# Entros Protocol: A Framework for Temporally-Consistent, Decentralized Proof-of-Personhood

Charles Hooper
github.com/entros-protocol

Original: June 27, 2025 Updated: August 7, 2026

**Abstract**

The proliferation of sophisticated AI and bot networks necessitates methods for verifying human uniqueness and liveness in digital ecosystems. Existing Proof-of-Personhood (PoP) solutions rely on centralized authorities, static biometrics, or socially correlatable data. These approaches create different privacy, security, and access tradeoffs. We introduce the Entros Protocol, a decentralized framework for PoP and Self-Sovereign Identity built on Solana. The core innovation is *temporal consistency*: the assertion that human identity is best proven by biological and behavioral change over time, not by one static secret. The framework captures multi-modal behavioral data during a configurable challenge, extracts a 308-dimensional feature vector, and produces a 256-bit locality-sensitive hash via SimHash. A Groth16 zero-knowledge proof verifies that consecutive fingerprints fall within a bounded Hamming distance without revealing either value. Attestations are anchored to non-transferable identity tokens (SPL Token-2022) with progressive Trust Scores. We provide formal security definitions, analyze replay, synthesis, and Sybil attacks, distinguish first-capture validation from returning-user continuity, and present benchmarks from a working Solana devnet implementation.

**Keywords:** Proof-of-Personhood, Decentralized Identity, Behavioral Biometrics, Zero-Knowledge Proofs, Groth16, SimHash, Liveness Detection, Temporal Consistency, Solana.

## 1 Introduction

The distinction between human and artificial actors in digital systems is increasingly blurred. Sybil attacks [1], where a single adversary creates numerous fake identities, undermine fair token distribution, democratic governance in DAOs, and the integrity of social platforms. The problem intensifies as generative AI produces increasingly realistic synthetic media.

Most current PoP systems rely on a single static biometric secret, such as an iris, palm print, or face. BrightID [6] takes a different approach using social graph analysis, which depends on coordinated verification events. These designs optimize for different properties. Entros explores a third axis: whether dynamic behavior can support revocable continuity and population-level uniqueness without exposing a permanent anatomical template.

The Entros Protocol operates on a different principle. A human is a continuous, dynamic process. Voice, motion, and touch can carry behavioral patterns that change over time while

retaining useful continuity. Entros tests whether those patterns can separate returning users, different users, and adaptive synthesis across consumer devices. The protocol treats that separation as an empirical requirement, not an assumption.

Instead of asking *"What is your secret?"*, the protocol asks *"Are you still you?"*.

### 1.1 Contributions

1. A multi-modal behavioral capture protocol (the *Liveness Interlock*) that extracts a 308-dimensional feature vector from voice, motion, and touch data captured simultaneously over a configurable window.
2. A locality-sensitive hashing pipeline (*SimHash*) that produces a 256-bit behavioral fingerprint. The research program measures whether same-person and cross-person distances form a usable operating region. Operational thresholds remain private.
3. A Groth16 zero-knowledge circuit that proves two Poseidon-committed fingerprints fall within a bounded Hamming distance, without revealing either fingerprint.
4. A non-transferable on-chain identity token (the *Entros Anchor*) with a progressive Trust Score based on successful verification history and account age.
5. A graduated trust model that separates first-capture validation, returning-user continuity, and portable wallet state.
6. A working implementation deployed on Solana devnet with end-to-end browser-based verification.

### 1.2 Paper Organization

Section 2 defines the Temporal-Biometric Hash pipeline. Section 3 presents the ZK circuit and on-chain verification. Section 4 details the economic model. Section 5 describes the Entros Anchor and Trust Score. Section 6 provides formal security analysis including the graduated trust model. Section 7 surveys related work. Section 8 presents implementation status and benchmarks.

## 2 The Temporal-Biometric Hash

### 2.1 Design Objectives

The Temporal-Biometric Hash (TBH) pipeline targets five properties. The first three require population and adversarial evidence before they become protocol guarantees.

**Definition 1** (TBH Requirements)**.** *A TBH scheme is a tuple of algorithms (*Challenge, Capture, Extract, Hash, Commit*) satisfying:*

1. ***Uniqueness.*** *Fingerprints from distinct individuals have high expected Hamming distance:* $\mathbb{E}[d_H(F_A, F_B)] \approx n/2$ *for n-bit fingerprints,* $A \neq B$.
2. ***Temporal Consistency.*** *Fingerprints from the same individual across sessions have bounded distance:* $d_H(F_t, F_{t+\Delta}) \in [\delta_{\min}, \delta_{\max}]$ *with high probability.*

3. ***Spoof Resistance.*** *Generating a fingerprint $F'$ such that $d_H(F', F_{target}) < \delta_{\max}$ requires knowledge of the target's behavioral characteristics across multiple modalities.*

4. ***Privacy.*** *The client-side fingerprint $F_T$ is not transmitted. On-chain state can include Poseidon commitments, zero-knowledge proofs, and an encrypted baseline blob. The validation service receives phrase audio, derived features, bounded time-series contours, a coarse curve outline, and request metadata. It derives a separate server-side fingerprint for cross-wallet analysis (Section 6.8).*

5. ***Efficiency.*** *Client hashing and proving run on consumer hardware. Private validation runs in the server tier.*

## 2.2 Challenge Generation

The protocol issues a nonce-seeded challenge consisting of two components:

**Phonetic phrase.** A 5-word phrase drawn uniformly at random from a curated 1,357-word neutral-vocabulary English dictionary (e.g., *"elephant mountain coffee yellow bicycle"*). The vocabulary, combinatorial structure ($1357^5 \approx 4.6 \times 10^{15}$ phrases), and rationale for choosing real words over nonsense syllables are discussed in Section 2.2.1.

**Lissajous curve.** A parametric curve $\gamma(t) = (A\sin(at + \delta), B\sin(bt))$ with random parameters $a, b, \delta$. The user traces this curve on-screen, producing kinematic data shaped by involuntary motor control patterns.

### 2.2.1 Phrase Vocabulary Selection

The original protocol design (June 2025 – April 2026) specified a 70-syllable nonsense vocabulary on the theory that non-semantic phrases would (a) prevent dictionary-based audio deepfake attacks by enlarging the synthesis target space and (b) elicit distinctive prosodic variation that text-to-speech systems would struggle to reproduce. Empirical deployment in early 2026 forced a revision of both claims.

**Threat-model evolution.** The "dictionary-based deepfake" model assumes adversaries pre-synthesize libraries of TTS audio for known phrases. Real-time streaming TTS has made that pattern obsolete. As of 2026, Cartesia Sonic Turbo and ElevenLabs Flash v2.5 generate arbitrary text at $\leq$100 ms time-to-first-audio for sub-cent unit cost, and self-hosted XTTS-v2 runs at RTF 0.3$\times$ on commodity GPUs. The ASVspoof 5 benchmark [25] abandoned the pre-synthesis library attack class entirely, focusing on real-time synthesis as the empirically dominant threat. Combinatorial vocabulary size ($70^5$ vs. $1357^5$) does not affect an attacker who never needs to pre-compute.

**Prosodic discrimination is vocabulary-independent.** Modern deepfake-detection literature [26] extracts the human-vs-synth signal from cycle-level perturbation statistics. These include jitter, shimmer, harmonic-to-noise ratio, and microtremor $F_0$ measured over voiced segments. These features are physical correlates of vocal-fold biomechanics and laryngeal control, independent of the lexical identity of the spoken content. The same prosodic asymmetry that distinguishes a human from a synthesizer on "elephant mountain coffee" also distinguishes them on "ba le fa ki te". Choosing nonsense provides no incremental discrimination on the prosodic axis the literature treats as load-bearing.

**ASR accuracy is vocabulary-dependent and asymmetric in the defender's favor.** The protocol's content-binding check requires the validation server to verify that the audio

matches the issued phrase. Both Whisper [27] and Wav2Vec2-Phoneme [28] exhibit substantially higher error rates on out-of-distribution input than on natural language. Whisper's autoregressive decoder hallucinates training-corpus filler such as "Thanks for watching" on nonsense input. Tests observed an approximate 30 % false-reject rate on clean human speech of nonsense syllables. Wav2Vec2-Phoneme operates in the right primitive through CTC forced alignment, but its baseline phoneme error rate compounds against the per-phoneme matching metric. This yields a discrimination gap of only 10–15 percentage points between right and wrong content, which is too narrow to threshold reliably. On real English words, Whisper-tiny.en operates in its training distribution with WER $\approx$ 5–6 % on LibriSpeech test-clean, and word-level Levenshtein on a curated dictionary gives a discrete signal whose collision probability between two random 5-word phrases is $< 0.1$ %.

**Curated real-word vocabulary.** The shipped implementation uses a 1,357-word neutral-vocabulary English dictionary curated by length (4–8 letters), syllable count (1–3), VADER-positive sentiment, hand-blocklist content safety filters, and homophone/substring-collision pruning. The $1357^5 \approx 4.6 \times 10^{15}$ challenge space preserves per-session unpredictability and content binding. It does not stop real-time synthesis. The shipped Whisper-tiny.en and word-level Levenshtein pipeline produced a 95 percentage-point separation between right-phrase and wrong-content calibration samples. This measurement covers phrase matching, not human-versus-synthetic classification.

## 2.3 Multi-Modal Data Acquisition

Three sensor streams are captured simultaneously over a configurable window (default: 12 seconds):

- $S_{\text{audio}}$: Microphone input, band-limited and decimated to a canonical 16 kHz on the device, capturing voice prosody.
- $S_{\text{motion}}$: IMU accelerometer/gyroscope at 60–100 Hz on mobile; mouse pointer dynamics on desktop.
- $S_{\text{touch}}$: Pointer/touch events including coordinates, pressure, and contact area from the digitizer.

## 2.4 Feature Extraction

Raw time-series data is distilled into a 308-dimensional feature vector $\mathbf{v} \in \mathbb{R}^{308}$ through three parallel pipelines.

### 2.4.1 Speaker Features ($\mathbf{v}_{\text{audio}} \in \mathbb{R}^{170}$)

**Pre-processing.** Audio is captured at 16 kHz and normalized to RMS 0.05 at the SDK source so amplitude features and downstream spectral features are stable across mic gains and device hardware. A pre-emphasis filter with coefficient 0.97 is applied before MFCC framing.

**Fundamental frequency, perturbation, and LTAS (44 features).** $F_0$ statistics and delta, jitter measures (local, RAP, PPQ5, DDP), shimmer measures (local, APQ3, APQ5, DDA), harmonics-to-noise ratio (HNR), voicing ratio, amplitude moments, and Long-Term Average

Spectrum (LTAS) statistics (spectral centroid, rolloff, flatness, spread). These features measure pitch behavior, cycle-level perturbation, noise, and spectral shape. Their discrimination against current synthesis systems is an empirical property of the complete validator.

**MFCCs and delta-MFCCs (72 features).** Mel-Frequency Cepstral Coefficients capture vocal-tract spectral envelope shape across mel-warped frequency bands. Per-coefficient statistics (mean, variance, skewness, kurtosis) over the capture window are computed for the 12 cepstral coefficients $C_1$–$C_{12}$, plus their temporal first derivatives over the same window. MFCCs and delta-MFCCs are the dominant speaker-discrimination signal in modern speech-recognition literature.

**LPC coefficient statistics (24 features).** Linear predictive coding [9] reduces each frame to coefficients describing the vocal tract's all-pole filter. Per-coefficient mean and variance over 12 LPC coefficients capture the acoustic resonance system independent of the spoken content.

**Formant trajectories (16 features).** $F_1 / F_2 / F_3$ absolute formant frequencies and their per-trajectory derivatives, plus selected formant bandwidths. Absolute formant frequencies carry vocal-tract length information directly tied to the speaker's anatomy.

**Voice quality (9 features).** Cepstral peak prominence (CPP) restricted to the $F_0$ quefrency band, spectral tilt, H1-H2 harmonic ratio, and sub-band energy ratios. These features distinguish breathy, creaky, and modal phonation modes which are stable per speaker.

**Pitch contour shape (5 features).** Discrete cosine transform coefficients of the $F_0$ contour over voiced regions encode prosodic curve shape independent of absolute pitch.

#### 2.4.2 Kinematic Features ($\mathbf{v_{kin}} \in \mathbb{R}^{81}$)

**Jerk, jounce, and path dynamics (54 features).** The third (jerk) and fourth (jounce) time derivatives of pointer coordinates, plus path curvature, directional entropy, speed and acceleration profiles, micro-correction frequency, pause ratios, path efficiency, segment length distribution, speed jitter variance, normalized path length, and angle autocorrelation. These features describe motor-control patterns that the validator evaluates alongside the other modalities [10].

**FFT band energies (12 features).** Cooley-Tukey FFT band energy across four frequency bands (0–2, 2–6, 6–12, 12–30 Hz) for each of three accelerometer axes.

**Physiological tremor peak (2 features).** Peak frequency and amplitude in the 4–12 Hz band of the motion magnitude—the band where physiological hand tremor concentrates.

**Cross-axis covariance (6 features).** Pairwise covariance for six selected IMU axis pairs captures coupled motor patterns that uncoupled per-axis features miss.

**Direction-reversal and angular dynamics (7 features).** Per-axis direction-reversal rate (mean and variance per accelerometer axis), mean angular velocity, and magnitude autocorrelation at lags 1, 5, 10, 25.

#### 2.4.3 Touch Features ($\mathbf{v_{touch}} \in \mathbb{R}^{57}$)

**Velocity, acceleration, pressure, contact area (36 features).** Touch coordinate velocity and acceleration, pressure statistics, contact area statistics, path jerk, and per-signal jitter variance. These features reflect fine motor control patterns of the fingertip or stylus.

**Pressure dynamics (4 features).** Pressure first-derivative mean, variance, skewness, kurtosis capture how the user's grip changes through a stroke.

**Contact and area dynamics (4 features).** Contact aspect ratio mean and variance, area first-derivative mean and variance.

**Path geometry (3 features).** Trajectory curvature mean, variance, and skewness over voiced motion frames.

**Velocity autocorrelation (3 features).** Touch velocity autocorrelation at lags 1, 3, 5.

**Inter-touch and stroke statistics (7 features).** Inter-touch gap mean, variance, skew, kurtosis (4), path efficiency (1), per-stroke total path length mean and variance (2).

### 2.5 Feature Fusion and SimHash

**Normalization.** The client normalizes each modality group before computing its local fingerprint. The validation service independently applies bounded normalization, modality weighting, and optional population whitening before its cross-wallet projection. Both paths sanitize non-finite values. The server parameters remain private because they form part of the active detection policy.

**Concatenation.** The three vectors are concatenated: $\mathbf{v}_{\text{fused}} = [\mathbf{v}_{\text{audio}} \| \mathbf{v}_{\text{kin}} \| \mathbf{v}_{\text{touch}}] \in \mathbb{R}^{308}$.

**SimHash [2].** The fused vector is projected onto 256 deterministic random hyperplanes $\{\mathbf{h}_1, \ldots, \mathbf{h}_{256}\}$. The Temporal Fingerprint is:

$$F_T[i] = \begin{cases} 1 & \text{if } \mathbf{v}_{\text{fused}} \cdot \mathbf{h}_i \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

for $i \in \{1, \ldots, 256\}$. By the properties of SimHash, $\Pr[F_T^{(A)}[i] \neq F_T^{(B)}[i]] = \frac{1}{\pi} \arccos\left(\frac{\mathbf{v}_A \cdot \mathbf{v}_B}{\|\mathbf{v}_A\| \|\mathbf{v}_B\|}\right)$, so similar feature vectors produce fingerprints with small Hamming distance.

### 2.6 Relationship to Cancelable Biometrics and Fuzzy Extractors

The problem of protecting biometric templates while enabling matching has a substantial academic history. *Fuzzy extractors* [14] derive cryptographic keys from noisy biometric inputs by correcting errors within a tolerance threshold. *Cancelable biometrics* [15] apply non-invertible transforms to biometric templates so that a compromised template can be revoked and replaced. Both approaches assume the biometric signal is fundamentally *static*—the same fingerprint or iris captured repeatedly with sensor noise.

Entros's behavioral biometrics present a different challenge. The signal is *inherently nonstationary*: voice prosody shifts with health, touch dynamics change with device, kinematic patterns evolve with habit. The "errors" between sessions are not noise to be corrected but genuine temporal variation that carries identity information. Fuzzy extractors' error-correction model does not apply because the variation is structured, not random. Cancelable transforms do not apply because the template itself drifts by design.

Entros uses SimHash to map vector similarity into Hamming distance, then uses Groth16 to prove that the distance falls inside a configured range. Population evaluation must determine whether real behavioral captures occupy a useful range. A user can rotate the Poseidon salt and commitment without changing the underlying fingerprint. Recent work on practical fuzzy extractors for iris biometrics [16] achieves 105 bits of entropy with a 92% true accept rate, providing a useful benchmark. Entros's 256-bit SimHash must be evaluated by effective entropy under adversarial feature distributions, not by bit length.

### 2.7 Poseidon Commitment

The fingerprint $F_T$ is private. A public commitment is computed as:

$$H_{\text{TBH}} = \mathsf{Poseidon}(\mathsf{pack}_{\text{lo}}(F_T), \mathsf{pack}_{\text{hi}}(F_T), s) \tag{2}$$

where $s$ is a 248-bit random salt, and $\mathsf{pack}_{\text{lo/hi}}$ pack the 256 bits into two BN254 field elements. The Poseidon hash [3] is chosen for ZK circuit efficiency ($\sim$300 R1CS constraints per hash vs. $\sim$25,000 for SHA-256).

## 3 ZK Self-Proof: Verification without Disclosure

### 3.1 Circuit Definition

The Hamming distance circuit is a Groth16 [4] arithmetic circuit over BN254 with $\sim$2,010 R1CS constraints. It proves three statements simultaneously:

1. $\mathsf{Poseidon}(\mathsf{pack}(F_T^{\text{new}}), s^{\text{new}}) = c^{\text{new}}$
2. $\mathsf{Poseidon}(\mathsf{pack}(F_T^{\text{prev}}), s^{\text{prev}}) = c^{\text{prev}}$
3. $\delta_{\min} \leq d_H(F_T^{\text{new}}, F_T^{\text{prev}}) < \delta_{\max}$

**Public inputs:** $c^{\text{new}}, c^{\text{prev}}, \delta_{\max}, \delta_{\min}$
**Private witnesses:** $F_T^{\text{new}}[256], F_T^{\text{prev}}[256], s^{\text{new}}, s^{\text{prev}}$

The Hamming distance is computed via bitwise XOR and popcount, expressed as R1CS constraints: for each bit position $i$, $d_i = F_T^{\text{new}}[i] + F_T^{\text{prev}}[i] - 2 \cdot F_T^{\text{new}}[i] \cdot F_T^{\text{prev}}[i]$, then $d_H = \sum_{i=1}^{256} d_i$.

Clients default to $\delta_{\min} = 3$ and $\delta_{\max} = 96$. The verifier program enforces $\delta_{\min} \geq 3$ and $\delta_{\max} \leq 96$ on submitted proofs. The circuit accepts distances in $[\delta_{\min}, \delta_{\max})$. Population evaluation must establish the false-match and false-reject behavior of the configured defaults.

**Soundness guarantees.** Groth16 provides computational knowledge soundness under the Generic Group Model and the $q$-Power Knowledge of Exponent assumption [4]. For Entros's circuit, this means no probabilistic polynomial-time adversary can produce a valid proof for a false statement, except with negligible probability. The current single-contributor Phase 2 setup adds a development trust assumption. A completed multi-party ceremony would reduce that risk to collusion among all ceremony contributors.

### 3.2 On-Chain Verification

Proof generation runs client-side using snarkjs (WASM). The proof is serialized into 256 bytes with 4 public inputs (32 bytes each).

On-chain verification uses the `groth16-solana` crate, implementing the BN254 pairing check within Solana's compute budget (<200K compute units). The verification program:

1. Validates a challenge nonce (single-use, time-limited to 5 minutes)
2. Executes the Groth16 pairing check
3. If valid: creates a `VerificationResult` PDA as an audit trail
4. If invalid: reverts the entire transaction (challenge nonce preserved for retry)

### 3.3 Trusted Setup

Groth16 requires a structured reference string (SRS) from a trusted setup ceremony. Phase 1 uses the Hermez community Powers of Tau, a multi-contributor circuit-agnostic setup. Phase 2 currently has a single contributor. A multi-party computation ceremony with independent contributors will precede mainnet deployment. One honest contributor is sufficient for soundness [12].

## 4 Economic Model

### 4.1 The Entros Token

The planned economic layer uses a native utility token, a standard SPL mint launched on a public bonding curve. The identity Anchor described in Section 5 is a separate Token-2022 mint carrying the NonTransferable extension [13]. The two assets use different standards and are not connected on-chain today.

1. **Staking.** Validators stake Entros tokens as slashable collateral to participate in the Anonymity Ring.
2. **Delegation.** Holders who do not run a node delegate stake to a validator, sharing the accuracy-weighted rewards and the slashing risk.
3. **Verification capacity.** Integrators stake Entros for discounted or unlimited verifications via capacity tiers.
4. **Economic governance.** Holders direct treasury allocation, the verification fee, validator admission policy, and ecosystem funding. Detection parameters remain outside token voting. A governed calibration process can version safe public changes without publishing exploitable thresholds.

None of these token mechanisms are active in the current devnet deployment. The deployed programs charge a configurable SOL verification fee. The registry also contains SOL deposit scaffolding, but it does not select validators, assign work, slash deposits, or distribute rewards.

### 4.2 User-Pays Model

Wallet-connected users pay the configurable protocol fee for each verification. Program initialization uses a 0.005 SOL default, and the authority can update it. Integrators can read on-chain Anchor state without a protocol billing relationship. The fee bounds attempt volume and does not determine whether a capture passes.

### 4.3 Validation Cycle

The protocol fee from each verification is collected into the protocol treasury on-chain. As the validator network decentralizes, a share of fees will be distributed to validators in proportion to validation accuracy. Independent ground-truth benchmarks will measure that accuracy instead of verification throughput.

### 4.4 Slashing

The design specifies a probabilistic audit mechanism: a configurable fraction of validations is re-scored against independent ground-truth benchmarks, with divergence from the audited outcome resulting in slashing. The audit is benchmarked against ground truth rather than inter-validator agreement by design, so that correlated drift—the whole set relaxing its threshold together under sustained load—is detectable, not just an individual outlier. Reward is weighted by audited accuracy, not verification throughput. This mechanism is specified but not yet implemented in the current devnet deployment.

## 5 The Entros Anchor

### 5.1 Non-Transferable Identity Token

The Entros Anchor is implemented using SPL Token-2022 with the `NonTransferable` mint extension. Each wallet maps to exactly one Anchor via a Program Derived Address (PDA).

The on-chain data structure stores: `owner` (Pubkey), `creation_timestamp` (i64), `last_verification_timestamp` (i64), `verification_count` (u32), `trust_score` (u16), `current_commitment` ([u8;32]), and a rolling window of the 52 most recent verification timestamps for Trust Score computation.

### 5.2 Progressive Trust Score

The Trust Score is a function of verification span and account age, not raw verification count. The program divides the previous 84 days into twelve seven-day bins. A week contributes once when it contains at least one successful verification. The formula combines two active components and one deprecated component:

**Weekly span score.** Each active week receives `base_trust_increment` $\times\ (12 - k)/12$, where $k = 0$ is the current week. More recent weeks receive more weight. Multiple verifications inside one week do not increase that week's contribution.

**Regularity bonus (deprecated).** The regularity bonus is set to zero. The legacy model rewarded low gap variance, which paid for machine scheduling precision. Deprecating it removes this automated script reward.

**Age bonus.** $\lfloor\sqrt{\min(\text{age\_days}, 365)}\rfloor \times 4$, using deterministic integer square root. This rewards account longevity without rewarding high-frequency checks.

The score is capped at a configurable maximum (currently 10,000) and computed on-chain during the `update_anchor` instruction, reading parameters from a cross-program PDA.

### 5.3 Verification Modes

Wallet-connected mode is the primary flow. A first verification requires a validator-signed receipt before the program mints an Entros Anchor. A returning verification supplies a Groth16 continuity proof before the program updates the Anchor and Trust Score. SAS issuance is a separate best-effort step when the executor and credential authority are configured.

Walletless mode is a secondary SDK and relayer design for applications that do not onboard a user wallet. The behavioral fingerprint remains in a device-local encrypted baseline.

Clearing storage resets that continuity. This mode does not create a portable user-owned Anchor or Trust Score. The reference application does not currently offer it. Its first-capture decision remains an open measurement question.

# 6 Security Analysis

## 6.1 Threat Model

**Definition 2** (Adversary)**.** *We consider a computationally-bounded adversary $\mathcal{A}$ with the following capabilities:*

1. *$\mathcal{A}$ has full access to the protocol source code, circuit definitions, and feature extraction pipeline (open source).*
2. *$\mathcal{A}$ can generate arbitrary synthetic sensor data (audio, motion, touch) and submit it through the Pulse SDK.*
3. *$\mathcal{A}$ can create arbitrary Solana wallets and fund them with SOL.*
4. *$\mathcal{A}$ cannot break the discrete logarithm assumption on BN254, the collision resistance of Poseidon, or the knowledge soundness of Groth16.*
5. *$\mathcal{A}$ cannot access another user's device storage (no physical access to encrypted fingerprints).*

## 6.2 Replay Attacks

**Theorem 1** (Replay Resistance)**.** *An adversary replaying a previously-captured fingerprint $F_T$ verbatim is rejected except with negligible probability, under the knowledge soundness of Groth16.*

*Proof.* A replayed fingerprint produces $d_H(F_T, F_T) = 0 < \delta_{\min} = 3$. The circuit outputs `false` for the range check $\delta_{\min} \leq d_H < \delta_{\max}$. By the knowledge soundness of Groth16, no valid proof exists for a false statement. The on-chain verifier rejects the transaction. The program consumes a challenge nonce after a successful verification and rejects later reuse. A failed transaction preserves the nonce for a legitimate retry until its 5-minute expiry. □

## 6.3 Synthetic Data Attacks

**Claim 1** (Coordinated Synthesis Research Question)**.** *Can the complete validation stack distinguish coordinated synthetic evidence from genuine captures across supported devices and users?*

The defense is layered:

**Feature-level.** The 308-dimensional feature vector summarizes voice, motion, and touch behavior. The private validator evaluates its distributions and relationships against the active policy. No feature count or individual statistic establishes synthesis resistance by itself.

**Joint projection.** SimHash projects the concatenated vector onto shared hyperplanes, so each output bit depends on features from all modalities at once and no single modality determines the fingerprint alone. The projection mixes the modalities. It does not test a relationship between them, and it is not the layer that establishes one.

**Entropy scoring.** The extraction pipeline measures Shannon entropy and jitter variance per sensor stream. Synthetic data with low or uniform entropy is flagged before reaching the hashing stage.

Entros treats synthesis resistance as an empirical property of the complete validation stack. The red-team program measures each attack class against the deployed configuration. It does not derive a joint success rate from assumed per-modality probabilities.

**Empirical context.** Serwadda and Phoha [17] found that spoofing mouse dynamics required per-target training and achieved limited success. Their result supports behavioral dynamics as one component of a measured, multi-signal validation stack. Entros tests coordinated synthesis against the complete stack rather than estimating protection from independent modality probabilities.

Pouw et al. [23] observed alignment between vocal amplitude and upper-limb kinetics in a controlled sustained-vocalization study. That result motivates future event-locked research. It does not validate the current lagged $F_0$-to-acceleration statistic, which remains research telemetry.

**Voice modality evidence.** Entros extracts jitter, shimmer, HNR, spectral, and contour features rather than relying only on voice timbre. ASVspoof research [11] motivates those measurements. Current TTS families and recording conditions can change their value, so Entros measures voice ablations and coordinated attacks against the full stack. Behavioral biometric systems using ZK verification have reported false accept rates below 1% in their own settings [18]. Those results do not establish an Entros rate.

### 6.4 Sybil Attacks

Each wallet maps to one Entros Anchor through PDA derivation. This rule prevents duplicate Anchors for one wallet. It does not establish that different wallets belong to different people.

The active detection layer compares a new server-derived fingerprint with recent fingerprints registered to other wallets. Similarity can trigger the configured duplicate-identity policy. The population operating curve remains under evaluation, so Entros does not present that comparison as a completed uniqueness guarantee.

Trust Score rewards successful verification across weekly time bins and account age. The configurable SOL fee bounds the volume of attempts that reach those checks. Neither mechanism replaces duplicate-identity detection.

#### 6.4.1 Layered Sybil Resistance

Entros combines three controls. Behavioral comparison addresses duplicate-wallet similarity. History and fees shape how applications consume that signal:

- **Behavioral comparison.** The private validator compares each new fingerprint with bounded recent fingerprints associated with other wallets. The population evaluation measures false matches, missed duplicates, device effects, and threshold behavior.
- **Verification history.** Trust Score rewards successful verification across distinct weekly bins. Frequent checks inside one week do not increase that week's contribution.
- **Attempt-volume bound.** The configured SOL fee and per-wallet limits constrain request volume. They do not decide whether a capture passes.

These controls are complementary. The cross-wallet comparison remains the direct uniqueness mechanism. Its effectiveness depends on the measured same-person and cross-person distributions. The architecture supports updated projections and risk policies as that evidence grows.

### 6.5 Privacy

**Theorem 2** (Zero-Knowledge Privacy)**.** *The on-chain verifier learns only that the Hamming distance between two fingerprints falls within* $[\delta_{\min}, \delta_{\max})$*. It learns neither fingerprint, neither salt, nor any feature vector.*

*Proof.* Groth16 reveals no private witness beyond the public statement. The public inputs are the two Poseidon commitments and the distance bounds. The fingerprints and salts remain private witnesses. Commitment privacy depends on Poseidon preimage resistance and secret, high-entropy salts. □

The client discards raw motion and touch after feature extraction. The validation service processes phrase audio for transcription and then discards it. On-chain state includes commitments, proofs, and encrypted baseline material.

**SimHash reversibility.** Recent work has demonstrated pre-image attacks on locality-sensitive hashes [21]. This shows that SimHash fingerprints can contain recoverable information about their inputs. Entros does not publish the plaintext fingerprint. The protocol publishes its Poseidon commitment and a proof that reveals only the Hamming distance range. The fingerprint stored locally for re-verification uses AES-256-GCM with a non-extractable `CryptoKey` in IndexedDB. In wallet-connected mode, a per-wallet PDA stores another encrypted copy. Its key derives from a domain-separated Ed25519 `signMessage` signature. The blob contains a 32-byte fingerprint and a 32-byte commitment salt. It contains neither raw sensor data nor the 308 statistical features. Wallet compromise can expose the fingerprint and salt, so Entros does not treat SimHash as a privacy-preserving representation. The Poseidon commitment provides that boundary. AES-256 remains quantum-resistant under standard assumptions about Grover's algorithm. The Ed25519-derived key path shares Solana's post-quantum migration concern.

### 6.6 Attack Cost Under the Detection Stack

The protocol does not claim to make spoofing impossible. The primary defense is detection: every capture is scored against the detection layers described above before it reaches the chain. Economic cost is a secondary layer that bounds the volume of attempts an adversary can mount, not the mechanism that decides whether any single capture passes. The defense is layered:

- **Feature-level:** The private validator scores the 308 derived features and bounded same-window signals against measured policy.
- **Circuit-level:** The circuit rejects exact replays and fingerprint pairs outside the configured Hamming interval.
- **Entropy scoring:** Low-entropy synthetic data is flagged before hashing.

- **Economic:** Each verification costs SOL and each wallet requires funding, which bounds attempt volume. Trust Score rewards months of consistency over bursts.

### 6.7 Graduated Trust Model

First-time verification establishes a behavioral baseline after the private validator accepts the capture and signs a mint receipt. With no prior fingerprint, the Hamming distance circuit does not run. This creates a first-capture policy result, not a cryptographic proof of sensor provenance or human presence.

Temporal consistency applies from the second verification onward. Each returning session checks behavioral drift against the stored fingerprint.

**Definition 3** (Trust Tiers)**.** *The protocol defines three trust tiers based on verification history:*

1. ***First-capture validation*** *(first verification). The server policy accepted one capture. No continuity proof exists yet. A wallet-connected mint also requires a validator-signed receipt.*
2. ***Device-bound continuity*** *(returning walletless verification). A Groth16 proof places the new fingerprint inside the configured distance interval from a device-local baseline.*
3. ***Portable wallet continuity*** *(wallet-connected verification). An on-chain Anchor records accepted verification history, the current commitment, and Trust Score.*

Tier 2 uses walletless mode, which the protocol defines but the reference application does not offer. The live wallet-connected flow starts at Tier 1 and adds Tier 3 continuity on return.

In wallet-connected mode the user pays the per-verification fee, so each Anchor carries a recurring cost to its holder, and integrators read on-chain state for free.

The Anchor exposes Trust Score and `last_verification_timestamp`. Trust Score describes accepted verification history. The timestamp describes recency. An integrator can require a fresh verification before a gated action and then read both fields. The result remains a protocol policy signal rather than authenticated sensor provenance.

Clearing local storage removes walletless continuity. Applications that require returning history can reject a first-capture result or require the wallet-connected tier.

### 6.8 Browser Trust Model and Server-Side Validation

The browser performs sensor capture, feature extraction, SimHash computation, Poseidon commitment, and Groth16 proof generation. It keeps an encrypted baseline locally. The validation service processes phrase audio and the statistical summary. The protocol persists commitments, proofs, encrypted baseline material, and account state. The browser remains an untrusted execution environment. An adversary controlling it can override sensor APIs, manipulate feature extraction, or submit pre-computed proofs from synthetic data. The deployed browser tier combines automation-framework signals, challenge-response phrase checks, private statistical validation, and cross-wallet registry analysis. Planned progressive server challenges, WebAuthn request binding, and native attestation will strengthen client integrity and limit pre-composed evidence.

The ZK proof provides a deterministic guarantee: the Hamming distance either falls within $[\delta_{\min}, \delta_{\max})$ or the proof is invalid. This is necessary but not sufficient. A valid proof

confirms the *mathematical relationship* between two fingerprints but cannot confirm the *provenance* of the underlying sensor data.

The protocol implements a two-level validation architecture:

**Level 1 (client-side, deterministic).** The Groth16 proof, as currently implemented. Provides mathematical certainty that the Hamming distance constraint is satisfied.

**Level 2 (server-side, statistical).** Before proof generation, the client transmits the 308-dimensional feature vector, phrase audio, bounded contours, curve outline, and request metadata. The server applies private statistical models and challenge-response checks. Its same-window analysis runs over client-supplied signals. It provides statistical evidence rather than sensor provenance.

The feature vector contains fixed-size statistics rather than raw motion or full-resolution touch streams. The request also carries bounded $F_0$ and acceleration contours for same-window analysis. The ZK proof hides the client fingerprint. These inputs support statistical validation but do not authenticate sensor origin.

The feature vector could alternatively be processed within a Trusted Execution Environment (TEE) where even the server operator cannot inspect individual feature vectors, providing an additional privacy guarantee for high-sensitivity deployments.

### 6.9 Device and App Integrity

Browser-based sensor APIs do not authenticate the physical source of submitted data. Entros therefore treats browser verification as a risk-scored tier and is researching stronger request binding for native clients.

App Attest and Play Integrity can return vendor-signed evidence about a recognized app instance and device environment. The planned native tier will bind that evidence to a server nonce and the final request digest, then verify freshness and replay state server-side. This strengthens confidence that an approved app instance authorized the request. It does not by itself prove sensor origin, human presence, or uniqueness.

WebAuthn can add registered-credential authorization and continuity evidence to browser submissions. Progressive server challenges can also limit evidence prepared before a session. Entros evaluates these controls as separate layers because each establishes a different property.

The native application also provides direct access to IMU and touch capabilities that browsers restrict. Those signals support behavioral validation without turning device attestation into a personhood claim.

### 6.10 Empirical Adversarial Validation

The protocol evaluates defined attack classes through a continuous internal red-team program. The harness submits synthesized evidence to the deployed validation service and records denominators, observed pass rates, and enforcement state. A validator rejection stops the reference flow before on-chain submission. Results apply only to the tested attack class and configuration.

The attack taxonomy spans eight tiers ordered by sophistication. Results for the first three tiers (the highest-priority attacks implementable without external TTS models) are summarized in Table 1.

| Tier | Attack class | Attempts | Tier 1 pass rate |
|---|---|---|---|
| T1 | Procedural synthesis | 2,000 | 0% |
| T2 | Multi-strategy parameter variation | 4,000 | 0% |
| T3a | Unconstrained feature optimization | 1,000 | 0% |
| T3b | Constrained feature optimization | 9,000 | 0% |

Table 1: Red team campaign results against live Tier 1 enforcement.

T1 exercises trivial procedural synthesis (sine-wave harmonics with additive noise). T2 extends this with four waveform strategies (harmonic, sawtooth, filtered noise, pulse train), four motion patterns (tremor, Brownian, circular, static), and parameter sampling across the full human voice range (80–350 Hz fundamental). T3 reconstructs the SimHash hyperplanes from the open-source SDK constants and uses hill-climbing optimization to craft 308-dimensional feature vectors targeting valid fingerprint distances, optionally constrained to published human feature ranges.

T3b applies distributional constraints derived from published voice science norms, maintaining physiologically plausible feature values throughout the optimization. Inter-feature consistency checks (e.g., perturbation measure ratios inherent to vocal fold mechanics) prevent the optimizer from producing individually plausible but structurally impossible feature combinations.

T4 extends the program to modern neural voice synthesis, which the tiers above deliberately exclude. T4a paired pre-recorded human voice with procedural motion and moved from a 100% pass counterfactual to 0% across four progressive defense waves, closing at 1,000 attempts with a 95% confidence interval of [0%, 0.37%]. T4b synthesized the issued challenge phrase in real time across two TTS model families and 58 voices. None of 200 attempts cleared the tested full-stack configuration. T5 remains open. T6–T8 remain planned and do not start before T5 meets its closure criteria. Aggregate results are published at `https://entros.io/security`. Attack implementation code remains private.

## 7 Related Work

**Worldcoin** [5] uses iris scanning to create a unique biometric identifier per person. The approach provides strong uniqueness guarantees through a dedicated hardware device (the Orb), which enforces a controlled capture environment. The tradeoff is a permanent anatomical template: because an iris scan cannot be changed, it cannot be revoked if the template is ever exposed. Entros's behavioral signature drifts naturally over time, making re-verification both the consistency check and the revocation mechanism.

**BrightID** [6] verifies uniqueness through social graph analysis, where users vouch for each other in verification parties. The approach trades dedicated hardware for coordination overhead. Entros targets verification on a consumer device without a coordinated social event.

**Reclaim Protocol** [7] proves ownership of existing web2 accounts via TLS session proofs. It answers "do you control this account?" Entros targets a complementary behavioral personhood signal.

**Traditional CAPTCHA** (reCAPTCHA [8], hCaptcha, Turnstile) provides session-level bot

detection using behavioral signals, browser fingerprinting, and centralized classifiers. Entros is researching whether a private behavioral baseline and portable wallet history can add continuity to that session-level model. The current first-capture path has not established a CAPTCHA-equivalent operating curve.

**VeryAI** uses palm print biometrics with on-device processing. Its palm print is a static anatomical identifier. Entros targets a complementary form of returning-user continuity through behavioral change across sessions.

**Behavioral biometrics with ZK proofs.** Hamm et al. [18] demonstrate continuous authentication using interactive and non-interactive ZK proofs over behavioral features, achieving a false accept rate of 0.65% and false reject rate of 0.48%. Their system validates that ZK verification of behavioral biometrics is practical, though their architecture targets session-level continuous authentication rather than cross-session identity persistence. Multi-modal fusion approaches for behavioral authentication [20] confirm that combining touch, keystroke, and accelerometer data improves both accuracy and spoofing resistance over single-modality systems.

**Formal frameworks for proof of personhood.** Choudhuri et al. [19] provide a cryptographic formalization of proof of personhood, defining ideal functionalities for Sybil resistance, authenticated personhood, and unlinkability. Their framework assumes trusted authorities issue personhood credentials. Entros is testing whether behavioral evidence can support a personhood credential without an anatomical issuer. Mapping each Entros claim to this framework remains future work.

**Data-handling boundary.** Raw motion and full-resolution touch remain on the device. The validation service processes phrase audio, derived features, bounded contours, a coarse curve outline, and request metadata. Persistent protocol state can include commitments, proofs, encrypted baseline material, and account data. Deployers must assess the biometric-data rules that apply to their jurisdiction and use case.

# 8 Implementation and Benchmarks

The devnet implementation has three Anchor programs, a Groth16 circuit, a published TypeScript SDK, a private Rust validation service, a Rust gateway and relayer, and a hosted wallet-connected application. A separate voter-weight addin is deployed on devnet as an on-chain prototype. The walletless API exists in the SDK and relayer, but the reference application does not offer that mode. Repository test suites and CI gates cover each component. Exact counts remain in the versioned test output instead of this paper.

The protocol fee treasury is live on devnet. Program initialization uses a 0.005 SOL default, and the authority can update the value. The transaction transfers the configured fee atomically with a successful mint or update. Treasury state is publicly auditable on Solana Explorer.

The devnet programs remain upgradeable under the current development authority. The mainnet plan adds hardware-backed custody, an observable upgrade delay, and an independent recovery path before external value depends on that authority.

## 8.1 Performance Benchmarks

Benchmarks measured on Chrome 132 (M1 MacBook Pro) and Safari (iPhone 15 Pro Max):

- Behavioral capture: 7,000–12,000 ms (configurable)
- Feature extraction (308 dimensions): ∼45 ms
- SimHash (256-bit): <1 ms
- Poseidon commitment: ∼3 ms
- Groth16 proof generation (WASM): ∼850 ms
- On-chain verification: ∼123K compute units
- **Total (excluding capture): ∼900 ms**

The total pipeline from button click to on-chain proof depends on the configured capture window and on network confirmation, plus ∼900 ms of computation. On mobile (iPhone 15 Pro Max, Safari), all three sensor streams (audio, IMU motion, touch) capture simultaneously. Safari returns the hardware's native 48 kHz whatever rate the page requests. The SDK band-limits and decimates any capture at or above that rate to a canonical 16 kHz before feature extraction, so the browser's own resampler stops influencing the fingerprint. Hardware that delivers below 16 kHz, such as a narrowband Bluetooth headset, is passed through at its native rate rather than upsampled, since upsampling would invent detail the microphone never captured. Proof generation completes within the same time budget via snarkjs WASM.

**Comparative context.** Groth16 proof generation at ∼850 ms compares favorably to PLONK-based systems, which require ∼2.5 s for equivalent circuit sizes [22]. On-chain verification at ∼123K compute units fits comfortably within Solana's 200K default budget; PLONK verification would exceed it. Poseidon commitment at ∼3 ms reflects the hash's ZK-optimized design (∼300 R1CS constraints vs. ∼25,000 for SHA-256 in-circuit [3]).

### 8.2 Desktop vs. Mobile Verification

Desktop verification operates with reduced sensor modalities. Mouse pointer dynamics serve as a proxy for hand movement, but capture wrist and finger motion rather than the arm and trunk movement available via mobile accelerometers. No touch pressure data is available from standard mice or trackpads. The effective dimensionality of the behavioral fingerprint is lower on desktop.

Published research quantifies the gap. Multi-modal touch and IMU fusion on mobile devices reports EER below 1% [24]. Desktop-only behavioral authentication (keystroke dynamics and mouse movement) reports EER in the 6–13% range across multiple studies. The difference reflects the richer sensor environment available on mobile: accelerometer, gyroscope, magnetometer, and capacitive touch digitizer with pressure sensitivity.

The current web flow accepts desktop captures with fewer sensor capabilities. A returning desktop verification can still produce a fingerprint and ZK continuity proof. The validator records the device context and applies its configured policy. Population evaluation must measure desktop and mobile distributions separately before assigning a comparative assurance level.

The native mobile application is the planned higher-assurance client and targets the Solana dApp Store. It provides direct IMU, microphone, touch, native proving, and Mobile Wallet Adapter integration. Representative-device testing and request-bound platform attestation remain release work.

# 9 Conclusion and Future Work

The Entros Protocol presents a framework for Proof-of-Personhood through temporal behavioral continuity. It combines private validation, committed behavioral fingerprints, zero-knowledge continuity proofs, and portable wallet history. Population separation and first-capture liveness remain measured protocol requirements.

The cryptographic proof establishes a bounded relationship between two committed fingerprints. The private validator decides whether a submitted capture satisfies the active statistical policy. First verification relies on that policy and a signed receipt. Returning verification adds the continuity proof. Fees bound request volume after detection.

**Future work:**

- Multi-contributor trusted setup ceremony for Groth16 Phase 2 before mainnet.
- External security audit of all on-chain programs, the ZK circuit, and the executor node.
- Full on-chain token economics: validator staking, delegation, capacity tiers, and economic governance wired to the utility token.
- Cross-chain reads of the Solana attestation from other chains, via an attestation relay rather than a protocol redeployment.
- Formal analysis of SimHash collision probability bounds under adversarial feature distributions.
- Cross-wallet fingerprint comparison is implemented in the private validation service. The registry keeps bounded recent server-derived fingerprints and compares a new wallet against other wallets. Population evaluation must establish an operating curve before this becomes a uniqueness claim.
- Server-side feature validation is implemented as described in Section 6.8. The client and on-chain protocol are public. Validation models, thresholds, and active detection policy remain private.
- Adversarial testing continues against defined attack classes. Entros publishes aggregate denominators and observed outcomes after each safe, completed campaign. T5 remains open, and T6 remains blocked on its closure criteria.
- Request-bound native attestation. Research App Attest and Play Integrity as stronger app and device integrity evidence for the native tier.

The client SDK, circuit definitions, and on-chain programs are open source and published as a defensive disclosure. The private validation and red-team repositories remain closed as defense-in-depth.

# References

[1] J. R. Douceur, "The Sybil Attack," in *Proc. IPTPS*, 2002.

[2] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proc. STOC*, 2002.

[3] L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, and M. Schofnegger, "Poseidon: A new hash function for zero-knowledge proof systems," in *Proc. USENIX Security*, 2021.

[4] J. Groth, "On the size of pairing-based non-interactive arguments," in *Proc. EUROCRYPT*, 2016.

[5] World Foundation, "World Whitepaper," 2023. Available: https://whitepaper.world.org

[6] BrightID, "BrightID: A decentralized, open-source social identity network," 2020. Available: https://brightid.org

[7] Reclaim Protocol, "Reclaim Protocol Documentation," 2024. Available: https://docs.reclaimprotocol.org

[8] Google, "reCAPTCHA Enterprise Documentation," 2023. Available: https://cloud.google.com/security/products/recaptcha

[9] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[10] C. Shen, Z. Cai, and X. Guan, "Continuous authentication for mouse dynamics: A pattern-growth approach," in *Proc. IEEE/IFIP DSN*, 2012.

[11] X. Wang, J. Yamagishi, M. Todisco, et al., "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, 2020.

[12] S. Bowe, A. Gabizon, and I. Miers, "Scalable multi-party computation for zk-SNARK parameters in the random beacon model," *IACR ePrint 2017/1050*, 2017.

[13] Solana Labs, "SPL Token-2022 Program," 2023. Available: https://github.com/solana-program/token-2022

[14] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. EUROCRYPT*, 2004.

[15] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614–634, 2001.

[16] B. Fuller et al., "Fuzzy extractors are practical: Cryptographic strength key derivation from the iris," in *Proc. ACM CCS*, 2025. IACR ePrint 2024/100.

[17] A. Serwadda and V. V. Phoha, "When kids' toys breach mobile phone security," in *Proc. ACM CCS*, 2013.

[18] D. Hamm, E. Kupris, and T. Schreck, "Always authenticated, never exposed: Continuous authentication via zero-knowledge proofs," in *Proc. STM*, Springer, 2025.

[19] A. R. Choudhuri, S. Garg, K. Lee, H. Montgomery, G. V. Policharla, and R. Sinha, "A cryptographic framework for proof of personhood," IACR ePrint 2026/333, 2026.

[20] A. Mahfouz, H. Mostafa, T. M. Mahmoud, et al., "M2auth: A multimodal behavioral biometric authentication using feature-level fusion," *Neural Computing and Applications*, vol. 36, pp. 21781–21799, 2024.

[21] S. Paik, C. Hwang, S. Kim, and J. H. Seo, "On the reversibility of locality-sensitive hashing-based biometric template protections," *IEEE Trans. Dependable and Secure Computing*, 2025.

[22] A. Gabizon, Z. J. Williamson, and O. Ciobotaru, "PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge," IACR ePrint 2019/953, 2019.

[23] W. Pouw et al., "The human voice aligns with whole-body kinetics," in *Proc. Royal Society B*, 2025.

[24] G. Stragapede et al., "BioMoTouch: Touch-Based Behavioral Authentication Using Motion and Touch Sensor Fusion," *arXiv:2604.07071*, 2025.

[25] X. Wang, H. Delgado, H. Tak et al., "ASVspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale," *arXiv:2408.08739*, 2024.

[26] K. Verma et al., "Pitch Imperfect: Detecting Audio Deepfakes Through Acoustic Prosody Analysis," *arXiv:2502.14726*, 2025.

[27] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv:2212.04356*, 2022.

[28] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020.