

Arush Singh

PROTOCOL ENGINEER

ZERO-KNOWLEDGE SPECIALIZATION

[github.com/04arush](#) · [linkedin.com/in/04arush](#)
[x.com/04arush](#) · [04arush.github.io](#)
[04arush@gmail.com](#) · New Delhi, India

PROFILE

Protocol engineer specialising in **Zero-Knowledge Cryptography**, with hands-on circuits in **Noir** and **Circom** and on-chain verification via **Barretenberg UltraHonk** and **Groth16** verifiers. Authored a Merkle Sum Tree proof-of-solvency, private credit verification for lending, and commit-nullifier voting. Strong in **arithmetic circuits**, hash and witness parity, and the Solidity integration around them, with a systems-level track in **Rust**.

SKILLS

Protocol & Cryptography

Rust **Go (Cosmos SDK)** Poseidon Merkle Sum Trees P2P Architecture Consensus
Tokio Alloy

Zero-Knowledge

Noir **Barretenberg (bb)** **Circom** **Rust (arkworks)** UltraHonk Groth16 / SNARKs
STARKs

Smart Contracts & Security

Solidity **Foundry** DeFI OpenZeppelin Chainlink VRF / Automation
Fuzz / Invariant Testing Slither Viem / Wagmi

Languages & Tooling

TypeScript Node.js React Linux / Shell Git SQL (PostgreSQL) NoSQL (MongoDB)
Python

EXPERIENCE

Blockchain Protocol Engineer

Oct 2026 - Present

Cubane · Hybrid · Part-time

[Tech used: Rust · Go · Solidity · Noir · TypeScript]

- Design and develop scalable blockchain architecture for Cubane's platform, working directly with the founders and the product and design teams.
- Build secure smart contracts, decentralised protocol components and off-chain integrations.
- Monitor and optimise protocol performance for efficiency, security and reliability.

PROJECTS

Zero-Knowledge Proof of Solvency

[GitHub](#)

Noir · Merkle Sum Trees · Poseidon · Barretenberg · Solidity · Foundry

- Noir circuit proves a user's balance is correctly included in a **Merkle Sum Tree** whose root sum matches the custodian's declared liability, without exposing balances or secrets. Sums are bound into each **Poseidon** node hash.
- Range-constrains every balance to $[0, 2^{64})$, closing the **negative-balance attack** that masks a reserve deficit.
- On-chain **UltraHonk** verification through `proveAndRecord()` with **100% line, function and branch coverage**. Design inspired by Summa (PSE).

ZK Undercollateralized Lending Protocol

[GitHub](#)

Noir · Barretenberg · UltraHonk · Solidity · Foundry · Next.js

- ▶ Noir circuit proves `credit_score ≥ 700` and `income ≥ 50,000` with both values **private** and thresholds public, so a borrower qualifies for collateral without revealing credit data.
- ▶ Pipeline compiles the circuit, generates the verification key with Barretenberg (bb) and emits `UltraVerifier.sol`, which `LendingPool.borrow()` calls before any funds move.
- ▶ In-browser proving via Noir WASM and Barretenberg WASM (10 to 30 s), so private inputs never leave the device.

ZKVote: Commit-Nullifier Voting DApp

[GitHub](#)

Circom · Groth16 · snarkjs · Poseidon · Solidity · Foundry · React

- ▶ a **Circom** circuit binds a Poseidon commitment to the vote and derives a per-proposal **nullifier**, hiding who voted for what.
- ▶ **Groth16** proofs are generated in-browser with snarkjs and verified on-chain; `ZKVoting.sol` checks deadline, commitment and nullifier state before verification.
- ▶ Append-only nullifier registry blocks double-voting. Foundry-tested; React/ethers.js frontend with a Node/MySQL metadata API.

HACKATHONS

ETHOnline 2026

Participation Award

Built **StreamProof**, a private royalty-tier verification protocol: a Noir circuit (Poseidon2 Merkle inclusion + threshold proof) verified on-chain via UltraHonk on Arc Testnet.

Frostbyte Hackathon

Finalist · Participation Award

Qualified with the **Employee Payroll Manager** (Chainlink Automation), then advanced to the finale with the **ZK lending protocol**.

EDUCATION

Bachelor's — Computer Application

Aug 2023 - Ongoing

Indira Gandhi National Open University (IGNOU)
New Delhi, Delhi, IN

CERTIFICATIONS

Noir Programming & ZK Circuits	Cyfrin Updraft · Apr 2026
Fundamentals of Zero-Knowledge Proofs	Cyfrin Updraft · Mar 2026
Advanced Foundry	Cyfrin Updraft · Mar 2026
Foundry Fundamentals	Cyfrin Updraft · Mar 2026
Rust Programming Basics	Cyfrin Updraft · Feb 2026
Solidity Smart Contract Development	Cyfrin Updraft · Jan 2026
Blockchain Basics	Cyfrin Updraft · Dec 2025