52 [1, 5]	4.7 [1, 10]	2.33 [1, 5]	4.8 [1, 10]	—	—
$(2, 6)$	2.82 [1, 5]	6.4 [1, 14]	2.88 [1, 5]	6.8 [2, 14]	2	2

Table S3 replaces the point predictions with distributions. Two ensembles are drawn 200 times per cell: a composition of 2^w *independent* uniform random maps, and a composition of 2^w counter-translated copies of *one* uniform random map, which is exactly the

dependence structure the section map has. At every cell the two ensembles give cycle-count means within 0.2 of each other, longest-cycle means within 0.4 at the displayed precision, and 5–95% intervals that coincide or differ by one at an endpoint; with 200 draws per cell that is no large difference, not a demonstration of equivalence, and we do not claim more. The draws are in the artifact for anyone who wants a tolerance test. The one-dimensional intervals are $[1, 2]$ for both cycle count and longest cycle at $w = 8$ and $w = 10$, which is where every measured one-dimensional row sits; the 2D-CCC at $(2, 6)$, with two cycles and longest cycle 2, sits inside the $[1, 5]$ and $[1, 14]$ intervals. The heuristic is therefore consistent with the measurements, but at these widths its intervals are wide enough that it would be consistent with a good deal else; it explains the order of magnitude, not the value.

S5. Multi-step structure and readout sensitivity at 16 bits

Table S4. Structural statistics of the base map T at $b = 16$, truncation. Every column is a property of the uncoupled realization T_w, not of the Weyl-coupled system. A *component* is one cycle together with every state draining into it. The basin column is the largest such component as a fraction of the state space. The tree column is the largest set draining into a *single* cyclic node, so it is at most the basin, and it is smaller whenever the component has more than one cyclic node. The two coincide for the affine controls, whose single component has a single cyclic node. Tail length is the number of steps from a state to the first cyclic state on its orbit, and the mean is over all states.

			cycles		largest		tail	
map	α_T	β_T	count	nodes	basin	tree	max	mean
doubling	0.5000	1.53e-05	1	1	1.000	1.000	16	15.0
affine, a=6	0.5000	1.53e-05	1	1	1.000	1.000	16	15.0
logistic	0.4374	1.21e-03	3	79	0.751	0.724	277	108.6
+ 1D-ECE	0.3794	5.57e-03	5	365	0.847	0.160	272	48.7
sine	0.4484	4.55e-03	6	298	0.737	0.575	429	162.0
+ 1D-ECE	0.3806	3.97e-03	6	260	0.886	0.285	249	81.6
tent	0.5000	5.10e-03	5	334	0.808	0.296	319	104.2
+ 1D-ECE	0.3825	3.92e-03	6	257	0.559	0.211	269	68.4
Singer	0.5770	6.93e-03	6	454	0.575	0.567	593	168.0
+ 1D-ECE	0.6025	3.48e-03	6	228	0.978	0.806	468	210.0
LPWLCM [9]	0.6795	1.26e-02	5	826	0.559	0.167	338	80.1
Chebyshev	0.4129	2.59e-03	5	170	0.983	0.508	182	73.0
cubic	0.6577	6.65e-03	10	436	0.209	0.204	320	96.1
2D-CCC [11]	0.5541	1.42e-03	9	93	0.927	0.727	192	57.9

Two quantities the main paper discusses without tabulating are reported in Tables S4 and S5: how far states are from the cyclic core, and how much the choice of readout matters.

Components, basins and tails

Every weakly connected component of a functional graph contains exactly one cycle, so the number of components is the number of cycles. Table S4 reports both that count and the number of cyclic nodes, which are different quantities: the logistic map has 3 cycles of lengths 17, 23 and 39, hence 79 cyclic nodes. It also separates the largest *component basin*, which aggregates every state draining into any node of one cycle, from the largest *attached tree*, which drains into a single cyclic node. The two differ substantially: for the LPWLCM the largest basin holds 55.9% of the state space while the largest tree holds 16.7%. Basin sizes are checked to partition the state space.

Tail lengths reach 593 steps for the Singer map and average between 49 and 210 steps over the non-affine rows, so the graphs are deep as well as sparse in cyclic states. Component counts run from 1 for the closed-form instances to 10 for the cubic map, and the largest basin drains between 20.9% and all of the space. None of this is visible in α_T.

Table S5. Output-equivalence classes at $b = 16$, truncation, post-update observation, under three readouts of word 0: the decimal byte readout of Appendix A.1 of the main paper, the high eight bits, and the low eight bits. Counts are by exact partition refinement and are run to a stopping certificate: $t^* = \min\{t : c_t = c_{t+1}\}$, after which the count is final. By Lemma 1 of the main paper every count is bounded by $|\operatorname{Im} T| = \alpha_T 2^{16}$. For the 2D-CCC, which runs at $w = 8$ here, the decimal readout has an alphabet of only four symbols (Section 4.11 of the main paper), so its rows measure the readout as much as the map.

		$\lfloor x\,10^{14} \rfloor \bmod 256$		high 8 bits		low 8 bits	
map	$\lvert\operatorname{Im} T\rvert$	classes	t^*	classes	t^*	classes	t^*
doubling	32 768	512	2	32 768	8	128	1
affine, a=6	32 768	512	2	32 768	4	128	1
logistic	28 663	28 638	4	26 695	15	28 644	4
+ 1D-ECE	24 862	24 836	4	24 826	4	24 838	4
sine	29 385	29 374	4	27 607	14	29 372	4
+ 1D-ECE	24 940	24 906	4	24 900	4	24 903	4
tent	32 768	32 768	4	32 768	10	32 768	4
+ 1D-ECE	25 067	25 044	5	25 037	4	25 047	4
Singer	37 817	37 793	4	32 287	62	37 795	4
+ 1D-ECE	39 488	39 412	4	39 394	4	39 411	4
LPWLCM [9]	44 532	44 501	5	41 798	16	44 499	8
Chebyshev	27 059	27 028	4	26 136	8	26 975	4
cubic	43 102	43 054	4	37 974	14	43 064	5
2D-CCC [11]	36 314	30 350	15	35 578	6	35 578	6

The common comparison readout is degenerate at reduced precision

The byte map $F(s) = \lfloor D_w(s_0) \cdot 10^{14} \rfloor \bmod 256$ used by the surveyed schemes is not injective on the word when w is small, and the reason is arithmetic rather than dynamical. Since $10^{14} = 2^{14}5^{14}$, the scaled integer is $(2s_0 + 1)\,2^{13-w}\,5^{14}$, whose low $13 - w$ bits vanish for $w < 13$. Counting distinct outputs directly gives 1 value at $w = 4$ and 6, 4 at $w = 8$, 16 at $w = 10$, 64 at $w = 12$ and the full 256 from $w = 14$ upward.

This explains an anomaly in Table 7: the 2D-CCC at $b = 20$ runs at $w = 10$ and emits exactly 16 distinct symbols at $t = 1$, the whole alphabet available at that width; at $b = 16$ it runs at $w = 8$ and the alphabet is four symbols (Table S5). For every $d \geq 2$ row of both tables the readout, not the map, is the binding constraint, and those rows should be read as measurements of the pair rather than of the map. The one-dimensional rows of Table 7 run at $w = 20$, where the alphabet is full, so the low-width collapse is absent; that is weaker than saying they are unaffected by the readout, which Table S5 shows they are not.

Readout choice changes the answer, and horizon matters as much

Class counts are reported at a stopping certificate, $t^* = \min\{t : c_t = c_{t+1}\}$, after which the count is final. The horizons are not uniform. Most rows stabilize by $t^* = 4$ or 5 under the decimal readout, but the high-eight-bit readout needs $t^* = 14$ to 16 for several maps and $t^* = 62$ for the Singer map. A fixed short window would therefore misreport those rows: at ten observations the Singer high-eight count is 19 179, against 32 287 at stabilization.

With horizons matched, the readout still changes the answer. At $b = 16$ the Singer map gives 37 793 classes under the decimal readout and 32 287 under the top eight bits; the 2D-CCC gives 30 350 against 35 578, in the opposite direction; the tent map gives 32 768 under both. The effect is real but smaller than a finite-window comparison suggests, and we report the stabilized counts with their horizons rather than a common cutoff.

S6. The surrogate XOR image experiment

The surveyed designs also report a second battery, computed on encrypted images: entropy, histogram χ^2, adjacent-pixel correlation, PSNR, NPCR and UACI. Table S6 runs it. We wire each map to a 256×256 test image through the simplest possible cipher, a key

Table S6. Surrogate XOR image experiment, run as a contrast. No cipher from any surveyed paper is implemented here. Each map iterated in native binary64 from a fixed seed generates a key stream under the readout of Appendix A.1 of the main paper, combined with the plaintext by XOR; this is a surrogate for the key-generation path of [12], not the published cipher. Reported are ciphertext entropy $H = -\sum_v \hat{p}_v \log_2 \hat{p}_v$ over the empirical byte frequencies of the 65 536-pixel image (the plug-in estimator, whose expected value for a uniform source at this size is about $8 - 255/(2 \ln 2 \cdot 65\,536) \approx 7.997$, which is why every generator returns that figure), histogram χ^2 (255 degrees of freedom, ideal ≈ 255), horizontal and vertical adjacent-pixel correlation, PSNR between plaintext and ciphertext, NPCR and UACI. α_T is from enumeration and is not an image statistic. NPCR is exactly $1/65\,536$ and UACI exactly $1/(65\,536 \cdot 255)$ for every generator, because a fixed-stream XOR has no plaintext dependence beyond the changed position: a one-pixel change moves one pixel by one grey level, as in any synchronous stream cipher. Those two figures measure cipher architecture and say nothing about the map, which is the point. The last column records that $C \oplus K$ returned the plaintext byte for byte, with matching SHA-256; the protocol and a key-stream test vector are in Section S6.

generator	α_T	H	χ^2	corr_H	corr_V	PSNR	NPCR (%)	UACI (%)	$P' = P$
plaintext (*camera*)	–	6.744	165296	0.992	0.993	–	–	–	–
Logistic (baseline)	0.4373	7.997	234	-0.006	-0.004	7.09	0.00153	6.0×10^{-6}	exact
1D-ECE over logistic	0.3792	7.997	285	-0.000	-0.002	7.06	0.00153	6.0×10^{-6}	exact
Sine	0.4484	7.997	280	-0.002	-0.005	7.07	0.00153	6.0×10^{-6}	exact
Tent	0.5000	7.997	257	-0.003	0.006	7.06	0.00153	6.0×10^{-6}	exact
Singer	0.5773	7.998	217	0.002	0.002	7.07	0.00153	6.0×10^{-6}	exact
1D-ECE over Singer	0.6034	7.997	254	0.004	-0.003	7.07	0.00153	6.0×10^{-6}	exact
LPWLCM	0.6785	7.997	243	-0.022	0.006	7.07	0.00153	6.0×10^{-6}	exact
Chebyshev	0.4131	7.997	278	-0.003	-0.004	7.06	0.00153	6.0×10^{-6}	exact
Cubic	0.6579	7.997	282	0.003	-0.002	7.10	0.00153	6.0×10^{-6}	exact

stream XOR, and state plainly that this is a surrogate for the key-generation component of [12] rather than that cipher, which resizes a generated matrix and applies further steps.

Algorithm S1 Key-stream surrogate of Table S6. The generator is the only component audited in this paper; the two XOR steps are the simplest cipher that exposes it. Decryption re-runs KeyStream with the same inputs and never inverts f, so a non-injective map supports exact synchronous decryption. This is a surrogate for the key-generation path of [12], not that cipher.

procedure KeyStream(f, parameters, x_0, h, F, n)
  $x \leftarrow x_0$ ▷ native binary64; $x_0 = 0.4132876591$ in every run
  for $i = 1, \ldots, h$ **do** $x \leftarrow f(x) - \lfloor f(x) \rfloor$ ▷ burn-in; $h = 0$ here
  for $i = 1, \ldots, n$ **do**
    $x \leftarrow f(x) - \lfloor f(x) \rfloor$ ▷ one update
    $K[i] \leftarrow F(x)$ ▷ post-update readout; $F(x) = \lfloor x \cdot 10^{14} \rfloor \bmod 256$
  return K

procedure Encrypt(P, K)
  return $C = P \oplus K$ ▷ K laid over P row-major, one byte per pixel
procedure Decrypt(C, K)
  return $P' = C \oplus K$ ▷ $P' = P$ by $K \oplus K = 0$; checked byte for byte

The surrogate is Algorithm S1, and it is fully specified so that the round trip can be checked. The plaintext P is the top-left 256×256 crop of the *camera* test image in scikit-image, an 8-bit greyscale array with SHA-256 `a43b2498...b4425` given in full in the record. The key stream K is generated by iterating the map in native binary64 from $x_0 = 0.4132876591$ with no discarded iterations: byte i is $\lfloor x_{i+1} \cdot 10^{14} \rfloor \bmod 256$ where x_{i+1} is the state after the $(i+1)$-th update, so the readout is post-update here as in the enumeration. K fills the image row-major, one byte per pixel; $C = P \oplus K$ and $P' = C \oplus K$.

For NPCR and UACI the second plaintext differs from P at pixel $(128, 128)$ by an XOR with 1, under the same K. As a test vector, the first eight key-stream bytes of the logistic generator are 161, 111, 57, 195, 159, 2, 255, 225; the first thirty-two for every generator are in the record. For all nine generators $P' = P$ held byte for byte and the SHA-256 of P' equalled that of P, which is the correctness identity of XOR and requires neither T^{-1} nor Φ^{-1}; a non-injective generator supports exact synchronous decryption because decryption re-runs the generator forward.

Every generator returns ciphertext entropy of 7.997 or 7.998 bits, a histogram χ^2 between 217 and 285 against an ideal of about 255, adjacent-pixel correlations within ± 0.023 of zero, and PSNR of 7.06 to 7.10 dB. The image fractions behind those generators run from 0.379 to 0.679. The battery is flat across a range of 0.30 in α_T, which is the same result as Table 8 for the randomness battery, where seven of nine pass the full battery, obtained for a different set of statistics computed on a different object; Table 8 of the main paper and Table S6 put the two batteries beside the same image fractions.

NPCR is exactly $1/65\,536$ of the pixels, that is $100/65\,536 \approx 0.00153\%$, and UACI exactly $1/(65\,536 \cdot 255)$, that is $100/(65\,536 \cdot 255) \approx 6.0 \times 10^{-6}\,\%$, for every generator, one changed pixel differing by one grey level, and that is correct rather than a defect: under a fixed key stream a one-pixel plaintext change moves exactly one ciphertext pixel, which is the behaviour of any synchronous stream cipher, including secure ones. Designs that report NPCR near 99.6% obtain it from a stage that makes the ciphertext depend on the plaintext beyond the changed position, such as value diffusion, plaintext-dependent control of the generator, or chaining; a fixed position permutation on its own would merely relocate the difference. The figure is therefore a statement about cipher architecture, and it is the clearest instance in either battery of a number that is fixed by the architecture and independent of the generator.

Neither battery is a bad test of what it tests. The finding is narrower: the quantities these designs report are computed on the output of a complete system at deployed precision, they are insensitive to the one-step reachability of the map inside it across the entire range we measured, and a reader given only those figures could not distinguish a generator that reaches 38% of its state space from one that reaches 68%. The audit reports what they do not.

S7. A measurement that did not work

Before the enumeration we attempted an output-level comparison: iterate each map at native double precision, apply the decimal byte readout, and measure the first-order difference distribution over 2^{20} bytes, the difference being $(z_{t+1} - z_t) \bmod 256$, so that the alphabet has 256 symbols. The five maps then under test were the logistic, sine, tent, Singer and LPWLCM realizations at their parameters of Appendix A.2 of the main paper; the record is `battery_native.json`. All five, including the plain logistic map, returned empirical difference entropies of 7.9998 bits, where the statistic is $-\sum_v \hat{p}_v \log_2 \hat{p}_v$ over the empirical frequencies $\hat{p}_v$ of the 256 possible modular differences in 2^{20} consecutive pairs. That is not a ceiling, since exactly balanced counts would give 8; it is close to the value expected from the finite-sample bias of a uniform 256-symbol source at this window, about $8 - 255/(2 \ln 2 \cdot 2^{20}) \approx 7.9998$. The statistic did not discriminate between the tested cases at the reported precision.

We report this because the negative result is informative about method rather than about maps. These readouts multiply the state by 10^{14} or 10^{15} and take a residue, retaining only the low bits of the mantissa. For the trajectories we tested those bits were empirically uniform to the resolution of this statistic; we do not claim this for smooth maps in general, and a constant map is an immediate counterexample. What the negative result shows is

that this statistic, on these readouts and these trajectories, carries no information about the map that produced them.

S8. The LPWLCM quantizer gap, branch by branch

Table S7 explains it. Both quantizers share the decoder of Appendix A.1, so for a given state the same branch is evaluated and the two runs differ only in the encoder; the difference must therefore lie in how the three branches' images overlap. The two linear branches, x/p on $[0.5, 0.6)$ and $(1-x)/(1-p)$ on $[0.6, 1)$, have slopes $5/3$ and $5/2$, so each is injective on its block at every width and its image is an arithmetic progression of cells with a rational step. Under truncation those two progressions are disjoint: their intersection is 0 at both widths. Under round-to-nearest each progression shifts by a different offset, the two interleave differently, and they overlap in 4 369 cells at $w = 16$ and 69 905 at $w = 20$, most of which the logistic branch already reaches. Inclusion–exclusion over the three branch images reproduces the union exactly at every row of the table, and the union difference, 710 states at $w = 16$ and 11 336 at $w = 20$, is the quantizer gap. The new overlap is most of that difference but not all of it: at $w = 16$ its net contribution is $4\,369 - 3\,679 = 690$ of the 710, and at $w = 20$ $69\,905 - 58\,800 = 11\,105$ of the 11 336, the remaining 20 and 231 states coming from small changes in the other terms. It is a lattice-alignment effect of the two rational slopes on the midpoint grid, not a property of the map's dynamics, and it is stable in w because the slopes are.

Table S7. Branch-resolved image of the LPWLCM under the two quantizers, from `quantizer_mismatch.py`. The three branches, logistic on $[0, 0.5)$, x/p on $[0.5, p)$ and $(1-x)/(1-p)$ on $[p, 1)$ with $p = 0.6$, each map their own block of states; the columns give the size of each branch's image, the pairwise and triple intersections, and the union, which is $|\operatorname{Im} T|$. Both linear branches are injective on their blocks at both widths and under both quantizers. Under truncation their images are disjoint; under rounding they overlap, and that overlap accounts for most of the quantizer gap in Table 5 of the main paper, 690 of 710 states at $w = 16$ and 11 105 of 11 336 at $w = 20$.

w	quantizer	$\lvert\operatorname{Im} B_0\rvert$	$\lvert\operatorname{Im} B_1\rvert$	$\lvert\operatorname{Im} B_2\rvert$	$B_0 \cap B_1$	$B_0 \cap B_2$	$B_1 \cap B_2$	triple	union	α_T
16	trunc	28 663	6 554	26 214	5 504	11 395	0	0	44 532	0.6795
16	round	28 662	6 554	26 214	5 489	11 429	4 369	3 679	43 822	0.6687
20	trunc	458 588	104 858	419 430	88 217	183 194	0	0	711 465	0.6785
20	round	458 589	104 858	419 430	88 101	183 542	69 905	58 800	700 129	0.6677

S9. Boundary-focused precision test at wide widths

Agreement at $w = 16$ does not by itself establish agreement at $w = 28$, where the grid is 4 096 times finer and the enhancer of [8] multiplies its argument by $\alpha\pi \approx 9\,870$ before taking a sine, so that a change in the last bits of the argument can cross a cell boundary. A full 40-digit recomputation above $w = 16$ is out of reach, but a targeted one is not: only a state whose binary64 output lies within the arithmetic error of a cell boundary can change successor under higher precision, so it suffices to find those states and recompute them (`boundary.py`, record `boundary.json`). At $w = 16$ a tolerance of 10^{-6} cells flags no state at all for the logistic or the enhanced logistic map. For the logistic map that is consistent with the exhaustive comparison above, which covered it; the enhanced logistic map was not in that comparison, so for it this is the only precision evidence we have at $w = 16$. At $w = 28$ the logistic map has 5 432 states within 10^{-5} cells of a boundary and ten of them change successor at 40 digits, a mismatch rate of $10/5\,432 \approx 1.8 \times 10^{-3}$ among the candidates, which normalized by the whole state space is $10/2^{28} \approx 3.7 \times 10^{-8}$; no claim is made about states outside the candidate set. The enhancer is the case that matters, and there the procedure gives an estimate rather than a bound. Its argument to the sine reaches about 8×10^4, so the binary64 error in the output is of order 10^{-2} cells at $w = 28$; two per

cent of its states lie within that distance of a boundary. We recomputed a uniform random sample of 20 000 of them at 40 digits and 265 changed successor, a conditional proportion of 0.0133 with an approximate 95% interval of 0.0117 to 0.0149. Scaling by the candidate count gives an estimated 71 000 changed successors in 2^{28} states, interval 62 000 to 80 000, that is, an estimated fraction of changed successors of 2.7×10^{-4}, interval 2.3 to 3.0×10^{-4}, within the candidate model. Two successor arrays differing at M inputs have image sizes differing by at most M, so this is a proxy for an upper limit on the change in α_T, not a measurement of that change, which may be zero; and the lower endpoint of the interval is not a lower bound on any image-fraction change. At $w = 24$ the same procedure gives an estimated 318 changes and 1.9×10^{-5}.

Three qualifications belong with those numbers, and they are the reason we call them estimates. The scaling is a sample proportion and carries the interval above. The candidate set is defined by an order-of-magnitude error estimate, not a certified error bound, so a state outside it could in principle change and would not be counted. And the experiment was run on the enhanced logistic map; the enhanced sine and tent maps have different expressions and we have not measured them. What follows is therefore narrower than a bound: the enhanced-map values at $b = 28$ are results for the specified binary64 realization whose arithmetic uncertainty is plausibly of the same order as the three-way agreement of Section 4.4, which is why that agreement is quoted at $b = 24$, where the estimate is an order of magnitude smaller. The environment was Python 3.12.3, NumPy 2.4.4 and mpmath 1.3.0, recorded in the artifact beside the counts.

S10. Burn-in support over h

Table S8 gives the image fraction after h transitions for every one-dimensional realization at $b = 20$. Section 4.15 of the main paper quotes the logistic row and states the observation convention.

Table S8. State-support fraction $I_h/N = |\operatorname{Im} T_w^h|/N$ after h transitions, at $b = 20$ under truncation, with $I_0 = N$ and $I_1/N = \alpha_T$ (Table 5 of the main paper). If h transitions are discarded before post-update emission, the number of distinguishable output streams is at most I_{h+1} (Lemma 1 of the main paper); the table lists selected h, so the required $h+1$ value is not always displayed. Two of the surveyed designs discard a fixed prefix, one of them 500 iterates. The affine controls halve their support at every step until one state remains, as Proposition 6 of the main paper requires; every finite realization eventually stabilizes on its cyclic set. The doubling and affine rows are the closed-form controls of Propositions 5 and 6 of the main paper.

map	h=1	h=2	h=8	h=32	h=128	h=500	h=1000
doubling	0.5000	0.2500	0.0039	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
affine, a=6	0.5000	0.2500	0.0039	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
logistic	0.4373	0.3294	0.1283	0.0355	0.0080	0.0028	0.0021
+ 1D-ECE	0.3792	0.2583	0.0919	0.0259	0.0057	0.0011	0.0008
sine	0.4484	0.3327	0.1329	0.0387	0.0093	0.0021	0.0010
+ 1D-ECE	0.3796	0.2586	0.0915	0.0256	0.0059	0.0018	0.0013
tent	0.5000	0.3317	0.1167	0.0391	0.0112	0.0028	0.0010
+ 1D-ECE	0.3802	0.2596	0.0925	0.0254	0.0056	0.0016	0.0011
Singer	0.5773	0.4746	0.2667	0.0933	0.0224	0.0051	0.0027
+ 1D-ECE	0.6034	0.4349	0.1620	0.0445	0.0104	0.0018	0.0012
LPWLCM [9]	0.6785	0.5370	0.2538	0.0826	0.0214	0.0056	0.0030
Chebyshev	0.4131	0.2743	0.0900	0.0217	0.0041	0.0008	0.0008
cubic	0.6579	0.4905	0.1986	0.0580	0.0128	0.0039	0.0027