Verified Randomness at Scale: The Eormen Certified Entropy Block

Leroy A. Palmer
palmer@eormen.com eormen.com

April 2026

Abstract

This paper defines the Eormen Certified Entropy Block (CEB): a fixed 1 GiB binary artefact produced by the Eormen Edge-of-Chaos Entropy Engine v5.2.0, post-processed through HMAC-DRBG in accordance with NIST SP800-90A, and subjected to a mandatory three-tier statistical validation pipeline before delivery. The three tiers are Dieharder v3.31.1 (17 test configurations), the NIST SP800-22 Rev. 1a battery (15 tests), and the Eormen Internal Validation Suite v6.1.0 (8 analysis categories). A block is delivered only if it passes all three tiers against fully disclosed thresholds; failed candidates are recorded and destroyed. No partial-pass certificate is ever issued.

The contribution of the Eormen product is governance per delivered unit. Every certified block is accompanied by a complete, cryptographically bound record of its passage through the full pipeline: the Dieharder results for all 17 tests, the SP800-22 results for all 15 tests, the Eormen Internal Validation results for all eight categories, three SHA-256 hashes (data-only, metadata-only, complete-file), a 128-bit nonce, a generation timestamp in UTC, and a detached GPG signature. The evidence is bound by hash to the exact bytes the customer receives, and the customer can re-run any tier against the delivered file and recover the same numbers at any future time. This level of per-block governance is not provided by any commodity entropy source, by any operating-system random device, or by any other commercial entropy product known to the authors at the time of writing.

We report complete validation results for five CEBs (EORM-2026-0003, 0009, 0018, 0029, 0035). Every statistical claim in this paper is independently verifiable from the delivered `.bin` file using openly available tools. Cross-block distinctness is evaluated through a six-level evidentiary framework (L0–L5) covering cryptographic file identity, identifier uniqueness, pairwise Hamming distance, zero-lag Pearson correlation, XOR transform stability, lagged cross-correlation across 110 pairs and three lag sets, and SHA-256 chunk collision search at chunk sizes of 4 096, 16 384 and 65 536 bytes. Every block pair passes at every level.

The governing equations and internal parameters of the Edge-of-Chaos Engine are Eormen trade secrets and are not disclosed in this paper; all validation thresholds, pipeline configurations and numerical results are fully disclosed.

Disclosure boundary. The Eormen Edge-of-Chaos Entropy Engine governing equations, internal parameters, seed schedule, warm-up procedure and lattice geometry are Eormen trade secrets and are not disclosed in this paper. The Eormen Internal Validation thresholds, the Dieharder and NIST SP800-22 configurations, and every numerical result reported herein are fully disclosed.

Contents

List of Tables

1 Introduction

1.1 The role of entropy in modern systems

A random bit generator underpins more of modern computing than its commodity status suggests. Cryptographic key generation, nonces and initialisation vectors, differential-privacy noise, Monte Carlo integration and stochastic simulation all rely on a supply of bits that is statistically indistinguishable from a uniform random source. The quality of those bits is rarely visible to the application above, and when it fails, the failure is rarely visible until a specific exploit surfaces.

The 2008 Debian OpenSSL defect [3] reduced the effective entropy of generated SSH and SSL keys to the process identifier, a number small enough to enumerate on a laptop. Heninger et al. [4] subsequently showed that a small but non-negligible fraction of Internet-facing RSA and DSA keys shared prime factors, a consequence of insufficient entropy at the moment of key generation. In each case the consumer of the random bits could not detect the defect: no frozen, auditable unit of entropy was available to inspect.

1.2 Eormen Certified Entropy Blocks

An Eormen Certified Entropy Block is a 1 GiB file of bytes that has passed three independent statistical test batteries and a cryptographic provenance procedure. Every block is delivered with:

- the full Tier 1 (Dieharder v3.31.1) results for 17 tests;
- the full Tier 2 (NIST SP800-22 Revision 1a) results for 15 tests;
- the full Tier 3 (Eormen Internal Validation v6.1.0) results for eight analysis categories;
- three SHA-256 hashes (data-only, metadata-only, complete file) and a 128-bit nonce;
- a detached GPG signature on every delivered artefact.

A customer who possesses the delivered `.bin` file can reproduce every hash, re-run every test, and verify every signature using openly available tools. The trust required of the Eormen generator itself is therefore bounded: the customer needs to trust only the disclosed test configurations, their public implementations, and the bytes of the delivered file.

1.3 Scope and purpose

This paper documents the three-tier validation pipeline, reports the complete results for five certified blocks, presents the cross-block distinctness evidence, and explains the per-block governance regime that accompanies every delivered unit. It does not describe the Eormen Edge-of-Chaos Entropy Engine, which is a trade secret. The paper claims statistical indistinguishability from a uniform random source under the disclosed tests; it does not claim true randomness in a physical sense, and it does not claim cryptographic security beyond what the statistical results support.

The distinguishing contribution of the Eormen product is not a claim about a superior byte stream. It is the governance apparatus: every delivered block carries a complete record of passage through three independent test batteries, bound by hash to the exact bytes the customer receives,

signed, timestamped, and independently re-verifiable at any future time. No commodity entropy source and no operating-system random device known to the authors provides per-read evidence of this scope.

1.4 How to read this document

Sections 2 and 3 describe the generation architecture and the validation pipeline at the level necessary to understand what the tests measure. Sections 4, 5 and 6 describe the three test tiers with their full configurations and pass criteria. Section 7 reports the results for the five featured blocks. Section 8 establishes cross-block independence and distinctness. Section 9 explains the governance gap between a certified Eormen delivery and a commodity entropy source. Sections 10 and 11 discuss what the results do and do not demonstrate.

Every numerical claim in Sections 7 and 8 is independently verifiable from the delivered `.bin` files using openly available tools (Dieharder v3.31.1, an SP800-22 Rev. 1a conforming implementation, and Python with NumPy, zlib, bzip2, lzma and SHA-256).

2 Generation architecture

2.1 Two-stage architecture

The Eormen generator is a two-stage pipeline. The first stage is the Eormen Edge-of-Chaos Entropy Engine, a proprietary dynamical-systems core that produces a candidate byte stream from a per-block input. The second stage is a standard HMAC-DRBG post-processor instantiated per NIST SP800-90A [1], seeded from the output of the first stage and clocked at the block rate. The post-processor output is the 1 GiB payload that is written to the block file.

The two stages are auditable independently: the HMAC-DRBG post-processor can be checked against its public specification, while the Edge-of-Chaos entropy source is auditable under non-disclosure through its statistical signature (§2.5).

2.2 Block format

A certified block is a single file of exactly 1073741888 bytes, composed of:

- a 1 GiB payload of exactly 1073741824 bytes, and
- a 64-byte metadata trailer appended at the end of the file.

The trailer carries the identifiers that bind the payload to its generation event: the 128-bit nonce, the generation timestamp, the version string of the validation pipeline, and the Dieharder, NIST and Internal Validation verdict flags. All numerical fields in the trailer are little-endian.

2.3 Deterministic reproducibility

The generator is deterministic at the block level: a given per-block input reproduces the identical 1 GiB payload, byte for byte. The data-only SHA-256 hash is therefore a unique fingerprint of a regenerated block, and a replay of the generation procedure reproduces the hash. Customers under audit can be supplied with a replay artefact on request, subject to the non-disclosure terms of the Eormen audit programme.

2.4 Three-tier hash verification

Each delivered block is bound by three SHA-256 [5] hashes, computed independently and published in the delivery manifest:

1. **Data-only SHA-256:** the hash of the first 1073741824 bytes (the payload).

2. **Metadata-only SHA-256:** the hash of the 64-byte trailer.

3. **Complete-file SHA-256:** the hash of all 1073741888 bytes.

A customer who possesses only the `.bin` file can independently reproduce each hash and verify all three against the delivery manifest. A modification to any byte of the payload is detected by a change in the data-only SHA-256 and in the complete-file SHA-256. A modification to any byte of the trailer is detected by a change in the metadata-only SHA-256 and in the complete-file SHA-256.

2.5 What is not disclosed

The Eormen Edge-of-Chaos Entropy Engine is a trade secret. The following are not disclosed, either in this paper or in any other Eormen publication:

- the governing equations of the dynamical core;
- the dimension of the dynamical state;
- the warm-up protocol and duration;
- the seed schedule and the seed-derivation function;
- the lattice size, geometry and coupling topology;
- the numerical method, step size and arithmetic precision;
- any parameter that would reduce the effective state space of the engine.

Customers or regulators wishing to audit the engine under non-disclosure are directed to the Eormen audit programme, the terms of which are published separately.

3 Validation pipeline

3.1 Pipeline architecture

Every candidate block is subjected to three independent test tiers before it is certified:

- **Tier 1, Dieharder v3.31.1**: a suite of classical randomness tests maintained as a successor to Marsaglia's Diehard battery. Seventeen tests are executed per block. Described in Section 4.
- **Tier 2, NIST SP800-22 Revision 1a**: the National Institute of Standards and Technology's statistical test suite for random number generators. Fifteen tests are executed per block. Described in Section 5.

- **Tier 3, Eormen Internal Validation v6.1.0**: an Eormen in-house suite organised into eight analysis categories. All thresholds are disclosed in Section 6.

The tiers run sequentially. Tier 1 reads the data-only payload (metadata stripped). Tier 2 reads the same data-only payload. Tier 3 reads the same data-only payload and additionally consults the metadata trailer for provenance binding. Each tier's raw output is serialised to a JSON artefact carried in the delivery package.

The three tiers are independent in the sense that no tier depends on another tier's output, and no tier can mask the failure of another. A failure in any tier halts the pipeline and the candidate is rejected. A partial-pass certificate is never issued.

3.2 Zero-tolerance pass criteria

Pass condition 3.1 (Tier gating)**.** A candidate block is certified if and only if it passes all three tiers in full, under the configurations documented in Sections 4, 5 and 6. The pass criteria for each tier are fixed independently of any particular generator and are frozen at each version increment of the Internal Validation Suite.

This gating rule differs from majority-vote or probabilistic-pass criteria used elsewhere in the field. It is stricter, by design: it reduces the type II error rate (false positives) at the cost of a higher type I error rate (rejection of acceptable blocks). The cost is paid in compute time, not in shipped quality.

3.3 What happens when a block fails

Candidate blocks that fail any tier are recorded in the Eormen failure register with the test name, the failed metric, the recorded value, the threshold and the timestamp. The failed block itself is destroyed after the failure record is written. No failed block is ever silently replaced, renamed or reused. The delivery manifest for a certified block records its generation timestamp and nonce; the generation-to-delivery ratio is tracked separately in the production register and is available to customers under audit.

3.4 Validation timescale

Table 1: Measured wall-clock durations of Tier 2 (NIST SP800-22) and Tier 3 (Eormen Internal Validation v6.1.0) per block. Tier 1 (Dieharder) runs under a separate containerised harness and its timing is reported in the delivery manifest. Processing rate is the Tier 3 throughput.

Block	NIST duration (s)	Internal duration (s)	Internal rate (MB/s)
EORM-2026-0003	8346	4067	0.250
EORM-2026-0009	8322	4021	0.250
EORM-2026-0018	8327	4041	0.250
EORM-2026-0029	8269	4036	0.250
EORM-2026-0035	8234	4023	0.250

Tier 1 runs in a separate containerised harness and its wall-clock duration is recorded in the delivery manifest. Total validation time is substantially greater than the generation time, which is the intended relationship: validation is the bottleneck, not generation.

4 Tier 1: Dieharder statistical test suite

4.1 Background

Dieharder is a maintained successor to George Marsaglia's Diehard battery of randomness tests, extended and packaged by Robert G. Brown at Duke University. The battery executes a sequence of stylised experiments against a stream of bytes, and for each experiment reports a p-value under the null hypothesis that the stream is drawn from a uniform independent source. A p-value very close to zero or very close to one is evidence against the null; a p-value in a plausible middle band is consistent with the null.

Tier 1 uses Dieharder version 3.31.1 [2], frozen for the duration of the validation campaign documented in this paper. The version string is recorded in the `dieharder_test_metadata` field of every Tier 1 results file. Dieharder is selected as the first tier because its tests span a wide range of statistical signatures (bit patterns, spacing, rank statistics, geometric embeddings, runs and streaks) and its assessment logic is transparent. It is a broad statistical screen, not a cryptographic security proof; its role is to reject candidates before the more targeted Tier 2 and Tier 3 analyses are applied.

4.2 Test selection and execution

Seventeen Dieharder tests are executed per candidate block. The test names, in the order in which they are executed and recorded in the results file, are given in the first column of table 2 (Section 7.2). The `diehard_runs`, `diehard_craps` and `marsaglia_tsang_gcd` experiments each produce two reported statistics, and these appear as separate rows; the `diehard_2dsphere` and `diehard_3dsphere` tests are invoked with `ntup` = 2 and 3 respectively.

Dieharder is invoked in file-input mode with the 1 GiB data-only payload (the 64-byte metadata trailer is stripped before the stream is passed to the test harness). The invocation uses Dieharder's raw binary file generator (`-g 201`) with the payload bound by `-f`. No internal Dieharder seed is consumed from the stream; the test driver's own PRNG is left at its default configuration, which controls only auxiliary sampling within each test and does not inject any bits into the stream under test.

4.3 Assessment criteria

Pass condition 4.1 (Dieharder)**.** A block passes Tier 1 if and only if every one of the 17 executed tests returns a reported p-value inside the closed interval $[10^{-6}, 1 - 10^{-6}]$ and no test returns an assessment of `FAILED`.

The p-value window is symmetric about 1/2. A p-value at the boundary is implausible under the null at a rate of 2×10^{-6}, well below the significance levels used in conventional statistical practice. Full per-test results for the five featured blocks are reported in Section 7.2.

5 Tier 2: NIST SP800-22 statistical test suite

5.1 Background

NIST Special Publication 800-22 Revision 1a, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications" [6], specifies fifteen statistical tests designed to detect departures from uniform randomness that are relevant to cryptographic use. The suite is widely adopted as a minimum bar for evaluating the statistical quality of candidate random number generators, and is referenced under a range of national and industry certification regimes.

SP800-22 is a statistical suite and not an entropy-source assessment. A separate document, SP800-90B [7], governs the estimation of min-entropy from a physical entropy source. Tier 2 is concerned only with the statistical indistinguishability of the delivered byte stream from a uniform independent source. Claims concerning min-entropy of the Eormen Edge-of-Chaos Engine are out of scope for this paper.

The version used is SP800-22 Revision 1a, and the specification string together with the significance level ($\alpha = 0.01$) is recorded in the `test_configuration` field of every Tier 2 results file.

5.2 Implementation over the full block

Tier 2 operates on the full 1 GiB data-only payload, so that exactly 2^{33} bits are presented to the suite. The implementation is an SP800-22 Rev. 1a conforming test harness; its version string, execution timestamp and wall-clock duration are recorded in the `test_configuration` field of each block's results file and reported in Section 3.4.

5.3 The fifteen tests

The fifteen NIST SP800-22 tests, in the order in which they are recorded in the results file, are listed in the first column of table 4 (Section 7.3). Each test is described in Section 2 of SP800-22 Rev. 1a and returns a p-value, or, for tests that report multiple p-values, a representative test statistic summarised to a single p-value by the harness as documented in the specification.

5.4 Pass criterion

Pass condition 5.1 (NIST SP800-22 Rev. 1a)**.** A block passes Tier 2 if and only if every one of the 15 executed tests returns a p-value satisfying $p \geq 0.01$. A single test with $p < 0.01$ rejects the block.

The threshold $\alpha = 0.01$ is the SP800-22 Rev. 1a canonical significance level. Full per-test results for the five featured blocks, including the minimum p-value observed across the corpus, are reported in Section 7.3.

6 Tier 3: Eormen Internal Validation Suite

6.1 Design principles

The Eormen Internal Validation Suite version 6.1.0 is an in-house statistical battery designed to complement Tiers 1 and 2 at the scale of 1 GiB blocks. Its design principles are:

- **Full-block measurement.** Every one of the 2^{30} bytes of the data-only payload contributes to every category that operates over the full block; no category subsamples the stream.
- **Independent perspectives.** Each of the eight categories measures a different statistical property: byte histogram, entropy at multiple scales, autocorrelation, spectral flatness, pattern runs, compression resistance, binary matrix rank, and chunk-level duplication.
- **Quantitative scoring with frozen thresholds.** Numeric scores, not just verdicts, are reported in the delivery package. Thresholds are fixed at each version increment and disclosed in full in this section.

The 64-byte metadata trailer is excluded from the statistical measurements and is consulted only for provenance binding.

6.2 Category 1: frequency distribution

The first analysis computes the 256-bin byte histogram of the payload and compares it to the uniform distribution. The reported quantities are the Pearson chi-square statistic (on df = 255 degrees of freedom) and its associated p-value, the maximum absolute deviation of any single bin frequency from the expected frequency ($1/256 = 0.00390625$), the standard deviation of the bin frequencies, and a Gini-style uniformity index.

Threshold. Pass requires the chi-square p-value to satisfy $p > 0.01$.

6.3 Category 2: entropy measurements

The second analysis computes the Shannon entropy of the payload at the byte level, under the theoretical maximum of 8.000 bits per byte. Shannon entropy is reported to 12 decimal places. A multi-scale analysis repeats the measurement over windows of 1 KB, 4 KB, 16 KB, 64 KB, 256 KB and 1 MB, reporting the minimum, maximum and standard deviation of the per-window entropy to detect concentrations of lower entropy within the block. Bit-position entropies are reported independently for each of the eight bit positions within the byte. Byte-pair entropy and conditional entropy are reported as second-order measures.

Threshold. Pass requires the Shannon entropy of the full block to satisfy $H > 7.999$ bits per byte.

6.4 Category 3: correlation analysis

The third analysis computes the Pearson autocorrelation of the byte sequence at seventeen lag distances drawn on the geometric grid $k = 2^j$ for $j = 0, 1, \ldots, 16$, that is $k \in \{1, 2, \ldots, 65536\}$ with successive values doubling. The reported quantities are the autocorrelation at every lag, the

maximum absolute autocorrelation across all lags, the autocorrelation decay rate, and the list of lags, if any, flagged as significant by the analyser.

Threshold. Pass requires the maximum absolute autocorrelation across all seventeen lags to satisfy $|r|_{\max} < 0.001$.

6.5 Category 4: spectral analysis

The fourth analysis computes the discrete Fourier transform of the byte sequence over 1 MiB windows, and aggregates the power spectrum across the block. The reported quantities are the total spectral power, the spectral flatness (the ratio of the geometric to the arithmetic mean of the power spectrum, which tends to unity for white noise), the peak-to-average ratio, and the FFT window size. Spectral flatness and peak-to-average ratio are reported as quantitative scores; no hard pass threshold is applied, and the measurements are interpreted jointly with Category 5.

6.6 Category 5: pattern analysis

The fifth analysis computes run-length distributions at each of the eight bit positions, reports the longest consecutive run of identical bits, computes the byte-level approximate entropy, and reports a 9-bit non-overlapping template match count. Run and pattern statistics are reported as quantitative scores and are interpreted jointly with Category 3.

6.7 Category 6: compression resistance

The sixth analysis applies three independent compression algorithms to the full 1 GiB payload at their maximum compression level:

- zlib (Deflate), level 9;
- bzip2 (Burrows-Wheeler), level 9;
- lzma (Lempel-Ziv-Markov), level 9.

The reported quantity is the compression ratio ρ = compressed_size/original_size for each algorithm. A ratio $\rho > 1$ means that the compressed output is larger than the input, which is the expected behaviour for incompressible data under a compressor that writes non-trivial framing and dictionary overhead.

Threshold. Pass requires the compression ratio for each of the three algorithms to satisfy $\rho > 0.999$.

6.8 Category 7: binary matrix rank

The seventh analysis partitions the bitstream into non-overlapping 32×32 binary matrices. There are exactly $2^{23} = 8388608$ such matrices in 1 GiB. The rank of each matrix is computed over GF(2), and the observed rank distribution is compared to the theoretical distribution by the Pearson chi-square goodness-of-fit test. The theoretical probabilities used by the analyser are: $\Pr[\text{rank} = 32] \approx 0.2888$, $\Pr[\text{rank} = 31] \approx 0.5776$, $\Pr[\text{rank} = 30] \approx 0.1284$ and $\Pr[\text{rank} \leq 29] \approx 0.0052$. The analyser reports the chi-square statistic, the chi-square p-value, and the full observed rank distribution.

6.9 Category 8: chunk uniqueness

The eighth analysis partitions the payload into non-overlapping chunks at three chunk sizes: 4096 bytes ($2^{18} = 262144$ chunks), 16384 bytes ($2^{16} = 65536$ chunks) and 65536 bytes ($2^{14} = 16384$ chunks). Each chunk is hashed under SHA-256 and the number of SHA-256 collisions is counted separately at each size. The analyser reports an integer score on the interval $[0, 1000]$, bound to bands:

- score $= 1000$: EXCELLENT, zero duplicates at any size;
- score ≥ 950: GOOD;
- score ≥ 900: ACCEPTABLE;
- score ≥ 800: CONCERNING;
- score < 800: FAILED.

Threshold. Pass requires score $= 1000$, equivalent to zero SHA-256 collisions at all three chunk sizes.

6.10 Summary pass criteria

Pass condition 6.1 (Eormen Internal Validation v6.1.0)**.** A block passes Tier 3 if and only if all of the following hold, jointly:

- frequency-distribution chi-square p-value > 0.01;
- full-block Shannon entropy > 7.999 bits per byte;
- maximum absolute autocorrelation over the seventeen lags < 0.001;
- zlib, bzip2 and lzma compression ratios each > 0.999;
- chunk-uniqueness score $= 1000$.

A single criterion outside its window rejects the block. The spectral-analysis and pattern-analysis measurements are reported as quantitative scores and do not carry an independent pass threshold.

7 Results: five certified entropy blocks

This section reports the full results of the three-tier validation pipeline, as applied to the five featured blocks. Every number reported below is extracted from the delivered results JSON files in the corresponding `delivery/EORM-2026-XXXX/` directory and can be reproduced by a customer from the delivered `.bin` file using the openly available tools named in Section 10.3.

7.1 Block identification

The five featured blocks are EORM-2026-0003, EORM-2026-0009, EORM-2026-0018, EORM-2026-0029 and EORM-2026-0035. Their provenance identifiers (128-bit nonce, generation timestamp, data-only SHA-256, complete-file SHA-256) are reported in table 7 (Section 8).

7.2 Tier 1 results: Dieharder

All $17 \times 5 = 85$ Tier 1 evaluations returned an assessment of `PASSED` with a p-value strictly inside $[10^{-6}, 1 - 10^{-6}]$. No `WEAK` verdicts and no `FAILED` verdicts were recorded for any block. The per-test p-values for all five blocks are reported in table 2, and the per-block summary (test counts and p-value distributional statistics) is reported in table 3.

Table 2: Dieharder v3.31.1 p-values for each of the 17 executed tests, for all five featured blocks. Every test returned assessment `PASSED` for every block; pass criterion is $p \in [10^{-6}, 1 - 10^{-6}]$ with no `FAILED` verdicts.

Test	0003	0009	0018	0029	0035
diehard_birthdays	0.3030	0.9307	0.4050	0.0375	0.7326
diehard_operm5	0.1062	0.5961	0.2224	0.7290	0.1438
diehard_rank_32x32	0.9658	0.8292	0.3464	0.8484	0.8242
diehard_rank_6x8	0.0746	0.5441	0.0730	0.9945	0.9633
diehard_bitstream	0.4490	0.7676	0.9787	0.9947	0.8292
diehard_count_1s_str	0.9701	0.0091	0.4627	0.7792	0.4363
diehard_count_1s_byt	0.8138	0.4962	0.5785	0.9620	0.9622
diehard_parking_lot	0.2729	0.4683	0.2785	0.5738	0.9647
diehard_2dsphere (ntup=2)	0.9882	0.3067	0.9289	0.1184	0.0781
diehard_3dsphere (ntup=3)	0.9498	0.7574	0.6763	0.4364	0.1007
diehard_squeeze	0.4884	0.9463	0.9629	0.9662	0.8105
diehard_runs	0.5972	0.9343	0.7500	0.8892	0.9562
diehard_runs	0.2125	0.8083	0.9406	0.7419	0.4784
diehard_craps	0.4522	0.8795	0.6947	0.8020	0.9255
diehard_craps	0.4177	0.9810	0.8030	0.9464	0.5817
marsaglia_tsang_gcd	0.1456	0.3293	0.0626	0.6929	0.1575
marsaglia_tsang_gcd	0.7733	0.1238	0.9232	0.3128	0.8685

Table 3: Dieharder per-block summary. Seventeen tests executed per block, zero `WEAK` verdicts, zero `FAILED` verdicts.

Block	passed	weak	failed	$p_{\min}$	$\bar{p}$	$p_{\max}$	Verdict
EORM-2026-0003	17	0	0	0.0746	0.5283	0.9882	PASSED
EORM-2026-0009	17	0	0	0.0091	0.6299	0.9810	PASSED
EORM-2026-0018	17	0	0	0.0626	0.5934	0.9787	PASSED
EORM-2026-0029	17	0	0	0.0375	0.6956	0.9947	PASSED
EORM-2026-0035	17	0	0	0.0781	0.6361	0.9647	PASSED

7.3 Tier 2 results: NIST SP800-22

All $15 \times 5 = 75$ Tier 2 evaluations returned $p \geq 0.01$, the SP800-22 Rev. 1a canonical significance threshold. The minimum p-value across the corpus is reported in the per-block summary. The per-test p-values for all five blocks are reported in table 4, and the per-block summary is reported in table 5.

7.4 Tier 3 results: Eormen Internal Validation Suite

All five blocks satisfy the Tier 3 pass criteria of Section 6.10. The key per-block metrics from the eight analysis categories are reported in table 6. Every value reported in that table lies inside its disclosed acceptance window; specifically, for every block the full-block Shannon entropy exceeds 7.999 bits per byte, the maximum absolute autocorrelation over the seventeen geometric lags is below 0.001, the frequency-distribution chi-square p-value exceeds 0.01, the zlib, bzip2 and

Table 4: NIST SP800-22 Rev. 1a p-values for each of the 15 executed tests, for all five featured blocks. Every test returned `PASSED` for every block at significance $\alpha = 0.01$ (pass criterion: $p \geq 0.01$).

Test	0003	0009	0018	0029	0035
monobit_test	0.9363	0.3428	0.4232	0.0849	0.4724
frequency_within_block_test	0.3428	0.1585	0.6057	0.3463	0.9850
runs_test	0.6790	0.1481	0.8652	0.3547	0.5390
longest_run_of_ones_test	0.9058	0.7231	0.0181	0.6340	0.0508
binary_matrix_rank_test	0.9455	0.6932	0.5789	0.8265	0.5735
discrete_fourier_transform_test	0.5506	0.4377	0.6451	0.4441	0.5687
non_overlapping_template_matching_test	0.5835	0.3223	0.4584	0.1976	0.2662
overlapping_template_matching_test	0.1454	0.6087	0.6053	0.9776	0.2040
maurers_universal_statistical_test	0.8793	0.2978	0.6742	0.3541	0.8369
linear_complexity_test	0.6980	0.9625	0.1410	0.5596	0.9930
serial_test	0.7712	0.3324	0.6529	0.5095	0.8117
approximate_entropy_test	0.1103	0.2323	0.4536	0.9997	0.5265
cumulative_sums_test	0.9999	0.9999	0.9999	0.9999	0.9999
random_excursions_test	0.3539	0.0734	0.1017	0.1464	0.2165
random_excursions_variant_test	0.5644	0.1797	0.0660	0.1888	0.0188

Table 5: NIST SP800-22 Rev. 1a per-block summary.

Block	executed	passed	failed	$p_{\min}$	$\bar{p}$	$p_{\max}$	Verdict
EORM-2026-0003	15	15	0	0.1103	0.6311	0.9999	PASSED
EORM-2026-0009	15	15	0	0.0734	0.4342	0.9999	PASSED
EORM-2026-0018	15	15	0	0.0181	0.4859	0.9999	PASSED
EORM-2026-0029	15	15	0	0.0849	0.5083	0.9999	PASSED
EORM-2026-0035	15	15	0	0.0188	0.5375	0.9999	PASSED

lzma compression ratios each exceed 0.999, and the chunk-uniqueness score is 1000 with zero duplicate chunks at every tested chunk size.

Table 6: Eormen Internal Validation Suite v6.1.0 key metrics for the five featured blocks. Thresholds are documented in Section 6. Every metric lies inside its disclosed acceptance window.

Metric	0003	0009	0018	0029	0035
Shannon entropy (bits/byte)	7.999999818373	7.999999821965	7.999999810361	7.999999839232	7.999999821654
Byte-pair entropy (bits)	7.999978077	7.999978086	7.999978099	7.999977750	7.999977898
Min block entropy (bits/byte)	7.999768334	7.999775994	7.999773779	7.999765865	7.999778509
Frequency χ^2 statistic (df = 255)	270.362	265.020	282.272	239.307	265.474
Frequency χ^2 p-value	0.2431	0.3201	0.1158	0.7517	0.3131
Max |byte-freq deviation|	5.895e-06	5.303e-06	4.991e-06	5.814e-06	4.760e-06
Max $\|r\|$ (autocorr., 17 lags)	8.318e-05	1.099e-04	1.035e-04	1.316e-04	5.627e-05
Autocorrelation sum	2.100e-08	4.200e-08	3.400e-08	4.700e-08	1.300e-08
Spectral flatness	0.974331	0.974382	0.974426	0.974385	0.974405
zlib ratio (compressed/raw)	1.000305	1.000305	1.000305	1.000305	1.000305
bzip2 ratio	1.004412	1.004431	1.004416	1.004421	1.004421
lzma ratio	1.000050	1.000050	1.000050	1.000050	1.000050
Matrix rank χ^2 p-value	0.4885	0.0393	0.0826	0.0297	0.0097
Chunk uniqueness score	1000	1000	1000	1000	1000
Duplicate chunks (all sizes)	0	0	0	0	0

7.5 Cross-block consistency

The spread of values across the five blocks is consistent with five independent draws from a uniform random source. Every metric reported in table 6 varies from block to block by a quantity comparable to its null standard error; no metric drifts systematically across the five blocks, and no pair of blocks shows a suspiciously small spread. A more formal cross-block comparison,

including the six-level distinctness framework, is reported in Section 8.

8 Cross-block independence and distinctness

8.1 Why distinctness matters

A customer who receives five certified 1 GiB blocks has a legitimate question to ask: are these genuinely five independent blocks, or could they be the same material packaged under five different identifiers? Section 7 establishes that each block *individually* passes the three-tier validation pipeline. This section establishes, independently of the pipeline, that the five featured blocks are *distinct* and *statistically independent* from each other to a level consistent with five fresh draws from a uniform random source.

The evidence is arranged in six levels, each of which a customer can reproduce from the five delivered `.bin` files using only openly available tools:

L0 File byte-exactness (SHA-256).

L1 Identifier uniqueness.

L2 Pairwise Hamming distance, zero-lag Pearson correlation, and XOR transform stability.

L3 Lagged cross-correlation at eleven log-spaced lags ($k = 2^j$ for $j = 0, 1, \ldots, 10$).

L4 Cross-block SHA-256 chunk collision search.

L5 Overall adjudication.

8.2 Provenance-level evidence (L0, L1)

Every certified block carries four identifiers established at generation time and cryptographically bound to the block data: (i) the 128-bit nonce, (ii) the generation timestamp in UTC, (iii) the data-only SHA-256 hash, and (iv) the complete-file SHA-256 hash. Table 7 lists each identifier for the five featured blocks. All five values in every column are distinct.

Table 7: Provenance identifiers for the five featured blocks. Each column contains five distinct values, confirming provenance-level distinctness.

Block ID	Nonce (hex, 128 bits)	Generation (UTC)	Data SHA-256 (first 12)	File SHA-256 (first 12)
EORM-2026-0003	416d1f1e2ef20146278f5958b008e142	2026-04-03T12:36:23Z	4da97f79b6ac	3d2ed2650a63
EORM-2026-0009	2ecd71f95ed66671e1a5faecdd3706db	2026-04-03T19:56:55Z	3ef63f88cc82	11ffaa02cef4
EORM-2026-0018	5bc2c80bde397b2b2498a1f75543ae71	2026-04-04T15:17:04Z	4e02daa34a84	73de889c6847
EORM-2026-0029	8873acd506f26c9a6efcb5e1a628f76c	2026-04-04T19:30:35Z	c89982f551c0	4cb11ab75aa0
EORM-2026-0035	4ba4aa6f0da5c750d73b02111a6c222c	2026-04-04T23:21:38Z	5e429e44ff2a	e6e84cc51b12

An adversary attempting to pass off a duplicated block as genuinely distinct would, at a minimum, have to fabricate consistent values in every identifier column. The data-only SHA-256 collision requirement alone is cryptographically infeasible under current assumptions.

8.3 Pairwise Hamming distance

For each of the $\binom{5}{2} = 10$ unordered pairs, we compute the Hamming distance between the two 2^{33}-bit streams, byte by byte. For two independent uniformly random streams, the theoretical Hamming distance follows a binomial distribution with mean $2^{32} = 4\,294\,967\,296$ bits and

standard deviation $\sqrt{2^{30}} \approx 46\,341$ bits (proportion $0.500000 \pm 5.39 \cdot 10^{-6}$ at one sigma). A duplicated block gives a Hamming distance of exactly zero; a bit-inverted block gives exactly 2^{33}.

Table 8: Pairwise Hamming distances between the five featured blocks. Expected mean for independent streams: $2^{32} = 4\,294\,967\,296$ bits; expected standard deviation: $\sqrt{2^{30}} \approx 46\,341$ bits (proportion $0.5 \pm 5.39 \cdot 10^{-6}$).

Pair	Hamming distance (bits)	Proportion	Dev. from 2^{32}	z
0003 vs 0009	4 294 982 270	0.500001743	+14974	+0.323
0003 vs 0018	4 294 923 522	0.499994904	−43774	−0.945
0003 vs 0029	4 294 958 446	0.499998970	−8850	−0.191
0003 vs 0035	4 294 914 074	0.499993804	−53222	−1.148
0009 vs 0018	4 294 929 412	0.499995590	−37884	−0.818
0009 vs 0029	4 294 960 452	0.499999203	−6844	−0.148
0009 vs 0035	4 294 935 516	0.499996300	−31780	−0.686
0018 vs 0029	4 294 983 188	0.500001850	+15892	+0.343
0018 vs 0035	4 294 900 280	0.499992198	−67016	−1.446
0029 vs 0035	4 294 963 112	0.499999513	−4184	−0.090

Every observed $|z|$ is below 1.5. The observed values are wholly consistent with the independent-stream null hypothesis.

8.4 Pairwise cross-correlation

We interpret each block as a sequence of $n = 2^{30}$ unsigned bytes and compute the Pearson correlation coefficient between the two sequences at zero lag. For independent uniform byte streams of length n, the null distribution of r has mean zero and standard deviation $\sigma_r = 1/\sqrt{n} \approx 3.052 \cdot 10^{-5}$. We repeat the calculation at non-zero lags $k \in \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$ to detect shifted duplication or lagged structure.

Table 9: Pairwise Pearson correlation at zero lag and across tested lags $k \in \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$. Null standard deviation: $\sigma_r = 1/\sqrt{2^{30}} \approx 3.052 \cdot 10^{-5}$. Threshold: $|z| \le 4.0$. Global maximum across $10 \times 12 = 120$ tests: $|z|_{\max} = 3.024$.

Pair	r (zero-lag)	$\lvert z\rvert$ at lag 0	max $\lvert r\rvert$ (lags 1–1024)	lag*	$\lvert z\rvert$ at max $\lvert r\rvert$
0003 vs 0009	$-3.310e-05$	1.085	$6.017e-05$	16	1.972
0003 vs 0018	$+3.930e-05$	1.288	$5.988e-05$	128	1.962
0003 vs 0029	$-1.119e-06$	0.037	$4.276e-05$	8	1.401
0003 vs 0035	$+1.351e-05$	0.443	$6.580e-05$	1024	2.156
0009 vs 0018	$-9.433e-06$	0.309	$5.021e-05$	1	1.645
0009 vs 0029	$-2.877e-05$	0.943	$3.874e-05$	1024	1.269
0009 vs 0035	$-1.934e-05$	0.634	$5.625e-05$	16	1.843
0018 vs 0029	$-4.770e-05$	1.563	$9.229e-05$	512	3.024
0018 vs 0035	$+6.198e-05$	2.031	$6.700e-05$	512	2.195
0029 vs 0035	$-2.077e-05$	0.680	$4.022e-05$	1024	1.318

Across all $10 \times 12 = 120$ (pair, lag) tests, the largest observed $|z|$ is 3.024, below the $|z| \le 4.0$ threshold fixed for this analysis. No pair exhibits correlation at any tested lag beyond what is expected from uniform independent streams.

8.5 Cross-block chunk collision search

For each chunk size $k \in \{4\,096, 16\,384, 65\,536\}$ bytes, we partition every block into $2^{30}/k$ non-overlapping chunks and compute the SHA-256 hash of each chunk. We then search the combined set of $5 \cdot 2^{30}/k$ hashes for any value that appears more than once.

A duplicated or near-duplicated chunk inserted into two blocks would produce a SHA-256 collision. For independent uniform streams, the birthday-bound probability of observing a collision is astronomically small even at the smallest chunk size.

Table 10: Cross-block chunk collision search (SHA-256). For each chunk size k, every block is partitioned into $2^{30}/k$ non-overlapping chunks and each chunk is hashed. The search is across all chunks of all five blocks. Zero collisions are observed at any chunk size.

Chunk k (bytes)	Chunks per block	Total chunks	Unique SHA-256	Duplicate hashes	$\log_{10}$ birthday bound
4 096	262 144	1 310 720	1 310 720	0	−9852.22
16 384	65 536	327 680	327 680	0	−39445.87
65 536	16 384	81 920	81 920	0	−157816.89

Zero SHA-256 collisions are observed across 1 720 320 total chunks.

8.6 Exclusive-OR transform stability test

For two independent uniform byte streams A and B, the stream $A \oplus B$ (byte-wise XOR) is itself a uniform byte stream, because the XOR of two independent uniform random variables on $\mathrm{GF}(2)^8$ is uniform on $\mathrm{GF}(2)^8$. We derive the ten pairwise XOR streams and apply the Eormen Internal Validation metrics directly to each derived stream.

This test detects failure modes not caught by the previous levels: if $A = B$ then $A \oplus B$ is the all-zeros stream; if A and B are related by any affine transform, $A \oplus B$ is structured; if A and B were drawn from correlated seeds, $A \oplus B$ carries a subtle statistical bias.

Table 11: Statistics of the ten pairwise XOR streams. For independent uniform byte streams A and B, the stream $A \oplus B$ is itself uniform on $\{0, \ldots, 255\}$. The derived streams must therefore pass the same entropy, chi-square and compression tests as the original blocks.

Pair ($A \oplus B$)	Shannon (bits/byte)	χ^2 stat	χ^2 p	zlib ratio	Assessment
0003 XOR 0009	7.999999822642	264.003	0.3360	1.000162	PASS
0003 XOR 0018	7.999999791509	310.351	0.0101	1.000162	PASS
0003 XOR 0029	7.999999839179	239.382	0.7507	1.000162	PASS
0003 XOR 0035	7.999999835664	244.623	0.6687	1.000162	PASS
0009 XOR 0018	7.999999809790	283.132	0.1090	1.000162	PASS
0009 XOR 0029	7.999999856352	213.823	0.9714	1.000162	PASS
0009 XOR 0035	7.999999813791	277.177	0.1625	1.000162	PASS
0018 XOR 0029	7.999999828822	254.790	0.4919	1.000162	PASS
0018 XOR 0035	7.999999841946	235.268	0.8072	1.000161	PASS
0029 XOR 0035	7.999999859001	209.884	0.9821	1.000162	PASS

Every XOR stream passes with Shannon entropy ≥ 7.9999998 bits per byte, χ^2 p-values in $[0.0101, 0.9821]$, and zlib compression ratios of 1.000161–1.000162, identical to the compression ratios observed on the source blocks themselves.

8.7 Summary of distinctness evidence

Observation 8.1 (Overall adjudication)**.** Across all six evidentiary levels, the cross-block distinctness of the five featured blocks is consistent with the null hypothesis of statistically independent high-entropy streams. To defeat this chain of evidence an adversary would have to simultaneously: (i) forge four independent provenance identifiers, (ii) forge SHA-256 pre-image matches, (iii) synthesise byte streams whose pairwise Hamming distances match the binomial null to within 1.5σ, (iv) avoid pairwise correlations at all tested lags, (v) avoid cross-block chunk

Table 12: Summary of cross-block distinctness evidence. All six evidentiary levels adjudicate PASS for the five featured blocks.

Level	Test	Theoretical pass condition	Observed result	Reproducible	Outcome
L0	File byte-exactness (SHA-256)	Observed = recorded for every file	5/5 match	Yes	PASS
L1	Identifier uniqueness	5/5 unique values on four identifiers	5/5 unique on all four	Yes	PASS
L2a	Pairwise Hamming	$\|z\| \leq 4.0$ for all 10 pairs	$\max \|z\| = 1.446$	Yes	PASS
L2b	Zero-lag Pearson r	$\|z\| \leq 4.0$ for all 10 pairs	$\max \|z\| = 2.031$	Yes	PASS
L2c	XOR transform stability	Shannon $\geq$ 7.99999, $\chi^2 p \in [0.001, 0.999]$, zlib $\in [0.9995, 1.0015]$	$\text{Shannon}_{\min} = 7.9999998$, $p \in [0.0101, 0.9821]$, $\text{zlib}_{\max} = 1.000162$	Yes	PASS
L3	Lagged cross-correlation	$\|z\| \leq 4.0$ for all 110 tests	$\max \|z\| = 3.024$	Yes	PASS
L4	Cross-block chunk SHA-256 collisions	0 collisions at $k \in \{4\,096, 16\,384, 65\,536\}$	0 / 1 720 320 total chunks	Yes	PASS

collisions at three chunk sizes, and (vi) produce ten pairwise XOR streams each indistinguishable from a fresh entropy block. No known construction achieves all six simultaneously.

9 Per-block governance and the commodity entropy gap

9.1 The certification regime as the product

The preceding sections report that five Eormen Certified Entropy Blocks pass every test in a three-tier pipeline comprising Dieharder v3.31.1 (17 tests), NIST SP800-22 Rev. 1a (15 tests), and the Eormen Internal Validation Suite v6.1.0 (8 analysis categories). The pipeline enforces a zero-tolerance pass gate: a single failure in any tier rejects the candidate, the failure is recorded with the test name, the failed metric, the recorded value, the threshold, and the timestamp, and the candidate block is destroyed. No partial-pass certificate is ever issued.

What distinguishes an Eormen Certified Entropy Block from any other source of random bytes is not a claim about a superior statistical profile. It is the governance regime that accompanies every delivered unit:

- the 1 073 741 824-byte payload;
- a 128-bit nonce, a generation timestamp in UTC, an engine version and a pipeline version embedded in the 64-byte metadata trailer;
- a data-only SHA-256 hash, a metadata-only SHA-256 hash, and a complete-file SHA-256 hash;
- a detached GPG signature over the complete file;
- the full Dieharder result record for all 17 tests;
- the full NIST SP800-22 result record for all 15 tests;
- the full Eormen Internal Validation result record for all 8 categories;
- hash binding tying every result record to the data-only hash of the delivered file.

The customer can recompute the three SHA-256 hashes, verify the GPG signature, and re-run any test from any tier against the delivered `.bin` file at any future time. Every statistical claim

Eormen makes about a delivered block is independently verifiable from the delivered bytes alone, using openly available tools.

This level of per-block governance is, to the authors' knowledge, without precedent. No commodity entropy source, no operating-system random device, and no other commercial entropy product known to the authors at the time of writing delivers per-unit evidence of this scope. A consumer of `/dev/urandom`, for example, receives a byte stream with no attached test results, no hash binding, no nonce, no timestamp, no signature, and no guarantee that any particular read would pass Dieharder, NIST SP800-22, or any other battery if tested. The user must either trust the kernel random subsystem unconditionally or perform and archive the testing themselves.

This is the gap the Eormen product fills. A customer who requires auditable, frozen, re-verifiable entropy, for example for cryptographic key ceremonies, long-term archival signatures, or compliance-driven random number requirements, can point to the delivered certificate and the hash-bound evidence rather than to an implicit trust assumption about a transient kernel stream.

10 Discussion

10.1 What the results demonstrate

The five featured blocks jointly pass 85 Dieharder evaluations, 75 NIST SP800-22 Rev. 1a evaluations, and the Eormen Internal Validation pass conditions of Section 6.10 for every block. Under the six-level cross-block distinctness framework of Section 8, every pair passes at every level. No evaluation across any of the three tiers or the distinctness framework returns a failure verdict. Under the published pass criteria, and in the absence of any partial-pass certificate, each of the five blocks meets the conditions to be certified and delivered.

What the evidence supports, precisely, is this: under the disclosed three-tier pipeline and the disclosed thresholds, the byte streams of the five delivered blocks are statistically indistinguishable from a uniform independent source, and are statistically indistinguishable from one another pairwise under five independent evidentiary levels.

10.2 Governance as the distinguishing contribution

Statistical quality alone is a necessary but insufficient condition for the Eormen product. The testing protocol documented in this paper, applied to every block before delivery and bound to the delivered bytes by hash, is itself the product. The three-tier pipeline with zero-tolerance gating, the failure register, the provenance metadata, the triple SHA-256 hashing, and the detached GPG signature collectively provide a level of per-unit governance that is not available from any commodity entropy source.

A consumer who draws 1 GiB from `/dev/urandom` may receive bytes that would pass the same tests, but has no way to know this without performing and archiving the tests independently. A consumer who receives an Eormen Certified Entropy Block does know, because the evidence arrives with the block and is tied to its exact bytes. This distinction matters wherever auditability, traceability, or regulatory compliance requires that the quality of consumed entropy be demonstrable after the fact.

10.3 Independent verification

Every validation number in this paper is independently verifiable from the delivered `.bin` files and the disclosed pipeline configurations using Dieharder v3.31.1, an SP800-22 Rev. 1a conforming implementation, and Python with NumPy, zlib, bzip2, lzma and SHA-256. A customer under audit may request a replay artefact for any specific block under the non-disclosure terms of the Eormen audit programme; the replay artefact regenerates the block byte for byte and confirms the three delivered SHA-256 hashes.

10.4 The role of failure

A validation pipeline that never rejects a candidate is statistically suspect: either its thresholds are too loose, or its generator is too narrow to exercise the full range of the tests, or both. The Eormen pipeline records every failed candidate to the production register with test name, failed metric, recorded value, threshold and timestamp, and destroys the failed block after the record is written. The production register is available to customers under audit. The presence of failures in the register is treated as evidence that the pipeline is exercising its thresholds rather than rubber-stamping its inputs.

11 Conclusion

This paper has defined the Eormen Certified Entropy Block, fully disclosed its three-tier validation pipeline and the Tier 3 thresholds, and reported the complete test results for the five featured blocks. The blocks pass every tier of the pipeline and every level of the cross-block distinctness framework. Every numerical claim in the paper is independently verifiable by a customer from the delivered `.bin` file using openly available tools; no claim depends on disclosure of the Eormen Edge-of-Chaos Entropy Engine, which remains a trade secret.

The contribution this paper documents is the per-block governance regime: a zero-tolerance three-tier pipeline, cryptographic provenance binding, and a complete evidence package delivered with every certified block. This level of per-unit certification is not provided by any commodity entropy source or operating-system random device known to the authors. For any application where the quality of consumed entropy must be demonstrable, auditable, and independently re-verifiable, the Eormen Certified Entropy Block provides that assurance by construction.

A Glossary

Certified Entropy Block (CEB) A 1 GiB file together with its 64 B metadata trailer that has passed the full three-tier Eormen validation pipeline.

Data-only SHA-256 The SHA-256 hash of the first 1073741824 B of the file (the 1 GiB payload), excluding the metadata trailer.

Complete-file SHA-256 The SHA-256 hash of the entire 1073741888 B file.

Eormen Internal Validation Suite A suite of eight analysis categories (version 6.1.0) applied as Tier 3 of the validation pipeline. All thresholds are fully disclosed in Section 6.

Hamming distance (bitwise) The number of bit positions at which two equally sized bit strings differ.

Pearson correlation r The normalised covariance between two sequences of real-valued observations.

Cross-correlation at lag k The Pearson correlation between A_i and B_{i+k} for $i = 0, \ldots, n - 1 - k$.

XOR transform stability The property that, for independent uniform byte streams A and B, the stream $A \oplus B$ is itself uniform.

Birthday bound An upper bound on the probability of a hash collision given N random draws from a domain of size M.

References

[1] Elaine Barker and John Kelsey. Recommendation for random number generation using deterministic random bit generators. Technical Report SP 800-90A Revision 1, National Institute of Standards and Technology (NIST), 2015.

[2] Robert G. Brown. *Dieharder: A Random Number Test Suite (v3.31.1)*. Duke University, Department of Physics, 2020. Version 3.31.1.

[3] Debian Project. DSA-1571-1 openssl: predictable random number generator. Debian Security Advisory DSA-1571-1, 2008.

[4] Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. Mining your Ps and Qs: Detection of widespread weak keys in network devices. In *Proceedings of the 21st USENIX Security Symposium*, pages 205–220, 2012.

[5] National Institute of Standards and Technology. Secure hash standard (shs). Technical Report FIPS PUB 180-4, National Institute of Standards and Technology (NIST), 2015.

[6] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, and San Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical Report SP 800-22 Revision 1a, National Institute of Standards and Technology (NIST), 2010.

[7] Meltem Sönmez Turan, Elaine Barker, John Kelsey, Kerry A. McKay, Mary L. Baish, and Mike Boyle. Recommendation for the entropy sources used for random bit generation. Technical Report SP 800-90B, National Institute of Standards and Technology (NIST), 2018.