The Library of Binary

Every page of 32,000 bytes, and the problem of finding one

The operator
github.com/library-of-binary/bynry-protocol

July 24, 2026

Abstract

A page is 32,000 bytes; the Library is every page that could exist. We set out the arithmetic of that claim: the size of the space, why storage is a category error rather than an engineering problem, why almost every page is indistinguishable from noise, and why short words are nevertheless inevitable while any particular sentence is not. We then state the two questions the project actually asks — what can be found in the space, and whether the findable pages share a structure — and describe the frozen 1,000,000-page calibration. That calibration fixes the *marginal* behaviour of each measurement, which suffices to test the first question; the second additionally requires a joint baseline that does not yet exist, so it is posed here rather than answered. We claim no search advantage of any kind, quantum or classical. The contribution is not a discovery that random data is random; that is the null hypothesis, and it is reported as such.

1 The address space

In plain terms. A page is just a very long number written in base 256. Because the number and the page are the same object, naming a page and producing it are the same act — so nothing has to be stored for everything to exist.

A page is a string of $n = 32{,}000$ bytes. Each position independently takes one of 256 values, so the Library contains exactly

$$256^{32{,}000} \;=\; \left(2^8\right)^{32{,}000} \;=\; 2^{256{,}000} \;\approx\; 10^{77{,}064} \tag{1}$$

pages. The count is not the interesting part. The interesting part is that the base-256 numeral reading

$$N(b_0 b_1 \cdots b_{n-1}) \;=\; \sum_{i=0}^{n-1} b_i \cdot 256^{\,n-1-i} \tag{2}$$

is a bijection onto $\{0, 1, \ldots, 256^n - 1\}$. Address and content are the same object in two notations: to name the number is to produce the page, and no page exists that cannot be named.

Because bytes carry no genre, a page is simultaneously text, program, image and sound. Those are not properties of the page but of the interpretation applied to it.

Remark 1. *Borges' [11] library contains roughly* $10^{1{,}834{,}097}$ *books. That count exceeds ours only because his object is larger — a book of* 1,312,000 *characters against our 32,000-byte page [4]. Counts of unequal objects say nothing about completeness. Measured per symbol the comparison reverses, a byte spanning* 256 *values against his* 25*: the set of all book-length byte strings is* $\approx 10^{3{,}159{,}611}$*. Nothing longer is missing from either collection, since a work that outgrows one page exists as a sequence of pages.*

2 Why nothing can be stored

In plain terms. Writing the Library down is not hard, it is arithmetically impossible: there is not enough matter. And almost every page is essentially incompressible: no description of it is meaningfully shorter than the page itself.

Proposition 1. *The Library cannot be stored in the observable universe, and the shortfall is not close.*

Argument. Storing every page requires $32{,}000 \cdot 8 \cdot 2^{256{,}000}$ bits — the count of pages multiplied by the size of one. Writing one bit per atom, the universe supplies about 10^{80}. The shortfall in orders of magnitude is

$$\log_{10}(32{,}000 \cdot 8) + \log_{10}\left(2^{256{,}000}\right) - 80 \;\approx\; 76{,}989, \tag{3}$$

so no arrangement of matter suffices. One bit per atom is an illustrative rate, not a physical information bound; substituting a rigorous one changes nothing. The holographic (Bekenstein) bound [5, 23] caps the information in the observable universe at about 10^{122} bits — forty orders of magnitude above one bit per atom, and still short of the required count by essentially the same 76,989 orders of magnitude. (An earlier version of this argument dropped the per-page factor and reported 76,984; the bits-per-page term contributes the missing $\log_{10} 256{,}000 \approx 5.4$.) □

The sharper statement is Kolmogorov's [21, 22]. Let $K(x)$ be the length of the shortest program producing x. Counting gives the standard bound: there are 2^n strings of length n but fewer than 2^{n-k} programs shorter than $n - k$ bits, so

$$\Pr_{x \sim \{0,1\}^n}[\, K(x) < n - k \,] \;<\; 2^{-k}. \tag{4}$$

Fewer than one page in 2^{100} is compressible by more than 100 bits. For a typical page no description is more than a machine-dependent additive constant shorter than the page itself [22]. The address is not a pointer to content stored more compactly elsewhere; there is nowhere more compact for it to live.

3 Two alphabets

Folded inside the binary Library is the ASCII Library, in which every byte is one of the 95 printable characters. Its pages are base-95 numerals and there are $95^{32{,}000} \approx 10^{63{,}287}$ of them. Within the binary Library the printable pages are vanishingly rare:

$$\Pr[\text{page is entirely printable}] \;=\; \left(\tfrac{95}{256}\right)^{32{,}000} \;\approx\; 10^{-13{,}777}. \tag{5}$$

4 The typical page

In plain terms. Almost every page is indistinguishable from static. That is not an opinion about most pages — it is provable about nearly all of them, and it is the baseline every claim in this paper is measured against.

Let $\hat{H}$ be the empirical byte-level Shannon entropy [29] of a page,

$$\hat{H} \;=\; -\sum_{v=0}^{255} \hat{p}_v \log_2 \hat{p}_v \;\le\; 8 \quad \text{bits/byte}, \tag{6}$$

with equality exactly when every byte value is equally frequent. For a uniformly drawn page the empirical byte distribution concentrates on the uniform, and $\hat{H}$ on its maximum, by the method of types [13, Ch. 11]: an empirical distribution ε-far from uniform has probability exponentially small in n (Sanov's theorem), so

$$\Pr\Big[\,|\hat{H} - 8| > \varepsilon\,\Big] \;\le\; 2\,e^{-c\,n\,\varepsilon^2}. \tag{7}$$

At $n = 32{,}000$ a page measuring even 7.9 is already astronomically atypical. This is why the Library looks like static: structure is not merely uncommon, it is exponentially suppressed. Whether a byte string departs from uniformity at all is what the standard statistical randomness test batteries [28] are built to measure — and for almost every page there is no systematic departure of the kind they look for, only the occasional flag a genuinely uniform sample throws by chance.

This is also what the project measures against rather than what it reports. A frozen run over 1,000,000 pages, generated by the pinned rule $\text{seed}_i = \text{SHA256}(\texttt{"BYNRY\_LENS\_CALIBRATION\_V1"} \,\|\, \text{u64\_le}($ and completed before public v1 scanning began, found the longest run of identical bytes to be 5 and detected 0 repeating periods. Those are properties of the null hypothesis, not findings.

5 Words are inevitable; meaning is not

In plain terms. Short words turn up constantly, because there are so many places for them to appear. Any particular sentence, in practice, will not — not because it is absent, but because no feasible search reaches it. Both facts come from the same calculation, and the gap between them is the whole shape of the project.

Fix a five-letter word. At a given offset of a random ASCII page it appears with probability $95^{-5} \approx 1.3 \times 10^{-10}$. A page offers approximately 32,000 offsets, so

$$\Pr[\text{word appears somewhere}] \;\approx\; 1 - \left(1 - 95^{-5}\right)^{32{,}000} \;\approx\; \tfrac{1}{240{,}000}. \tag{8}$$

Any given short word therefore exists on an enormous number of pages. Note the statement this licenses: the EXPECTED number of pages to scan before meeting one is 240,000, not a guarantee of meeting it within that many. At $p = 1/240{,}000$ per page, 240,000 independent pages succeed with probability $1-(1-p)^{1/p} \approx 1-e^{-1} \approx 0.63$. Waiting times, not certainties.

Now fix a meaningful 100-character sentence. The same calculation gives $\approx 10^{-193}$ per page. The number of pages containing it remains staggering, $\approx 10^{63{,}094}$, yet the probability that any page one happens upon contains it is for every practical purpose zero. Note the scope: this is a statement about BLIND SEARCH. The address of a page containing any given sentence is computed directly in time linear in n (Section 6) — what is unreachable is arriving at one by looking.

Remark 2. *This is the Library's central asymmetry, and its central joke: everything exists, and existence is worthless. The value was never in the pages. It is in the coordinates.*

6 Search is easy; discovery is hard

In plain terms. Going from text to its address takes microseconds. Going the other way — finding, among pages nobody wrote, the rare ones where something surfaced by itself — is the hard direction, and the only one worth calling discovery.

Search — given bytes, produce the page containing them — is trivial: the bijection of Section 1 computes the address directly, in time linear in n. The inverse problem is the hard one: identifying, among pages nobody has written, the rare ones on which structure surfaced unaided.

That is what mining is: rejection sampling of the atypical set. Pages are derived, measured through fixed lenses, and the exponentially rare outliers kept. Two design properties matter for the integrity of the resulting record.

Content-blind issuance. The reward is a proof-of-work check over the derived page and never inspects what the page contains. A remarkable find earns exactly what a dull one does. The protocol consequently offers no reward premium for a remarkable page over a dull one. That removes the protocol's own incentive to fabricate; it does not remove reputational or off-protocol ones, and what actually prevents fabrication is that every published find is re-derived from its coordinates by the verifier. Deriving the page is work the miner does regardless; scanning it through the lenses is additional work — about 35× the derivation — but that work is content-independent, so no page earns a premium for what it turns out to contain.

Frozen thresholds. All notability thresholds were calibrated over the 1,000,000-page run of Section 4 and fixed before public v1 scanning began. A threshold cannot be moved to fit a result, and a negative result therefore remains reportable.

7 The two questions

> **In plain terms.** The first question — what turns up — is mostly arithmetic and is already answered. The second — whether the pages that turn up resemble each other more than chance allows — is the real one, and it is stated here so that it can come back negative.

(A) What can be found? Answerable directly, and largely by arithmetic. At least one word of five or more letters from the pinned corpus appears on roughly one page in 190; at least one of six or more, one in 14,000 (both measured empirically on the raw byte stream). A *particular* five-letter word is far rarer: the idealised printable-ASCII calculation above puts it near one page in 240,000. The two figures live in different populations — the corpus rate on the raw, case-folded bytes the lens actually reads, the particular-word rate on uniform printable ASCII — so only the direction of the comparison, not the exact ratio, transfers. A particular 100-character sentence will, for every practical purpose, never be reached *by blind search* — though its address, like any page's, is computed directly from its bytes (Section 6).

(B) Do the findable pages share a structure? The substantive question, the one requiring volume, and the one still open.

State it carefully, because the obvious statement overreaches. The calibration run fixed the *marginal* behaviour of each of the 7 patterns measurements — one frozen histogram per measurement — and nothing more. It did not fix their *joint* distribution, and the composite score is accordingly a monotone ranking index, not a joint surprisal: the measurements are correlated on structured pages, since one physical construction (a repeating byte pattern, say) can legitimately move several of them at once.

The hypothesis is that pages clearing one threshold clear others more often than the calibrated *joint* null predicts — not merely more often than the product of the marginals, since ordinary pages already carry cross-metric dependence and would clear that weaker

bar on their own. Testing it therefore requires a joint baseline that does not yet exist and must be constructed from per-page calibration data, pre-registered before it is applied. That construction is outstanding work, stated here rather than assumed, because a joint claim asserted against a baseline that was never measured is precisely the failure this design exists to prevent.

Note that (B) cannot be answered by a leaderboard. The words threshold clears about one page in 190 — itself a calibrated empirical rate, not an exact constant, since a pre-registered replication measured a close but different rate — and the count in any finite run is a random variable around it; that is arithmetic, not evidence. Only the correlation structure carries information the threshold choice did not put there.

8 Quantum sampling: a different door

In plain terms. This project claims and implements no quantum search advantage. That is a statement about what is built here, not about what is possible: a lens is a Boolean predicate, so amplitude amplification would in principle find a clearing page in $O(1/\sqrt{a})$ evaluations where blind search needs $O(1/a)$. What the processor is used for here is different — it changes where the bytes come from: measured in a suitable basis, the circuits used here produce correlated, non-uniform outcomes, so the pages carry rhythms that uniform sampling would essentially never produce. That is a claim about provenance, not speed — and it is a claim about these circuits and this readout, not about entanglement as such.

No search advantage is claimed here, and none is implemented. It would be wrong to say none is *known*: each frozen lens is a Boolean predicate over pages, and amplitude amplification [12] solves exactly that problem with a quadratic reduction in expected evaluations, $O(1/\sqrt{a})$ against $O(1/a)$ for a marked fraction a. The reason it is irrelevant here is concrete rather than principled: the oracle would have to evaluate the whole derivation — SHA-256 into ChaCha20 into a 32,000-byte page — and then the lens, coherently, in superposition over nonces. That circuit is far outside anything current hardware can hold. The honest claim is therefore narrow: this sampler is not faster, and the speedup that does exist is unreachable at this scale. The processors used here therefore do not locate structured pages faster [18] — not as a statement about quantum computers in general, but about what this work does with them. What changes is the sampling distribution: measuring the circuits used here, in the computational basis, draws bits from a correlated, non-uniform distribution. That a quantum circuit can be measured to sample a distribution no efficient classical process reproduces is the premise of random-circuit sampling [3, 8]; those circuits are random where these are structured, and the property relied on here is only the non-uniformity of the readout, not a classical-hardness claim. (Entanglement alone does not imply this: $CZ\,|{+}{+}\rangle$ is entangled yet yields four equiprobable computational-basis outcomes. The property being relied on belongs to these specific circuits and this readout, not to entanglement as a category.), so the pages that emerge carry rhythms uniform sampling would essentially never produce.

A July 2026 alpha sampled 10 pages on IBM hardware across three circuit families. All 10 cleared the patterns threshold; a matched control of 500 pseudorandom pages cleared it at approximately 1%. The effect is circuit-structured: entropy deficit separated by family across two orders of magnitude (≈ 15, ≈ 325, $\approx 2{,}082$), on one backend within one session, which is consistent with the circuit groups rather than with a single device-wide bias.

That grouping has a limit worth stating plainly. It is not read off the outcome measurements: each signed record commits the job's request parameters, and the family is recovered by matching those committed bytes to the known circuit templates. What is missing is a single explicit, pre-registered family *label* bound into the record — so the grouping depends

on the operator's template set rather than on a field the record certifies on its own. Binding such a label into each signed job record is outstanding work.

Family	Entropy deficit	Metrics outside calibration	Pages
Brick-layer	≈ 15	1	4
Quantum walk	≈ 325	4	3
Partial GHZ	$\approx 2{,}082$	5	3

Table 1: The alpha, counted against the measured calibration extents. The number of measurements falling outside the observed 1,000,000-page range differs by circuit family; the families are not ordered by circuit depth, and no depth ordering is claimed.

Two limits attach. No computational advantage is claimed — the shallow circuits affordable here are not run to compute anything faster — so the property of interest is provenance rather than a practical quantum advantage [19]: bytes originating in a physical measurement rather than a pseudorandom generator. Because provenance can be imitated, it is recorded as ARCHIVED EVIDENCE — the protocol's weakest tier, deliberately not its ATTESTED tier, which would require an independently verifiable attestation — and never described as verified. Second, quantum finds carry attribution only — credited to the operator who ran the job, never to the device — and cannot mint anything.

Remark 3. 10 *pages is an alpha, not a result. What would make it one is a prediction with a* defined *ordering — some property of the circuits fixed in advance against which the deficit is expected to track — registered before the next run rather than read off these same three points.*

9 What would falsify this

Question (B) is left *unsupported* if, once the archive is large and the joint baseline of Section 7 has been constructed and pre-registered, the observed joint behaviour of the 7 patterns measurements over mined pages is consistent with that baseline. A stronger *negative* conclusion — that the excess correlation is absent, not merely undetected — requires a pre-registered smallest relevant effect and an equivalence test with adequate power; without that, consistency means the signal is not supported, not that it is ruled out. The per-measurement thresholds were fixed in advance precisely so that this outcome stays reportable, and it will be reported.

Note the order of operations, which is itself part of the claim: the joint baseline must be registered before it is compared against, or the comparison becomes unfalsifiable in the ordinary way. Until then the honest position is that (B) is open and untested, not that it is supported.

10 The coin beneath the library

In plain terms. BYNRY is a program on a general-purpose chain; the chain provides agreement on state, and a proof-of-work rule governs only how coins are issued. Crucially, the content of a page cannot change the reward for deriving it — so a fabricated find earns no extra coin, and it is the server's re-derivation of every claimed page, not any absence of motive, that keeps the research record honest.

BYNRY is a program deployed on Solana, a general-purpose smart-contract chain in the line of Ethereum [33]; consensus, ordering and finality are supplied by the underlying chain,

and what the program adds is a permissionless way to search the Library and a tamper-evident record of what turned up. Coin issuance is a proof-of-work rule in the tradition of Bitcoin [25]: a miner picks coordinates, derives the page there, and competes to clear a difficulty target. The proof-of-work relation is over the coordinates and a *commitment* to the derived page, $H(\text{challenge}, \text{identity}, \text{nonce}, \text{output\_commit})$; a separate proof establishes that output_commit = Commit(CanonicalPage(seed)). The two stages matter: at filing time an attacker can hash a chosen commitment without deriving any page, which is the source of the claim-flooding cost of Section 11.

The property that makes the research layer trustworthy is that *the content of a page cannot increase the coin it earns*: a page that spells a word, or clears a structure threshold, mints exactly what an apparent-noise page mints for the same work. This removes the incentive to fabricate an interesting find for profit — though it does not remove attribution, leaderboards, or off-protocol motives, and scanning a page through the lenses is itself real work (about 35× the derivation). What actually forecloses a false report is re-derivation: the server re-derives every claimed page from its coordinates and rejects any that does not reproduce, so a claimed find that was not really there simply fails to verify.

11 Five constraints that shape the design

In plain terms. The protocol's shape is forced by five constraints of four different kinds: two are counting lower bounds, one is a definitional incompatibility, one is an information-theoretic indistinguishability, and one is an economic tradeoff rather than a proved impossibility. Each mechanism below is the construction the design ledger *adopts* to live within one of them — not a proven optimum.

11.1 An address cannot be shorter than its page (a counting bound)

No lossless fixed-length address shorter than a full 32,000-byte page can name every page: by pigeonhole a code of 2^b addresses cannot injectively cover $2^{256{,}000}$ pages when $b < 256{,}000$. A 256-bit seed reaches at most 2^{256} of the $2^{256{,}000}$ pages — a fraction of $2^{-(256{,}000-256)}$.

Construction. Mining does not enumerate the Library; it samples a sublibrary that is pseudorandom under the standard assumptions on its primitives. Coordinates are expanded through SHA-256 [1] and ChaCha20 [26]; the pages are then pseudorandom rather than independent-uniform, and steering the search toward a chosen page is assumed hard under the preimage-resistance of the hash and the pseudorandomness of the stream cipher. (The standards specify the primitives; they do not by themselves prove the composed sampler pseudorandom, and that composition is an assumption, not a theorem, here.) Re-derivation needs only that this mapping is *deterministic*: a verifier recomputes any claimed page from its coordinates alone, with nothing stored, whether or not the coordinate-to-page map is injective. The address-is-content bijection of Section 1 concerns the full Library; the protocol samples a generated sublibrary and rests on determinism, not injectivity.

11.2 A program cannot be both immutable and crypto-agile (a definitional incompatibility)

A deployed verifier cannot be simultaneously permanently immutable and automatically able to migrate away from a primitive that a future adversary breaks: immutability forbids the upgrade, agility requires it.

Construction. The design ledger resolves the tension with verifier-sunset epochs. Each acceptance path — including a deterministic, authority-independent fallback — is pinned

with an expiry, and *every* path is gated by that sunset. The fallback stays executable for the deployment's life, but its authorization to mint expires with each pinned policy, so emission *fails closed* until a successor policy is pinned. The system can therefore be frozen against tampering yet still be forced to stop, rather than silently accept a broken primitive, when a policy lapses. Claimant and manifest signatures use Ed25519 [7, 20] under *strict* verification (`verify_strict`), which rejects the small-order and non-canonical keys a permissive check would accept [15]. The randomness beacon — drand [31], a threshold-BLS network [9] — uses BLS signatures rather than Ed25519; verifiable delay functions [10] and publicly-verifiable secret sharing [30] are alternative unbiasable-randomness constructions the design does not adopt.

11.3 An unverified permissionless claim cannot be cheap, capacity-reserving, and denial-of-service-proof at once (an economic tradeoff)

If a claim can be filed without a full proof at filing time, and anyone may file it, then three properties cannot hold together: the claim is cheap, it reserves scarce capacity, and it is immune to economic denial of service. Cheapness invites flooding; reserving capacity for a cheap claim hands an attacker a denial-of-service lever. This is a cost, not an impossibility, so it is answered with a price rather than a proof.

Construction. A per-shard escalating bond schedule makes capacity elastic and append-only, and a *saturated* epoch fails closed *for emission* while settlement and bond accounting continue. The price is stated explicitly as $\text{cost\_to\_saturate} = \text{shard\_count} \cdot \sum_{0 \le i < \text{shard\_capacity}} \text{bond}(i)$, the posted-bond sum for the deployed policy — the cost of exhausting an epoch's ticket capacity, not a universal lower bound on every attack (canonical valid tickets refund). A later page-binding proof validates a ticket after filing; the authorization to submit membership is a separate mechanism, admin-attested on the current devnet.

11.4 A signature cannot show that a search happened (an indistinguishability)

Participation in the research layer means reporting a stretch of the space that was searched, most of which yields nothing. A signature over such a report authenticates its author and protects its integrity; it does not, and cannot, establish that the pages were ever computed. Consider a miner who honestly scans a window and finds nothing notable, and one who computes nothing and signs the same declaration. Both produce the same signed fields over the same window and the same hash of an empty record set: the two transcripts are identical, so their distributions coincide and no verifier separates them with probability better than chance. This is a property of what digital signatures provide — origin authentication and integrity, not the truth or timeliness of the signed statement [2, 14] — not a defect of any particular scheme. Verifying the claim directly costs what producing it cost: re-deriving a window is the same work as scanning it, measured here at ≈ 1.9 ms per page, so an indexer keeping pace with k volunteers must out-compute all k. A succinct proof over the scan would change this, at a per-window cost far above the budget such a report is worth.

Construction. The design gives up on proving coverage and instead refuses to let an unprovable number pass as a proven one. Three things follow. First, the window is *declared and signed before any page in it is derived*, and the completion signature commits to the record set and the aggregate counts, so a miner cannot choose a window after seeing what is in it, nor restate its contents afterwards. Second, every record a report does carry is re-derived byte-for-byte at ingest against the canonical evaluation, and a report containing a single record that does not reproduce is refused *entirely* — so what the report asserts about pages it found is checked in bulk, never trusted. Third, and structurally, the archive keeps

two separate quantities: the union of declared windows, which is what participants say they searched, and the set of pages the archive re-derived for itself, which is a lower bound on work actually done. They are held in different data structures with different shapes — an interval union against a point set — so a consumer cannot add them or mistake one for the other, and every published figure is labelled with which one it is. How far the second falls below the first is a property of the epoch, not a constant: under a hard proof-of-work target only the rare threshold-clearing page leaves a record and the proven figure is a small fraction of the attested one, while under a permissive target — as on the current test network — every derived page is a solution, every page leaves a record, and the two coincide. In both regimes the proven quantity is the only coverage-shaped number in the system that cannot be asserted for free.

11.5 An exact perpetual set needs $\Omega(n)$ information somewhere (a counting bound)

An exact, perpetually reconstructable set has an unavoidable information cost. For an arbitrary — here, effectively pseudorandom — set of n elements drawn from a 2^{256}-element universe, a lossless representation needs $\log_2 \binom{2^{256}}{n} \approx n(256 - \log_2 n)$ bits, i.e. $\Omega(n)$ in n; a succinct-verification scheme does not remove this. Zero-knowledge proofs compress the on-chain *verification* of a set-transition to one root and a constant-size proof, but the data must still be retrievable for the set to be reconstructable.

Construction. Output-set membership is committed as an on-chain root (with a live count, bounded at $2^{40} - 1$ members in V1), and validated by a succinct set-transition proof — a zero-knowledge VM guest wrapped in a Groth16 receipt [16] — that a new root extends the old, over an authenticated dictionary keyed by a Merkle root [24]. The $\Omega(n)$ obligation is met on chain directly: the set reconstructs from the ordered transition history alone — every settled batch carries its inserted winner set. Each epoch's research object *must* publish its post-state snapshot and transition witness; those snapshots are mandatory listed artifacts, not the primary reconstruction path, and losing one is a recorded availability degradation event, never a silent gap. Stronger availability — erasure-coded [27] shards with data-availability sampling [17] — is noted in the ledger as future work, not the V1 mechanism. The archive is a hash chain anchored on chain, so truncation *through the latest authenticated on-chain checkpoint* is detectable. A local write-ahead head also detects a truncated suffix unless an attacker rewrites both the log and that head; what is absent before the next checkpoint is *externally authenticated* detection. Anchoring is currently an operator action.

12 Measurement without moving the target

> **In plain terms.** The research claims rest on thresholds that are immutable within a lens version and were fixed before public scanning at that version began. That ordering — calibrate, freeze, then scan — is what stops a bar from being drawn around a result after the fact.

Every page a miner derives is scanned by a fixed battery of lenses; the structure lens alone takes 7 separate readings. Each reading's notability threshold was calibrated over a frozen run of 1,000,000 pages and is immutable within a lens version — a changed threshold creates a *new* version with its own parallel record stream — and the lens-v1 thresholds were fixed before public v1 scanning began. Freezing the thresholds first is the pre-registration that answers the standard objection to any outlier search: with 7 readings over many pages, some page clears some threshold by chance — the multiple-comparisons problem that false-discovery-rate control [6] exists to address — so a threshold chosen after seeing the data would

manufacture significance. Fixing it first makes a cleared threshold a *descriptive notability* result under the frozen empirical calibration — an occurrence, not by itself a p-value, an experiment-wide significance, or a confirmed discovery.

Confirmed search is a separate, preregistered layer. There the protocol controls false discovery online with e-LOND [34] over conditionally valid e-values [32] at ordered positions, with per-epoch guarantees under its admission assumptions. Two V1 qualifications are load-bearing: on devnet the registration of a search is governance-gated, and an e-value's conditional validity rests on reviewed registered templates rather than on a verification that proves them mathematically valid. This is distinct from the descriptive lens readings on mined pages, to which it is not automatically applied. The frozen thresholds also make *negative* results reportable: because the bar cannot move, a campaign that clears nothing is evidence about the named SHA-256/ChaCha20 sampler at that lens version, threshold, and sample size — not about the whole Library.

13 Provenance, not advantage

In plain terms. Quantum records in the archive are evidence of *where bytes came from*, never of a faster search. No computational advantage is claimed, and the record does not prove a quantum job ran — it binds a claimant to a page, and carries the archive's weakest evidentiary tier.

The archive admits pages sampled on quantum hardware, and is careful about what such a record establishes. No computational advantage is claimed: a lens is a Boolean predicate, and while amplitude amplification [12] gives a quadratic *query-complexity* improvement — $O(1/\sqrt{a})$ oracle uses against $O(1/a)$ repetitions — realizing it would need a reversible oracle over the full derivation and the lens coherently in superposition, which is (by an uncited engineering assessment, not a cited result) far beyond current hardware. What a quantum run changes is the *distribution* the bytes are drawn from, and certifying even that is subtle [18]. Such a record therefore carries the archive's *weakest* tier, `ARCHIVED_EVIDENCE`: the claimant signature and verified decode bind the claimant and the page but do *not* prove that the submitter ran a quantum job; the `ATTESTED` tier, which requires an independent attestation document with a verified signer, is reserved and is not claimed for these records. The separation observed in the July 2026 alpha — 10 hardware pages clearing the patterns threshold against about 1% of 500 controls, ordered by circuit family — is a measured pulse at $n = 10$, not a proof.

14 The honest limits

In plain terms. What the protocol does not establish is stated as plainly as what it does. The second research question is open, the quantum records are archived evidence rather than attested, and on the current test network several checks are still run by the operator rather than by the chain.

Three limits follow from the constraints above and are worth stating without hedging. First, the joint question of Section 7 is *open*: the marginal calibration fixes each reading alone, but the joint baseline that would let excess cross-lens correlation be tested does not yet exist, so the second question is posed, not answered. Second, a quantum record carries the `ARCHIVED_EVIDENCE` tier and never proves hardware origin. Third, the perpetual-set obligation is standing, not solved: the data-availability duty is an ongoing cost the network must keep paying, and on the current devnet several verification seams — epoch settlement,

challenge distribution, and membership-submission authorization among them — are still admin-attested rather than fully chain-verified. None of these is hidden by the design; each is a place the construction is honest about the boundary it works against.

Remark 4. *The mechanisms above are stated at the level the design ledger fixes them; the exact constructions — the bond schedule, the verifier-policy state machine, the e-LOND accountant, the archive's two-archive admission and retrieval rules, and the emission arithmetic — are specified in the protocol's decision ledger and* `specs/`*, and a per-spec treatment is left to a companion document.*

References

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

[2] Digital signature standard (dss). Technical Report FIPS PUB 186-5, National Institute of Standards and Technology, 2023.

[3] Frank Arute, Kunal Arya, Ryan Babbush, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574:505–510, 2019.

[4] Jonathan Basile. *Tar for Mortar: The Library of Babel and the Dream of Totality.* punctum books, 2018.

[5] Jacob D. Bekenstein. Universal upper bound on the entropy-to-energy ratio for bounded systems. *Physical Review D*, 23(2):287–298, 1981.

[6] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57(1):289–300, 1995.

[7] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2(2):77–89, 2012.

[8] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, et al. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14:595–600, 2018.

[9] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *Public Key Cryptography — PKC 2003*, pages 31–46. Springer, 2003.

[10] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In *Advances in Cryptology — CRYPTO 2018*, pages 757–788. Springer, 2018.

[11] Jorge Luis Borges. La biblioteca de babel. In *El Jardín de senderos que se bifurcan.* Editorial Sur, Buenos Aires, 1941.

[12] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Information*, volume 305 of *Contemporary Mathematics*, pages 53–74. American Mathematical Society, 2002. arXiv:quant-ph/0005055.

[13] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley-Interscience, 2 edition, 2006.

[14] Quynh Dang. Recommendation for digital signature timeliness. Technical Report NIST Special Publication 800-102, National Institute of Standards and Technology, 2009.

[15] Henry de Valence. ZIP 215: Explicitly defining and modifying Ed25519 validation rules, 2020. Zcash Improvement Proposal.

[16] Jens Groth. On the size of pairing-based non-interactive arguments. In *Advances in Cryptology — EUROCRYPT 2016*, pages 305–326. Springer, 2016.

[17] Mathias Hall-Andersen, Mark Simkin, and Benedikt Wagner. Foundations of data availability sampling. Cryptology ePrint Archive, Paper 2023/1079, 2023.

[18] Dominik Hangleiter and Jens Eisert. Computational advantage of quantum random sampling. *Reviews of Modern Physics*, 95:035001, 2023.

[19] Nils Herrmann, Daanish Arya, Marcus W. Doherty, et al. Quantum utility — definition and assessment of a practical quantum advantage. *arXiv preprint arXiv:2303.02138*, 2023.

[20] Simon Josefsson and Ilari Liusvaara. Edwards-curve digital signature algorithm (EdDSA). Technical Report RFC 8032, Internet Engineering Task Force, 2017.

[21] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7, 1965.

[22] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 4 edition, 2019.

[23] Seth Lloyd. Ultimate physical limits to computation. *Nature*, 406:1047–1054, 2000.

[24] Ralph C. Merkle. A digital signature based on a conventional encryption function. In *Advances in Cryptology — CRYPTO '87*, pages 369–378. Springer, 1988.

[25] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[26] Yoav Nir and Adam Langley. ChaCha20 and Poly1305 for IETF protocols. Technical Report RFC 8439, Internet Engineering Task Force, 2018.

[27] Irving S. Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[28] Andrew Rukhin, Juan Soto, James Nechvatal, et al. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical Report SP 800-22 Rev. 1a, National Institute of Standards and Technology, 2010.

[29] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[30] Ewa Syta, Philipp Jovanovic, Eleftherios Kokoris Kogias, Nicolas Gailly, Linus Gasser, Ismail Khoffi, Michael J. Fischer, and Bryan Ford. Scalable bias-resistant distributed randomness. In *IEEE Symposium on Security and Privacy*, pages 444–460, 2017.

[31] The drand Project. drand: A distributed randomness beacon daemon, 2020. League of Entropy.

[32] Vladimir Vovk and Ruodu Wang. E-values: Calibration, combination and applications. *Annals of Statistics*, 49(3):1736–1754, 2021.

[33] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger, 2014. Ethereum Yellow Paper.

[34] Ziyu Xu and Aaditya Ramdas. Online multiple testing with e-values. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 238 of *PMLR*, pages 3997–4005, 2024.