RATS
Internet-Draft
Intended status: Informational
Expires: 9 November 2026

C. Ayerbe Posada
ULISSY s.r.l.
M. Usama Sardar
TU Dresden
8 May 2026

TRIP: Trajectory-based Recognition of Identity Proof
draft-ayerbe-trip-protocol-04

Abstract

This document specifies the Trajectory-based Recognition of Identity Proof (TRIP) protocol, a decentralized mechanism for establishing claims of physical-world presence through cryptographically signed, spatially quantized location attestations called "breadcrumbs." Breadcrumbs are chained into an append-only log, bundled into verifiable epochs, and distilled into a Trajectory Identity Token (TIT) that serves as a persistent pseudonymous identifier.

The protocol employs a Criticality Engine grounded in statistical physics to distinguish biological movement from synthetic trajectories. Power Spectral Density (PSD) analysis detects the 1/f signature of Self-Organized Criticality in human mobility through the PSD scaling exponent alpha. A six-component Hamiltonian energy function scores each breadcrumb against the identity’s learned behavioral profile in real time.

This revision (-03) addresses three areas identified through expert review by researchers in the statistical physics community: it replaces informal terminology with standard spectral analysis nomenclature; it provides the analytical and numerical bridge between the Levy flight displacement exponent and the PSD scaling exponent; and it introduces a convergence analysis framework for quantifying the minimum trajectory length required for reliable single-trajectory classification. Additionally, this revision removes Passive Verification mode entirely, requiring all Attestation Results to be bound to Relying Party nonces via the Active Verification Protocol.

TRIP is designed to be transport-agnostic and operates independently of any particular naming system, blockchain, or application layer.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Copyright Notice

Table of Contents

1. Introduction

Conventional approaches to proving that an online actor corresponds to a physical human being rely on biometric capture, government-issued documents, or knowledge-based challenges. Each technique introduces a centralized trust anchor, creates honeypots of personally identifiable information (PII), and is susceptible to replay or deepfake attacks.

TRIP takes a fundamentally different approach: it treats sustained physical movement through the real world as evidence of embodied existence. A TRIP-enabled device periodically records its position as a "breadcrumb" -- a compact, privacy-preserving, cryptographically signed attestation that the holder of a specific Ed25519 key pair was present in a particular spatial cell at a particular time. An adversary who controls only digital infrastructure cannot fabricate a plausible trajectory because doing so requires controlling radio-frequency environments (GPS, Wi-Fi, cellular, IMU) at many geographic locations over extended periods.

This document specifies the data structures, algorithms, and verification procedures that constitute the TRIP protocol. It intentionally omits transport bindings, naming-system integration, and blockchain anchoring, all of which are expected to be addressed in companion specifications.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

Terms defined in the RATS Architecture [RFC9334] (Attester, Evidence, Verifier, Attestation Result, Relying Party) are used throughout this document with their RFC 9334 meanings. Additional terms specific to TRIP:

Breadcrumb: A single, signed attestation of spatiotemporal presence. The atomic unit of TRIP Evidence.

Trajectory: An ordered, append-only chain of breadcrumbs produced by a single identity key pair.

Epoch: A bundle of breadcrumbs (default 100) sealed with a Merkle root, forming a verifiable checkpoint.

Trajectory Identity Token (TIT): A pseudonymous identifier derived from an Ed25519 public key paired with trajectory metadata.

Criticality Engine: The analytical subsystem that evaluates trajectory statistics for signs of biological Self-Organized Criticality (SOC). In RATS terms, the Criticality Engine is a component of the Verifier.

PSD Scaling Exponent (alpha): The slope of the Power Spectral Density of the displacement time series in log-log space. This quantity is referred to in the spectral analysis literature as the "spectral exponent" or "scaling exponent." Human mobility produces alpha values in the range [0.30, 0.80], corresponding to 1/f pink noise -- the spectral signature of systems operating at criticality, as demonstrated by Parisi’s work on scale-free correlations in biological systems [PARISI-NOBEL].

Hamiltonian (H): A weighted energy function that quantifies how much a new breadcrumb deviates from the identity’s learned behavioral profile.

Anchor Cell: An H3 cell where an identity has historically spent significant time (e.g., home, workplace).

Flock: The set of co-located TRIP entities whose aggregate movement provides a reference signal for alignment verification.

Proof-of-Humanity (PoH) Certificate: A compact Attestation Result containing only statistical exponents derived from the trajectory, with no raw location data.

2. RATS Architecture Mapping

TRIP implements the Remote ATtestation procedureS (RATS) architecture defined in [RFC9334]. This section provides the normative mapping between TRIP components and RATS roles, establishes the attestation topology, and frames the detailed protocol mechanics defined in subsequent sections.

2.1. Role Mapping

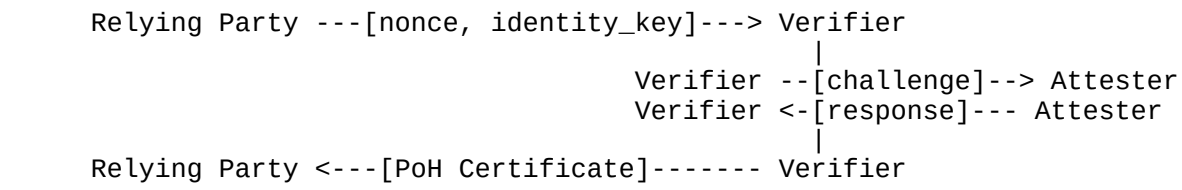
RATS Role	TRIP Component	Description
Attester	TRIP-enabled mobile device	Collects breadcrumbs, signs them with the identity Ed25519 private key, chains them into the append-only trajectory log, and transmits H3-quantized Evidence to the Verifier.
Evidence	Breadcrumbs and epoch records	H3-quantized spatiotemporal claims including cell identifiers, timestamps, context digests, chain hashes, and Ed25519 signatures. Evidence is transmitted from Attester to Verifier.
Verifier	Criticality Engine	Receives Evidence, performs chain verification, computes PSD scaling exponents (Section 7.1), fits Levy flight parameters (Section 8.1), evaluates the six-component Hamiltonian (Section 9), and produces Attestation Results.
Attestation Result	PoH Certificate and trust score	Contains only statistical exponents (alpha, beta, kappa) and aggregate scores. No raw Evidence (cell IDs, timestamps, chain hashes) is included in the Attestation Result. See Section 10.
Relying Party	Any service consuming PoH Certificates	Evaluates the Attestation Result against its own policy. Does not receive or process raw Evidence.

Table 1: TRIP-to-RATS Role Mapping

2.2. Attestation Topology

TRIP's Active Verification Protocol (Section 12.2) implements the RATS *background-check model*: the Relying Party initiates verification by sending the identity's public key and a freshness nonce to the Verifier, which challenges the Attester, evaluates the Evidence, and returns a PoH Certificate (Attestation Result) to the Relying Party.

Background-Check Model (normative):



The background-check model is the REQUIRED attestation topology for TRIP. It ensures that every Attestation Result is bound to a specific Relying Party challenge, preventing replay of certificates across contexts.

A *passport-model* deployment, where the Attester obtains a PoH Certificate in advance and presents it directly to Relying Parties, is not prohibited but provides weaker freshness guarantees. In the passport model, the certificate's validity duration (field 11 of the PoH Certificate, Section 10) and the chain head hash (field 13) provide the only freshness binding. Relying Parties accepting passport-model certificates SHOULD require short validity durations.

TRIP proposes the use of post-handshake attestation via [I-D.fossati-seat-expat] for integration with standard RATS attestation flows.

2.3. Evidence Flow

H3-quantized Evidence is transmitted from the Attester to the Verifier. This is an explicit design choice: the Verifier requires access to the full breadcrumb chain to compute PSD scaling exponents, fit Levy flight parameters, and evaluate the Hamiltonian.

Privacy preservation derives from the H3 quantization transform applied by the Attester before any data leaves the device, NOT from data locality. Raw GPS coordinates MUST NOT be transmitted. The quantization transform is lossy and irreversible.

The Verifier MUST NOT forward raw Evidence to Relying Parties. Only the Attestation Result (PoH Certificate) is disclosed to Relying Parties.

2.4. Verifier Trust Model

The Relying Party MUST trust the Verifier that produced the Attestation Result. The TRIP protocol supports multiple independent Verifiers. An Attester MAY submit Evidence to more than one Verifier. A Relying Party MAY accept Attestation Results from any Verifier it trusts.

Each Verifier MUST have its own Ed25519 key pair. The Verifier signs PoH Certificates with its private key (field 14). Relying Parties verify this signature against the Verifier’s published public key.

3. Breadcrumb Data Structure

A breadcrumb is encoded as a CBOR map [RFC8949] with the following fields:

Key	CBOR Type	Description
0	uint	Index (sequence number)
1	bstr (32)	Identity public key (Ed25519)
2	uint	Timestamp (Unix seconds)
3	uint	H3 cell index
4	uint	H3 resolution (7-10)
5	bstr (32)	Context digest (SHA-256)
6	bstr (32) / null	Previous block hash
7	map	Meta flags
8	bstr (64)	Ed25519 signature

Table 2: Breadcrumb CBOR Fields

3.1. Spatial Quantization

The H3 geospatial indexing system [H3] partitions the Earth’s surface into hexagonal cells at multiple resolutions. TRIP employs resolutions 7 through 10:

Resolution	Avg. Area	Edge Length	Use Case
7	~5.16 km^2	~1.22 km	Rural / low-density
8	~0.74 km^2	~0.46 km	Suburban / general
9	~0.11 km^2	~0.17 km	Urban / high-density
10	~0.015 km^2	~0.07 km	Default / standard verification

Table 3: H3 Resolution Parameters

A conforming implementation MUST quantize raw GPS coordinates to an H3 cell before any signing or storage operation. Raw coordinates MUST NOT appear in breadcrumbs or in any protocol message transmitted between TRIP entities.

H3 resolution is a configurable protocol parameter. Implementations SHOULD default to resolution 10. Deployments MAY select alternative resolutions based on jurisdictional requirements, population density, and use-case sensitivity. Lower resolutions (larger cells) provide stronger location privacy at the cost of reduced spatial discrimination for trust computation.

3.2. Context Digest Computation

The context digest binds ambient environmental signals to the breadcrumb without revealing them. The digest is computed as follows:

1. Construct a pipe-delimited string of tagged components in the following order:
    * "h3:" followed by the H3 cell hex string
    * "ts:" followed by the timestamp bucketed to 5-minute intervals (floor(Unix_minutes / 5) * 5)
    * "wifi:" followed by the first 16 hex characters of SHA-256(sorted comma-joined BSSIDs), if Wi-Fi scan data is available
    * "cell:" followed by the first 16 hex characters of SHA-256(sorted comma-joined tower IDs), if cellular data is available
    * "imu:" followed by the first 16 hex characters of SHA-256(IMU vector string), if inertial sensor data is available
2. Compute SHA-256 over the UTF-8 encoding of the resulting string.

Absent components MUST be omitted entirely, not represented as empty strings.

3.3. Signature Production

The signable payload is the deterministic CBOR encoding (per Section 4.2 of [RFC8949]) of a CBOR map containing fields 0 through 7, with map keys sorted in ascending integer order. The Ed25519 signature [RFC8032] is computed over the raw bytes of this CBOR encoding and stored at key 8.

```
signable_payload = CBOR-Deterministic(fields[0..7])
signature        = Ed25519-Sign(private_key, signable_payload)
```

Deterministic CBOR encoding ensures that any conforming implementation produces identical byte sequences for the same logical content, which is essential for reproducible signature verification across heterogeneous platforms.

3.4. Block Hash and Chaining

The block hash is the SHA-256 digest of the complete deterministic CBOR encoding of the breadcrumb (fields 0 through 8 inclusive, i.e., including the signature):

```
BreadcrumbHash(B) = SHA-256(CBOR-Deterministic(B[0..8]))
B[N+1].field[6]   = BreadcrumbHash(B[N])
B[0].field[6]     = null
```

Each breadcrumb at index > 0 MUST carry the block hash of its immediate predecessor in field 6, forming an append-only hash chain. The genesis breadcrumb (index 0) MUST set field 6 to null (CBOR simple value 22).

4. Chain Management

4.1. Location Deduplication

Proof-of-Trajectory requires demonstrated movement. A conforming implementation MUST reject a breadcrumb if the H3 cell is identical to the immediately preceding breadcrumb. Implementations SHOULD also enforce a cap (default 10) on the number of breadcrumbs recordable at any single H3 cell to prevent stationary farming.

4.2. Minimum Collection Interval

Breadcrumbs SHOULD be collected at intervals of no less than 15 minutes. An implementation MAY allow shorter intervals during explicit "exploration" sessions but MUST NOT accept intervals shorter than 5 minutes.

4.3. Chain Verification

A Verifier MUST check:

1. Index values form a contiguous sequence starting at 0.

2. Timestamps are monotonically non-decreasing.

3. Each previousHash matches the block hash of the prior breadcrumb.

4. Each Ed25519 signature verifies against the identity public key and the canonical signed data.

5. Epochs

An epoch seals a batch of breadcrumbs (default 100) under a Merkle root. The epoch record is a CBOR map containing:

Key	Type	Description
0	uint	Epoch number
1	bstr (32)	Identity public key
2	uint	First breadcrumb index
3	uint	Last breadcrumb index
4	uint	Timestamp of first breadcrumb
5	uint	Timestamp of last breadcrumb
6	bstr (32)	Merkle root of breadcrumb hashes
7	uint	Count of unique H3 cells
8	bstr (64)	Ed25519 signature over fields 0-7

Table 4: Epoch CBOR Fields

The Merkle tree MUST use SHA-256 and a canonical left-right ordering of breadcrumb block hashes. An epoch is sealed when the breadcrumb count reaches the epoch size threshold.

6. Trajectory Identity Token (TIT)

A TIT is the externally presentable identity derived from a TRIP trajectory. It consists of:

* The Ed25519 public key (32 bytes).
* The current epoch count.
* The total breadcrumb count.
* The count of unique H3 cells visited.
* A trust score (see Section 11).

A TIT SHOULD be encoded as a CBOR map for machine consumption and MAY additionally be represented as a Base64url string for URI embedding.

7. The Criticality Engine

The Criticality Engine is the core analytical component of the TRIP Verifier. It evaluates whether a trajectory exhibits the statistical signature of biological Self-Organized Criticality (SOC) -- the phenomenon where living systems operate at the boundary between order and chaos, producing scale-free correlations that are mathematically distinct from synthetic or automated movement.

The theoretical foundation rests on three pillars:

First, Parisi's demonstration [PARISI-NOBEL] that flocking organisms such as starling murmurations exhibit scale-free correlations [CAVAGNA-STARLINGS] where perturbations propagate across the entire group regardless of size. Crucially, Ballerini et al. showed that these interactions are topological (based on nearest k neighbors) rather than metric (based on distance) [BALLERINI-TOPOLOGICAL]. TRIP exploits this through Power Spectral Density analysis (Section 7.1): human movement produces characteristic 1/f pink noise that synthetic trajectories cannot replicate.

Second, Barabasi et al.'s discovery [BARABASI-MOBILITY] that human displacement follows truncated Levy flights with approximately 93% predictability [SONG-LIMITS]. TRIP learns each identity's mobility profile -- displacement distribution, anchor transition patterns, and circadian rhythms -- and detects deviations from these learned baselines (Section 8).

Third, a six-component Hamiltonian energy function (Section 9) that combines spatial, temporal, kinetic, flock-alignment, contextual, and structural analysis into a single anomaly score for each incoming breadcrumb. The Hamiltonian provides real-time detection while the PSD and mobility statistics provide aggregate trajectory assessment.

7.1. Power Spectral Density Analysis

The primary diagnostic is the Power Spectral Density (PSD) of the displacement time series. Given a trajectory of N breadcrumbs with displacements d(i) between consecutive breadcrumbs, the PSD is computed via the Discrete Fourier Transform:

```
S(f) = |DFT(d)|^2

where d = [d(0), d(1), ..., d(N-1)]
and d(i) = haversine_distance(cell(i), cell(i-1))
```

The PSD is then fitted to a power-law model:

```
S(f) ~ 1 / f^alpha
```

The exponent alpha is the PSD scaling exponent -- referred to in the spectral analysis literature as the "spectral exponent" or "scaling exponent." In the context of human mobility, this quantity captures the degree of long-range temporal correlation in movement patterns. The theoretical significance of this exponent derives from Parisi’s work demonstrating that biological systems operating at criticality produce characteristic scale-free correlations [PARISI-NOBEL]. The PSD scaling exponent is the critical diagnostic:

Alpha Range	Noise Type	Classification
0.00 - 0.15	White noise	Synthetic / automated script
0.15 - 0.30	Near-white	Suspicious (possible sophisticated bot)
0.30 - 0.80	Pink noise (1/f)	Biological / human
0.80 - 1.20	Near-brown	Suspicious (possible replay with drift)
1.20+	Brown noise	Drift anomaly / sensor failure

Table 5: PSD Scaling Exponent Classification

A conforming implementation MUST compute the PSD scaling exponent over a sliding window of the most recent 64 breadcrumbs (minimum) to 256 breadcrumbs (recommended). The alpha value MUST fall within [0.30, 0.80] for the trajectory to be classified as biological.

The key insight is that automated movement generators lack the long-range temporal correlations ("memory") inherent in a system operating at criticality. A random walk produces white noise (alpha near 0). A deterministic replay produces brown noise (alpha near 2). Only a biological system operating at the critical point produces pink noise in the characteristic [0.30, 0.80] range.

NOTE: The alpha range [0.30, 0.80] is a protocol-specified classification boundary constructed from combined literature. The boundaries are informed by empirical studies demonstrating 1/f-like spectral properties in human GPS trajectories [VADAI-GPS] and general spectral characteristics of human physical activity [MACZAK-SPECTRAL]. Deployments MAY adjust these boundaries based on population-specific calibration data, provided that the biological range remains centered near alpha = 0.55 and excludes the white noise (alpha < 0.15) and brown noise (alpha > 1.20) regions.

7.2. Criticality Confidence Score

The Criticality Confidence is a value in [0, 1] computed from the PSD scaling exponent and the goodness-of-fit (R-squared) of the power-law regression:

```
alpha_score = 1.0 - |alpha - 0.55| / 0.25

criticality_confidence = alpha_score * R_squared

where:
  0.55 is the center of the biological range
  0.25 is the half-width of the biological range
  R_squared is the coefficient of determination of the
    log-log linear regression
```

A criticality_confidence below 0.5 SHOULD trigger elevated monitoring. A value below 0.3 SHOULD flag the trajectory for manual review or additional verification challenges.

7.3. Levy-PSD Bridge

This section establishes the mathematical relationship between the truncated Levy flight displacement exponent beta (Section 8.1) and the PSD scaling exponent alpha (Section 7.1). Previous revisions of this specification asserted both as independent properties of human mobility. This section demonstrates that they are related through the spectral properties of heavy-tailed random processes.

7.3.1. Analytical Relationship

For a stationary stochastic process with displacement increments drawn from a symmetric stable distribution with stability index mu (where mu = beta - 1 for the Levy flight exponent beta used in Section 8.1), the Power Spectral Density of the cumulative displacement series scales as:

$$S(f) \sim f^{-\alpha}$$

where $\alpha = 2 - \mu = 2 - (\beta - 1) = 3 - \beta$

For pure (non-truncated) Levy flights.

However, human displacement follows TRUNCATED Levy flights with an exponential cutoff at distance kappa (Section 8.1). The truncation modifies the spectral relationship: at low frequencies (long time scales), the exponential cutoff causes the process to resemble Brownian motion (alpha approaching 2), while at high frequencies (short time scales), the pure Levy scaling dominates. For the intermediate frequency range relevant to TRIP’s sliding window (64-256 breadcrumbs collected at 15-minute intervals, spanning approximately 16-64 hours of movement data), the effective PSD scaling exponent is:

```
alpha_eff ~ (3 - beta) * g(N, kappa, delta_t)

where:
  beta    = Levy flight exponent (typically 1.50 - 1.90)
  g(...)  = correction factor for truncation and finite window
  N       = number of breadcrumbs in the analysis window
  kappa   = truncation distance (km)
  delta_t = mean inter-breadcrumb interval

For typical human values:
  beta  ~ 1.75  =>  3 - beta = 1.25
  g(...)  ~ 0.4 - 0.6  (empirically observed)
  alpha_eff ~ 0.50 - 0.75

This falls squarely within the biological range [0.30, 0.80].
```

7.3.2. Empirical Evidence

The analytical relationship above is supported by empirical studies:

* Vadai et al. [VADAI-GPS] analyzed GPS trajectory data and demonstrated 1/f-like spectral characteristics in human daily motion, with PSD scaling exponents consistent with the predicted range.

* Maczak et al. [MACZAK-SPECTRAL] studied 42 human subjects and found spectral exponents close to 1 in general physical activity data, confirming the presence of long-range temporal correlations consistent with Self-Organized Criticality.

* The original Levy flight analysis by Gonzalez, Hidalgo, and Barabasi [BARABASI-MOBILITY] reported beta values of approximately 1.75 +/- 0.15 across a population of 100,000 mobile phone users, which through the bridge equation predicts alpha_eff in [0.40, 0.70] -- consistent with the biological classification range.

7.3.3. Numerical Validation

Implementers SHOULD validate the Levy-PSD bridge for their specific deployment by conducting Monte Carlo simulations:

1. Generate 10,000 synthetic trajectories using truncated Levy flights with beta drawn uniformly from [1.50, 1.90] and kappa drawn from a log-normal distribution matching the target population.

2. Quantize each trajectory to H3 resolution 10 and apply the deduplication rules of Section 4.1.

3. Compute the PSD scaling exponent alpha for each synthetic trajectory.

4. Verify that the (beta, alpha) pairs fall within the expected relationship with the correction factor g in the range [0.3, 0.7].

Additionally, generate control trajectories from:

* Pure random walks (expected: alpha near 0)

* Deterministic replays of recorded trajectories (expected: alpha near 2)

* Correlated random walks with Gaussian increments (expected: alpha outside [0.30, 0.80])

The Monte Carlo validation confirms that the [0.30, 0.80] classification boundary correctly separates biological from synthetic trajectories with quantifiable error rates (see Section 7.4).

7.4. Convergence Analysis

The PSD scaling exponent, Levy flight parameters, and flock alignment metrics are fundamentally ensemble properties derived from statistical physics. Applying them to a single trajectory raises the question: how many breadcrumbs are required for these ensemble properties to converge on an individual trajectory with a given confidence level?

This section provides guidance on convergence behavior. Definitive false positive and false negative rates require empirical validation against real-world datasets (e.g., GeoLife, MDC), which is planned for a companion publication. The framework below describes the expected convergence properties and the protocol’s mitigation strategies.

7.4.1. Convergence Regimes

The reliability of TRIP’s statistical classifiers depends on trajectory length. Three regimes are identified:

Breadcrumbs	Regime	PSD Reliability	Levy Fit Reliability
0 - 63	Bootstrap	Not computed (insufficient data for DFT)	Not reliable
64 - 199	Provisional	Computed but with wide confidence intervals; alpha estimate variance ~ 0.15	Beta estimated but kappa poorly constrained
200+	Stable	Alpha estimate variance < 0.05; R-squared meaningful	Both beta and kappa well-constrained

Table 6: Convergence Regimes

The trust scoring formula (Section 11) incorporates profile maturity through the factor min(breadcrumb_count / 200, 1.0), which scales Hamiltonian weights during the bootstrap and provisional regimes.

7.4.2. Composition of Independent Tests

TRIP does not rely on any single statistical test. The six-component Hamiltonian (Section 9) combines independent classifiers: spatial Levy fit, temporal Markov properties, kinetic transition analysis, flock alignment, IMU cross-correlation, and chain structural integrity. Even if each individual test has a significant error rate on a short trajectory, the composition of independent tests reduces the combined error probability.

For k independent tests each with false positive rate p_i, the probability that a synthetic trajectory passes ALL tests simultaneously is:

P(false_positive_all) = product(p_i, i=1..k)

For k = 6 tests each with p_i = 0.1 (conservative):
P(false_positive_all) = 0.1^6 = 10^{-6}

In practice the tests are not perfectly independent (spatial and kinetic components share displacement data), so the actual combined false positive rate will be higher than the product bound. Empirical measurement is required.

7.4.3. Error Cost Asymmetry

TRIP's classification errors have asymmetric costs:

* *False negative* (human classified as bot): Low cost. The identity accumulates more breadcrumbs and is reclassified correctly as the trajectory lengthens. No permanent damage occurs.

* *False positive* (bot classified as human): Higher cost, but requires simultaneous spoofing across all six Hamiltonian components -- spatial displacement statistics, temporal circadian patterns, Markov transition probabilities, flock alignment, IMU cross-correlation, and chain timing regularity. This represents a significantly harder adversarial problem than defeating any single test.

7.4.4. Minimum Breadcrumbs for Classification

Based on the convergence analysis above, the minimum trajectory lengths for classification decisions are:

* *64 breadcrumbs*: Minimum for PSD computation. Sufficient for preliminary screening (reject obvious bots) but not for positive human classification.

* *100 breadcrumbs*: Minimum for handle claiming (Section 11). The Levy fit becomes usable and the Markov transition matrix begins to stabilize.

* *200 breadcrumbs*: RECOMMENDED for reliable positive human classification. At this length, the PSD alpha estimate has variance below 0.05 and the Levy parameters are well-constrained.

* *256+ breadcrumbs*: Sufficient for high-confidence classification suitable for high-stakes Relying Party decisions.

Determining precise false positive and false negative rates at each breadcrumb count requires empirical validation. Implementers SHOULD conduct the Monte Carlo simulations described in Section 7.3.3 and test against publicly available human mobility datasets to establish ROC curves and confidence intervals for their specific deployment parameters.

8. Mobility Statistics

This section defines the mobility model that enforces known constraints of human movement, as established by Barabasi et al. [BARABASI-MOBILITY].

8.1. Truncated Levy Flights

Human displacement between consecutive recorded locations follows a truncated power-law distribution:

```
P(delta_r) ~ delta_r^(-beta) * exp(-delta_r / kappa)
```

where:

```
  delta_r = displacement distance (km)
  beta    = power-law exponent (typically 1.50 - 1.90)
  kappa   = exponential cutoff distance (km)
```

The exponent beta captures the heavy-tailed nature of human movement: most displacements are short (home to office) but occasional long jumps (travel) follow a predictable distribution. The cutoff kappa is learned per identity and represents the characteristic maximum range.

A conforming implementation MUST maintain a running estimate of beta and kappa for each identity by fitting the displacement histogram using maximum likelihood estimation over the most recent epoch (100 breadcrumbs).

A new displacement that falls outside the 99.9th percentile of the fitted distribution MUST increment the spatial anomaly counter.

The relationship between beta and the PSD scaling exponent alpha is established in Section 7.3. Implementations SHOULD verify internal consistency between the fitted beta value and the observed alpha value; a discrepancy exceeding the expected range of the correction factor g (Section 7.3.1) MAY indicate data quality issues or adversarial manipulation of one metric.

8.2. Trajectory Predictability

Research has demonstrated that approximately 93% of human movement is predictable based on historical patterns [SONG-LIMITS]. TRIP exploits this by maintaining a Markov Transition Matrix over anchor cells:

```
T[a_i][a_j] = count(transitions from a_i to a_j)
                / count(all departures from a_i)
```

where a_i, a_j are anchor cells.

An anchor cell is defined as any H3 cell where the identity has recorded 5 or more breadcrumbs. The transition matrix is rebuilt at each epoch boundary.

The predictability score Pi for an identity is the fraction of observed transitions that match the highest-probability successor in the Markov matrix. Human identities converge toward Pi values in the range [0.80, 0.95] after approximately 200 breadcrumbs. Deviations below 0.60 are anomalous.

8.3. Circadian and Weekly Profiles

The implementation SHOULD maintain two histogram profiles:

* A circadian profile C[hour] recording the probability of activity in each hour of the day (24 bins).

* A weekly profile W[day] recording the probability of activity on each day of the week (7 bins).

These profiles provide the temporal baseline for the Hamiltonian temporal energy component (Section 9.2).

9. The Six-Component Hamiltonian

To assess each incoming breadcrumb, the Criticality Engine computes a weighted energy score H that quantifies how much the breadcrumb deviates from the identity’s learned behavioral profile. High energy indicates anomalous behavior; low energy indicates normalcy.

```
H = w_1 * H_spatial
  + w_2 * H_temporal
  + w_3 * H_kinetic
  + w_4 * H_flock
  + w_5 * H_contextual
  + w_6 * H_structure
```

Component	Weight	Diagnostic Target
H_spatial	0.25	Displacement anomalies (teleportation)
H_temporal	0.20	Circadian rhythm violations
H_kinetic	0.20	Anchor transition improbability
H_flock	0.15	Misalignment with local human flow
H_contextual	0.10	Sensor cross-correlation failure
H_structure	0.10	Chain integrity and timing regularity

Table 7: Hamiltonian Component Weights

Weights are modulated by the profile maturity m, defined as min(breadcrumb_count / 200, 1.0). During the bootstrap phase (m < 1.0), all weights are scaled by m, widening the acceptance threshold for new identities.

9.1. H_spatial: Displacement Anomaly

Given the identity's fitted truncated Levy distribution P(delta_r), the spatial energy for a displacement delta_r is the negative log-likelihood (surprise):

H_spatial = -log(P(delta_r))

where P(delta_r) = C * delta_r^(-beta) * exp(-delta_r / kappa) and C is the normalization constant.

Typical displacements yield H_spatial near the identity's historical baseline. A displacement that exceeds the identity's learned kappa cutoff by more than a factor of 3 produces an H_spatial value in the CRITICAL range.

9.2. H_temporal: Rhythm Anomaly

H_temporal = -log(C[current_hour]) - log(W[current_day])

Activity at 3:00 AM for an identity with a 9-to-5 circadian profile yields high H_temporal. Activity at 8:00 AM on a Tuesday for the same identity yields low H_temporal.

9.3. H_kinetic: Transition Anomaly

```
from_anchor = nearest anchor to previous breadcrumb
to_anchor   = nearest anchor to current breadcrumb
H_kinetic   = -log(max(T[from_anchor][to_anchor], epsilon))
```

where epsilon = 0.001 (floor to prevent log(0))

A home-to-office transition at 8:00 AM yields low H_kinetic. An office-to-unknown-city transition yields high H_kinetic.

9.4. H_flock: Topological Alignment

Inspired by Parisi's finding that starlings track their k nearest topological neighbors (k approximately 6-7) rather than all birds within a metric radius [BALLERINI-TOPOLOGICAL], the flock energy measures alignment between the identity's velocity vector and the aggregate velocity of co-located TRIP entities.

```
v_self  = displacement vector of current identity
v_flock = mean displacement vector of k nearest
          co-located identities (k = 7)

alignment = dot(v_self, v_flock)
            / (|v_self| * |v_flock|)

H_flock = 1.0 - max(alignment, 0)
```

When flock data is unavailable (sparse network or privacy constraints), the implementation SHOULD fall back to comparing the current velocity against the identity's own historical velocity distribution at the same location and time-of-day.

H_flock defeats GPS replay attacks: an adversary replaying a previously recorded trajectory will find that the ambient flock has changed since the recording, producing a misalignment signal.

9.5. H_contextual: Sensor Cross-Correlation

```
H_contextual = divergence(
  observed_imu_magnitude,
  expected_imu_magnitude_for(gps_displacement)
)
```

Implementations that lack IMU access MUST set H_contextual = 0 and SHOULD increase the weights of other components proportionally.

9.6. H_structure: Chain Structural Integrity

* Inter-breadcrumb timing regularity: excessively uniform intervals suggest automation.

* Hash chain continuity: any break in the chain produces maximum H_structure.

* Phase-space smoothness: the velocity-acceleration phase portrait of a human trajectory traces smooth loops, while bots produce either chaotic blobs or tight limit cycles.

9.7. Alert Classification

The total Hamiltonian H maps to an alert level. The baseline H_baseline is the rolling median of the identity's own recent energy values, making the threshold self-calibrating per identity:

H Range	Level	Action
[0, H_baseline * 1.5)	NOMINAL	Normal operation
[H_baseline * 1.5, 3.0)	ELEVATED	Increase sampling frequency, log
[3.0, 5.0)	SUSPICIOUS	Flag for review, require reconfirmation
[5.0, infinity)	CRITICAL	Freeze trust score, trigger challenge

Table 8: Hamiltonian Alert Levels

10. Proof-of-Humanity Certificate

A PoH Certificate is a compact, privacy-preserving Attestation Result (in the RATS sense) asserting that an identity has demonstrated biological movement characteristics. It contains ONLY statistical exponents derived from the trajectory -- no raw location data, no GPS coordinates, no cell identifiers.

The certificate is encoded as a CBOR map:

Key	Type	Description
0	bstr (32)	Identity public key
1	uint	Issuance timestamp
2	uint	Epoch count at issuance
3	float	PSD scaling exponent alpha
4	float	Levy beta exponent
5	float	Levy kappa cutoff (km)
6	float	Predictability score Pi
7	float	Criticality confidence
8	float	Trust score T
9	uint	Unique cell count
10	uint	Total breadcrumb count
11	uint	Validity duration (seconds)
12	bstr (16)	Relying Party nonce (REQUIRED)
13	bstr (32)	Chain head hash at issuance (REQUIRED)
14	bstr (64)	Verifier Ed25519 signature

Table 9: PoH Certificate CBOR Fields

Fields 12 and 13 are REQUIRED in all PoH Certificates. Every certificate MUST be issued in response to an Active Verification request (Section 12). There is no passive issuance mode.

A Relying Party receiving a PoH Certificate can verify:

1. The Verifier signature (field 14) is valid against a trusted Verifier public key.

2. The PSD scaling exponent alpha (field 3) falls within [0.30, 0.80].

3. The criticality confidence (field 7) exceeds the Relying Party's policy threshold.

4. The trust score (field 8) meets application requirements.

5. The certificate has not expired (field 1 + field 11 > current time).

6. The nonce (field 12) matches the Relying Party's original challenge.

7. The chain head hash (field 13) provides freshness binding.

The certificate reveals NOTHING about where the identity has been -- only that it has moved through the world in a manner statistically consistent with a biological organism.

11. Trust Scoring

```
T = 0.40 * min(breadcrumb_count / 200, 1.0)
  + 0.30 * min(unique_cells / 50, 1.0)
  + 0.20 * min(days_since_first / 365, 1.0)
  + 0.10 * chain_integrity
```

chain_integrity = 1.0 if chain verification passes, else 0.0
T is expressed as a percentage in [0, 100].

The threshold for claiming a handle (binding a human-readable name to a TIT) requires breadcrumb_count >= 100 and T >= 20.

A trajectory that fails the criticality test (alpha outside [0.30, 0.80]) MUST have its trust score capped at 50, regardless of other factors.

12. Replay Protection and Active Verification

TRIP provides replay protection at two distinct layers: protection of the Evidence chain against tampering, and protection of Attestation Results against replay to Relying Parties. All Attestation Results MUST be issued via the Active Verification Protocol described in this section.

12.1. Chain-Level Replay Protection

The monotonically increasing index and the chaining via the previous block hash field provide replay protection within a single trajectory. A replayed breadcrumb will fail the chain integrity check. Cross-trajectory replay will fail Ed25519 signature verification.

12.2. Active Verification Protocol

The Active Verification Protocol provides cryptographic freshness guarantees by binding the Attestation Result to a Relying Party-supplied nonce, the current chain head, and the current time. This is the ONLY verification mode supported by TRIP. There is no passive verification mode.

The protocol proceeds as follows:

1. The Relying Party generates an unpredictable nonce (RECOMMENDED: 16 bytes from a cryptographically secure random number generator) and sends a Verification Request to the Verifier:

```
VerificationRequest = {
  0 => bstr .size 32,   ; identity public key
  1 => bstr .size 16,   ; nonce
  2 => uint,            ; request timestamp
  3 => uint,            ; requested freshness window (seconds)
}
```

2. The Verifier delivers a Liveness Challenge to the Attester via a real-time channel (e.g., WebSocket push, push notification):

```
LivenessChallenge = {
  0 => bstr .size 16,   ; nonce (from Relying Party)
  1 => bstr .size 32,   ; verifier identity (public key)
  2 => uint,            ; challenge timestamp
  3 => uint,            ; response deadline (seconds)
}
```

3. The Attester constructs and signs a Liveness Response binding the nonce to the current chain state:

```
LivenessResponse = {
  0 => bstr .size 16,   ; nonce echo
  1 => bstr .size 32,   ; chain_head_hash
  2 => uint,            ; response timestamp
  3 => uint,            ; current breadcrumb index
  4 => bstr .size 64,   ; Ed25519 signature over fields 0-3
}
```

4. The Verifier validates the Liveness Response by checking:

   * The Ed25519 signature (field 4) is valid against the identity's public key over fields 0-3.

   * The nonce echo (field 0) matches the original Verification Request.

   * The chain_head_hash (field 1) is consistent with the Verifier's stored trajectory state.

   * The response timestamp (field 2) is within the deadline.

   * The breadcrumb index (field 3) matches or exceeds the Verifier's last known index.

5. Upon successful validation, the Verifier produces a fresh PoH Certificate with field 12 set to the nonce and field 13 set to the chain_head_hash, signs it, and returns it to the Relying Party.

6. The Relying Party verifies the PoH Certificate per Section 10, confirming that field 12 matches its original nonce.

If the Attester does not respond within the deadline, the Verifier MUST return an error. The Verifier MUST NOT issue a PoH Certificate without a valid Liveness Response.

12.3. Active Verification CDDL

The following CDDL [RFC8610] schema defines the Active Verification messages:

```
; Active Verification Protocol CDDL Schema

verification-request = {
  0 => bstr .size 32,        ; identity_key
  1 => bstr .size 16,        ; nonce
  2 => uint,                 ; request_timestamp
  3 => uint,                 ; freshness_window_seconds
}

liveness-challenge = {
  0 => bstr .size 16,        ; nonce
  1 => bstr .size 32,        ; verifier_key
  2 => uint,                 ; challenge_timestamp
  3 => uint,                 ; response_deadline_seconds
}

liveness-response = {
  0 => bstr .size 16,        ; nonce_echo
  1 => bstr .size 32,        ; chain_head_hash
  2 => uint,                 ; response_timestamp
  3 => uint,                 ; current_breadcrumb_index
  4 => bstr .size 64,        ; ed25519_signature
}
```

13. Security Considerations

13.1. GPS Replay Attacks

An adversary records a legitimate trajectory and replays the GPS coordinates on a different device. TRIP detects this through multiple channels:

* H_flock: the ambient flock has changed since the recording.

* H_contextual: unless the adversary also replays Wi-Fi BSSIDs, cellular tower IDs, and IMU data, the context digest will not match.

* H_structure: the timing regularity of a replay is typically either too perfect or shifted in a detectable pattern.

13.2. Synthetic Walk Generators

* PSD scaling exponent test: random walk generators produce white noise (alpha approximately 0). Brownian motion generators produce alpha approximately 2. Neither falls in the biological [0.30, 0.80] range.

* Levy flight fitting: synthetic displacements rarely match the truncated power-law distribution with biologically plausible beta and kappa values.

* Predictability test: synthetic trajectories either show near-zero predictability (random) or near-perfect predictability (scripted), both outside the human [0.80, 0.95] range.

13.3. Emulator Injection

An adversary runs the TRIP client on an Android/iOS emulator with spoofed GPS. Detection relies on H_contextual (emulators lack real IMU data) and context digest (emulators lack real Wi-Fi and cellular data).

13.4. Device Strapping (Robot Dog Attack)

An adversary straps a phone to a mobile robot or drone. This is the most sophisticated attack because it produces real GPS, Wi-Fi, cellular, and IMU data from actual physical movement. Mitigation relies on PSD analysis (robotic movement lacks 1/f noise), phase-space smoothness (H_structure), and circadian profiles. This attack remains an active area of research.

13.5. Verifier Compromise

A compromised Verifier could issue fraudulent PoH Certificates. Mitigations: Relying Parties SHOULD accept certificates from multiple independent Verifiers; key rotation and revocation procedures SHOULD be established; the Active Verification Protocol ensures even a compromised Verifier cannot produce a valid certificate without the Attester’s cooperation.

13.6. Denial of Service

Verifiers SHOULD rate-limit requests per identity and per Relying Party. The Active Verification Protocol’s real-time requirement provides an inherent rate limit on valid completions.

13.7. Statistical Classifier Limitations

The Criticality Engine applies ensemble statistical properties (PSD scaling exponent, Levy flight parameters, flock alignment) to individual trajectories. As discussed in Section 7.4, the reliability of these classifiers depends on trajectory length. Implementers MUST be aware that:

* Classification confidence is low during the bootstrap regime (fewer than 64 breadcrumbs) and moderate during the provisional regime (64-199 breadcrumbs).

* The alpha range [0.30, 0.80] is a protocol-specified boundary informed by, but not directly taken from, a single peer-reviewed source. Deployments SHOULD calibrate this range against population-specific data.

* The composition of six independent Hamiltonian components provides defense in depth, but the actual combined error rate depends on the degree of independence between components, which requires empirical measurement.

* Definitive ROC curves and confidence intervals require validation against real-world human mobility datasets. This validation is planned for a companion publication and is outside the scope of this protocol specification.

14. Privacy Considerations

14.1. Quantization-Based Privacy

TRIP's privacy model is based on lossy spatial quantization, not on data locality. H3-quantized Evidence is transmitted from the Attester to the Verifier. Raw GPS coordinates MUST NOT be transmitted. At the default resolution 10, each cell covers approximately 15,000 m^2, providing meaningful ambiguity in populated areas.

14.2. Verifier Data Handling

The Verifier MUST NOT forward raw Evidence to Relying Parties. The Verifier MUST disclose its data retention policy. The Verifier SHOULD retain only statistical aggregates and MAY discard individual breadcrumbs after incorporation. The Verifier MUST support data deletion where required by law.

14.3. Relying Party Data Minimization

The PoH Certificate reveals statistical exponents and aggregate counts. A Relying Party does NOT learn which cities or specific locations the identity has visited, the identity's home or workplace, the identity's daily schedule, or any raw trajectory data.

14.4. Trajectory Correlation and Sybil Resistance

A single physical entity operating multiple TRIP identities simultaneously constitutes a Sybil attack. TRIP raises the cost: each identity requires a separate physical device, weeks of sustained movement, and independent trajectory accumulation. The H_flock component provides detection of co-located trajectories with identical displacement vectors.

14.5. Population Density Considerations

In sparsely populated areas, even cell-level granularity may narrow identification. Implementations SHOULD use lower resolution in rural areas and MAY allow users to override to a lower resolution at any time.

15. Deployment Considerations

15.1. Multiple Verifier Deployments

Any entity that implements the verification procedures defined in this specification MAY operate as a TRIP Verifier. An Attester MAY submit Evidence to more than one Verifier.

15.2. Verifier Interoperability

All conforming Verifiers MUST implement chain integrity verification, PSD scaling exponent classification, and the PoH Certificate format. Two Verifiers processing the same Evidence SHOULD produce consistent alpha, beta, and kappa values within numerical precision bounds.

15.3. Transport Binding

This specification does not mandate a specific transport. Implementations MAY use HTTPS, WebSocket, CoAP, or any transport providing confidentiality and integrity protection. The Active Verification Protocol requires a real-time channel.

15.4. Naming System Integration

The binding of human-readable names to TRIP identities is outside the scope of this specification and is expected to be addressed in a companion document.

15.5. Accessibility and Low-Mobility Users

TRIP does not require geographic travel. It requires sustained physical existence over time. A person who remains in a single H3 cell generates a valid trajectory; the trust scoring formula assigns 20% weight to temporal continuity and 40% to breadcrumb count, both accumulating regardless of spatial diversity. The context digest provides environmental diversity even without movement. The Hamiltonian is self-calibrating per identity. For stationary users, implementations SHOULD supplement spatial PSD with temporal PSD (analyzing inter-breadcrumb timing patterns). Deployments MUST NOT impose minimum spatial diversity requirements that would exclude users with mobility limitations.

16. IANA Considerations

This document has no IANA actions at this time. Future revisions may request CBOR tag assignments and a media type registration for application/trip+cbor.

17. References

17.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/rfc/rfc8032>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/rfc/rfc8610>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/rfc/rfc8949>.

[RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <https://www.rfc-editor.org/rfc/rfc9334>.

17.2. Informative References

[BALLERINI-TOPOLOGICAL]
Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Lecomte, V., Orlandi, A., Parisi, G., Procaccini, A., Viale, M., and V. Zdravkovic, "Interaction ruling animal collective behavior depends on topological rather than metric distance", DOI 10.1073/pnas.0711437105, 2008, <https://doi.org/10.1073/pnas.0711437105>.

[BARABASI-MOBILITY]
Gonzalez, M.C., Hidalgo, C.A., and A.-L. Barabasi, "Understanding individual human mobility patterns", DOI 10.1038/nature06958, 2008, <https://doi.org/10.1038/nature06958>.

[CAVAGNA-STARLINGS]
Cavagna, A., Cimarelli, A., Giardina, I., Parisi, G., Santagati, R., Stefanini, F., and M. Viale, "Scale-free correlations in starling flocks", DOI 10.1073/pnas.1005766107, 2010, <https://doi.org/10.1073/pnas.1005766107>.

[H3] Uber Technologies, "H3: Uber’s Hexagonal Hierarchical Spatial Index", 2023, <https://h3geo.org/>.

[I-D.fossati-seat-expat]
Sardar, M. U., Fossati, T., Reddy.K, T., Sheffer, Y., Tschofenig, H., and I. Mihalcea, "Remote Attestation with Exported Authenticators", Work in Progress, Internet-Draft, draft-fossati-seat-expat-02, 26 February 2026, <https://datatracker.ietf.org/doc/html/draft-fossati-seat-expat-02>.

[I-D.ietf-rats-eat]
Lundblade, L., Mandyam, G., O’Donoghue, J., and C. Wallace, "The Entity Attestation Token (EAT)", Work in Progress, Internet-Draft, draft-ietf-rats-eat-31, 6 September 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-rats-eat-31>.

[I-D.lkspa-wimse-verifiable-geo-fence]
Krishnan, R., Smith, N., Lopez, D., Prasad, A., and S. Addepalli, "Privacy Preserving Verifiable Geofencing with Residency Proofs for Sovereign Workloads", Work in Progress, Internet-Draft, draft-lkspa-wimse-verifiable-geo-fence-04, 1 March 2026, <https://datatracker.ietf.org/doc/html/draft-lkspa-wimse-verifiable-geo-fence-04>.

[I-D.mandyam-rats-proxlocclaim]
Mandyam, G., "The Proximate Location Claim", Work in Progress, Internet-Draft, draft-mandyam-rats-proxlocclaim-01, 17 January 2024, <https://datatracker.ietf.org/doc/html/draft-mandyam-rats-proxlocclaim-01>.

[I-D.richardson-rats-pop-endorsement]
Richardson, M., "Proof of Position for Auditor managed Endorsements", Work in Progress, Internet-Draft, draft-richardson-rats-pop-endorsement-00, 13 May 2025, <https://datatracker.ietf.org/doc/html/draft-richardson-rats-pop-endorsement-00>.

[MACZAK-SPECTRAL]
Maczak, B., "General spectral characteristics of human activity", DOI 10.1038/s41598-024-54165-4, 2024, <https://doi.org/10.1038/s41598-024-54165-4>.

[PARISI-NOBEL]
The Nobel Foundation, "Nobel Prize in Physics 2021: Giorgio Parisi", 2021, <https://www.nobelprize.org/prizes/physics/2021/parisi/facts/>.

[SONG-LIMITS]
Song, C., Qu, Z., Blumm, N., and A.-L. Barabasi, "Limits of Predictability in Human Mobility", DOI 10.1126/science.1177170, 2010, <https://doi.org/10.1126/science.1177170>.

[VADAI-GPS]
Vadai, G., Antal, A., and Z. Gingl, "Spectral Analysis of Fluctuations in Humans' Daily Motion", DOI 10.1142/S0219477519500287, 2019, <https://doi.org/10.1142/S0219477519500287>.

Appendix A. Relationship to Other Geo-Location Proposals

Several Internet-Drafts proposed in the RATS and WIMSE working groups address aspects of location attestation. This appendix positions TRIP relative to four such proposals.

A.1. EAT Location Claim

The Entity Attestation Token [I-D.ietf-rats-eat] defines a location claim that conveys a single point-in-time geographic position in WGS-84 coordinates. The claim provides no temporal depth, no privacy quantization, and no behavioral context. TRIP could emit its Trajectory Identity Token or Proof-of-Humanity Certificate as an EAT claim, providing longitudinal behavioral evidence within an EAT-formatted Attestation Result. The two are complementary.

A.2. Proximate Location Claim

The Proximate Location Claim [I-D.mandyam-rats-proxlocclaim] defines evidence of relative position derived from secure ranging between two devices, typically using ultra-wideband (UWB) radio per FiRa Consortium specifications. Proximate Location is instantaneous and pairwise; TRIP is longitudinal and self-attested. Proximate Location requires dedicated radio hardware in both endpoints; TRIP uses commodity GNSS and ambient signals already present on consumer mobile devices.

A.3. Verifiable Geofencing for Workloads

The Verifiable Geofencing draft [I-D.lkspa-wimse-verifiable-geo-fence] targets confidential computing workloads, providing cryptographically verifiable proof-of-residency that binds a workload identity to a host platform and a geographic boundary. Verifiable Geofencing answers "where is this workload executing?". TRIP answers "who is the human or autonomous entity behind this identity, and how has that entity moved over time?". The two address orthogonal concerns: residency versus operator integrity.

A.4. Proof of Position for Auditor Endorsements

The Proof of Position draft [I-D.richardson-rats-pop-endorsement] defines a mechanism by which a human auditor establishes physical contact with a device, typically via USB or serial console, and produces an endorsement asserting properties that the device cannot self-claim, including its physical location. Proof of Position is one-time, human-mediated, and locally delivered. TRIP is continuous, self-attested, and remotely verifiable.

A.5. Comparison Summary

Proposal	Temporal Model	Evidence Source	Primary Question
EAT Location	Instantaneous	Self-reported coordinate	Where is this device right now?
Proximate Location	Instantaneous	Secure ranging hardware	Is device X near device Y?
Verifiable Geofence	Continuous (workload lifetime)	TPM + sensor attestation	Is this workload inside the boundary?
PoP Endorsement	One-time	Human auditor, side-channel	Did a trusted human visit this device?
TRIP	Longitudinal	Self-attested behavioral trajectory	Has this identity moved like a biological entity over time?

Table 10

TRIP is the only proposal in this set that uses behavioral trajectory over time as its source of evidence. The others provide instantaneous location claims, jurisdictional residency proofs, or one-time endorsements. TRIP is intended to coexist with these specifications, supplying a distinct dimension of evidence -- continuity of physical existence -- that none of them addresses.

Appendix B. History

Version -01 introduced a Criticality Engine grounded in Giorgio Parisi’s Nobel Prize-winning work on scale-free correlations [PARISI-NOBEL] and Albert-Laszlo Barabasi’s research on the fundamental limits of human mobility [BARABASI-MOBILITY].

Version -02 formalized the mapping to the RATS Architecture [RFC9334], introduced the Active Verification Protocol with cryptographic freshness guarantees, and corrected the privacy model.

This revision (-03) addresses three substantive issues identified through expert review by researchers working in the statistical physics of complex systems:

First, it replaces the informal term "Parisi Factor" with the standard spectral analysis term "PSD scaling exponent alpha," properly attributing the theoretical foundation to Parisi's work without conflating tribute with established nomenclature.

Second, it provides the missing analytical and numerical bridge between the Levy flight displacement exponent beta (Section 8.1) and the PSD scaling exponent alpha (Section 7.1). Previous revisions asserted both as independent properties of human mobility without demonstrating their mathematical relationship.

Third, it introduces a convergence analysis framework (Section 7.4) that addresses the fundamental question of applying ensemble statistical properties to single trajectories, including guidance on minimum trajectory length and error rate estimation.

Additionally, this revision removes Passive Verification mode entirely. All Attestation Results MUST now be bound to Relying Party nonces via the Active Verification Protocol (Section 12), eliminating the replay vulnerability identified in -02 review.

B.1. Changes from -02

This section summarizes the substantive changes from draft-ayerbe-trip-protocol-02:

* Replaced the term "Parisi Factor" with the standard spectral analysis term "PSD scaling exponent alpha" throughout the document. The theoretical contribution of Parisi's work is acknowledged in the motivation and terminology, not in the variable naming.

* Added Section 7.3 (Levy-PSD Bridge) providing the analytical relationship between the Levy flight displacement exponent beta and the PSD scaling exponent alpha, with supporting references to empirical studies [MACZAK-SPECTRAL] [VADAI-GPS].

* Added Section 7.4 (Convergence Analysis) addressing the application of ensemble statistical properties to single trajectories, including minimum trajectory length guidance and error rate estimation framework.

* Removed Passive Verification mode entirely (Section 12). All Attestation Results MUST now be produced via the Active Verification Protocol with Relying Party-supplied nonces. The PoH Certificate fields for nonce (field 12) and chain head hash (field 13) are now REQUIRED, not optional.

* Updated the PoH Certificate (Section 10) to reflect mandatory Active Verification fields.

* Added references to recent empirical studies on spectral properties of human GPS trajectories.

B.2. Changes from -03

This section summarizes the changes from draft-ayerbe-trip-protocol-03:

* Added Appendix A (Relationship to Other Geo-Location Proposals) positioning TRIP relative to four parallel proposals: EAT Location Claim, Proximate Location, Verifiable Geofencing, and PoP Endorsement, per the commitment made on rats@ietf.org in February.

* Added RFC 8610 reference and citation in Section 12.3.

* Editorial cleanup of Section 16.

Appendix C. Acknowledgements

The TRIP protocol builds upon foundational work in cryptographic identity systems, geospatial indexing, statistical physics, and network science. The author thanks the contributors to the H3 geospatial system, the Ed25519 specification authors, and the broader IETF community for establishing the standards that TRIP builds upon. The Criticality Engine framework is inspired by the work of Giorgio Parisi on scale-free correlations in biological systems and Albert-Laszlo Barabasi on the fundamental limits of human mobility.

The authors thank Jun Zhang for raising critical questions about accessibility and the applicability of mobility models to users with limited physical mobility, leading to the Accessibility and Low-Mobility Users section.

The authors thank an anonymous reviewer from the statistical physics community for identifying three critical issues addressed in this revision: the need for standard spectral analysis terminology, the missing analytical bridge between Levy flight parameters and PSD scaling exponents, and the fundamental question of applying ensemble properties to single trajectories. These contributions led to Sections 6.3 and 6.4 and the new Section 13.7 on statistical classifier limitations.

Authors' Addresses

Camilo Ayerbe Posada
ULISSY s.r.l.
Via Gaetano Sacchi 16
00153 Roma RM
Italy
Email: cayerbe@gmail.com

Muhammad Usama Sardar
TU Dresden
Dresden
Germany
Email: muhammad_usama.sardar@tu-dresden.de