# Scalable Light Node Verification of Aggregated ZK Proofs

Ayush Gupta

*LayerEdge*

**Abstract**

This paper presents a protocol enabling Light Nodes to verify the correctness of large collections of zero-knowledge (ZK) proofs sourced from diverse blockchain environments and off-chain data producers. Proofs originate from Bitcoin Layer 2 (BTC-L2) solutions, Data Availability (DA) layers, AI-based chains, Decentralized Physical Infrastructure (DePin) networks, and Real-World Asset (RWA) platforms. These proofs are recursively aggregated into a single *root proof*, which is then anchored on the Bitcoin blockchain for global availability and immutability.

Crucially, Light Nodes do not need to verify all underlying proofs or store the full blockchain state. Instead, they verify only a randomly selected subset of intermediate aggregated proofs, achieving probabilistic integrity guarantees. With sufficiently many Light Nodes each conducting minimal and independent verifications, the probability that any invalid proof remains undetected diminishes exponentially. To incentivize their participation and ensure the long-term integrity and security of the system, Light Nodes are rewarded with tokens from both the LayerEdge protocol and its clients. This design leverages Bitcoin for final settlement, ZK proofs for succinct correctness, random sampling for scalability, and economic incentives for decentralized security.

## 1 Introduction

The proliferation of specialized blockchain solutions and off-chain computational frameworks has created a heterogeneous environment. Systems such as BTC-L2 protocols, DA layers, AI-oriented chains, DePin networks, and RWA platforms produce zero-knowledge proofs $\{\pi_{i,j}\}$, each attesting to correct execution or data integrity. Verifying every proof is resource-intensive, limiting scalability.

We address this challenge by:

(1) Aggregating all base ZK proofs into a single *root proof* via recursive, pairwise aggregation.

(2) Anchoring the root proof on the Bitcoin blockchain to leverage its robust security and global availability.

(3) Introducing *Light Nodes*, resource-constrained participants that verify only a randomly chosen subset of intermediate proofs. Collectively, many Light Nodes approximate full verification with high probability.

(4) Rewarding Light Nodes with tokens both from LayerEdge and its clients to ensure ongoing participation and incentivize sustained network integrity.

# 2 System Model

## 2.1 Entities and Proof Sources

Define:

$$\mathcal{C} = \{\mathcal{C}_{\text{BTC-L2}}, \mathcal{C}_{\text{DA}}, \mathcal{C}_{\text{AI}}, \mathcal{C}_{\text{DePin}}, \mathcal{C}_{\text{RWA}}\},$$

where:

- $\mathcal{C}_{\text{BTC-L2}}$: Off-chain systems (e.g., rollups, sidechains) extending Bitcoin's functionality, providing ZK proofs of transaction batches.
- $\mathcal{C}_{\text{DA}}$: Data Availability layers ensuring that published data can be retrieved, accompanied by ZK proofs verifying data completeness.
- $\mathcal{C}_{\text{AI}}$: AI-focused chains generating proofs of model correctness and inference verification.
- $\mathcal{C}_{\text{DePin}}$: Decentralized physical infrastructure networks providing proofs verifying sensor data and resource allocation.
- $\mathcal{C}_{\text{RWA}}$: Real-world asset tokenization platforms attesting to proper representation of off-chain assets on-chain.

Each client $\mathcal{C}_i \in \mathcal{C}$ outputs proofs $\{\pi_{i,j}\}$, each validating a relation $S_{i,j}$. Collectively:

$$\Pi = \{\pi_{i,j}\}, \quad N = |\Pi|.$$

## 2.2 Aggregator

The aggregator collects all proofs and applies a ZK-proof aggregation operation $\oplus$ to produce a single root proof $\pi_{\text{root}}$. This succinctly represents the correctness of all $\pi_{i,j}$ proofs.

## 2.3 Light Nodes

Light Nodes are resource-constrained verifiers who:

1. Trust Bitcoin for final settlement.
2. Perform random sampling to verify a small subset of proofs.
3. Receive token rewards from LayerEdge and clients of LayerEdge, compensating them for their verification efforts and incentivizing sustained, honest participation.

# 3 Proof Aggregation

We assume a ZK-proof system that supports aggregation:

$$\pi_a \oplus \pi_b \to \pi_{a,b},$$

such that:

$$\mathcal{V}(\pi_{a,b}) = \text{True} \iff (\mathcal{V}(\pi_a) = \text{True} \wedge \mathcal{V}(\pi_b) = \text{True}).$$

Organize all base proofs as leaves of a full binary tree $\mathcal{T}$ of height $h = \lceil \log_2(N) \rceil$. Each internal node is formed by aggregating its two children. After $h$ rounds:

$$\pi_{\text{root}} = \bigoplus_{\pi_{i,j} \in \Pi} \pi_{i,j}.$$

$\pi_{\text{root}}$ now succinctly attests to the correctness of all $N$ individual proofs.

# 4 Settlement on Bitcoin

To achieve trust-minimized finality, the aggregator commits $\mathcal{H}(\pi_{\text{root}})$, a cryptographic hash of the root proof, in a Bitcoin transaction. Let $tx_{\text{BTC}}$ be the Bitcoin transaction containing this commitment. Once $tx_{\text{BTC}}$ is confirmed at Bitcoin block height $b$, the proof state is globally anchored. Any participant can reference $tx_{\text{BTC}}$ to confirm the canonical root proof.

# 5 Light Node Verification via Random Sampling

Verifying the entire proof tree is infeasible for each Light Node. Instead, we distribute verification tasks across numerous Light Nodes, each checking only one node in the aggregation tree. When many Light Nodes participate, the probability that a fraudulent proof remains undiscovered is negligible.

## 5.1 Random Selection Process

Let $\mathcal{T}$ contain $M$ nodes (both leaves and internal nodes corresponding to aggregated proofs). Each Light Node $C^{\ell}_{\text{light}}$ obtains a randomness seed $r_\ell$, derived from Bitcoin block headers or a Verifiable Random Function (VRF). It then selects a node $(k, m)$ from $\mathcal{T}$ using a pseudo-random process ensuring uniform or strategically biased coverage.

## 5.2 Verification and Detection

The Light Node requests the proof $\pi_m^{(k)}$ and verifies it:

$$\mathcal{V}(\pi_m^{(k)}) = \begin{cases} \text{True}, & \text{if valid,} \\ \text{False}, & \text{otherwise.} \end{cases}$$

If $\mathcal{V}(\pi_m^{(k)}) = \text{False}$, the node broadcasts an alert. This reveals a contradiction and exposes aggregator misbehavior or invalid proofs from clients.

## 5.3 Statistical Guarantees

One Light Node selecting uniformly at random has a $\frac{1}{M}$ chance of detecting any specific invalid node. With $L$ independent Light Nodes:

$$\Pr(\text{undetected invalid proof}) = \left(1 - \frac{1}{M}\right)^L \approx e^{-L/M}.$$

As $L$ grows, this probability drops exponentially, making it extremely unlikely for invalid proofs to remain hidden.

## 5.4 Incentivization of Light Nodes

To ensure the long-term integrity and security of the network, Light Nodes performing these verifications are rewarded. The reward mechanism can be structured as follows:

1. **LayerEdge Token Rewards**: Light Nodes receive a base reward in LayerEdge's native tokens (or other designated tokens of the LayerEdge ecosystem). These rewards are distributed periodically to active and honest Light Nodes.

2. **Client Contributions**: Clients of LayerEdge, such as BTC-L2s, DA layers, AI chains, DePin, and RWA chains, may also contribute tokens to a reward pool. The rationale is that these clients benefit directly from having their proofs verified and thereby share in maintaining the system's security.

3. **Performance-Based Bonuses**: Light Nodes that detect invalid proofs or consistently participate over long periods may receive additional bonuses, further aligning incentives for diligent verification.

This incentivization scheme ensures that as the ecosystem expands and more proofs are generated, there are always sufficient Light Nodes willing to participate, verifying portions of the aggregation tree and thereby maintaining integrity.

### 5.5 Minimal Resource Requirements

A single proof verification is $O(1)$ or $O(\log N)$ depending on the proof system, which is negligible compared to $N$. Thus, Light Nodes can run on low-power devices, and their incentive-driven participation scales with the growth of the network.

## 6 Conclusion

We have described a scalable, trust-minimized verification architecture that integrates proof aggregation, Bitcoin settlement, random sampling by Light Nodes, and explicit incentivization to ensure continuous integrity. By rewarding Light Nodes with tokens from LayerEdge and its clients, we create a vibrant ecosystem where security verifiers are economically motivated to participate and maintain high standards of correctness.

This architecture paves the way for a decentralized and economically sustainable verification model, enhancing security and trust across a wide range of blockchain and off-chain computational domains.