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

Charles Hooper
github.com/entros-protocol

Original: June 27, 2025 Updated: April 25, 2026

Abstract

The proliferation of sophisticated AI and bot networks necessitates robust methods for verifying human uniqueness and liveness in digital ecosystems. Existing Proof-of-Personhood (PoP) solutions rely on centralized authorities, invasive static biometrics, or socially-correlatable data, creating vulnerabilities in privacy, security, and accessibility. 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 not by a static secret, but by the bounded, chaotic drift of biological and behavioral patterns over time. The framework captures multi-modal behavioral data (voice prosody, hand tremor, touch dynamics) during a configurable behavioral challenge, extracts a 134-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 the protocol against replay, synthesis, and Sybil attacks, introduce a graduated trust model distinguishing first-time liveness checks from sustained temporal consistency, and present benchmarks from a working implementation deployed on Solana devnet.

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—iris (Worldcoin [5]), palm print (VeryAI), or face—that, once captured, serves as a permanent anatomical identifier. BrightID [6] takes a different approach using social graph analysis, which depends on coordinated verification events. These designs optimize for different properties: strong uniqueness guarantees at the cost of revocability, or social trust at the cost of coordination

overhead. Entros explores a third axis: consistency of dynamic behavior over time, which is both bounded enough to uniquely identify and variable enough to be naturally revocable.

The Entros Protocol operates on a different principle. A human is not a static data point; they are a continuous, dynamic process. The behavioral signature of a living human—the micro-perturbations in voice, the involuntary tremor in hand movement, the idiosyncratic pressure patterns of touch—drifts over time in a bounded, chaotic pattern that is unique to each individual. AI can mimic a snapshot of this signature. Sustaining a temporally-consistent imitation across weeks and months is computationally prohibitive relative to the value extractable from most Sybil attacks.

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 134-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 *Temporal Fingerprint* where intra-person Hamming distance is bounded ($\sim$20–65 bits) while inter-person distance approaches random ($\sim$128 bits).
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 that rewards sustained temporal consistency over time.
5. A graduated trust model that honestly distinguishes first-time liveness checks from sustained behavioral consistency, with integrator-controlled trust thresholds.
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 must satisfy five properties:

Definition 1 (TBH Requirements)**.** *A TBH scheme is a tuple of algorithms* ($\mathsf{Challenge}, \mathsf{Capture}, \mathsf{Extract}, \mathsf{Hash}, \mathsf{Commit}$ *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 fingerprint* F_T *is never transmitted. Only a Poseidon commitment* $H_{TBH} = \mathsf{Poseidon}(F_T, s)$ *is published on-chain. The underlying feature vector is transmitted to the validation server as a fixed-size statistical summary for provenance validation but is not stored (Section 6.8).*

5. ***Efficiency.*** *All operations run on consumer hardware within a browser context in under 5 seconds.*

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 ($1357^5 \approx 4.7 \times 10^{15}$ 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, δ. 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—a pattern that real-time streaming TTS has obsoleted. 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 [28] 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 [29] extracts the human-vs-synth signal from cycle-level perturbation statistics—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 [30] and Wav2Vec2-Phoneme [31] exhibit substantially higher error rates on out-of-distribution input than on natural language. Whisper's autoregressive decoder hallucinates training-corpus filler ("Thanks for watching") on nonsense input—observed at $\sim$30 % false-reject rate on clean human speech of nonsense syllables. Wav2Vec2-Phoneme operates in the right primitive (CTC forced alignment, not transcription) but its baseline phoneme error rate of 10–15 % on out-of-distribution phonotactics compounds against the per-phoneme matching metric, yielding a discrimination gap of only 10–15 percentage points between right and wrong content—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 combinatorial defense remains: $1357^5 \approx 4.7 \times 10^{15}$ unique 5-word phrases, infeasible to pre-synthesize. The discrimination gap on the shipped Whisper-tiny.en + word-level Levenshtein pipeline is approximately 95 percentage points (mean right-phrase distance 0.07 vs. mean wrong-content distance 1.0 across calibration trials)—an order-of-magnitude improvement over the v2 phoneme-level pipeline. Per-session unpredictability is preserved by uniform random sampling from the public dictionary; per Kerckhoffs's principle, the protocol's security depends on nonce freshness, the content-binding check itself, and the orthogonal Tier 1 acoustic and Tier 2 cross-modal defenses—not on the secrecy of the vocabulary.

2.3 Multi-Modal Data Acquisition

Three sensor streams are captured simultaneously over a configurable window (default: 7 seconds, extended to 12 seconds in the reference web application):

- S_{audio}: Microphone input at 16 kHz (or device-native rate), capturing voice prosody.
- S_{motion}: IMU accelerometer/gyroscope at 60–100 Hz on mobile; mouse pointer dynamics on desktop.
- S_{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 134-dimensional feature vector $\mathbf{v} \in \mathbb{R}^{134}$ through three parallel pipelines.

2.4.1 Speaker Features ($\mathbf{v}_{\text{audio}} \in \mathbb{R}^{44}$)

Fundamental frequency and perturbation. F_0 statistics and delta, jitter measures (local, RAP, PPQ5, DDP), shimmer measures (local, APQ3, APQ5, DDA), and harmonics-to-noise ratio (HNR). These capture physiological characteristics of the vocal tract and laryngeal control. F_0 is trivial for TTS engines to match, but jitter and shimmer measure involuntary micro-perturbations that synthetic speech produces with unnaturally low or uniform values. HNR detects synthetic audio because TTS engines produce unnaturally clean signals without the breath noise present in real speech.

Formant ratios and spectral shape. Formant frequency ratios (F_1/F_2, F_2/F_3) via linear predictive coding (LPC) analysis [9], and Long-Term Average Spectrum (LTAS) statistics (spectral centroid, rolloff, flatness, spread) capture vocal tract resonance geometry.

Statistical condensing. Voicing ratio, amplitude entropy, and per-feature moments (mean, variance, skewness, kurtosis) over the capture window produce the fixed-size vector $\mathbf{v}_{\text{audio}}$.

2.4.2 Kinematic Features ($\mathbf{v}_{\text{kin}} \in \mathbb{R}^{54}$)

Jerk and jounce analysis. The third (jerk) and fourth (jounce) time derivatives of pointer coordinates are computed. Involuntary human movements exhibit characteristic high-frequency jerk signatures that scripted movements lack.

Path dynamics. 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 capture habitual motor control patterns unique to each individual [10].

2.4.3 Touch Features ($\mathbf{v}_{\text{touch}} \in \mathbb{R}^{36}$)

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.

2.5 Feature Fusion and SimHash

Normalization. Each modality group is independently z-score normalized ($\mu = 0$, $\sigma = 1$) to ensure equal contribution regardless of raw magnitude. Non-finite values are sanitized to zero.

Concatenation. The three vectors are concatenated: $\mathbf{v}_{\text{fused}} = [\mathbf{v}_{\text{audio}} \| \mathbf{v}_{\text{kin}} \| \mathbf{v}_{\text{touch}}] \in \mathbb{R}^{134}$.

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 non-stationary*: 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 addresses this through a different construction: SimHash produces a locality-sensitive fingerprint where bounded drift maps to bounded Hamming distance, and a Groth16 circuit proves that this distance falls within the expected range. The Poseidon commitment provides the revocability property—a user can generate a new salt to produce a fresh commitment without changing their behavioral profile. Recent work on practical fuzzy extractors for iris biometrics [16] achieves 105 bits of entropy with 92% true accept rate, providing a useful benchmark: Entros's 256-bit SimHash must be evaluated not by bit-length but by effective entropy under adversarial feature distributions, a question we identify as future work.

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$1,996 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 $d_H = \sum_{i=1}^{256} d_i$.

The minimum distance constraint ($\delta_{\min} = 3$) prevents exact replay attacks. The threshold ($\delta_{\max} = 96$) rejects fingerprints from different people.

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—wrong Hamming distance or wrong Poseidon preimage—except with negligible probability. The structured reference string introduces a trust assumption: an adversary who knows the toxic waste from *all* Phase 2 ceremony contributors can forge proofs. The multi-party ceremony ensures this requires universal collusion among independent participants.

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—multi-contributor, production-grade, circuit-agnostic. Phase 2 currently has a single contributor. A multi-party computation ceremony with $\geq$10 independent contributors will precede mainnet deployment. The SRS is compromised only if *all* Phase 2 contributors collude [12].

4 Economic Model

4.1 The Entros Token

The protocol's economic security is anchored by a native utility token (SPL Token-2022 with Confidential Balances [13]):

1. **Staking.** Validators stake Entros tokens to participate in the Anonymity Ring.
2. **Verification capacity.** Integrators stake Entros for discounted or unlimited verifications via capacity tiers.
3. **Governance.** Token holders vote on protocol parameters.

4.2 User-Pays Model

In wallet-connected mode, the user pays a small protocol fee (∼0.005 SOL) per verification. This is trivial for any Solana user but creates real economic cost for bot farms attempting to maintain thousands of identities. Integrators read on-chain verification state for free via `verifyEntrosAttestation()`—no escrow, no API keys, no billing relationship. For walletless mode (liveness-check tier), the integrating application optionally funds verifications via the relayer API.

4.3 Validation Cycle

The protocol fee from each verification is collected into an on-chain treasury PDA. The treasury purchases Entros from the open market and distributes rewards to honest validators, creating buy pressure proportional to real verification volume.

4.4 Slashing

The design specifies a probabilistic audit mechanism: a configurable fraction of successful validations would trigger a secondary audit by an independent Anonymity Ring, with disagreement resulting in slashing of the primary Ring's stakes. 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_timest` (i64), `verification_count` (u32), `trust_score` (u16), `current_commitment` ([u8;32]), and a rolling window of the 10 most recent verification timestamps for Trust Score computation.

5.2 Progressive Trust Score

The Trust Score rewards consistency over time, not volume. A bot verifying 100 times in one day scores lower than a human verifying weekly for months. The formula combines three components:

Recency-weighted count. Each of the last 10 verification timestamps contributes $\frac{3000}{30+d_i}$ where d_i is the number of days since that verification, multiplied by a protocol-configurable base increment.

Regularity bonus. The standard deviation of inter-verification gaps is computed. Lower variance yields a higher bonus (up to 20 points), rewarding regular spacing.

Age bonus. $\lfloor\sqrt{\min(\text{age\_days}, 365)}\rfloor \times 2$, using deterministic integer square root. Diminishing returns prevent gaming via old unused accounts.

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 Walletless Mode

Wallet-connected mode is the primary flow. The user pays a small protocol fee, signs the transaction, mints an Entros Anchor, and builds an on-chain Trust Score queryable by any integrator. This creates economic cost for bot farms: each fake identity requires a funded wallet and per-verification fees.

Walletless mode is a secondary liveness-check tier. The user completes the behavioral challenge; the Pulse SDK generates the ZK proof; the relayer submits it on-chain. The behavioral fingerprint is stored locally (encrypted with AES-256-GCM, key as non-extractable `CryptoKey` in IndexedDB) for future re-verification. The identity is device-bound and ephemeral—clearing storage resets it. No on-chain Anchor, no portable Trust Score.

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. Additionally, challenge nonces are single-use and time-limited (5 minutes), preventing replay of the proof itself. □

6.3 Synthetic Data Attacks

Claim 1 (Multi-Modal Synthesis Difficulty)**.** *An adversary must simultaneously synthesize realistic data across all three modalities (voice, motion, touch) such that the fused 134-dimensional feature vector produces a SimHash fingerprint within Hamming distance $\delta_{\max}$ of the target.*

The defense is layered:

Feature-level. The 134-dimensional feature vector captures involuntary biological processes: vocal jitter/shimmer from laryngeal muscle micro-contractions, kinematic jerk from neuromuscular control loops, touch pressure from fingertip biomechanics. Each dimension presents a distinct synthesis challenge. Jitter measures, for instance, capture perturbation rates that TTS engines produce with unnaturally low variance [11].

Cross-modal correlation. SimHash projects the concatenated vector onto shared hyperplanes. Each output bit depends on features from all modalities jointly. An adversary cannot spoof modalities independently—the cross-modal correlations must be consistent.

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.

We do not claim synthesis is impossible. We claim it is expensive relative to the value extractable from most Sybil attacks, and that this cost increases with the number of identities maintained over time.

Empirical context. Serwadda and Phoha [17] demonstrated that spoofing mouse dynamics—even with full knowledge of the target's behavioral profile—required extensive per-target training and achieved limited success rates. Entros requires spoofing three modalities simultaneously. Under an independence assumption, if single-modality spoofing succeeds with probability p_v (voice), p_m (motion), and p_t (touch), the joint success rate is $p_v \cdot p_m \cdot p_t$. Even generous estimates of 0.3 per modality yield $\sim$2.7% joint success per attempt.

Voice modality resilience. Modern text-to-speech systems can clone voice timbre from seconds of reference audio. However, Entros's voice features specifically target involuntary laryngeal micro-perturbations (jitter, shimmer, HNR) rather than perceptual voice quality. ASVspoof challenge results [11] confirm that TTS outputs exhibit unnaturally low jitter variance and unnaturally high HNR compared to natural speech. While this gap is narrowing as synthesis quality improves, the multi-modal fusion ensures that voice is one signal among three, not a single point of failure. Behavioral biometric systems using ZK verification have independently demonstrated practical false accept rates below 1% [18].

6.4 Sybil Attacks

Each wallet maps to exactly one Entros Anchor (enforced by PDA derivation). Creating k fake identities requires:

1. k funded Solana wallets (SOL cost)
2. k independent behavioral profiles, each sustained across regular re-verifications
3. $k \times m$ verification fees over m re-verification cycles

The Trust Score penalizes new accounts (age bonus starts at 0) and irregular patterns (regularity bonus requires consistent spacing). An adversary building 1,000 identities with Trust Score > 500 over 3 months incurs costs that exceed the value of most airdrop allocations.

6.4.1 Layered Sybil Resistance

Entros's Sybil resistance operates through three independent layers, each raising the cost of maintaining duplicate identities:

- **Economic deterrence.** Each verification costs the user SOL. Each wallet requires funding. Maintaining thousands of fake identities over months requires sustained capital expenditure that scales linearly with the attack surface.
- **Temporal deterrence.** Trust Score rewards consistency over time. Weekly re-verifications across months carry more weight than bulk verifications in a single session. A bot farm must maintain each identity's behavioral signature across sessions and days, compounding the operational cost per identity.
- **Behavioral fingerprint comparison.** The server-side registry compares each new verification's SimHash fingerprint against existing entries. One person operating multiple wallets produces clustered fingerprints that the registry detects. The discriminative power of the 134-feature behavioral fingerprint is being empirically calibrated through data collection from diverse users.

These layers are complementary. Economic cost makes Sybil farming expensive. Temporal requirements make it slow. Behavioral comparison makes it detectable. An attacker must defeat all three simultaneously. The fingerprint registry's effectiveness improves as empirical data informs threshold calibration, and the architecture supports evolution toward probabilistic risk scoring as the user population 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. By the zero-knowledge property of Groth16, the proof reveals nothing beyond the truth of the statement. The public inputs are Poseidon commitments (computationally hiding under the discrete logarithm assumption on BN254) and the threshold parameters. The fingerprints and salts are private witnesses. □

Raw biometric data is destroyed after feature extraction. On-chain, only the Poseidon commitment is stored—computationally hiding under standard assumptions.

SimHash reversibility. Recent work has demonstrated pre-image attacks on locality-sensitive hashes [21], showing that SimHash fingerprints contain recoverable information about the original input. Entros's architecture addresses this at two levels: (1) the SimHash fingerprint is never transmitted or stored on-chain—only the Poseidon commitment is public, and the ZK proof reveals nothing beyond the Hamming distance range; (2) the fingerprint stored locally for re-verification is encrypted with AES-256-GCM using a non-extractable `CryptoKey` in IndexedDB, requiring device-level compromise to access. SimHash is not relied upon as a privacy-preserving representation; the Poseidon commitment provides that property.

6.6 Economic Sustainability of Attacks

The protocol does not claim to make spoofing impossible. It claims to make sustained spoofing *economically irrational* relative to the value it extracts. The defense is layered:

- **Feature-level:** Realistic multi-modal sensor data across 134 dimensions is hard to synthesize.
- **Circuit-level:** Replays ($d_H = 0$) and imposters ($d_H > \delta_{\max}$) are rejected.
- **Entropy scoring:** Low-entropy synthetic data is flagged before hashing.
- **Economic:** Each verification costs SOL. Each wallet requires funding. Trust Score rewards months of consistency over bursts.

6.7 Graduated Trust Model

First-time verification establishes a behavioral baseline. With no prior fingerprint, the Hamming distance circuit does not fire. The protocol functions as a multi-modal liveness check, relying on feature extraction quality to distinguish human from synthetic data.

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. ***Liveness*** *(first walletless verification). Multi-modal sensor data was captured from a likely human. No temporal consistency. Signal strength: low. Suitable for captcha-equivalent use cases.*
2. ***Device-bound consistency*** *(returning walletless verification). Behavioral drift matches a device-local fingerprint. Signal strength: medium. Suitable for session authentication, content gating.*
3. ***Portable identity*** *(wallet-connected with Trust Score). Persistent on-chain Anchor with months of behavioral consistency visible to all integrators. Signal strength: high. Suitable for airdrop eligibility, DAO governance, DeFi access controls.*

In wallet-connected mode, the user pays a protocol fee per verification, creating direct economic cost for bot farms. In walletless mode, the integrator optionally funds verifications and controls abuse exposure through per-IP rate limiting and minimum Trust Score requirements. The protocol provides the signal; the integrator sets the threshold.

A bot that clears local storage before each walletless verification is perpetually at Tier 1—a liveness check with no temporal history. High-value integrations can require Tier 2 or Tier 3, making this strategy ineffective for anything beyond basic captcha equivalence.

6.8 Browser Trust Model and Server-Side Validation

The current implementation executes the entire verification pipeline—sensor capture, feature extraction, SimHash computation, Poseidon commitment, and Groth16 proof generation—within the browser. This architecture maximizes privacy: raw biometric data never leaves the device, and the ZK proof is the only artifact transmitted. However, the browser is an

untrusted execution environment. An adversary controlling the browser can override sensor APIs (injecting synthetic audio via `getUserMedia`, dispatching fabricated `PointerEvents`), manipulate the feature extraction pipeline, or submit pre-computed proofs generated from optimized synthetic data.

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). The 134-dimensional feature vector is transmitted alongside the proof to a validation server. The server applies statistical analysis using models inaccessible to the client: cross-modality correlation coefficients (real humans exhibit involuntary correlations between voice, motion, and touch that independent synthetic generators do not reproduce), per-feature entropy distributions (synthetic data exhibits entropy profiles outside expected human ranges), and jitter variance ratio analysis (text-to-speech engines produce unnaturally low jitter variance compared to natural speech [**?**]). The server-side models constitute a shared secret: the adversary cannot reverse-engineer the validation criteria.

This architecture preserves the core privacy property. The feature vector is a fixed-size statistical summary (means, variances, spectral coefficients)—not raw time-series data. It cannot be used to reconstruct the original audio, motion, or touch signals. The ZK proof continues to ensure the fingerprint itself is never revealed. The feature vector provides a complementary signal for provenance validation without compromising the zero-knowledge property of the identity proof.

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 Attestation

Browser-based sensor APIs provide no guarantee that captured data originates from physical hardware. A complementary defense is device attestation: verifying the integrity of the execution environment before behavioral capture begins.

A native mobile application can perform deterministic checks unavailable to browser JavaScript: whether the device is rooted or jailbroken, whether the application binary has been tampered with or instrumented (e.g., Frida, Xposed), whether the execution environment is an emulator rather than physical hardware, and whether sensor APIs are being intercepted by hooking frameworks. Hardware attestation APIs (Android's Play Integrity, iOS's DeviceCheck) provide cryptographic proof of device integrity signed by the platform vendor.

This constitutes a *positive security model*: rather than detecting characteristics of bots (negative model, probabilistic), the system verifies characteristics known to be genuine (positive model, deterministic). Device attestation forms the foundation layer. Behavioral biometric verification with ZK proofs operates on top of it. The combination addresses both the provenance question (did this data come from a real device?) and the identity question (is this behavioral pattern consistent with the claimed identity?).

A native application also unlocks sensor modalities that browsers restrict: persistent accelerometer access with a single permission grant on iOS (browsers re-prompt each session), pressure-sensitive touch data on supported Android devices, and background re-verification without requiring the user to open a web page.

6.10 Empirical Adversarial Validation

The theoretical defenses described in Sections 6.1–6.9 are validated empirically through a continuous internal red team program. An adversarial testing harness submits synthesized feature vectors to the production validation service over HTTP, measuring per-tier pass rates against the live Tier 1 enforcement layer—the gate preceding on-chain submission. An attempt that fails Tier 1 cannot proceed to challenge fetch, ZK proof generation, or transaction submission, so a 0% Tier 1 pass rate implies a 0% on-chain anchor creation rate by construction.

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 134-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 (modern voice cloning via XTTS-v2, F5-TTS), T5 (coupled cross-modal synthesis), and T6–T8 (identity theft, replay perturbation, adaptive probing) are planned. Aggregate results are published at `https://entros.io/security`. Attack implementation code remains private per responsible-disclosure convention.

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 hardware cost for coordination overhead—users must attend verification events, and the trust model assumes non-colluding participants. Entros's approach requires neither coordination nor hardware: verification happens on a single device in 12 seconds.

Reclaim Protocol [7] proves ownership of existing web2 accounts via TLS session proofs. It answers "do you control this account?"—not "are you human?" Entros and Reclaim are complementary: Reclaim proves account ownership, Entros proves the account owner is human.

Traditional CAPTCHA (reCAPTCHA [8], hCaptcha, Turnstile) provides session-level bot detection using behavioral signals, browser fingerprinting, and centralized machine learning classifiers. The advantages are maturity and a vast training dataset (Google processes billions of sessions). The disadvantages are privacy concerns (behavioral data sent to Google), lack of identity persistence (no concept of "the same human returning"), binary output (pass/fail with no graduated trust), and vulnerability to captcha-solving services. Entros's first walletless verification provides comparable liveness detection; its graduated trust model provides capabilities CAPTCHA fundamentally cannot.

VeryAI uses palm print biometrics with on-device processing, sharing Entros's commitment to keeping raw biometric data on the device. Its palm print is a static identifier, which provides strong one-shot uniqueness but does not capture behavioral drift across sessions. Entros's behavioral temporal consistency model is complementary to this kind of static-biometric design: a palm print proves you exist; Entros proves you're the same human returning over months.

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 the first rigorous 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 derives personhood from behavioral biometrics without a trusted issuer, which is more decentralized but harder to formalize under their model. Mapping Entros's security properties to this framework is identified as future work.

Regulatory positioning. Proof-of-personhood systems that transmit or store biometric data have faced enforcement actions under GDPR and regional biometric-data laws, with cited concerns including collection, storage, and cross-border transfer of biometric templates. Entros's architecture is designed to avoid these triggers by construction: raw biometric data

never leaves the user's device, and only a 134-dimensional statistical summary and a zero-knowledge proof are transmitted. This places Entros's data flows closer to a standard web analytics fingerprint than to a biometric collection pipeline.

8 Implementation and Benchmarks

The protocol is deployed on Solana devnet with six components: three Anchor/Rust on-chain programs with full constraint validation and on-chain Trust Score; a Groth16/Circom circuit (1,996 constraints) with trusted setup; the Pulse SDK (TypeScript, published on npm, 60 SDK tests including an 8-phase adversarial pen test harness); a server-side validation service (Rust, 32 tests); an executor node (Rust, live on Railway) providing the relayer API with rate limiting and commitment registry; and a demo application (Next.js on Vercel) with walletless and wallet-connected flows. A Realms DAO voter weight plugin (38 tests) provides governance integration. Total test coverage: 155 tests across all repos.

The protocol fee treasury is live on devnet, collecting 0.005 SOL per verification. The fee is deducted atomically within the batched verification transaction and is admin-adjustable. Treasury balance is publicly auditable on Solana Explorer.

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 (134 dimensions): $\sim$45 ms
- SimHash (256-bit): $<$1 ms
- Poseidon commitment: $\sim$3 ms
- Groth16 proof generation (WASM): $\sim$850 ms
- On-chain verification: $\sim$180K compute units
- **Total (excluding capture): $\sim$900 ms**

The total pipeline from button click to on-chain proof takes approximately 11–16 seconds depending on the configured capture window, plus $\sim$900 ms of computation. On mobile (iPhone 15 Pro Max, Safari), all three sensor streams (audio, IMU motion, touch) capture simultaneously. Audio captures at the device-native 48 kHz and is processed identically. Proof generation completes within the same time budget via snarkjs WASM.

Comparative context. Groth16 proof generation at $\sim$850 ms compares favorably to PLONK-based systems, which require $\sim$2.5 s for equivalent circuit sizes [22]. On-chain verification at $\sim$180K compute units fits within Solana's 200K default budget; PLONK verification would exceed it. Poseidon commitment at $\sim$3 ms reflects the hash's ZK-optimized design ($\sim$300 R1CS constraints vs. $\sim$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% [27]. 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.

Entros accepts desktop verification as a valid but weaker signal. The verification produces a legitimate behavioral fingerprint and ZK proof regardless of device. Trust Score accumulates identically. The server-side validation applies the same checks. The practical difference is that desktop fingerprints have lower inter-person discriminability, and the Sybil registry threshold may need to be more conservative for desktop-only users.

The mobile application, targeting the Solana dApp Store, is the production target for strongest verification signal. Native sensor APIs provide sub-millisecond accelerometer timestamps, touch pressure data, and persistent background access without per-session permission prompts.

9 Conclusion and Future Work

The Entros Protocol presents a framework for Proof-of-Personhood through temporal behavioral consistency. By measuring bounded, chaotic drift in multi-modal biometric signals over time, it provides graduated trust guarantees that static biometrics and session-level captcha cannot.

The protocol is honest about its limitations. First-time verification is a liveness check, not a temporal consistency proof. The graduated trust model makes this explicit rather than presenting a false binary. The defense against sophisticated synthesis attacks is economic, not absolute—sustained spoofing at scale costs more than it extracts.

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.
- Entros utility token: SPL Token-2022 with Confidential Balances for validator staking, capacity tiers, and governance.
- Cross-chain deployment to Ethereum L2s after Solana mainnet stabilizes.
- Formal analysis of SimHash collision probability bounds under adversarial feature distributions.
- Cross-wallet fingerprint comparison is implemented in the server-side validation layer. The executor maintains a registry of SimHash fingerprints and compares each new verification against existing entries. If the Hamming distance between a new fingerprint and

any existing entry falls below $\delta_{\max}$, the verification is flagged as a potential duplicate identity. Empirical investigation of the persistence of involuntary behavioral features across deliberate behavioral modification is ongoing.

- Server-side feature validation is implemented as described in Section 6.8. The validation models, thresholds, and detection algorithms are proprietary—the protocol layer is open source for trust and auditability, the defense layer is private for security. This follows the emerging "immutable open source" model in decentralized systems: on-chain programs are transparent and immutable, off-chain defense logic is private.
- Adversarial testing was conducted across eight phases: exact replay (blocked by $\delta_{\min}$), naive synthesis (passes client-side pipeline), sustained re-verification (100% success rate without server-side validation), human-to-bot handoff, cross-modality correlation analysis, Sybil cost modeling, feature-level optimization (converges in 251 iterations), and full-pipeline random search (90% success rate without server-side validation). These results motivated the implementation of server-side validation as a required defense layer.
- Device attestation via native application. Hardware integrity verification before behavioral capture, implementing a positive security model as described in Section 6.9.

The protocol is open source and published as a defensive disclosure to establish prior art. Source code, circuit definitions, and SDK are available at `github.com/entros-protocol`.

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] "Countries that have banned or investigated Worldcoin," BitPinas, 2026. Available: https://bitpinas.com/learn-how-to-guides/list-countries-banned-investigated-worldcoin/

[24] Y. Zang et al., "SONAR: A Synthetic AI-Audio Detection Framework and Benchmark," *arXiv:2410.04324*, 2024–2025.

[25] Y. Chen et al., "VoiceRadar: A New Paradigm of Voice Deepfake Detection via Micro-Frequency Estimation," in *Proc. NDSS*, 2025.

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

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

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

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

[30] 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.

[31] 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.