ENI6MA Complexity White Paper: Proof, Entropy, and Matrix Offsets

August 25th 2025

by F.D. Rosario, Dr. Lin Wang, Dr. Francisco Perez
@ENI6MA.com

Abstract

We introduce **ENI6MA**, a stateless, witness-driven proof-of-knowledge (PoK) architecture that replaces long-lived secrets with **ephemeral, time-coupled evidence**. The system is realized through five cooperating strata: (i) a **two-way hash keystream and framing layer** for length-preserving, random-access encryption with authenticated headers; (ii) **entropy-pool generation and sealing** of uniform 512-bit values; (iii) a **deterministic runtime loader** that validates and exposes seeds in multiple numeric views; (iv) a **session entropy and offset computation** that mixes microsecond time with selected seeds to rotate **three concentric alphabet rings**; and (v) an —Π **control matrix** that deterministically extracts per-round offsets and drives a six-zone **manifold projection**. In each round, the prover reveals only a **witness set** (zone) and the verifier performs **set-membership checks**; acceptance is the Boolean accumulator $\Lambda = \bigwedge_i [\, s_i \in W_i \,]$. Verification is $O(R)$ time and $O(1)$ space, relying solely on symmetric primitives (PRF/XOF, rotations, constant-time scans).

Security arises from **combinatorial breadth** rather than heavy algebra. With canonical rings $(30, 30, 12)$, six hyperplanes, and a **fresh hidden color bijection each round**, the structure-free brute-force space for $R = 20$ is

$$N = \big(10{,}800 \times 6!\big)^{20} = (7{,}776{,}000)^{20} \approx 6.533 \times 10^{137},$$

a **138-digit** integer (exactly **60** trailing zeros) corresponding to $\log_2 N \approx$ **457.8** bits. Independently, under balanced foliation into $Z = 6$ zones across three rings, the blind-guess false-accept rate tightens as $p_{\mathrm{FA}} \approx Z^{-3R}$, yielding policy-tunable soundness with linear verification cost. Because rings are re-rotated and color morphisms are re-drawn **every round**, transcript marginals are flat and **non-reusable**; observers learn "accept/deny," not the private mappingachieving a **passive zero-knowledge** posture with minimal, deterministic audit artifacts.

Operationally, ENI6MA addresses the dominant failure modes of key-centric systems. There are **no persisted credentials** to exfiltrate or replay; captures are time-scoped transcripts that verify deterministically yet reveal no statistically binding facts about the witness. The same binary functions **online** (live clocks) or **offline** (bounded τ windows), making the design suitable for high-QPS services, PAM/JIT elevation, CI/CD robots, IoT/edge devices, and DDIL/air-gapped environments. Compared with passwords, passkeys, and PKI, ENI6MA **removes static credentials** and collapses breach blast radius. Compared with practical ZKPs/SNARKs, ENI6MA forgoes arithmetization, trusted setup, and bulky proofs, while delivering ≈**458-bit** brute-force difficulty at modest round counts and transcripts that are single-use, auditable, and statistically uninformative.

We conclude that ENI6MA provides a **human-accessible, post-quantum-friendly PoK substrate**simple to implement, economical to verify, and robust under passive observationreframing identity as **contextual work** rather than secret possession and aligning security economics with deployability at scale.

Keywords: proof of knowledge, stream framing, PRF/XOF, entropy pools, cyclic rotations, manifold foliation, set-membership verification, stateless security

1. Introduction

Modern enterprises struggle with a paradox: we must **prove** a person or agent knows *the right thing* (or is *present now*) **without** storing reusable secrets that can be phished, dumped, replayed, or extorted. **ENI6MA's Rosario–Wang Proof (RWP)** resolves that paradox by turning identity from a *possession* (passwords, keys, passkeys) into a **perception-bound proof of knowledge** that is ephemeral, auditably verifiable, and fast enough for everyday operations.

At its core, ENI6MA RWP produces a tiny, per-round "witness" that **proves membership** in a time-keyed, high-dimensional structure, without revealing the private mapping behind it. Each session is driven by three ingredients: a microsecond time coefficient τ, an embedded 512-bit prime π, and uniformly random entropy ε (often 256–512 bits). These are not static keys to be guarded for years; they are **session inputs** that drive a **deterministic, short-lived manifold** of symbols. From that manifold, the prover selects a visible **zone** consistent with a private string S (or policy token); the verifier merely checks that each character of S appears in the corresponding zone's witness string, **linear-time verification with minimal leakage**.

What the system does (operationally)

1. **Projects entropy into rotating "rings."** ENI6MA arranges alphabets (e.g., 30-char uppercase, 30-char lowercase, 12-char digits) as **cyclic character rings**. Time-mixed 3-digit slices derived from ε and τ yield per-ring

rotation offsets Θ. This means that even a 1-µs change in τ reshuffles which characters appear where.

2. **Foliates each ring into six balanced "hyperplanes."** After rotation, each ring is partitioned into six contiguous, near-equal slices (zones). For the canonical 30/30/12 rings, a zone's witness concatenates 5 uppercase + 5 lowercase + 2 numeric symbols, **12 total characters per zone per round**, with balanced exposure that prevents exploitable frequency bias.

3. **Captures a minimal "witness" per round.** The prover (human or agent) sees six colored zones; picking one zone is tantamount to saying "my i-th secret character is **in this set**." The verifier records only the zone's 12-character witness string X_i and (optionally) deterministic indices for audit.

4. **Verifies with a membership check, not a decrypt.** Let the secret be $s = s_1 \dots s_L$ and the collected witnesses $X_1, \dots, X_T$. The verifier accepts iff there are enough rounds and each secret character is contained in its paired witness:

$$\text{ACCEPT} \iff T \geq L \ \wedge \ \bigwedge_{i=1}^{L} \mathbf{1}\{\text{lower}(s_i) \in \text{lower}(X_i)\} = 1.$$

Accept if T is at least L and, for every i from 1 to L, the lowercase of the i-th secret character appears in the lowercase of the i-th witness string.

That's the whole check: **no long-lived keys**, no decrypting blobs, no server-side password equivalents, just a deterministic, time-keyed inclusion test.

Implications of Eni6ma (RWP) over ZKP and PKi

- **Eliminates "secret sprawl."** There are no long-lived passwords, API keys, JWT signing keys, or device seeds to steal at rest. Compromise of a database or vault doesn't gift an attacker replayable credentials.
- **Stops phishing and replay at the root.** A captured witness is **ephemeral** and tied to τ at microsecond granularity; even perfect screenshots are stale next round.
- **Linear-time, auditable verification.** The check is a handful of string operations and indexes, **O(n)** in the length of the claimed secret, suited for high-QPS services, PAM gates, CI/CD robots, and IoT edges.
- **Online and disconnected.** The same binary works **fully offline** (fixed τ windows) or online (live clocks), which is essential for air-gapped, DDIL, and regulated environments.
- **Accessible to humans, native for agents.** People can visually select zones; agents follow the same protocol programmatically. This unifies human login, machine auth, and just-in-time (JIT) privilege elevation.

- **Clear separation of concerns.** The math answers "Did the witness pass?" Governance answers "Should policy grant access?" This makes **compliance and audits** cleaner: transcripts reveal minimal, precise facts.

How RWP complexity scales to "cosmic" levels

Let us consider the attacker's problem not as algebra but as a combinatorial edifice that multiplies along three axes: time, topology, and hidden maps. Each round is built from (i) microsecond-scale timing that perturbs index selection, (ii) independent rotations of multiple alphabet rings via entropy/time mixing, and (iii) a private witness morphism that the verifier and prover share but never reveal. The upshot is a per-round hypothesis cloud that compounds across rounds: constant-time work for the twins, but an exponential "tower" for an adversary. In ENI6MA terms, freshness comes from a temporal nonce mixed with entropy to rotate the ring before each foliation; the same nominal input therefore traces a different valid path whenever the clock or context shifts, which kills replay by construction.

Two structural properties deny an attacker the usual footholds for correlation and frequency analysis. First, foliation produces near-balanced slices; in the canonical three-ring (30,30,12) design with six zones, each witness exposes ˜12 of 72 symbols per round, yielding a uniform success baseline of $(1/6)^R$ under random guessing across R rounds. Long-run marginals are therefore flat, and per-zone frequencies do not drift in a way that can be exploited. Second, the hidden morphism injects an additional factorial ambiguity (6! under six zones) and, under the product-manifold interpretation, explodes the key-recovery space: reconstructing the private maps amounts to searching a space on the order of 10,800!, about $2^{129\,\mathrm{k}}$ possibilities per map, pure preimage/key-search hardness with no helpful structure exposed on the wire.

Crucially, the privacy of transcripts is not rhetorical. The system's assurance envelope formalizes **passive zero-knowledge**: with $\mathcal{M}$ confined to the twins and one-time witnesses driven by fresh entropy and nonce, an eavesdropper's view is simulatable and "leaks nothing beyond acceptance." The per-attempt error budget is explicit, $\epsilon = C^{-L} + 2^{-\lambda}$, and the verifier's runtime is linear in the number of rounds, one rotation, one morphism lookup, one combine per round, so defenders scale assurance by dialing L (or the zone count C) without changing asymptotic cost.

The RWP Eni6ma Complexity Formulation

1. $A(\alpha) := \sum_{j=1}^{m} \log_2 |\alpha_j|$
   "A of alpha equals the sum, from j equals one to m, of log base two of each ring's size."

$A(\alpha)$ compresses a heterogeneous collection of ring sizes into a single "bit-mass" measure. Because $\log_2\big(\prod_j |\alpha_j|\big) = \sum_j \log_2 |\alpha_j|$, it faithfully represents the information content of choosing one symbol per ring in base-2 units.

Operationally, $A(\alpha)$ lets designers trade many small rings for a few large ones while keeping a clear accounting of total combinatorial capacity. It is the natural unit for tuning security knobs and for comparing alternative alphabets on an equal-bits footing.

2. $C_R = \Big(W\, 25^{\,m}\, \sigma\, \prod_{j=1}^{m} |\alpha_j|\Big)^{R} \Gamma$
   "C sub R equals, parenthesis W times twenty-five to the m times sigma times the product of ring sizes, close parenthesis, raised to the R, times Gamma."

C_R is the attacker's search space over R rounds. Per round, timing ambiguity W, balanced slicing 25^m, optional routing σ, and the raw ring product $\prod |\alpha_j|$ define the hypothesis count; raising to R captures independence across rounds, and Γ (e.g., 6!) accounts for the hidden witness morphism.

Intuitively, each round is cheap for the prover (local rotations and a private lookup) but expensive for the adversary (global enumeration without structure). The product-then-power form makes the "tower of power" explicit: small per-round factors amplify exponentially across rounds.

3. $S := \log_2 C_R = R(\log_2 W + m \log_2 25 + \log_2 \sigma + A(\alpha)) + \log_2 \Gamma$
   "S equals log base two of C sub R; that expands to R times the sum of the log terms plus log base two of Gamma."

S expresses security as effective bits, converting multiplicative structure into additive contributions. Each design choice, timing spread, slice count, routing, ring sizes, adds linearly to S, while the morphism contributes a constant offset $\log_2 \Gamma$.

This linear decomposition enables principled tuning: to gain ΔS bits, increase R or any log-term by $\Delta S/R$. It also clarifies sensitivity: doubling W adds R bits; adding one ring multiplies per-round hypotheses by its size, adding $R \log_2 |\alpha_{\text{new}}|$ bits.

4. $T_{\text{att}}(R) \in \Omega\big(C_R\big) = \Omega\big(2^S\big)$
   "Attacker time is in Omega of C sub R, which equals Omega of two to the S."

This lower bound formalizes that, absent exploitable structure, any successful attack scales at least linearly with the search space. If transcripts are simulatable and frequency-flat, generic enumeration (or its parallelized variants) sets the dominant cost.

Equating $\Omega(C_R)$ with $\Omega(2^S)$ connects combinatorics to information: S bits of uncertainty imply at least 2^S work. Thus, every additional bit in S doubles the asymptotic attack time.

5. $T_{\text{ver}}(R) \in \Theta(R)$
   "Verifier time is in Theta of R."

Verification performs constant work per round, one rotation, one slice membership, one combine, so total cost grows linearly with the number of rounds. There is no hidden superlinear term from keys or exponents.

This asymmetry, linear verification versus exponential attack growth, is the economic heart of the design. Operators dial R to meet policy while keeping verification predictable and low-latency.

6. $\Pr[\mathsf{ACCEPT}] \leq 6^{-R}$ and $I(\text{Transcript}; \text{Witness}) \approx 0$
   "The probability of accept is at most six to the minus R." "The mutual information between transcript and witness is approximately zero."

With six balanced zones per round, a blind guess succeeds with probability $1/6$; independence across rounds multiplies these odds, yielding 6^{-R}. This gives an explicit, policy-tunable false-accept bound free of algebraic assumptions.

The near-zero mutual information asserts passive zero-knowledge: observed transcripts convey no statistically binding facts about the private witness map. Flat marginals, time aliasing, and the hidden morphism jointly deny correlation and frequency footholds.

7. Canonical Alphabets instance
   The cardinal size of the projective alphabets contributes to the expansion of the attack space where the enumerated sets of :

$$AtoZ = |30|(\alpha_1)$$

$$atoz = |30|(\alpha_2)$$

$$0to9 = |12|(\alpha_3)$$

   compond the exponent such that $A(\alpha) = \log_2 30 + \log_2 30 + \log_2 12$; hence $S = R(\log_2 W + m \log_2 25 + \log_2 \sigma + A(\alpha)) + \log_2 \Gamma$
   "A of alpha equals log base two of thirty plus log base two of thirty plus log base two of twelve; S follows from the same formula."

The canonical triple contributes a fixed bit-mass $A(\alpha) \approx \log_2(10{,}800)$, giving a transparent baseline for capacity per round. Designers then add timing (W), routing (σ), and rounds (R) to reach target S.

This plug-and-play evaluation illustrates the method: compute $A(\alpha)$, select W, σ, choose R, and read off S. The outcome is an auditable, parametric security budget aligned with operational latency.

Complexity Statement (RWP, per-round hyperplane bijection)

Setting. Three concentric rings with alphabet sizes $\alpha = (30, 30, 12)$ yield $\prod_j |\alpha_j| = 10{,}800$. Each round independently draws a hidden color (hyperplane) bijection from S_6 ($|S_6| = 6! = 720$). For $R = 20$ rounds, an attacker observing only transcripts and lacking exploitable structure must enumerate the Cartesian product of ring rotations and color maps per round.

Search space.

$$N \;=\; \Big(\; \underbrace{10{,}800}_{\text{ring rotations}} \;\times\; \underbrace{6!}_{\text{color bijection}} \;\Big)^{\overbrace{20}^{\text{rounds}}} \;=\; (10{,}800 \cdot 720)^{20} \;=\; 7{,}776{,}000^{20}.$$

Asymptotic cost. On a unit-cost RAM model,

$$T_{\text{att}}(R) \in \Omega(N) \;=\; \Omega\big((10{,}800 \cdot 720)^{R}\big), \qquad T_{\text{ver}}(R) \in \Theta(R),$$

i.e., attack cost grows exponentially in R while verification remains linear.

Bit security and magnitude.

$$S \;:=\; \log_2 N \;=\; R\big(\log_2 10{,}800 + \log_2 720\big) \;\approx\; 20\,(13.39874 + 9.49185) \;\approx\; \mathbf{457.81}\ \text{bits.}$$

$$\log_{10} N \;=\; 20\big(\log_{10} 10{,}800 + \log_{10} 720\big) \;\approx\; 137.815 \;\Rightarrow\; N \approx 6.533186235 \times 10^{137}.$$

Decimal structure.

Since $7{,}776{,}000 = 2^8\,3^5\,5^3$, we have

$$N = (7{,}776{,}000)^{20} = 2^{160}\,3^{100}\,5^{60} = M \cdot 10^{60},$$

so N is a **138-digit** integer .

Putting these together yields the "brute-force tower of power": per-round hypothesis sets multiply across rounds while the transcript distribution remains flat and one-time. No stable co-occurrences persist for a signals analyst; time aliasing and re-randomized rotations de-correlate observations; and the private morphism keeps the mapping space astronomically large. Operationally, this achieves a stronger **non-leakage** posture than many practical ZK instantiations: transcripts are single-use, auditable, and statistically uninformative in the passive model, even as verification remains $O(L)$. (Formally, ENI6MA does not claim general NP zero-knowledge; its guarantees are scoped to identification with passive-ZK transcripts and information-theoretic soundness absent leakage.)

Let us make the magnitude explicit. With three rings $(30, 30, 12)$, $R = 20$ rounds, and a fresh hyperplane bijection each round, the attacker's structure-free space is

$$N \;=\; \big(10{,}800 \times 6!\big)^{20} \;=\; (7{,}776{,}000)^{20} \;\approx\; 6.533 \times 10^{137},$$

a **138-digit** integer with exactly **60 trailing zeros** (since $7{,}776{,}000 = 2^8 3^5 5^3 \Rightarrow N = 2^{160} 3^{100} 5^{60} = M \cdot 10^{60}$). In "security bits," $\log_2 N \approx \mathbf{457.8}$ bits, well beyond 256-bit brute force ($\approx 1.16 \times 10^{77}$, 78 digits) and even typical SNARK/STARK soundness targets tuned near 2^{-128} per proof. Crucially,

RWP attains this scale **without** heavy algebra: per round it exposes flat, slice-balanced marginals and a re-drawn, hidden bijection; across rounds, independence multiplies these factors, while verification remains $O(R)$.

This yields an operational non-leakage posture that many practical ZK deployments cannot match in the passive-eavesdropper model. Standard zero-knowledge systems prove statements about **fixed** witnesses; their transcripts are sound and zero-knowledge under assumptions, but the public structure persists across sessions. By contrast, RWP's transient, **session-rekeyed** geometry offers no stable co-occurrences for correlation or frequency analysis; transcripts are single-use and statistically uninformative, so the adversary is reduced to enumerating N possibilities, $\sim 10^{138}$, rather than leveraging reusable algebraic scaffolding. In short: **exponential attacker work** ($\approx$458 bits) versus **linear verification**, with **no stored secrets** and no exploitable statistical residue between sessions.

ENI6MA Use Case and Solutions (end-to-end)

- **Enterprise login without stored secrets.** Replace passwords, TOTP, and passkeys with session-fresh RWP. Nothing reusable sits on disk; nothing useful leaks in transit.
- **PAM/JIT control plane.** Put RWP as a **pre-authorization fence** before sudo, break-glass, KMS decrypts, cloud-role grants. On PASS, grant a **short-lived role** and auto-revoke; on FAIL, no credential exists to steal.
- **API calls and service-to-service access.** Agents prove presence now, governed by policy; no long-lived client secrets in CI/CD or IaC templates.
- **IoT/edge and air-gapped ops.** Offline-deterministic rounds with bounded τ make RWP practical where clocks are coarse and links are intermittent.
- **Incident response and duress.** Because proofs are time-scoped and transcript-auditable, responders can narrow blast radius to seconds-class windows. Optional duress mappings can accept while silently flagging.

Why it's "human-accessible" (and still post-quantum complete)

Humans are good at **perceiving membership** ("my character is in *that* zone"), not at remembering entropy safely. RWP embraces that: the UI presents six zones with concise witness strings; the user's cognitive task is to **recognize** rather than **recall**. This reduces error rates and eliminates unsafe secret handling. Under the hood, verification is **symmetric-only**, string-and-mod-arithmetic, **friendly to a post-quantum future** because no RSA/ECC hardness assumption is required. The security wall is combinatorial + time-variance, not factorization or discrete logs.

Performance and deployability

- **Throughput/latency.** Per round you do a few modulo operations, rotations on strings of length $\leq$30, and a membership test in a 12-char witness, microseconds on commodity CPUs. Proofs finish in a handful of rounds; verification stays **O(L)**.
- **Auditability.** Optional index triples and timestamps yield deterministic replay and **provable compliance artifacts** without exposing the private morphism.
- **Interoperability.** The same primitive works for consumers, workforce, service identities, and robots, with the **same verifier**. Online or offline, cloud or on-prem.

A compact mental model

Think of ENI6MA as a **hologram**: a rich, time-varying internal state collapses each round into a tiny, verifiable projection (the witness). The verifier checks only **membership**; the manifold's hidden offsets, orders, and permutations stay private. Because the manifold changes every microsecond and each round multiplies the hypothesis count, **learning today's state does not predict the next**. That is why we say **"cosmic-scale security"**: with ordinary parameters, the effective brute-force landscape eclipses classical 128-bit spaces, yet the honest prover and verifier do constant-time work per character.

Bottom line: ENI6MA RWP gives companies a **key-less, replay-proof, and human-accessible** proof-of-knowledge that **removes stored secrets**, **collapses attack surface**, and **hardens privileged operations**, all with **linear-time verification** and transcripts fit for compliance. It's a practical bridge from fragile secret possession to resilient **proof-of-presence** in the milliseconds that matter.

Rosario-Wang Proof Use Case : Digital Identity Via iPoK

ENI6MA's Rosario–Wang Proof (RWP) solves a pervasive problem: proving *knowledge or presence now* without keeping long-lived secrets that can be phished, replayed, or exfiltrated. The design is pragmatic and operations-ready: (a) keep cryptographic state **minimal and ephemeral**, (b) make verification **linear-time and auditable**, and (c) run the **same binary online or disconnected**. Each session yields a compact **transcript** keyed to microsecond time τ, an embedded 512-bit prime π, and uniformly random 512-bit entropy ε. High-dimensional randomness is reduced to **bounded cyclic rotations** over fixed **alphabet rings**, then **foliated** into six balanced zones; a prover selects the zone consistent with a private string S. The verifier accepts if, and only if, each secret character is contained in its round's **witness set**. Below we formalize the mechanics and give tight, human-readable (TTS) glosses for every equation.

How Eni6ma (RWP) Addresses the Gap in Cryptography

RWP **removes stored secrets**, collapsing the primary breach vector. It is **phish- and replay-resistant** because witnesses expire with τ. Verification is **O(L)** and fits high-throughput PAM fences, CI/CD, and IoT edges. The **same primitive** serves humans (zone recognition) and agents (API flow), online or air-gapped. Finally, transcripts are **minimal yet deterministic**, enabling compliance evidence without exposing private mappings. In short: **human-accessible proof-of-knowledge** with **cosmic-scale effective search** for adversaries, and **constant, auditable work** for defenders.

To move from outcomes that matter to operators we can implement, we make the business guarantees, no stored secrets, replay resistance, and auditable proof, explicit as parameters. Each session is parameterized by three levers: time τ (binding proofs to microseconds), an entropy element ε (freshness and diversification), and an embedded 512-bit prime π (namespacing and key derivation). These drive cyclic "rings" $\alpha_0, \alpha_1, \alpha_2$ whose rotations by offsets $\Theta_{r,j}$ produce the human-legible witnesses that the verifier records.

Operational requirements map directly to these symbols. Scalability becomes a verification rule that is linear in the secret length L: we evaluate membership $s_i \in W_i$ for each round and accumulate the verdict $\Lambda = \bigwedge_i [\, s_i \in W_i \,]$. Auditability becomes determinism: given the transcript and the routing matrix $\mathcal{M}$, any party can reconstruct witness sets and re-check Λ without seeing the private morphism. Replay resistance becomes a probability bound: under balanced foliation into Z zones and independence across three rings, the false-accept rate tightens to $p_{\mathrm{FA}} \approx Z^{-3R}$ over R rounds. Tuning Z and R thus trades user effort for measurable assurance.

Implementation choices then follow. A keyed XOF/PRF acting as a two-way keystream turns (π, H) into random-access blocks for encrypted pools and transcripts; constant-time rotations and fixed-length scans enforce uniform cost; domain tags ensure that contexts remain disjoint. Because the verifier handles small arrays, the same code path serves human clients, automated agents, and offline verifiers. In short, the symbols below are not abstract decoration; they are the minimal hooks that realize ENI6MA's risk guarantees as testable and portable computations, small enough to fit latency budgets, and precise enough for compliance.

Key derivation (time-scoped, stateless)

We compress π and τ into a per-session master key, then derive per-ring prongs:

$$K_{\mathrm{master}} = \mathrm{KDF}\big(\mathrm{enc}_{512}(\pi) \parallel \mathrm{le}_{64}(\tau) \parallel \text{"eni6ma.v1"}\big).$$

"K master equals K-D-F of pi encoded to 512 bits, concatenated with tau in little-endian 64 bits, concatenated with the label eni-six-ma dot v-one."

$$K_j = \mathrm{PRF}\big(K_{\mathrm{master}} \parallel \text{"col"}\ j\big), \quad j \in \{0, 1, 2\}.$$

"K sub jay equals P-R-F of K master concatenated with the string 'col' and the index j, for j equals zero, one, or two."

Entropy reduction and rotation offsets

Let ε expose eight 64-bit limbs $(w_0, \dots, w_7)$. We reduce to a 64-bit mixer via Horner's rule with base $B = 2^{16} + 1$:

$$H(K_j, \varepsilon) = \Big(\big(\cdots ((w_0 \oplus K_j)B + w_1)B + \cdots \big)B + w_7 \Big) \bmod 2^{64}.$$

"H of K sub j and epsilon equals a Horner sum: start with w-zero xor K-j, multiply by B, add w-one, repeat through w-seven, then take modulo two to the sixty-four."

Per-round, per-ring rotation offsets are:

$$\Theta_{r,j} = \big(H(K_j, \varepsilon_r) + r \big) \bmod |\alpha_j|.$$

"Theta sub r comma j equals H of K-j and the r-th entropy block plus r, modulo the size of ring alpha sub j."

Rings, foliation, and witnesses

We maintain three cyclic rings $\alpha_0, \alpha_1, \alpha_2$ with sizes $30, 30, 12$ and a foliation factor $Z = 6$. After rotation by $\Theta_{r,j}$, we choose a zone index:

$$z_r = \mathrm{PRF}(K_2, \varepsilon_r \oplus \mathrm{le}_{64}(\tau)) \bmod Z.$$

"z sub r equals P-R-F of K-two and epsilon-r xor tau, modulo Z."

Apply the rotation and select the zone slice (balanced, near-equal):

$$\alpha_j^{(r)} = \mathrm{rot}\big(\alpha_j, \Theta_{r,j}\big), \qquad S_{r,j} = \mathrm{slice}\big(\alpha_j^{(r)}, z_r, \lceil |\alpha_j| / Z \rceil\big).$$

"Alpha sub j to the r equals ring alpha-j rotated by Theta r j. S sub r j equals the slice of that rotated ring at zone index z-r with length equal to ceiling of the ring size divided by Z."

Concatenate per-ring slices to form the **witness** for round r:

$$W_r = S_{r,0} \parallel S_{r,1} \parallel S_{r,2}.$$

"W sub r equals S r zero concatenated with S r one concatenated with S r two."

Acceptance rule (membership, not decryption)

For a secret string $S = s_1 \cdots s_L$, the verifier checks **membership**:

$$\mathsf{ACCEPT} \iff \bigwedge_{i=1}^{L} \mathbf{1}\{\, s_i \in W_i \,\} = 1.$$

"Accept if and only if, for every i from one to L, the indicator that s sub i is in W sub i equals one."

This check is **linear time** in L: it is string rotation plus membership in small sets. No long-term secrets are revealed or stored; the transcript records only $(i, W_i, z_i, \Theta_{i,\cdot})$ as needed for audit.

False-accept bound and security growth

Under uniform slicing and independence across rings, the per-round false-accept probability is the product of slice densities:

$$p_{\mathrm{FA,round}} \approx \prod_{j=0}^{2} \frac{|S_{r,j}|}{|\alpha_j|} \approx \left(\frac{1}{Z}\right)^3 = \frac{1}{216}.$$

"p sub F-A per round is approximately the product over j of slice size divided by ring size, which is about one over Z cubed; with Z equals six, that is one over two-sixteen."

Over R independent rounds,

$$p_{\mathrm{FA,total}} \approx \left(\frac{1}{Z}\right)^{3R}.$$

"Total false-accept probability is approximately one over Z to the power three-R."

Measuring strength in bits:

$$\mathrm{bits} \;=\; -\log_2 p_{\mathrm{FA,total}} \;=\; 3R \log_2 Z.$$

"Bits equals minus log base two of the total false-accept probability, which equals three times R times log base two of Z."

With $Z = 6$ and $R = 10$, this yields about $3 \times 10 \times \log_2(6) \approx 77.6$ bits of **soundness**, before accounting for microsecond time windows, private morphisms between colors (adding a 6! ambiguity), and route-ordering, all of which further multiply the attacker's search.

E–T–Π control matrix (routing the work)

We summarize per-round control as an —Π triple (q_r, r_r, s_r) that selects entropy, time, and prime-mixed columns for deterministic replay:

$$\mathcal{M}_r = (q_r, r_r, s_r) = \big(\mathrm{idx}(\varepsilon_r),\ \mathrm{le}_{64}(\tau) \bmod R,\ H(K_0, \varepsilon_r) \bmod 3\big).$$

"Script M sub r equals the triple of q-r, r-r, s-r, where q-r is the index of the r-th entropy block, r-r is tau modulo R, and s-r is H of K-zero and epsilon-r modulo three."

This matrix is the **route table** the verifier can reconstruct from the transcript to show exactly how each W_r was produced, useful for audits, duress analytics, and incident forensics.

2. Notation

- $\pi \in \{0,1\}^{512}$: embedded 512-bit prime seed
- $\varepsilon \in \{0,1\}^{512}$: 512-bit entropy element; limbs $(w_0, \ldots, w_7) \in \mathrm{u64}^8$
- τ: session time (microseconds), with derived indices as specified
- $\alpha_0, \alpha_1, \alpha_2$: uppercase, lowercase, numeric rings; sizes $30, 30, 12$
- $Z = 6$: number of zones (foliation factor)
- KDF, PRF: key derivation and pseudorandom function/XOF
- $\Theta_{r,j}$: rotation offset for round r and ring j
- $\mathcal{M}$: —Π matrix with per-round triples (q, r, s)

3. Two-Way Hash Keystream and Stream Framing

This pattern seals artifacts and session data with a keyed XOF/PRF keystream derived from an embedded prime and a public header. Because encryption and decryption are the same XOR pass, binaries remain minimal, constant-time, and auditable. Random access to blocks allows partial reads, resumable transfers, and efficient archival verification. Relative to PKI, there are no long-lived private keys, certificate chains, or rotation ceremonies to manage. Relative to ZKPs/SNARKs, there is no arithmetization, trusted setup, or bulky proof object, just linear passes over bytes. And compared with Proof-of-Work/Stake, there is no consensus burn: framing is local, deterministic, and energy-light. The result is a transport and storage substrate that preserves confidentiality (IND-CPA with unique headers), supports AEAD/MAC for integrity, and provides provenance via structured headers. In the full system, framed artifacts carry entropy pools and transcripts; because headers encode domain tags and nonces, streams are domain-separated and replay-detectable, making this layer the reliable "I/O sheath" around ENI6MA's cryptographic core.

3.1 Equations

Key derivation from embedded prime and public header:

$$K = \mathrm{KDF}(\pi, H), \quad H = (\texttt{file\_id} \,\|\, \texttt{chunk\_size} \,\|\, \texttt{session\_nonce} \,\|\, T).$$

"K equals K-D-F of pi and H; H is file-id concatenated with chunk-size, session-nonce, and timestamp T."

Keystream block and encryption per counter i:

$$\mathcal{K}_i = \mathrm{PRF}_K(H, i), \qquad C_i = P_i \oplus \mathcal{K}_i.$$

"Script K sub i equals P-R-F keyed by K of H and counter i. Cipher chunk C sub i equals plaintext P sub i XOR keystream K sub i."

Decryption symmetry:

$$P_i = C_i \oplus \mathcal{K}_i.$$

"Plaintext P sub i equals ciphertext C sub i XOR keystream K sub i."

3.2 Steps

1. Serialize header H and write to output.
2. Compute $K = \mathrm{KDF}(\pi, H)$.
3. For each chunk index $i = 0, 1, 2, \ldots$, derive $\mathcal{K}_i$, write $C_i = P_i \oplus \mathcal{K}_i$.
4. Decrypt by the same steps with C in place of P.

3.3 Discussion

Treat a keyed XOF (e.g., BLAKE3 in keyed mode) as a PRF to obtain random-access keystream blocks indexed by i under a per-file namespace H. If the PRF is secure and keystream positions are never reused under the same key, the construction achieves IND-CPA. Practically, using a *unique* `file_id` and *fresh* `session_nonce` guarantees stream independence across objects and runs; T makes replayed headers obvious and aids provenance.

The framing is length-preserving and chunk-oriented, fitting memory-mapped files and streaming pipelines. Because XOR is its own inverse, a single routine suffices for encryption and decryption, valuable in minimal binaries. Integrity is orthogonal: add a MAC over H and all ciphertext chunks (or use AEAD wrapping) to detect header or payload tampering. In multiparty or archival settings, the header doubles as metadata for audit (who produced the stream, when, and with what chunk size).

Performance stems from: (i) sequential PRF expansion; (ii) fixed-size, cache-friendly chunks; (iii) zero padding overhead. The result is competitive with

stream ciphers while preserving deterministic re-derivation from (π, H). Recommended practices: (1) **Never** reuse H with the same π; (2) Include explicit domain tags inside H (e.g., "eni6ma.pool.v1") for clean separation; (3) MAC the entire frame if authenticity matters; (4) Choose chunk sizes to balance throughput and latency (e.g., 1–64 KiB).

Use-cases: encrypting entropy pools at rest; sealing configuration/state blobs; transporting short session transcripts between cooperating processes. Because the keystream is random access, partial reads or out-of-order fetches (e.g., HTTP range) remain efficient, all that's needed is H and the counter i. The approach generalizes to structured objects by hashing domain-specific header fields into H. In sum, a lightweight, audit-friendly, invertible framing that underpins ENI6MA's sealed artifacts.

4. Entropy Pool Generation and Encryption

Here ENI6MA manufactures high-grade, uniform 512-bit values once, then ships them only as sealed ciphertext. This decouples *generation* (with logs, audits, and hardware randomness) from *use* (lightweight, offline devices), avoiding endpoint RNG fragility. Against PKI, the benefit is stark: pools are not secrets of enduring value; compromise of a pool file yields nothing without the embedded prime and header. Against ZKPs/SNARKs, witness material is trivial to prepare, no constraint systems, no ceremony, so the cost of "freshness at scale" is near zero. And unlike PoW/PoS, security does not rely on externalized energy or stake; unpredictability is intrinsic to the sampled space. Practically, the pool supports fleet provisioning, DDIL deployments, and deterministic replays for compliance. In the whole design, the pool is the raw entropy reservoir that —Π will route into offsets, ensuring each session's manifold is both unbiased and independently sampled, while keeping the operational footprint, bandwidth, CPU, and storage, small.

4.1 Equations

Uniform sampling over the 512-bit upper half:

$$\varepsilon = (w_0, \ldots, w_7), \quad w_k \leftarrow \mathrm{RNG}(), \;\; w_7 \leftarrow w_7 \vee 2^{63}.$$

"Epsilon equals eight 64-bit limbs w zero through w seven; each limb is sampled from a cryptographic RNG; set the top bit of w seven."

Birthday bound on collisions for n draws into 2^{512} space:

$$\Pr[\text{collision}] \leq \frac{n(n-1)}{2^{513}}.$$

"Probability of any collision is at most n times n minus one divided by two to the five-hundred-thirteen."

Encryption of serialized pool P under header H:

$$C = \mathrm{Enc}_K(H, P) \;\text{ with }\; K = \mathrm{KDF}(\pi, H).$$

"Ciphertext C equals Enc with key K of header H and plaintext P, where K equals K-D-F of pi and H."

4.2 Steps

1. For each entry, sample eight u64s; set MSB of the most significant limb.
2. Serialize little-endian to 64 bytes per entry; concatenate to form P.
3. Choose fresh `file_id`, `session_nonce`; build H.
4. Encrypt P as in §3; optionally compute MAC over $H \| C$.
5. Persist header+payload; record pool size N.

4.3 Discussion

The pool is a bank of uniform 512-bit values sampled from $[2^{511}, 2^{512}-1]$. For practical pool sizes (e.g., thousands to millions), birthday collisions are negligible. Enforcing the MSB avoids accidental truncation or leading-zero ambiguities in downstream decimal processing. Serialization fixes byte order, enabling cross-platform reproducibility.

The encrypted pool separates *manufacture* from *use*: generation can be centralized (with audit logs) or CI-driven; consumption occurs on endpoints that only need the embedded π to decrypt locally. Because the payload is length-preserving and blockless, devices can stream-decrypt in constant memory. A MAC (or AEAD) gives tamper evidence and guards against file substitution.

Operationally: version pool format; include a domain tag and creation timestamp in H; log `file_id` and N for inventory. Rotation policies should regenerate pools periodically; old pools can be retained encrypted for forensic reproducibility without exposing raw values. When many devices share identical ciphertext pools, compromise of one runtime reveals only its working set, not the global stock, assuming no plaintext pools are stored.

Use-cases: (i) *Provisioning*: ship a single encrypted pool image to fleets; (ii) *A/B testing*: compare offset distributions across pools; (iii) *Air-gapped sites*: decrypt on-host without calling out to key servers. The pool underpins —Π extraction (next sections) and can be repurposed for other randomness-hungry subsystems if application domains are strictly separated in H.

5. Encrypted Entropy Pool Runtime

The runtime loader reverses framing, validates invariants (uniqueness, top-bit set, count), and exposes entries in multiple numeric views (limbs, big-ints, fixed-width). Its contribution is *deterministic, constant-space* availability of

high-quality entropy at the edge, no online key servers, no shared vaults. Compared to PKI, there is nothing equivalent to a master key or CA hierarchy to exfiltrate; compared to ZKPs/SNARKs, there is no per-session proving cost explosion; compared to PoW/PoS, there is no global coordination or latency penalty. Security gains flow from narrow interfaces: the loader never reveals private morphisms, and logs record only reproducible metadata (header digests, indices). In the cohesive system, this runtime guarantees that the later —Π and projection stages begin from validated randomness, preventing statistical skew that could otherwise leak structure. Operationally, it simplifies fleet hygiene, immutable pool images, reproducible validation, and telemetry ready for audit, while keeping the trust surface symmetric-only and inspectable.

5.1 Equations

Deterministic load with embedded π:

$$\mathrm{Load}(F;\pi) \to \mathcal{E} = (\varepsilon_0, \ldots, \varepsilon_{N-1}), \quad \text{via } P = \mathrm{Dec}_K(H, C),\ K = \mathrm{KDF}(\pi, H).$$

"Load of file F with pi returns the sequence of epsilons by decrypting ciphertext C under key K; K equals K-D-F of pi and header H."

Validation predicate:

$$\mathrm{Valid}(\mathcal{E}) \equiv \Big(\forall i : \varepsilon_i \in \mathcal{D}\Big) \wedge \Big(\big|\{\varepsilon_i\}\big| = N\Big) \wedge (N = \hat{N}),$$

where $\mathcal{D} = \{x \in \{0,1\}^{512} : x_{511} = 1\}$.
"Valid of script E holds if every epsilon is in the 512-bit set with top bit one, all entries are unique, and the count equals expected N-hat."

5.2 Steps

1. Parse header H, derive K, stream-decrypt to P.
2. Rehydrate entries into limbs / BigUint / fixed-width $U512$.
3. Run validation; expose pool metadata and accessors.

5.3 Discussion

A single loader routine reverses the framing and returns validated entries in multiple numeric views. Fixed-width limbs ease constant-time arithmetic; big integers integrate with higher-level libraries; a compact $U512$ type feeds the —Π Horner path. The loader can be purely streaming: read a chunk, XOR, parse whenever a full 64-byte entry comes into view.

Validation defends invariants assumed downstream. Enforcing MSB and uniqueness avoids skew in decimal block extraction. Counting N catches truncated or padded files; MAC verification (if used) prevents silent corruption and

swap attacks. If pools are mirrored across environments, the same binary will accept or reject deterministically given identical F and π.

Practicalities: check file permissions before decryption; consider memory-mapping large pools for fast random access; cache parsed entries if the application repeatedly samples the same ε_i. Emit a status record (header digest, N, validation flags) useful in telemetry and incident response. Because the pool is sealed at rest, compromise of at-rest images leaks nothing without the running binary.

6. Session Entropy and Offset Computation

This pattern binds microsecond time τ to a selected pool element to produce per-ring rotation offsets $\Theta_{r,j}$. The effect is *time-coupled non-replayability*: even perfect captures expire as τ advances. Against PKI, this eliminates nonce bookkeeping, token lifetimes, and key rotation complexities, freshness is inherent. Against ZKPs/SNARKs, there is no circuit regeneration or prover overhead; offsets are computed with simple modular arithmetic in $O(R)$. Against PoW/PoS, liveness and finality are local: a verifier decides instantly without waiting for blocks or validators. Relevance to the whole: offsets are the clocked gears that re-index the alphabet rings, ensuring every round's witness is both unpredictable and reconstructible from transcript metadata. This gives defenders tunable assurance via parameters (window size, round count) while preserving human usability. By confining state to (τ, i), the verifier remains memory-light and deterministic, a prerequisite for high-QPS services and air-gapped verifiers alike.

6.1 Equations

Session time and index selection:

$$\tau = (\text{now}_\mu) \bmod 1000, \qquad i = \tau \bmod N.$$

"Tau equals current microsecond count modulo one-thousand; index i equals tau modulo N."

Decimal chunking of E into 3-digit blocks b_k:

$$b_k = 100\, d_{3k} + 10\, d_{3k+1} + d_{3k+2} \in \{0, \ldots, 999\}.$$

"b sub k equals one-hundred times digit three-k plus ten times digit three-k plus one, plus digit three-k plus two."

Rosario modulo index and rotation:

$$\kappa_{r,j} = \big((\tau \bmod U) + r \cdot M + j\big) \bmod U, \;\; U = 25, \; M = 3,$$

$$\Theta_{r,j} = b_{\kappa_{r,j}+1} \bmod |\alpha_j|.$$

"Kappa r comma j equals tau mod U plus r times M plus j, all mod U; Theta r comma j equals block at kappa plus one modulo the ring size."

6.2 Steps

1. Fix session τ (microseconds mod 1000) and select entropy index $i = \tau \bmod N$.
2. Convert $E = \varepsilon_i$ to decimal digits; form 3-digit blocks b_k.
3. For round r and ring $j \in \{0, 1, 2\}$, compute $\kappa_{r,j}$ and $\Theta_{r,j}$.

6.3 Discussion

This layer binds time and randomness into stable, per-round offsets. Keeping τ constant during the session avoids jitter in the visible slices; advancing τ between sessions drives diversity. Decimal block extraction offers high-variance indices without big-integer divisions in hot paths; it is easy to audit and port. The modulo window $U = 25$ and the 3-ring stride $M = 3$ spread attention across blocks while reserving block zero for protocol use.

Offsets $\Theta_{r,j}$ are cyclic shifts in $\mathbb{Z}_{|\alpha_j|}$. Under uniform E, blocks b_k approximate uniform over $[0, 999]$, yielding near-uniform rotations modulo 30 or 12. The design is resilient to display or transport: given (τ, i) and the pool entry, any verifier can recompute offsets and reconstruct the same per-round vistas.

Operationally, log (τ, i) and the computed $\Theta_{r,j}$ for reproducibility; never log secret characters. If the UI cycles rounds on a timer, r increments while τ is fixed; if re-authenticating later, a new τ and (likely) a different i will scramble rotations. Parameter tuning: larger U increases the spread in κ, smaller U increases locality. For accessibility, ring sizes can change (e.g., add symbols), requiring only updates to modulo reductions.

7. —Π Matrix Generation

The —Π control matrix deterministically routes (ε, τ, π) into low-range indices using Horner-style reductions and lightweight mixing. Conceptually, it is a reproducible "switchboard" that selects which entropy columns feed which rings in which rounds. Advantages over PKI are immediate: no certificate path discovery or revocation checks, just arithmetic. Over ZKPs/SNARKs, the matrix avoids trusted setup and polynomial commitments; its outputs are tiny, and recomputation is $O(R)$. Versus PoW/PoS, the matrix provides auditability without global consensus, transcripts carry enough to replay the routing exactly. System relevance: —Π is the bridge between sealed entropy and visible witnesses, guaranteeing that any verifier can reconstruct the same vistas from a transcript without learning the private morphism. Because it is SIMD/GPU-friendly, it scales to large deployments; because it is label- and domain-separated, it avoids

cross-protocol interference. In short, —Π supplies deterministic control without statefulness, reinforcing ENI6MA's stateless security model.

7.1 Equations

Enhanced time mixing and dynamic prime:

$$\tau' = \mathsf{Enh}(\tau, \pi), \qquad \pi' = \mathsf{Dyn}(\pi, \tau').$$

"Tau-prime equals Enhance of tau and pi; pi-prime equals Dynamic mix of pi with tau-prime."

Horner-style reduction over $B = 2^{64}$ with position mix:

$$g(x) = \Big(\cdots ((x_7 B + x_6)B + x_5) \cdots + x_0 \Big) \bmod \tau' \ \oplus \ f(\pi', \mathrm{pos}).$$

"g of x equals Horner accumulation of eight 64-bit limbs base two to the sixty-four, reduced modulo tau-prime, XOR a position-dependent mix of pi-prime."

Per-round projections to target alphabets:

$$q = g(e_i) \bmod 30, \quad r = g(e_i) \bmod 30, \quad s = g(e_i) \bmod 12.$$

"q equals g of e sub i mod thirty; r equals g of e sub i mod thirty; s equals g of e sub i mod twelve."

Matrix assembly:

$$\mathcal{M}(e_{1..n}; \pi, \tau) = \big((q_1, r_1, s_1), \ldots, (q_n, r_n, s_n)\big).$$

"Calligraphic M of the entropy sequence equals the list of triples q r s for rounds one through n."

7.2 Steps

1. Compute τ' and π' per session.
2. For each selected entropy value e_i, evaluate $g(e_i)$.
3. Reduce to (q_i, r_i, s_i) and record $\mathcal{M}$ alongside (τ', π') metadata.

7.3 Discussion

The —Π stage extracts low-range offsets from 512-bit inputs using only additions, shifts, and modular reduction, GPU- and SIMD-friendly. Horner accumulation over $B = 2^{64}$ is cache-efficient and branch-free; a small position-dependent perturbation derived from π' decorrelates patterns across limbs. The range $\{30, 30, 12\}$ matches the three ring sizes; alternate rings (symbols, punctuation) are plug-compatible by swapping modulus values.

Session-specific τ' and π' inject non-repeatability under the same τ or π alone. Recording provenance (τ', π', pool index, count n) enables remote checkers or auditors to reconstruct $\mathcal{M}$ without sharing secrets. Statistical sanity checks (e.g., chi-square over q, r, s) can be run over large logs to confirm near-uniformity and absence of bias from implementation mistakes.

In practice: precompute $\mathcal{M}$ for the planned number of rounds; export as a compact structure (e.g., JSON with base-10 integers). Because each row depends solely on e_i (plus session parameters), independent threads or cores can compute rows in parallel. If higher statistical independence is desired, substitute the perturbation f with a keyed PRF call; throughput costs increase modestly.

8. Interactive Proof-of-Knowledge via Manifold Projection

ENI6MA collapses the hidden states into *witness sets*, balanced slices of rotated rings, so the prover's action is a membership choice, and the verifier's job is the Boolean accumulator $\Lambda = \bigwedge_i [s_i \in W_i]$. This yields *linear-time verification* with *passive zero-knowledge*: observers learn "accept/deny," not the secret or morphism. Compared to PKI, no private key ever exists at rest; compared to ZKPs/SNARKs, there are no large proof objects, no verifier cryptography beyond hashing and membership checks; compared to PoW/PoS, there is no throughput/finality tax, decisions are instantaneous and local. In the cohesive system, manifold projection operationalizes identity as *perception-bound work* rather than secret possession, unifying human and agent ceremonies under the same verifier. Soundness scales exponentially with rounds while compute and memory remain flat, enabling seconds-class ceremonies for login, PAM/JIT elevation, CI/CD gates, and DDIL contexts. The projection is thus the capstone: simple to implement, hard to counterfeit, and trivially auditable.

8.1 Equations

Foliation of each ring α_j into $Z = 6$ zones after rotation by $\Theta_{r,j}$: write $|\alpha_j| = Zq_j + \rho_j$ with remainder ρ_j. Then:

$$\text{slice}_{r,j,z} = \alpha_j[s : e),\ s = z\,q_j + \min(z, \rho_j),\ e = s + q_j + [z < \rho_j].$$

"Slice equals alpha sub j from start s to end e; s equals z times q sub j plus the smaller of z and rho sub j; e equals s plus q sub j plus one if z is less than rho sub j."

Witness for zone z at round r:

$$W_{r,z} = \text{slice}_{r,0,z} \parallel \text{slice}_{r,1,z} \parallel \text{slice}_{r,2,z}.$$

"W sub r comma z equals the concatenation of the three ring slices for that zone."

Acceptance for secret string $S = (S_1, \ldots, S_n)$:

$$\mathsf{ACCEPT} \iff \forall i \in \{1, \ldots, n\} : \; S_i \in W_{i,z_i}.$$

"Accept if and only if for every round i, the secret character S sub i is a member of the witness W sub i at the chosen zone z sub i."

Boolean accumulator form:

$$\Lambda = \bigwedge_{i=1}^{n} [\, S_i \in W_{i,z_i} \,].$$

"Lambda equals the logical AND over i from one to n of the statement S sub i is in W sub i z sub i."

8.2 Steps

1. For each round r, rotate each ring by $\Theta_{r,j}$ and compute six slices per ring.
2. Build $W_{r,z}$ for $z \in \{0, \ldots, 5\}$; render the six zone options.
3. Prover selects zone z_r consistent with secret S_r; record selection.
4. After n rounds, compute Λ; accept iff $\Lambda = 1$.

8.3 Discussion

The verifier never learns S; it verifies *containment* decisions only. Each round's view is a projection (rotation + foliation) of a product of three rings into six balanced slices, yielding a simple yes/no predicate. If a cheating prover picks zones uniformly at random and slices are equalized, the per-round success probability is $\approx 1/Z$ and the chance of n consecutive successes is Z^{-n} (e.g., 6^{-n}). Raising n thus exponentially suppresses false accepts while keeping each interaction cognitively light.

UI/UX can present slices as columns or panels; accessibility options (font choice, contrast, speech prompts) are compatible since verification depends only on set membership, not on secret transport. The ceremony is *stateless*: transcripts need not store secrets, only zone selections and session parameters necessary to reconstruct views. Optional rate limiting and lockouts handle online adversaries; offline attackers cannot regenerate the same witnesses without the session parameters and pools.

Security derives from (a) independent, session-specific rotations; (b) unbiased ring partitions; (c) an acceptance predicate that is auditable and minimal. Because witnesses are ephemeral views, replaying a previous round under a different τ or $\mathcal{M}$ fails: constructed $W_{r,z}$ won't match prior layouts. For defense-in-depth, server-side verifiers should MAC transcripts and headers before storage or transport; device-only verifiers can keep verification entirely local.

9. Security and Complexity

Stream confidentiality. Under the PRF assumption, the keystream is computationally indistinguishable from random and XOR encryption achieves IND-CPA if H is unique per stream. AEAD or MAC provides integrity and replay detection for headers and payloads.

Entropy quality. With 512-bit entries and enforced MSB, collision is negligible for realistic pool sizes (cf. §4). Pool validation preserves uniformity assumptions used by offset extraction.

Session unpredictability. The functions Enh and Dyn bind τ and π into session-specific parameters; recomputing $\mathcal{M}$ with fresh τ remodels witnesses, frustrating transcript replay.

Verifier runtime. All operations are linear in the number of rounds and ring elements shown per round. Let n be rounds; verification cost is $O(n)$ set-membership checks on small arrays. Memory is $O(1)$ beyond headers and a few ring buffers.

Soundness accumulator. Writing the acceptance as :

$$\Lambda = \bigwedge_{i=1}^{20} [\, S_i \in W_{i,z_i} \,]$$

"Lambda equals AND over $i = 1$ to 20 of S_i being a member of W_{i,z_i}."
and assuming equalized slices with $Z = 6$, a random guesser's acceptance probability is 6^{-20}. With $n = 20$, false accept $\approx 1/3{,}656{,}158{,}440{,}062{,}976$; this gives us 1 in 3.7 quadrillion chance of guess with $n = 20$, $\approx 2.7351 \times 10^{-16}$.

10. Implementation Guidance

- **Headers & domains.** Include explicit version and domain tags in H.
- **MAC/AEAD.** Prefer AEAD for pool files and remote transcripts; at minimum, MAC $H \| C$.
- **Logging.** Log $(\tau, i, U, M, \Theta_{r,j})$ for reproducibility; never log S or zone-to-character mappings.
- **Accessibility.** Permit alternate renderings (voice readouts of slices, high-contrast modes) without changing acceptance logic.
- **Tuning n.** Choose rounds to meet target false-accept; e.g., consumer unlock $n = 6$–8, admin operations $n = 8$–10.
- **Parallelism.** Compute $\mathcal{M}$ rows independently; precompute slices per round.
- **Testing.** Run statistical tests over q, r, s across large sessions to confirm near-uniformity.

11. Example Use-Cases

1. **Human unlock ceremony.** A user enters a 6–8 round session; zone selections are checked locally, and no reusable secret is ever stored.
2. **Device JIT authorization.** Headless agent computes $\mathcal{M}$, returns compact transcript for a policy engine to grant a short-lived capability.
3. **Sealed configuration.** Configuration blobs are framed and encrypted under §3; devices decrypt on demand with embedded π.
4. **Kiosk/AR workflows.** Visual slices are rendered; users speak the zone name. Accessibility is first-class because only membership matters.

12: The Eni6ma Advantage

ENI6MA composes standard, auditable primitives, PRF-class stream encryption, uniform 512-bit sampling, and cyclic rotations, with a minimalist manifold-projection verifier to yield a **stateless proof-of-knowledge substrate**. Identity is enacted as a sequence of **audited set-membership decisions**: an acceptance is the logical conjunction of per-round checks; a failure produces **no reusable residue**. Because witnesses are derived from fresh entropy and time, transcripts function as receipts rather than credentials. The machinery compiles into **small, constant-time binaries** that admit formal review and deploy cleanly across human and agent ceremonies.

Compared with **zero-knowledge proofs (ZKPs)** and other heavy cryptographic proofs, ENI6MA emphasizes **operational simplicity and latency**. It avoids circuit arithmetization, trusted setups, polynomial-commitment backends, and large proof objects; both proving and verification are **O(L)** in the number of rounds with symmetric-crypto–like inner loops. There are **no long-lived private keys** to escrow, rotate, or exfiltrate; replay is structurally invalidated by time/entropy coupling. This makes ENI6MA **post-quantum friendly in practice** (relying chiefly on hash/PRF hardness), suitable for edge devices and high-throughput services without specialized hardware. Whereas many ZK deployments trade speed for privacy, ENI6MA delivers **passive zero-knowledge transcripts by construction**, observers learn only "accepted/denied," not the private map or secret symbol. It further eliminates PKI lifecycle burdens, supports **mutual liveness** through bidirectional challenge–response, and composes naturally with existing federation (OIDC/SAML) by minting context-bound tokens from one-time witnesses.

Additional benefits include: (i) **tunable assurance** via simple parameters (ring size and round count) that link usability to measurable error; (ii) **accessibility-aware UX** (e.g., reduced color cardinality) without abandoning formal bounds; (iii) **selective disclosure** and attribute attestations without exposing PII; (iv) **secretless auditing**, logs store verifiable receipts, not keys; and (v) **defense-in-depth engineering**: per-user diversified builds, constant-time tables, and remote attestation to harden the private morphism. In sum,

ENI6MA replaces heavyweight, key-centric proofs with a **fast, stateless, and replay-proof** evidence layer that is easier to audit, cheaper to run, and safer to operate at scale.

13. Our Contribution

The Rosario–Wang Proof (RWP) contributes a stateless, witness-driven proof-of-knowledge that replaces long-lived secrets with **ephemeral, time-coupled evidence** assembled entirely from symmetric, auditable primitives. Its distinctive advance is to recast verification as **set-membership over a time-keyed projection** rather than decryption or circuit evaluation.

High-dimensional randomness ε, microsecond time τ, and an embedded 512-bit prime π are routed by an —Π control matrix into **rotations of fixed alphabet rings** and a **six-zone foliation**. In each round the prover is shown a compact "witness set" W_i and selects the zone consistent with a private symbol. Verification is the Boolean accumulator

$$\Lambda = \bigwedge_{i=1}^{R} [\, s_i \in W_i \,],$$

so acceptance is the conjunction of audited membership decisions. Under balanced slicing with three rings and $Z = 6$ zones, the **false-accept bound** admits a closed form $p_{\mathrm{FA}} \approx Z^{-3R}$; for $R = 10$ this yields ≈ 77.6 bits of soundness **before** time-binding and private morphisms.

1. **Statelessness with deterministic auditability.** RWP's transcripts are **receipts, not credentials**: they log time, indices, and witness strings sufficient to reconstruct each round's vista, while the private morphism remains non-exportable. Compromised logs do not enable replay.

2. **Linear-time, constant-space verification.** Because Λ is an AND of per-round membership checks, verifier cost is $O(R)$ and memory is $O(1)$, well suited to high-QPS services, PAM pre-authorization fences, CI/CD robots, and DDIL/air-gapped environments.

3. **Symmetric-only, post-quantum-friendly design.** RWP avoids trusted setup, polynomial-commitment backends, and large proof objects. It relies on PRF/XOF expansion, cyclic rotations, and small-set tests; **time/entropy coupling** defeats replay, and **domain separation** yields session independence.

4. **Human–agent unification.** The same verifier serves perceptual ceremonies for humans and API-level ceremonies for agents, producing **secretless, reproducible compliance artifacts**.

RWP offers a practical alternative to heavyweight ZK and PKI-centric approaches: it delivers **exponential security growth with linear verification**

cost, removes long-lived keys (and their lifecycle burdens), and preserves privacy by revealing only accept/deny outcomes. Operationally, the construction compiles into small, constant-time binaries that run online (live clocks) or offline (bounded τ windows), enabling consistent deployment across consumer, enterprise, and defense workloads.

14. Conclusion

ENI6MA reframes digital identity as **contextual work** rather than secret possession. We have presented a complete, auditable substrate in which ephemeral evidence, derived from a two-way hash keystream, encrypted entropy pools, a deterministic loader, a time-scoped offset system, and the —Π control matrix, drives a lightweight interactive proof. The verifier's job is intentionally simple: evaluate set-membership across per-round witnesses and accumulate a Boolean verdict. This yields **linear-time verification** and **constant space**, yet offers **exponentially growing soundness**: with $Z = 6$ and R rounds, the false-accept probability scales as $p\_\text{FA}! \approx ! Z^{-3R}$, giving practical assurance levels at single-digit round counts and substantially higher margins once time-binding and private morphisms are included.

Operationally, ENI6MA targets the failure modes that dominate today's incidents. There are **no long-lived secrets at rest** to exfiltrate or replay; captures are time-scoped transcripts that do not reveal the private mapping. The same binary runs online (live clocks) and offline (bounded windows), enabling deployments from cloud PAM fences and CI/CD robots to DDIL and air-gapped sites. Because the primitive is **symmetric-only** and relies on XOF/PRF hardness rather than discrete-log or factoring assumptions, it is **post-quantum friendly** in practice and amenable to constant-time, cache-tolerant implementations on commodity hardware.

Beyond cryptographic sufficiency, we emphasized **governance clarity and auditability**. The acceptance predicate Λ separates "soundness of proof" from "policy to grant," producing transcripts that are minimal yet deterministic for compliance, forensics, and duress analytics. Parameters such as ring sizes, round counts, and foliation factor Z expose a clean knob between usability and assurance; accessibility-aware renderings (e.g., reduced color cardinality, speech prompts) are first-class because correctness hinges only on membership, not on secret transport.

Limitations and future work are concrete. Side-channel resilience (timing, rendering), entropy quality audits, bias detection in ring partitions, and precise leakage accounting over long-horizon transcripts warrant continued study. Formal models for the private morphism (including 6! ambiguity under constrained UIs), empirical error curves for human provers, and end-to-end proofs of composability with hardware roots and remote attestation are active directions. We also plan standardized test vectors, FIPS-style validation artifacts, and integrations that mint short-lived federated tokens directly from ENI6MA transcripts.

In sum, ENI6MA offers a **stateless, human-accessible proof-of-knowledge** that collapses breach blast radius, resists phishing and replay by construction, and scales from individuals to high-QPS services with transparent assurance accounting. It provides a practical path from key hoarding to **proof-of-presence in the milliseconds that matter**, aligning security economics, usability, and verifiability in one cohesive architecture.

References and Bibliography

- [1] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.
- [2] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, 1976.
- [3] S. Goldwasser and S. Micali, "Probabilistic Encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, 1984.
- [4] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that Yield Nothing but their Validity or All Languages in NP Have Zero-Knowledge Proof Systems," *Journal of the ACM*, vol. 38, no. 3, pp. 691–729, 1991.
- [5] A. Fiat and A. Shamir, "How to Prove Yourself: Practical Solutions to Identification and Signature Problems," in *Proc. CRYPTO '86*, LNCS 263, pp. 186–194, 1987.
- [6] C.-P. Schnorr, "Efficient Identification and Signatures Using a Trapdoor Function," *Journal of Cryptology*, vol. 3, pp. 161–174, 1991.
- [7] T. P. Pedersen, "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing," in *Proc. CRYPTO '91*, LNCS 576, pp. 129–140, 1992.
- [8] R. C. Merkle, "A Digital Signature Based on a Conventional Encryption Function," in *Proc. CRYPTO '87*, LNCS 293, pp. 369–378, 1988.
- [9] NIST, *FIPS 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, Aug. 2015.
- [10] NIST, *SP 800-38A: Recommendation for Block Cipher Modes of Operation*, Dec. 2001.
- [11] NIST, *SP 800-38D: Recommendation for Galois/Counter Mode (GCM) and GMAC*, Nov. 2007.
- [12] IETF, *RFC 8439: ChaCha20 and Poly1305 for IETF Protocols*, Jun. 2018.
- [13] H. Krawczyk and P. Eronen, *RFC 5869: HMAC-based Extract-and-Expand Key Derivation Function (HKDF)*, May 2010.

- [14] NIST, *SP 800-90A Rev.1: Deterministic Random Bit Generators (DRBGs)*, Jun. 2015.
- [15] NIST, *SP 800-108: Recommendation for Key Derivation Using Pseudorandom Functions*, Oct. 2009.
- [16] S. Haber and W. S. Stornetta, "How to Time-Stamp a Digital Document," in *Proc. CRYPTO '90*, LNCS 537, pp. 437–455, 1991.
- [17] D. J. Bernstein, "The Salsa20 Family of Stream Ciphers," in *New Stream Cipher Designs*, LNCS 4986, pp. 84–97, 2008.
- [18] M. Bellare and C. Namprempre, "Authenticated Encryption: Relations Among Notions and Analysis of the Generic Composition Paradigm," *Journal of Cryptology*, vol. 21, pp. 469–491, 2008 (extended journal version of 2000 work).
- [19] J. O'Connor *et al.*, "BLAKE3: One Function, Fast Everywhere," 2020. (Project report/whitepaper).
- [20] P. C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," in *Proc. CRYPTO '96*, LNCS 1109, pp. 104–113, 1996.
- [21] J. Benaloh and M. de Mare, "One-Way Accumulators: A Decentralized Alternative to Digital Signatures," in *Proc. EUROCRYPT '93*, LNCS 765, pp. 274–285, 1994.
- [22] J. Camenisch and A. Lysyanskaya, "Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials," in *Proc. CRYPTO 2002*, LNCS 2442, pp. 61–76, 2002.
- [23] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short Proofs for Confidential Transactions and More," in *Proc. IEEE Symposium on Security and Privacy (S&P)*, pp. 315–334, 2018.
- [24] H. Krawczyk, "SIGMA: The SIGn-and-MAC Approach to Authenticated Diffie-Hellman," in *Proc. CRYPTO 2003*, LNCS 2729, pp. 400–425, 2003.
- [25] L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.
- [26] NIST, *FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM)*, 2024.
- [27] NIST, *FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA)*, 2024.

- [28] NIST, *FIPS 205: Stateless Hash-Based Digital Signature Standard (SLH-DSA)*, 2024.
- [29] I. Damgård, "On Σ-Protocols," Lecture Notes / Tutorial, 2012.
- [30] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, 1985.

Appendix A , Equations

1. $K = \mathrm{KDF}(\pi, H)$.
   "K equals K-D-F of pi and H."
2. $\mathcal{K}_i = \mathrm{PRF}_K(H, i)$.
   "Keystream K sub i equals P-R-F keyed by K of H and i."
3. $C_i = P_i \oplus \mathcal{K}_i,\ P_i = C_i \oplus \mathcal{K}_i$.
   "Cipher equals plaintext XOR keystream; plaintext equals ciphertext XOR keystream."
4. $\Pr[\text{collision}] \le \frac{n(n-1)}{2^{513}}$.
   "Probability of collision is at most n times n minus one over two to the five-hundred-thirteen."
5. $\tau = (\mathrm{now}_\mu) \bmod 1000,\ i = \tau \bmod N$.
   "Tau equals microsecond clock mod one-thousand; i equals tau mod N."
6. $\kappa_{r,j} = \big((\tau \bmod U) + r \cdot M + j\big) \bmod U$.
   "Kappa r j equals tau mod U plus r times M plus j, all modulo U."
7. $\Theta_{r,j} = b_{\kappa_{r,j}+1} \bmod |\alpha_j|$.
   "Theta r j equals block at kappa plus one modulo ring size."
8. $\tau' = \mathsf{Enh}(\tau, \pi),\ \pi' = \mathsf{Dyn}(\pi, \tau')$.
   "Tau-prime is Enhance of tau and pi; pi-prime is Dynamic of pi with tau-prime."
9. $g(x) = \mathrm{Horner}_{B=2^{64}}(x) \bmod \tau' \ \oplus\ f(\pi', \mathrm{pos})$.
   "g of x equals Horner base two to the sixty-four modulo tau-prime XOR f of pi-prime and position."
10. $(q, r, s) = (g(e_i) \bmod 30,\ g(e_i) \bmod 30,\ g(e_i) \bmod 12)$.
    "q and r are g of e sub i mod thirty; s is g of e sub i mod twelve."
11. $\mathrm{slice}_{r,j,z}$ as in §8.1.
    "Slice start s equals z times q sub j plus min of z and rho sub j; end e equals s plus q sub j plus one if z less than rho sub j."

12. $W_{r,z} = \text{slice}_{r,0,z} \| \text{slice}_{r,1,z} \| \text{slice}_{r,2,z}$.
"Witness W r z is the concatenation of three slices."

13. $\mathsf{ACCEPT} \Leftrightarrow \forall i : S_i \in W_{i,z_i}$.
"Accept if and only if S sub i is in W sub i z sub i for all i."

14. $\Lambda = \bigwedge_{i=1}^{n} [\, S_i \in W_{i,z_i} \,]$.
"Lambda equals AND over i from one to n of S sub i in W sub i z sub i."

Appendix B , Step-by-Step Variants (Expanded, inline LaTeX)

1) Accumulator verdict Λ

Inputs. A secret string $S = s\_1, \ldots, s\_L$, per-round witnesses $W\_1, \ldots, W\_T$ (each W_i is the concatenation of zone slices for round i), and optional zone indices $z\_1, \ldots, z\_T$. Assume $T \geq L$.

Predicate. The acceptance predicate is $\Lambda = \bigwedge\_i = 1^{L}, [, s\_i \in W\_i,]$.

Procedure.

1. Initialize $\Lambda \leftarrow 1$.
2. For $i = 1$ to L:
   - Compute a constant-time membership flag $b\_i \leftarrow \text{mem}(s\_i, W\_i)$, where $\text{mem}(x, W)$ returns 1 iff $x \in W$ and 0 otherwise (use fixed-length scans to avoid timing leakage).
   - Update $\Lambda \leftarrow \Lambda \wedge b\_i$.
3. **Decision:** accept iff $\Lambda = 1$.

Early reject (optional). For latency, you may reject on the first $b\_i = 0$; for side-channel discipline, prefer the constant-time accumulation above.

Edge cases.

- If $T > L$, either ignore surplus rounds or use them as redundancy; if $T < L$, fail with $\Lambda = 0$.
- If a duress map DM is active, evaluate a parallel flag $d = \bigvee\_i = 1^{L} [, s\_i \in \mathsf{DM}(W\_i),]$ and log (Λ, d).

Complexity. With witness length ℓ_W (typically $\ell\_W = 12$), each membership is $O(\ell\_W)$ and verification is $O(L, \ell\_W)$ time and $O(1)$ space. With balanced foliation Z, the false-accept probability for random guessing is $p\_\text{FA} \approx Z^{-3L}$.

Correctness check. When W_i are generated from the same (ε, τ, π) and routing as the transcript indicates, honest provers with correct s_i satisfy $s\_i \in W\_i$ for all i, hence $\Lambda = 1$.

```
# Accumulator verdict (constant-time style)
function VERIFY_ACCUMULATOR(S[1..L], W[1..T]) -> bool:
    if T < L: return false
    Lambda := 1
    for i in 1..L:
        bi := CT_MEMBER(S[i], W[i])   # returns 0/1 in constant time
        Lambda := Lambda AND bi
    return (Lambda == 1)
```

2) Keystream framing

Header and key. Serialize $H = (\texttt{file\_id}, |, \texttt{chunk\_size}, |, \texttt{session\_nonce}, |, T, |, \texttt{domain\_tag})$ and derive $K = \mathrm{KDF}(\pi, H)$, where π is the embedded 512-bit prime seed and `domain_tag` ensures domain separation (e.g., "eni6ma.pool.v1").

Block function. For chunk counter $i \in 0, 1, 2, \ldots$, compute $\mathcal{K}\_i = \mathrm{PRF}\_K(H, |, i)$. Encryption is $C\_i = P\_i \oplus \mathcal{K}\_i$; decryption is $P\_i = C\_i \oplus \mathcal{K}\_i$.

Procedure (encrypt).

1. Emit H.
2. Set $K \leftarrow \mathrm{KDF}(\pi, H)$.
3. For $i = 0, 1, 2, \ldots$:
   - Compute $\mathcal{K}\_i \leftarrow \mathrm{PRF}\_K(H, |, i)$.
   - Output $C\_i \leftarrow P\_i \oplus \mathcal{K}\_i$.

Procedure (decrypt).

1. Parse H.
2. Recompute $K \leftarrow \mathrm{KDF}(\pi, H)$.
3. For each i, compute $\mathcal{K}\_i$ as above and recover $P\_i \leftarrow C\_i \oplus \mathcal{K}\_i$.

Security notes.

- Ensure (π, H, i) triples are never reused; uniqueness of H per object/session enforces stream independence.
- For authenticity, compute $\mathrm{MAC}\_K(H, |, C)$ or use an AEAD construction; reject if verification fails.
- Random access follows from the PRF: any block i can be (de)crypted without processing prior blocks.

Complexity. Throughput is $O(n)$ in total bytes with constant extra space; per-block work is one PRF call plus an XOR over the chunk.

```
# Two-way hash keystream framing
function ENCRYPT_STREAM(pi, H, P[0..m-1]) -> (H, C[0..m-1], tag):
    K  := KDF(pi, H)
    for i in 0..m-1:
        Ki := PRF(K, H || i)
        C[i] := P[i] XOR Ki
    tag := MAC(K, H || C)     # optional but recommended
    return (H, C, tag)

function DECRYPT_STREAM(pi, H, C[0..m-1], tag) -> P[0..m-1] or FAIL:
    K := KDF(pi, H)
    if not VERIFY_MAC(K, H || C, tag): return FAIL
    for i in 0..m-1:
        Ki := PRF(K, H || i)
        P[i] := C[i] XOR Ki
    return P
```