A Framework for Designing Corruption-Resistant Operator Networks

Anthias Labs

Introduction

Anthias Labs created the following framework to help AVS teams design and implement corruption-resistant crypto protocols. Key research areas include dual token Proof-of-Stake models, ZK proving systems, and novel verification mechanisms such as proof of sampling. The following provides an overview of the steps for teams to ensure they assess.

1 Infrastructure Challenges

Infrastructure protocols and middleware protocols oftentimes require a network of operators (also known as nodes) to maintain healthy operation. To discourage malicious activities of the operators, which would undermine protocol integrity, there must be proper detection mechanisms and punishment mechanisms in place to dis-incentivize corruption.

1.1 Detection Mechanisms

1. Proof of sampling: Verifying that a subset of data or actions is representative of the whole, ensuring the integrity of the system.
2. ZK proving: Using zero-knowledge proofs to demonstrate the correctness of computations or actions without revealing sensitive information.
3. Optimistic proof / dispute: Assuming actions are valid unless challenged, with disputes resolved through a predefined process.

1.2 Punishment Mechanisms

1. Slashing staked collateral: Reducing or removing the staked assets of misbehaving actors as a penalty.
2. Slashing delegated staked collateral: Penalizing misbehaving actors by reducing or removing the staked assets delegated to them by others.

3. Permanent removal from the system: Banning malicious actors from participating in the protocol indefinitely.

4. Temporary bans from the system: Prohibiting misbehaving actors from participating in the protocol for a set period.

5. Slashing rewards: Withholding or reducing the rewards earned by misbehaving actors.

6. Damaging social reputation / capital: Publicly identifying and denouncing malicious actors, harming their standing within the community.

2 Designing an Operator Network

Designing a robust and corruption-resistant system requires a systematic approach that considers the roles and responsibilities of the operator network, the verification and detection mechanisms, and the penalty mechanisms.

2.1 Step 1: Defining Scope and Requirements

- Functionality: Determine what critical functions your operator network will perform. Will they be verifying transactions, building blocks, attesting data, monitoring smart contracts, or performing other roles?

- Decentralization: Evaluate whether your operator network needs to be decentralized to prevent central points of failure and reduce corruption risks.

- Scale and Accessibility: Decide on the size of the operator network and whether it should be open to anyone or restricted (e.g., permissioned or whitelisted). Consider if there should be barriers to entry, such as financial commitments or hardware requirements, which can vary significantly between protocols like Solana and Ethereum.

2.2 Step 2: Designing Verification Mechanisms

- Accuracy of Performance: Establish criteria to determine if an operator has fulfilled their responsibilities correctly. This might include automated checks or community reviews.

- Corruption Scenarios: Identify potential corruption scenarios specific to the roles and functions of the operators. What forms of corruption could occur, and how could they impact the network?

- Dispute Frequency and Resolution: Estimate how often disputes might arise and outline a process for resolution. Consider using mechanisms like optimistic proof or dispute resolution systems where actions are presumed valid unless challenged.

- Finality of Verification: Define the desired level of finality for verifications—how conclusive and irreversible should the validation of actions be?

2.3 Step 3: Designing Penalty Mechanisms

- Assessment of Corruption Risk: Continuously evaluate the corruption risk by analyzing the potential gains from corrupt activities versus the losses from penalties and lost opportunities. Implement a risk-reward framework to understand the attractiveness of corruption.
- Penalty Structures: Develop clear, stringent penalty mechanisms for deterrence, including slashing staked or delegated collateral, temporary or permanent exclusion from the network, and financial penalties. Consider also the impact of non-financial penalties like damage to social reputation.

3 Assessing Corruption Risk

Assessing corruption risk and implementing effective surveillance mechanisms are crucial for system security. This can be done by benchmarking potential earnings and costs for all possible corruption scenarios, and determining the relative cost-to-earnings ratio as a risk benchmark.

3.1 Benchmarking Earnings and Costs

Analyze the risk of corruption by comparing potential earnings from fraudulent activities against legitimate earnings. Direct gains may include profits from fraudulent transactions or system manipulation, while indirect gains, such as increased influence or control over the protocol, can lead to price dumping or market manipulation. Compare these corrupt gains against opportunity cost earnings, including block rewards, transaction fees, and other protocol incentives. Conduct a risk-reward analysis, weighing the severity and probability of punitive outcomes against the benefits of corrupt actions.

Conclusion

Operator network design is crucial to network security and success. At Anthias Labs, to support our AVS partners, we are building monitoring tools, such as real-time surveillance to assess operator activities and detect deviations from expected behavior. If you are interested in working with us, please refer to the contact information on our site.