

Shreyas Londhe

[in linkedin.com/in/shreyaslondhe](https://www.linkedin.com/in/shreyaslondhe) [✉ shreyas_londhe@outlook.com](mailto:shreyas_londhe@outlook.com) [G github.com/shreyas-londhe](https://github.com/shreyas-londhe) [T @shreyaslondhe](https://twitter.com/shreyaslondhe)

Applied cryptography engineer specializing in zero-knowledge proof systems with proven delivery on production-grade ZK stacks (ProveKit, zk-email). Experienced in performance tuning (constraint/memory reduction), recursive verification pipelines, and integrating real-world protocols (email/JWT) into ZK circuits for external clients.

SKILLS

Languages: Rust, Go, TypeScript, Solidity **ZK Frameworks:** Halo2, Circom, Noir, Plonky2, Barretenberg
Primitives: Sumcheck, Spartan, WHIR, FRI, IVC, R1CS, Plonkish, Homomorphic Encryption

PROFESSIONAL EXPERIENCE

- Applied Cryptography Engineer — ZkEmail** May 2024 – Present
- Implemented **layered witness building** with batched modular inverses, reducing witness generation from **10% to 1.2%** of total proving compute.
 - Built **SHA-256 compression blackbox**, cutting proving time from **22s to 9s (59% faster)** and peak memory from **10.4GB to 5.25GB (50% reduction)**.
 - Implemented witness-splitting for verifier-controlled challenges in LogUp/Spice—prior implementation let provers choose challenges, breaking range-check soundness.
 - Developed **zk-regex v2**, a regex-to-circuit compiler (Circom/Noir) with off-circuit NFA traversal and in-circuit proof.
 - Developed **jwt-tx-builder** for proving OAuth identity (e.g., Google Workspace membership) via JWT verification in ZK.
- Applied Cryptography Engineer — Aerius Labs** June 2023 – April 2024
- Developed **ZkSnap** end-to-end, a privacy-preserving voting protocol using IVC-based proof aggregation and additive homomorphic encryption.
 - Built the full Halo2 circuit stack with client-side proof generation under 30s and trustless recursive aggregation.
 - Authored technical whitepaper covering cryptographic primitives, protocol design, and threat model.
- Software Engineer — Nume Crypto** March 2022 – May 2023
- Architected smart contracts for L2 solution with focus on upgradability and storage efficiency.
 - Implemented Diamond Proxy pattern to bypass Ethereum's 24KB contract size limit.
 - Developed ZK circuits for protocol edge cases, deposits, and withdrawals.

OPEN SOURCE CONTRIBUTIONS

- WHIR** — ZK Polynomial Commitment Scheme
- Reduced multi-polynomial proof overhead with `batch_prove/batch_verify`, collapsing N separate proofs into one.
 - Enabled recursive verification by fixing transcript to constant length—previously variable-length transcripts blocked in-circuit verification.

GRANTS

- zkFOCIL** — Ethereum Foundation Research Grant
- Built ZK circuit to anonymize FOCIL includer identities, replacing public signatures with proofs to prevent coercion/legal targeting.
 - Built using Barretenberg stdlib and Ultra Honk with KZG and IPA backends.
- PLUME** — PSE Grant
- Implemented Plume nullifier scheme + secp256k1 hash-to-curve in Halo2 for anonymous double-spend prevention; passed PSE audit.