LOOKING FOR ATTENTION: RANDOMIZED ATTENTION TEST DESIGN FOR VALIDATOR MONITORING IN OPTIMISTIC ROLLUPS

A PREPRINT

Suhyeon Lee
Tokamak Network
suhyeon@tokamak.network
School of Cybersecurity
Korea University
orion-alpha@korea.ac.kr

June 12, 2025

ABSTRACT

Optimistic Rollups (ORUs) significantly enhance blockchain scalability but inherently suffer from the verifier's dilemma, particularly concerning validator attentiveness. Current systems lack mechanisms to proactively ensure validators are diligently monitoring L2 state transitions, creating a vulnerability where fraudulent states could be finalized. This paper introduces the Randomized Attention Test (RAT), a novel L1-based protocol designed to probabilistically challenge validators in ORUs, thereby verifying their liveness and computational readiness. Our game-theoretic analysis demonstrates that an Ideal Security Equilibrium, where all validators are attentive and proposers are honest, can be achieved with RAT. Notably, this equilibrium is attainable and stable with relatively low economic penalties (e.g., under $1000) for non-responsive validators and a low attention test frequency (e.g., under 1% per epoch). RAT thus provides a crucial, practical mechanism to enforce validator diligence, fortifying the overall security and integrity of ORU systems with minimizing additional costs.

Keywords attention test · Ethereum · economics · layer 2 · mechanism design · optimistic rollup · security

1 Introduction

Layer 2 scaling solutions, particularly Optimistic Rollups (ORUs), have become crucial for enhancing blockchain throughput and reducing transaction costs while inheriting the security of the underlying Layer 1 (L1). ORUs achieve scalability by processing transactions off-chain and posting only essential data and state commitments to L1. Their security relies on the "optimistic" assumption that state transitions posted by the proposer are valid, backed by a challenge period during which any validator can submit a Fraud Proof to dispute invalid states.

However, the fundamental security of ORUs hinges critically on the assumption that there exists at least one honest and diligent validator actively monitoring the L1, downloading L2 data, re-executing transactions, and submitting Fraud Proofs when necessary. This assumption is potentially fragile due to the well-known verifier's dilemma [1]. Validators may lack direct incentives to perform the costly verification work, especially if the proposer is perceived as honest or if they believe other validators will bear the cost (free-riding). This dilemma is particularly pronounced in ORUs because, during normal operation where no fraud occurs, validators have minimal, if any, on-chain interactions. This lack of observable activity makes it difficult to ascertain whether validators are genuinely online and performing their verification duties, or merely appearing to be so. Consequently, a lazy but online validator, failing to perform actual verification, creates a significant security vulnerability, as a malicious proposer could potentially finalize fraudulent states unchallenged. Existing liveness checks (e.g., network pings) are insufficient as they do not ascertain a validator's capability or willingness to perform the core verification tasks.

While the concept of an "attention test" to proactively verify validator engagement has been discussed within the blockchain community for several years as a potential solution, concrete protocol designs and analyses of their practical

implications have been notably absent. To address this critical gap, we propose RAT (Randomized Attention Test), the first comprehensive attention test mechanism specifically designed for Optimistic Rollups. RAT introduces probabilistic, individually-targeted challenges integrated into the proposer's state commitment process. When triggered, a randomly selected validator is required to independently compute the state transition and engage in a scheme with the proposer within a strict time limit. Failure to respond correctly or timely results in direct economic penalties, thus providing individual economic incentives for validators to maintain operational attentiveness.

Our main contributions are as follows:

- To the best of our knowledge, we are the first to detail a practical attention challenge mechanism, termed Randomized Attention Test (RAT), specifically designed to address the lazy validator problem in Optimistic Rollups.
- We present a comprehensive game-theoretic analysis demonstrating that RAT effectively incentivizes rational proposers and validators to converge towards an Ideal Security Equilibrium, where all validators remain attentive and proposers act honestly even under the weak randomness environment.
- Our findings indicate that this Ideal Security Equilibrium is attainable with RAT under realistic conditions, imposing only minimal operational overhead on the ORU system, primarily through modest, probabilistically applied economic penalties and low challenge frequencies.

The remainder of this paper is structured as follows. Section 2 briefly review the related works. Section 3 defines the problem scope. Section 4 introduces the design of the RAT protocol, a novel attention test mechanism. Section 5 explains the game model of validators and proposer strategies in RAT. Based on the model, Section 6 analyzes the conditions and robustness of the ideal strategic equilibrium by RAT. Section 7 investigates implications of the parameter setting and practicality for RAT based on the game theoretic results. Section 8 discusses related issues, and Section 9 concludes the paper.

2 Related Works

The development of optimistic rollups has spurred a wide array of research aimed at enhancing their economic viability, security, and operational scalability. This section reviews key areas of this prior work, highlighting common assumptions and identifying a nuanced challenge related to proactive validator oversight, which our present study seeks to address.

2.1 Economic Analysis in Rollups

Early works in the economic analysis of rollups have focused on establishing robust security guarantees through well-aligned incentive mechanisms. Tas et al. [2] propose a framework for *accountable safety* in rollups, which emphasizes the need for designs that hold participants economically accountable for misbehavior. Li [3] further explores the security of optimistic blockchain mechanisms, highlighting the importance of ensuring that validator incentives are strong enough to deter malicious actions. In a similar vein, Mamageishvili and Felten [4] analyzes incentive schemes for rollup validators, including a discussion of strategies under a conceptual attention test. These foundational studies establish the importance of economic incentives for honest participation and for penalizing malicious behavior. However, a subtle aspect of the verifier's dilemma persists: even with robust penalties for detected fraud, the incentive for an individual validator to incur the cost of active verification during periods of perceived sequencer honesty can be weak, especially if they assume others will bear this cost. This raises questions about the consistent, proactive engagement of validators, a prerequisite for these incentive systems to function effectively when fraud does occur.

2.2 Data Availability and Its Cost

Given that Optimistic Rollups are designed as a layer-2 scaling solution, minimizing the costs associated with data availability (DA) is a critical concern. Several studies have tackled this problem from different angles. Palakkal et al. [5] provide a systematization of compression techniques in rollups, outlining methods and inefficiencies in some rollups' practice. Mamageishvili and Felten [6] propose efficient rollup batch posting strategies on the base layer using call data, while Crapis et al. [7] offer an in-depth analysis of EIP-4844 economics and rollup strategies, including blob cost sharing. Complementary empirical investigations by Heimbach and Milionis [8], Huang et al. [9], and Park et al. [10] further document the inefficiencies in current DA cost structures and examine their impact on rollup transaction dynamics and consensus security. Lee [11] investigates blob sharing as a potential remedy for the dilemma faced by small rollups in the post-EIP-4844 era. These advancements are crucial for ensuring that validators *can* access the necessary data affordably and efficiently. Yet, the availability and affordability of data do not inherently guarantee that

every validator will consistently download, process, and verify it, especially if there are no direct, periodic checks on their actual engagement with this data beyond dispute scenarios.

2.3 Fraud Proof and Dispute Resolution Protocols

Prior research on fraud proofs in Optimistic Rollups has primarily focused on ensuring rapid dispute resolution and robust liveness of the dispute game itself. For example, BoLD [12] and Dave [13] propose protocols that optimize the dispute game to achieve low delays and cost-efficient on-chain verification. Concurrently, Berger et al. [14] analyze economic censorship dynamics to guarantee that disputes are resolved even under adversarial conditions. While these protocols are vital for the *reactive* security of ORUs by providing mechanisms to challenge and penalize identified fraudulent state transitions, some analyses, such as Lee [15], point to potential exploitabilities in current validator incentive structures by malicious proposers. This underscores that the effectiveness of the entire reactive framework ultimately hinges on the presupposition that at least one honest and diligent validator is actively monitoring the system, sufficiently motivated, and ready to initiate such a dispute. Our work focuses on this antecedent aspect: fostering an environment where validators are proactively and verifiably attentive.

3 Problem Statement and Design Goals

This section first outlines the core problems within Optimistic Rollups that necessitate the development of an attention test mechanism. Subsequently, we establish key design goals that such a mechanism should satisfy, which will serve as an evaluative framework for the solution proposed in this paper.

3.1 The Validator Liveness and Correctness Problem

The security model of Optimistic Rollups critically depends on the assumption that there exists at least one honest Validator who is actively monitoring the L1, downloading L2 data, executing state transitions, and submitting Fraud Proofs when necessary. However, several challenges arise:

- **The Lazy Validator Problem:** A validator might remain online and appear active but fail to perform the computationally intensive task of re-executing state transitions. Such a validator would not detect invalid state roots posted by a malicious Sequencer, undermining the rollup's security.
- **Limitations of Simple Liveness Checks:** Basic liveness checks (like monitoring network connectivity) are insufficient as they do not verify if the validator possesses the necessary data or has performed the required computations.
- **Reactive Nature of Fraud Proofs:** The Fraud Proof mechanism is inherently reactive. It only comes into play **after** an invalid state has been posted. There is no built-in mechanism to proactively ensure that validators are continuously ready and capable of performing their verification duties **before** a potential fraud occurs.

Therefore, a mechanism is needed to proactively and periodically verify that Validators are not only online (live) but also correctly processing the L2 state (correctness) and have access to the necessary data (data availability). We term such a mechanism an *Attention Test*.

3.2 Design Goals for Attention Tests

An effective attention test mechanism for ORU validators should possess the following properties:

1. **Verifiability:** The test should check the validator's ability to perform core validation tasks (like state computation), and the results must be objectively verifiable on L1.
2. **Unpredictability:** The timing of the test and the selection of the challenged Validator should be unpredictable to prevent gaming the system.
3. **Fairness:** All active validators should have a chance of being selected for a test, without bias towards specific individuals or groups.
4. **Efficiency:** The mechanism should incur low overhead in terms of L1 gas costs and latency added to the normal rollup operation. (This is often a trade-off against security).

The subsequent sections will introduce the RAT (Randomized Attention Test) protocol, designed to address the verifier's dilemma and validator liveness challenges outlined previously. We will then detail its operation and analyze how it fulfills the design goals established.

4 The RAT Protocol

To address the challenges outlined in Section 3, specifically the verifier's dilemma and the need for proactive validator monitoring, we propose the **R**andomized **A**ttention **T**est (RAT) protocol. RAT integrates a probabilistic challenge mechanism into the routine state commitment process of the optimistic rollup, leveraging the existing structure on the L1 blockchain to minimize additional costs.

4.1 System Model and Assumptions

This study operates within a standard ORU framework, which comprises the following key entities and components:

- An L1 Blockchain (e.g., Ethereum): It hosts the main ORU smart contract $\mathcal{SC}$. This contract handles state root submissions, dispute resolution logic, staking, and the RAT protocol execution.
- Optimistic rollup (L2): It processes transactions off-chain and periodically posts state commitments and transaction data (for data availability) to the L1 blockchain.
- Proposer (P): A single, designated entity responsible for ordering L2 transactions, computing the resulting state transitions, and submitting the corresponding state commitments to the L1 contract.
- Validators ($\mathcal{V}$): A set of N registered validators $\mathcal{V} = \{v_1, v_2, ..., v_N\}$, each identified by an L1 address and required to stake enough collateral managed by the L1 contract.

In addition to the system model, the protocol relies on the following standard security and operational assumptions:

- **L1 Security:** The underlying L1 blockchain provides security guarantees, including finality for confirmed transactions and resistance to censorship.
- **Cryptographic Primitives:** The cryptographic hash function $H(\cdot)$ (e.g., Keccak-256) used is collision-resistant and behaves like a random oracle for unpredictability purposes.
- **Network Model (Partial Synchrony):** The network ensures that messages broadcast by honest parties are eventually delivered to other honest parties within a finite, bounded time, although the exact bound may not be known a priori or may fluctuate.
- **Homogeneous Validators and Linear Costs:** The validators are homogeneous in terms of risk preference and face a common, linear cost for operation. This cost largely reflects metered, usage-based computational expenses (e.g., cloud infrastructure) proportional to their verification activities.
- **Dispute Game Security:** The dispute game of ORU systems are safe and live. If a validator disputes an incorrect state commitment by a proposer, the validator wins the dispute game. If a validator dispute a correct state commitment by a proposer, the proposer wins the dispute game.

4.2 Protocol Flow and Challenge Trigger

The RAT protocol introduces a probabilistic verification step into the standard optimistic rollup procedure (Commitment $\rightarrow$ Dispute window $\rightarrow$ Finalization) for state commitments. It aims to ensure validator attentiveness without significantly altering the core ORU dispute window.

Figure 1 illustrates this integrated flow. Deviating from the standard Optimistic Rollup procedure, following an L2 state transition, the proposer interacts with the L1 smart contract. With probability $1 - \pi_a$, the proposer posts its state commitment directly. Conversely, with probability π_a, the proposer initiates an attention test by submitting an "attention puzzle" to the L1 smart contract. The design of the attention puzzle requires to two key properties:

- **Commitment Binding:** The attention puzzle must be uniquely and verifiably bound to the specific state commitment (σ_P) being attested in that epoch. This ensures the test is relevant to the current state transition. This property is key to ensuring that the ORU's dispute window does not need to be extended for the attention test.
- **Knowledge Proof:** Only a validator possessing the correct L2 state transition information should be able to derive and submit the valid solution to the puzzle.

If an attention test is triggered, a target validator is selected by $\mathcal{SC}$ from the registered validator pool. Based on the target validator's response (or lack thereof) within a predefined time window, there are three possible outcomes:

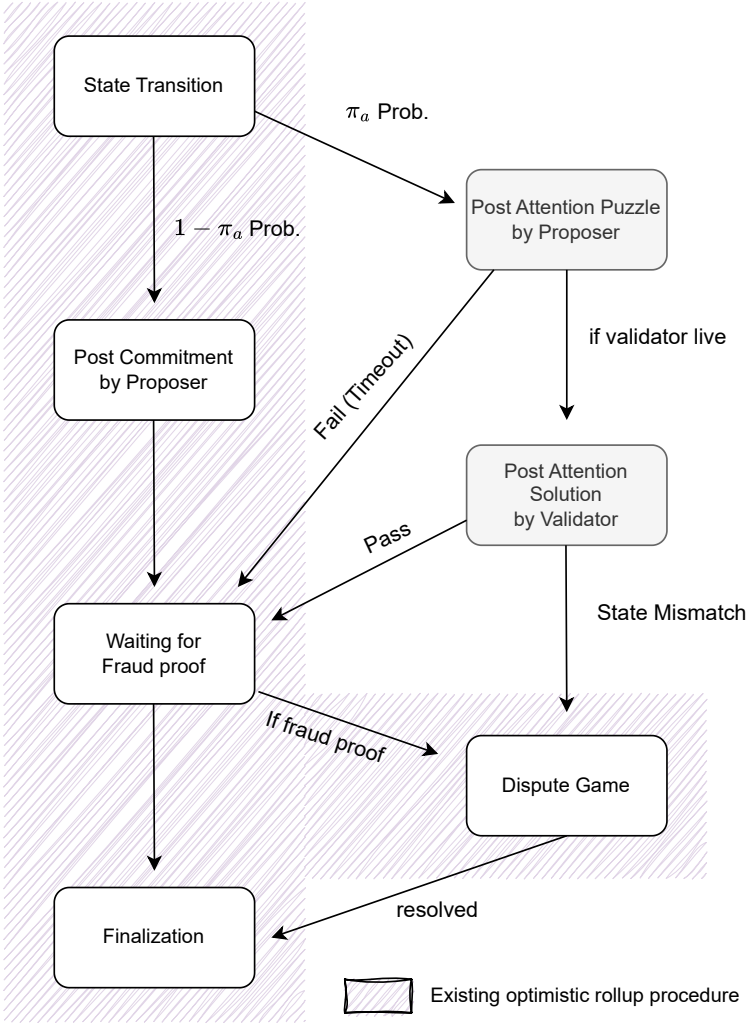

Figure 1: Overview of RAT Integrated with Optimistic Rollup Procedure

- **Pass:** The validator submits a solution and verified for the proposer's attention test in the contract. Then, the standard ORU procedure continues, awaiting potential fraud proofs during the remainder of the dispute window.

- **Fail (Timeout):** The validator does not submit any solution within the attention test timing window. The validator gets a specific penalty to the deposit. Then, the standard ORU procedure continues, awaiting potential fraud proofs during the remainder of the dispute window.

- **State Mismatch:** The validator submits a solution and the smart contract determines this solution does not correctly correspond to the attention puzzle binding to the specific state transition. This mismatch implies that at least one of them (the proposer and the target validator) has incorrect state transition information. Therefore, the L1 smart contract proceeds the dispute game to fix this issue.

The subsequent subsection will detail the specific mechanisms for implementing this RAT protocol flow and ensuring the aforementioned properties.

4.3 Attention Test Mechanism

Our mechanism is designed to be integrated into existing ORU protocols with minimal disruption to their core logic. A key aspect of this efficiency is leveraging the proposer's standard operational flow. In typical ORUs, the proposer periodically submits a state commitment to the L1 smart contract ($\mathcal{SC}$). This commitment is usually the state root (σ_P) of a Merkle Tree (or a similar structure like a Sparse Merkle Tree) that compactly represents the entire L2 state. Being a root hash, σ_P itself does not reveal the underlying L2 data but attests to it.

The RAT mechanism utilizes this existing state commitment process. The proposer always submits its computed state root σ_P along with associated metadata like the corresponding L2 block number (`L2blocknum`) to $\mathcal{SC}$, irrespective of whether an attention test is triggered for that epoch. It is $\mathcal{SC}$ that then probabilistically determines if an attention test should be initiated for this specific σ_P. This determination, along with the selection of a target validator, is made deterministically by $\mathcal{SC}$ based on the submitted σ_P and other on-chain data (e.g., current L1 block hash)[1].

If an attention test is triggered, the "attention puzzle" posed to the target validator is not to re-submit the entire state root (as the proposer has already submitted σ_P). Instead, to prove attentiveness more granularly yet efficiently, the target validator is required to submit the two direct child components, left and right child nodes/hashes, denoted (L_V, R_V), that constitute the state root σ_P. That is, if $H_{\text{tree}}(L_V, R_V) = C_P$, then the solution is valid. Only a validator that has correctly processed the L2 state transitions leading to σ_P would know these immediate constituent components. This two-component solution forms the core of our RAT interaction, depicted in Figure 2.

Phase 1: Proposer's State Commitment and Potential Attention Test Trigger by $\mathcal{SC}$ The Proposer (P) computes the L2 state root σ_P and the associated `L2blocknum` for the current epoch. P then sends an L1 transaction to $\mathcal{SC}$ containing σ_P and `L2blocknum`. This submission is identical to the standard ORU procedure where P posts its state commitment.

Upon receiving this transaction, $\mathcal{SC}$ first records σ_P and `L2blocknum`. Then, $\mathcal{SC}$ uses σ_P and a source of on-chain data to:

1. Probabilistically decide if an attention test is triggered with target probability π_a.
2. If triggered, probabilistically select a target validator v_i from the registered validator set.

If an attention test is triggered for a certain validator, $\mathcal{SC}$ emits an attention event. This event contains the Proposer's submitted state root σ_P, `L2blocknum`, and the target validator's address. This event signals the validator to respond and initiates a response timer, T_{response}. If no test is triggered, the process continues as a standard ORU state commitment.

Phase 2: Target Validator's Attention Solution Submission Upon recognizing the attention event specifically targeting it, the target validator should act. Its core task, which it should have already performed if diligent, is to independently process all L2 transactions up to `L2blocknum` to derive the state root and, crucially for this test, the two direct child components (L_V, R_V) that hash to form the Proposer's submitted σ_P.

Within the T_{response} window, the validator must submit its computed (L_V, R_V) as the "attention solution" to $\mathcal{SC}$ via an L1 transaction. Upon receiving this solution, $\mathcal{SC}$ performs the verification: it computes $H(L_V, R_V)$ using the L2-specific tree hashing function and compares the result against the σ_P previously submitted by the proposer and recorded by $\mathcal{SC}$ for this test. This comparison, along with timeout handling, determines the outcome of the Attention Test (Pass, Fail-Timeout, or State Mismatch).

In this structure, we can expect that the L1 gas cost will not be significant for the successful attention test outcome (Pass). It requires only on additional L1 transaction with two hash strings, one hash calculation, and one comparison of two hash strings.

5 Game Model

In this section, we develop a game-theoretic model, a game $\mathcal{G}$, before analyzing the strategic interactions by the RAT protocol.

[1] L1 blockhash is a manipulative value. However, as Obervation 2 in Section 6.3, manipulating this randmoness for not triggering attention tests for fraud does not affect the intended security of the protocol.

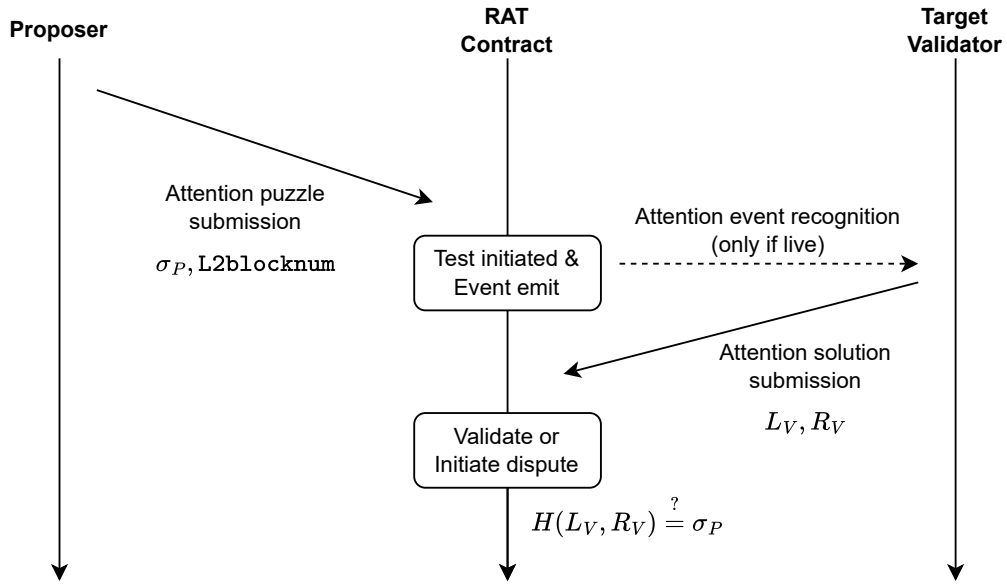

Figure 2: Interaction flow for an attention test

5.1 Players and Strategies

The game $\mathcal{G}$ involves two strategic player types:

- **Proposer (P):** This player is responsible for submitting L2 state commitments to the L1 contract. The proposer chooses a probability $\pi_p \in [0, 1]$ of submitting an honest state root (action H). Consequently, the probability of submitting a fraudulent state root (action F) is $1 - \pi_p$.
- **Representative Validator (V):** This player represents one of the N homogeneous validators. Each validator i chooses a probability $\pi_{v,i} \in [0, 1]$ of being online and actively verifying state transitions during a given period. The probability of being offline (not verifying) is $1 - \pi_{v,i}$. Due to the homogeneity assumption in Section 4.1, we will often focus on a symmetric strategy profile where $\pi_{v,i} = \pi_v$ for all i.

5.2 Key Parameters

The economic environment of game $\mathcal{G}$ is defined by the following parameters and understanding. A parameter written in uppercase signifies a large value.

- **Validator's Payoffs and Costs:**
  - f_v: Standard fee received by a validator for its validation activities unless the system fails to detect a fraud state commitment.
  - c_m: Marginal cost incurred by a validator *only when it is online* and actively verifying.
  - c_v: Operating cost incurred by a validator who chose π_v, and $c_v = \pi_v c_m$.
  - r_v: Total reward pool available if a fraudulent state root submitted by the proposer is successfully challenged.
  - r_v^*: The expected share of r_v for an individual attentive validator that successfully challenges a fraud. We define $r_v^* = \frac{r_v}{1+(N-1)\bar{\pi}_v}$, where $\bar{\pi}_v$ is the average probability that any other given validator in the system is online. In a symmetric equilibrium, $\bar{\pi}_v = \pi_v$.
  - c_{off}: Penalty incurred by a validator if it is selected for an Attention Test and fails to respond correctly (i.e., is offline or unresponsive).
  - C_{fail}: Penalty incurred by each validator if a proposer's fraud goes undetected and leads to a system failure.

Table 1: State-Dependent Payoff Structure for Validator and Proposer

Validator System State			**Proposer Action**	
Network Status	Validator	Attention Test	Honest	Fraud
Online	Online	Pass	$(f_v - c_v,\, f_p)$	$(f_v - c_v + r_v^*,\, -C_{\text{fraud}})$
		-	$(f_v - c_v,\, f_p)$	$(f_v - c_v + r_v^*,\, -C_{\text{fraud}})$
	Offline	Timeout	$(f_v - c_v - c_{\text{off}},\, f_p)$	$(f_v - c_v - c_{\text{off}},\, -C_{\text{fraud}})$
		-	$(f_v - c_v,\, f_p)$	$(f_v - c_v,\, -C_{\text{fraud}})$
Offline	Offline	Timeout	$(f_v - c_v - c_{\text{off}},\, f_p)$	$(-c_{\text{off}} - C_{\text{fail}},\, f_p + R_{\text{fraud}})$
		-	$(f_v - c_v,\, f_p)$	$(-C_{\text{fail}},\, f_p + R_{\text{fraud}})$

- **Proposer's Payoffs and Costs:**
  - f_p: Standard fee received by the proposer for submitting an honest state root.
  - C_{fraud}: Penalty incurred by the proposer if its fraudulent submission is detected and successfully challenged.
  - R_{fraud}: Illicit profit gained by the proposer if its fraudulent submission goes undetected and is finalized.
- **System Parameters:**
  - N: The total number of registered validators in the system.
  - π_a: The intended probability that an Attention Test is triggered for a given state submission. If triggered, one validator is chosen uniformly at random to be tested, so an individual validator is tested with probability π_a/N.

5.3 Payoff Structure and Utility Calculation Preliminaries

The payoffs for each player depend on their own actions, the actions of the other player(s), and the outcomes of probabilistic events like attention tests and fraud detection. We will use a detailed state-dependent payoff structure as Table 1 to derive the expected utilities. The core interactions determining payoffs include:

- **Fraud Detection:** An online validator is assumed to always detect a fraudulent submission (F) by the proposer. An offline validator never detects fraud.
- **Attention Test Outcome:** If an attention test is triggered (π_a) and a specific validator is challenged (probability $1/N$ for that validator):
  - If the challenged validator is online, it passes; no direct reward[2] or penalty from the test itself beyond normal operation.
  - If the challenged validator is offline, it incurs penalty c_{off} by timeout.
- **Consequences of Proposer's Fraud:**
  - If fraud is detected (e.g., by at least one online validator): The proposer incurs C_{fraud}. The successful challenger(s) share r_v (leading to r_v^* for an individual).
  - If fraud is not detected: The proposer gains R_{fraud}. All validators (or at least those implicated by the system failure rules) may incur C_{fail}.

Based on these interactions and the payoff table, we can calculate the expected utility for each player.

6 Equilibrium Analysis of the RAT Mechanism

This section analyzes the strategic interactions between the proposer and validators within the RAT-enhanced Optimistic Rollup framework. We first derive the expected utility functions for each player and then identify the conditions under which an Ideal Security Equilibrium – where all validators are attentive and the proposer is honest – can be achieved and maintained. We also examine the robustness of this equilibrium to potential exploitation of the weak randomness in RAT.

[2]We can consider f_v as an indirect reward for validation.

6.1 Expected Utility Functions

Firstly, Proposition 1 allows us to focus on symmetric Nash Equilibria, where $\pi_{v,i} = \pi_v$ for all validators i, and thus the expected online probability of other validators $\bar{\pi}_v$ can be set to π_v.

Proposition 1 (Symmetric Validator Strategy in Nash Equilibrium)**.** *Given the assumption of homogeneous validators, all validators will adopt the same strategy $\pi_v \in [0, 1]$ in any Nash Equilibrium of the game $\mathcal{G}$.*

Proof Sketch. This can be shown by contradiction. If two homogeneous validators adopted different strategies in equilibrium, at least one would have an incentive to switch, as they face identical optimization problems given the strategies of others. This allows us to focus on symmetric Nash Equilibria, where $\bar{\pi}_v = \pi_v$. □

6.1.1 Validator's Expected Utility

The expected utility for a representative validator depends on their own strategy (Online or Offline), the proposer's strategy (Honest or Fraudulent, with probability π_p of being honest), and the aggregate behavior of other validators (with probability π_v of being online, due to Proposition 1).

- **If the validator chooses to be Online (On):** They incur cost $c_v = c_m \pi_v = c_m \cdot 1 = c_m$. If the proposer is fraudulent (probability $1 - \pi_p$), they receive an expected reward share $r_v^* = r_v/(1 + (N - 1)\pi_v)$ for detecting fraud.
$$U_v(\mathsf{On}|\pi_p, \pi_v) = f_v - c_m + (1 - \pi_p) r_v^* \tag{1}$$
- **If the validator chooses to be Offline (Off):** They do not incur any operational cost ($c_v = c_m \pi_v = c_m \cdot 0 = 0$). They face a probability π_a/N of being selected for an Attention Test and incurring penalty c_{off} if they fail (regardless of proposer's action in this baseline model). If the proposer is fraudulent and the fraud is not detected by any other validator (probability $(1 - \pi_v)^{N-1}$), they incur penalty C_{fail}.
$$U_v(\mathsf{Off}|\pi_p, \pi_v) = \pi_p f_v - \frac{\pi_a}{N} c_{\text{off}} - (1 - \pi_p)(1 - \pi_v)^{N-1} C_{\text{fail}} \tag{2}$$

The overall expected utility for a validator choosing to be online with probability π_v (their own choice) is:

$$E[U_v](\pi_p, \pi_v) = \pi_v U_v(\mathsf{On}|\pi_p, \pi_v) + (1 - \pi_v) U_v(\mathsf{Off}|\pi_p, \pi_v) \tag{3}$$

6.1.2 Proposer's Expected Utility

The proposer's expected utility depends on their strategy (Honest or Fraudulent) and the validators' collective online probability (π_v).

- **If the proposer chooses to be Honest (H):**
$$U_p(\mathsf{H}) = f_p \tag{4}$$
- **If the proposer chooses to be Fraudulent F:**
$$U_p(\mathsf{F}|\pi_v) = (1 - (1 - \pi_v)^N)(-C_{\text{fraud}}) + (1 - \pi_v)^N (f_p + R_{\text{fraud}}) \tag{5}$$

The overall expected utility for a proposer choosing to be honest with probability π_p is:

$$E[U_p](\pi_p, \pi_v) = \pi_p U_p(\mathsf{H}) + (1 - \pi_p) U_p(\mathsf{F}|\pi_v) \tag{6}$$

6.2 Ideal Security Equilibrium

We are interested in an Ideal Security Equilibrium where all validators are online and attentive ($\pi_v = 1$), and the proposer acts honestly ($\pi_p = 1$). For this to be a Nash Equilibrium, neither party should have an incentive to unilaterally deviate.

6.2.1 Existence Conditions

We derive the conditions by checking each player's best response under the assumption that the other player is adhering to the ideal strategy.

Condition 1: Validator's Best Response to an Honest Proposer Given the proposer is honest ($\pi_p = 1$), a validator will choose to be Online ($\pi_v = 1$) if $U_v(\mathsf{On}|\pi_p = 1, \pi_v = 1) \geq U_v(\mathsf{Off}|\pi_p = 1, \pi_v = 1)$. (Here, $\pi_v = 1$ in the conditioning arguments refers to other validators also being online).

From Eq. 1, $U_v(\mathsf{On}|\pi_p = 1, \pi_v = 1) = f_v - c_m$. From Eq. 6.1.1, $U_v(\mathsf{Off}|\pi_p = 1, \pi_v = 1) = f_v - \frac{\pi_a}{N} c_{\text{off}} - (1 - 1)(1 - 1)^{N-1} C_{\text{fail}} = f_v - \frac{\pi_a}{N} c_{\text{off}}$. Thus, the condition is $f_v - c_m \geq f_v - \frac{\pi_a}{N} c_{\text{off}}$, which simplifies to:

$$c_m \leq \frac{\pi_a}{N} c_{\text{off}} \tag{7}$$

Condition 2: Proposer's Best Response to Attentive Validators Given all validators are online ($\pi_v = 1$), the proposer will choose to be Honest ($\pi_p = 1$) if $U_p(\mathsf{H}) \geq U_p(\mathsf{F}|\pi_v = 1)$. From Eq. 4, $U_p(\mathsf{H}) = f_p$. From Eq. 5, $U_p(\mathsf{F}|\pi_v = 1) = (1 - (1 - 1)^N)(-C_{\text{fraud}}) + (1 - 1)^N (f_p + R_{\text{fraud}}) = -C_{\text{fraud}}$. Thus, we get the below condition which is always true:

$$f_p \geq -C_{\text{fraud}} \tag{8}$$

Theorem 2 (Existence of Ideal Security Equilibrium)**.** *The Ideal Security Equilibrium, where all validators are attentive ($\pi_v = 1$) and the proposer is honest ($\pi_p = 1$), exists if and only if the following condition:*

$$c_m \leq \frac{\pi_a}{N} c_{off} \tag{9}$$

Observation 1. The primary condition for Ideal Security reveals a crucial insight: Validator's attentiveness in an honest environment is driven not by direct rewards (f_v, r_v), but by the economic trade-off involving operational costs (c_m) versus the risk and penalty of failing an Attention Test (π_a, c_{off}, N). Consequently, careful calibration of the RAT mechanism's parameters is crucial for guiding the system towards this desired equilibrium.

6.2.2 Stability Analysis for Ideal Security

We now examine whether players have an incentive to deviate from the pure strategies ($\pi_v = 1, \pi_p = 1$) that constitute the Ideal Security Equilibrium, assuming the conditions from Theorem 2 are met.

Validator's Incentive to Maintain $\pi_v = 1$ (given $\pi_p = 1$) The validator's expected utility when the proposer is honest ($\pi_p = 1$), as a function of their own online probability π_v, is

$$E[U_v](\pi_v|\pi_p = 1) = \pi_v U_v(O|\pi_p = 1, \pi_v) + (1 - \pi_v) U_v(\mathsf{Off}|\pi_p = 1, \pi_v). \tag{10}$$

Substituting the conditional utilities for $\pi_p = 1$, we get

$$E[U_v](\pi_v|\pi_p = 1) = \pi_v (f_v - c_m) + (1 - \pi_v)(f_v - \frac{\pi_a}{N} c_{\text{off}}). \tag{11}$$

Taking the derivative with respect to π_v:

$$\left.\frac{dE[U_v]}{d\pi_v}\right|_{\pi_p=1} = (f_v - c_m) - (f_v - \frac{\pi_a}{N} c_{\text{off}}) = \frac{\pi_a}{N} c_{\text{off}} - c_m \tag{12}$$

If the condition for the ideal security equilibrium in Theorem 2 holds:

- If $c_m < \frac{\pi_a}{N} c_{\text{off}}$, then $\frac{dE[U_v]}{d\pi_v} > 0$. This implies that $\pi_v = 1$ is the unique optimal strategy.
- If $c_m = \frac{\pi_a}{N} c_{\text{off}}$, then $\frac{dE[U_v]}{d\pi_v} = 0$. The validator is indifferent.

Proposer's Incentive to Maintain $\pi_p = 1$ (given $\pi_v = 1$) The proposer's expected utility when all validators are online ($\pi_v = 1$), as a function of their own honesty probability π_p, is $E[U_p](\pi_p|\pi_v = 1) = \pi_p f_p + (1 - \pi_p)(-C_{\text{fraud}})$. Taking the derivative with respect to π_p:

$$\left.\frac{dE[U_p]}{d\pi_p}\right|_{\pi_v=1} = f_p - (-C_{\text{fraud}}) = f_p + C_{\text{fraud}} \tag{13}$$

Given Condition C2 ($f_p \geq -C_{\text{fraud}}$), and standard assumptions $f_p \geq 0, C_{\text{fraud}} > 0$, this derivative is typically strictly positive, making $\pi_p = 1$ the unique optimal strategy.

Thus, when the conditions for the Ideal Security Equilibrium are (strictly) met, both players have a clear incentive to adhere to their pure strategies.

Table 2: State-Dependent Payoff Structure for Validator and Advanced Proposer Model

Validator System State			**Proposer Action**	
Network Status	Validator	Attention Test	H (Honest)	F (Fraud)
Online	Online	Pass	$(f_v - c_v,\, f_p)$	-
		-	$(f_v - c_v,\, f_p)$	$(f_v - c_v + r_v^*,\, -C_{\text{fraud}})$
	Offline	Timeout	$(f_v - c_v - c_{\text{off}},\, f_p)$	-
		-	$(f_v - c_v,\, f_p)$	$(f_v - c_v,\, -C_{\text{fraud}})$
Offline	Offline	Timeout	$(f_v - c_v - c_{\text{off}},\, f_p)$	-
		-	$(f_v - c_v,\, f_p)$	$(-C_{\text{fail}},\, f_p + R_{\text{fraud}})$

6.3 Robustness to Exploited Weak Randomness

We now consider a more adversarial scenario where a malicious proposer can exploit weaknesses in RAT's randomness generation. Specifically, we assume the proposer can ensure that **no attention test is triggered when they submit a fraudulent state root.** This allows us to test the robustness of the Ideal Security Equilibrium under stronger adversarial capabilities.

6.3.1 Analysis under Proposer's RAT Evasion

As presetned in Table 2, the core modification is that if the proposer is fraudulent $(1 - \pi_p)$, the effective attention test probability π_a becomes 0 for that instance concerning RAT penalties for the validator.

Validator's Expected Utility (re-evaluation for Ideal Equilibrium Check) When checking the validator's best response to an honest proposer ($\pi_p = 1$), the proposer is, by definition, honest. Thus, the assumption about RAT evasion during fraud does not alter the validator's utility calculation in this specific step. $U_v(\mathsf{On}|\pi_p = 1, \pi_v = 1) = f_v - c_m$. $U_v(\mathsf{Off}|\pi_p = 1, \pi_v = 1) = f_v - \frac{\pi_a}{N} c_{\text{off}}$ (since proposer is honest, π_a applies). The condition $c_m \leq \frac{\pi_a}{N} c_{\text{off}}$ remains the same.

Proposer's Expected Utility (re-evaluation for Ideal Equilibrium Check) When checking the proposer's best response to attentive validators ($\pi_v = 1$), if the proposer considers being fraudulent, they know they can evade RAT. However, attentive validators will still detect the fraud through their own verification, leading to the penalty $-C_{\text{fraud}}$. The evasion of RAT does not change the outcome of fraud detection by diligent validators. $U_p(\mathsf{H}) = f_p$. $U_p(\mathsf{F}|\pi_v = 1, \text{RAT evasion during fraud}) = -C_{\text{fraud}}$. The condition $f_p \geq -C_{\text{fraud}}$ remains the same.

Corollary 3 (Robustness of Ideal Security Equilibrium under Weak Randomness)**.** *The conditions for the Ideal Security Equilibrium stated in Theorem 2 ($c_m \leq \frac{\pi_a}{N} c_{off}$ and $f_p \geq -C_{fraud}$) remain sufficient even if a malicious proposer can fully evade the Randomized Attention Test when submitting a fraudulent state root.*

Observation 2. The randomness source for RAT, potentially derived from L1 block data or proposer inputs, might be exploitable. However, even if a malicious proposer successfully evades an attention test *when submitting a fraudulent state*, this exploitation does not alter the conditions required to achieve the Ideal Security Equilibrium. This is because the validator's incentive to be attentive and the proposer's incentive to be honest remain intact.

7 Design for Ideal Security

A key objective for system designers is to foster the **Ideal Security Equilibrium**, where all validators are attentive ($\pi_v^* = 1$) and the Proposer remains honest ($\pi_p^* = 0$). This section discusses how system parameters, particularly the attention test probability (π_a) and the penalty for failing an attention test (C_{off}), can be tuned to achieve this desirable state while considering practical operational constraints.

7.1 Consideration of Parameter Interplay

From Theorem 2, the condition for the Ideal Security Pure Strategy Nash Equilibrium is:

$$c_m \leq \frac{\pi_a C_{\text{off}}}{N} \tag{14}$$

where c_m is the marginal cost of validator attentiveness, π_a is the system-wide probability of an attention test being triggered per epoch, C_{off} is the penalty for a validator failing such a test (if selected), and N is the number of validators.

In a practical rollup deployment, c_m is largely determined by the rollup's transaction throughput, state complexity, and the operational costs of validator infrastructure. Similarly, N is often an emergent property reflecting the desired level of decentralization, or a target set by the system. Therefore, the primary levers for system designers to satisfy Condition (14) are π_a and C_{off}.

A higher π_a increases the likelihood of catching inattentive validators but also incurs greater L1 transaction overhead due to more frequent RAT interactions. Conversely, a higher C_{off} provides a stronger deterrent against inattentiveness, potentially allowing for a lower π_a. The maximum credible value for C_{off} is practically bounded by the validator's staked deposit, D_V (i.e., $C_{\text{off}} \leq D_V$). To minimize system overhead (by minimizing π_a), it is rational to set C_{off} to its maximum credible value, D_V. Rearranging Condition (14) to solve for the minimum required π_a, assuming $C_{\text{off}} = D_V$, we get:

$$\pi_a \geq \frac{c_m N}{D_V} \tag{15}$$

This equation forms the basis for our parameter tuning strategy.

7.2 Estimating Proper Attention Test Frequency

To ground our analysis and understand the practical range for π_a, we first need to estimate a representative marginal cost of attentiveness (c_m) per epoch. The value c_m represents the per-epoch operational cost that a lazy validator might attempt to shirk by not performing its duties (e.g., not re-executing L2 transactions, not maintaining readiness for L1 interactions).

7.2.1 Estimating Marginal Cost of Attentiveness

We can approximate c_m by considering the publicly available data regarding the monthly operational costs of running validator nodes for various prominent Optimistic Rollups. These costs typically encompass server instance specifications (CPU, RAM) and storage requirements. For instance, Table 3 outlines the required specifications and estimated operational costs for full nodes of three representative Optimistic Rollups.

Table 3: Estimated Monthly Operational Costs for Validator Nodes

Rollup	Official Min. Specs	AWS Instance Example	SSD Storage (Min.)	Instance Cost/Mo	Storage Cost/Mo	Total Cost/Mo
OP Mainnet [16]	16GB RAM, Latest CPU, 1.6TB SSD	r5.xlarge (4vCPU/32GB)	1.6TB	$140	$128	$268
Base [17]	32GB RAM (64GB rec.), Multi-core, NVMe, 2~3TB+ SSD	r5.2xlarge (8vCPU/64GB)	3TB	$280	$240	$520
Arbitrum [18]	16GB RAM, 4-core, NVMe, (t3.xlarge rec.)	t3.xlarge (4vCPU/16GB)	1TB	$140	$80	$220

Based on these figures, it is reasonable to estimate that the monthly operational cost for a validator node performing diligent verification could broadly range from approximately $200 to $600. Given this potential range, and adopting a deliberately pessimistic (or perhaps, realistically cautious) stance on operational expenditures for Optimistic Rollups, we set a representative monthly cost of **$600 per validator instance** for our analysis. This higher estimate robustly covers the active validation tasks a lazy validator might avoid.

To convert this monthly cost into a per-epoch marginal cost c_m, we assume an L2 state commitment (and thus a potential RAT challenge window or epoch) occurs every 10 minutes. The number of such epochs in a month (approximating a

month as 30 days) is:

$$\frac{30 \text{ days/month} \times 24 \text{ hours/day} \times 60 \text{ minutes/hour}}{10 \text{ minutes/epoch}} = 4320 \text{ epochs/month}$$

Therefore, the estimated marginal cost of attentiveness per epoch (c_m) is:

$$c_m \approx \frac{\$200 \text{ per month}}{4320 \text{ epochs/month}} \approx \$0.139 \text{ per epoch}$$

This estimated c_m will be used in the subsequent numerical analysis to determine the required attention test probability π_a.

7.2.2 Numerical Analysis on Attention Test Probability

With the estimated $c_m \approx \$0.139$ per epoch, we can now simulate the minimum required system-wide attention test trigger probability (π_a) per epoch, as a percentage. This numerical analysis explores how π_a varies with different levels of the attention test Penalty (C_{off}, which we assume can be set up to the validator's deposit D_V) and for several different numbers of validators (N).

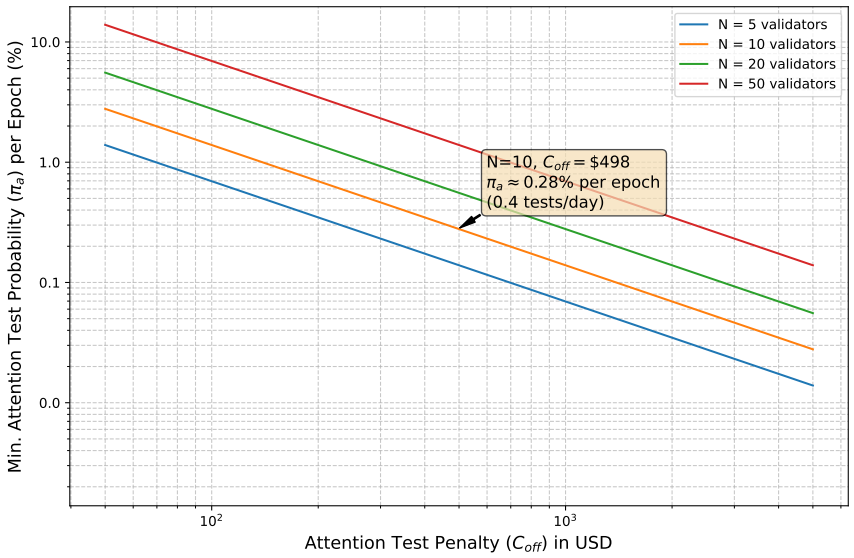

Figure 3: Minimum attention test probability ($\pi_a\%$) per epoch required for Ideal Security, as a function of attention test Penalty (C_{off}) and Number of Validators (N). Assumes $c_m \approx \$0.139$/epoch. Both axes are on a logarithmic scale.

The numerical analysis results, depicted in Figure 3, reveal several key insights for designing a RAT mechanism that promotes Ideal Security:

1. **Practical Trigger Frequencies:** The annotation for $N = 10$ and $C_{\text{off}} \approx \$498$ (as per the plot) shows a required system-wide $\pi_a \approx 0.28\%$ per epoch. With 10-minute epochs, this translates to approximately $0.28/100 \times (24 \times 6) \approx 0.4$ tests per day. The probability that a specific validator is directly challenged in any given epoch (assuming random selection among N validators when a system-wide test is triggered) is $\pi_a \times (1/N)$. Using Equation (2), this individual challenge probability becomes $(c_m N/C_{\text{off}}) \times (1/N) = c_m/C_{\text{off}}$. Therefore, an individual validator expects a direct challenge roughly every D_V/c_m epochs. For $C_{\text{off}} = \$498$ and $c_m \approx \$0.139$, this is about $498/0.139 \approx 10756$ epochs. With 10-minute epochs, this is approximately $10756 \times 10/(60 \times 24) \approx 74.7$ days.
2. **Feasibility and Trade-offs:** If C_{off} is too low relative to $c_m N$, the required π_a could exceed 100% (as seen for higher N and lower C_{off} values in the plot, where lines go above the 100% mark if not for the log scale's typical range for probabilities). This would indicate that Ideal Security is not achievable under those parameters without triggering a RAT (or multiple) every epoch, which is impractical. The plot helps visualize these feasibility boundaries. Designers must balance the desire for low L1 overhead (low π_a) with the need for sufficiently high validator deposits/penalties (C_{off}) and the inherent costs of validation (c_m).

3. **Impact of Validator Count (N):** For a fixed C_{off} and c_m, the required π_a increases linearly with the number of validators (N). This is because with more validators, the individual responsibility for upholding the Ideal Security condition (from the perspective of Equation (14)'s right-hand side) is effectively diluted if $\pi_a C_{\text{off}}$ remains constant. To compensate, π_a must increase proportionally with N if C_{off} is fixed, or C_{off} must increase. The plot shows distinct, parallel lines for different N values on the log-log scale, reflecting this relationship. For example, at $C_{\text{off}} = \$500$, π_a is approximately 0.28% for $N = 5$.

In conclusion, the numerical analysis underscores that a well-capitalized validator set is crucial for enabling a low-overhead RAT mechanism (low π_a) that can still effectively enforce validator attentiveness and achieve the Ideal Security equilibrium. The precise tuning of π_a will depend on the specific rollup's economic parameters, decentralization goals, and tolerance for L1 operational costs.

8 Discussion

8.1 L1 Cost Implications of RAT

Regarding the L1 gas overhead, when the proposer submits a state commitment, and the smart contract determines an attention test should be initiated with the test probability, the contract itself performs additional computations. These include calculating whether to trigger the test (e.g., using the commitment and the current L1 block hash as inputs to a pseudorandom function) and, if so, selecting a target validator from the registered set. This process adds a marginal computational overhead to the proposer's L1 submission transaction specifically in epochs where a RAT is triggered, primarily for these decision-making steps by the smart contract and for emitting the corresponding attention event. Subsequently, for a successful "Pass" outcome of the attention test, the direct L1 gas cost incurred by the challenged validator involves submitting one additional L1 transaction containing their solution. The on-chain verification by the SC for this scenario is then limited to one hash computation and one comparison against the stored commitment. These operations are computationally lightweight. Furthermore, our analysis indicates a low attention test frequency is enough for the ideal security equilibrium. It suggests that the incremental L1 gas cost for a successfully passed attention test is expected to be modest.

8.2 The Simplicity of the Validator Cost Function

Our analysis utilized a linear cost function for validator attentiveness, a reasonable first-order approximation, particularly given the metered nature of cloud services commonly used by validators. However, the true cost could exhibit non-linearities, such as initial setup costs or diminishing then increasing marginal costs, potentially resembling a sigmoid function. While a comprehensive analysis of such variations is extensive, we posit that as long as the cost function remains monotonically increasing, the fundamental structure of the derived Nash Equilibria (Ideal Security, System Failure, and potential MSNE) would likely persist, with shifts primarily occurring in the specific equilibrium points and the thresholds between them. A more granular understanding of these cost structures could further refine the practical parameter tuning of RAT, although the current model provides a robust foundation for understanding the core incentive dynamics.

8.3 Limitations of State Transition Tracking in Attention Tests

The RAT protocol, as designed, effectively verifies a validator's ability to track L2 state transitions and compute correct state roots, which is a critical aspect of attentiveness. Nevertheless, this does not comprehensively guarantee the validator's full capability to successfully generate and submit a valid fraud proof during an actual dispute, as highlighted by incidents like the Kroma Optimistic Rollup's VM misconfiguration issue in 2024 [19]. This distinction underscores that RAT primarily addresses validator liveness and computational correctness for state tracking, rather than the end-to-end integrity of the fraud proof submission pipeline, which can involve complex L1 contract interactions and specific local environment configurations. While RAT enhances security by ensuring foundational diligence, future research could explore extensions or complementary mechanisms to more holistically assess a validator's readiness to complete the entire fraud proof process, balancing the desire for comprehensive assurance against increased complexity and L1 overhead.

9 Conclusion

This paper introduced and detailed the Randomized Attention Test (RAT) protocol, a novel mechanism specifically designed to address the verifier's dilemma concerning validator attentiveness within Optimistic Rollups. By employing

probabilistic challenges, RAT proactively monitors and incentivizes validator diligence. Our game-theoretic analysis established the conditions for an Ideal Security Equilibrium, demonstrating that honest and attentive behavior from all participants can be encouraged. Significantly, this desirable state is shown to be practically achievable with modest economic penalties for non-compliance and a low challenge probability, thereby minimizing L1 gas overhead and operational expenses. The analysis also provided a framework for understanding the trade-offs inherent in tuning RAT's parameters. Ultimately, RAT is proposed as a vital complement to existing Fraud Proof mechanisms, enhancing the security posture of Optimistic Rollup systems by ensuring that validators are not just present, but actively and correctly performing their crucial verification duties. This work paves the way for more robust and trustworthy ORU implementations.

References

[1] L. Luu, J. Teutsch, R. Kulkarni, and P. Saxena, "Demystifying incentives in the consensus computer," in *Proceedings of the 22Nd acm sigsac conference on computer and communications security*, pp. 706–719, 2015.

[2] E. N. Tas, J. Adler, M. Al-Bassam, I. Khoffi, D. Tse, and N. Vaziri, "Accountable safety for rollups," *arXiv preprint arXiv:2210.15017*, 2022.

[3] J. Li, "On the security of optimistic blockchain mechanisms," *Available at SSRN 4499357*, 2023.

[4] A. Mamageishvili and E. W. Felten, "Incentive schemes for rollup validators," in *The International Conference on Mathematical Research for Blockchain Economy*, pp. 48–61, Springer, 2023.

[5] R. Palakkal, J. Gorzny, and M. Derka, "Sok: Compression in rollups," in *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 712–728, 2024.

[6] A. Mamageishvili and E. W. Felten, "Efficient rollup batch posting strategy on base layer," in *International Conference on Financial Cryptography and Data Security*, pp. 355–366, Springer, 2023.

[7] D. Crapis, E. W. Felten, and A. Mamageishvili, "Eip-4844 economics and rollup strategies," in *Financial Cryptography and Data Security. FC 2024 International Workshops* (J. Budurushi, O. Kulyk, S. Allen, T. Diamandis, A. Klages-Mundt, A. Bracciali, G. Goodell, and S. Matsuo, eds.), (Cham), pp. 135–149, Springer Nature Switzerland, 2025.

[8] L. Heimbach and J. Milionis, "The early days of the ethereum blob fee market and lessons learnt," in *Proceedings of Financial Cryptography 2025*, 2025. Presented at Financial Cryptography 2025, to appear.

[9] Y. Huang, S. Wang, Y. Huang, and J. Tang, "Two sides of the same coin: Large-scale measurements of builder and rollup after eip-4844," *arXiv preprint arXiv:2411.03892*, 2024.

[10] S. Park, B. Mun, S. Lee, W. Jeong, J. Lee, H. Eom, and H. Jang, "Impact of eip-4844 on ethereum: Consensus security, ethereum usage, rollup transaction dynamics, and blob gas fee markets," *arXiv preprint arXiv:2405.03183*, 2024.

[11] S. Lee, "180 days after eip-4844: Will blob sharing solve dilemma for small rollups?," in *Proceedings of the DICG Workshop, in association with the 45th IEEE International Conference on Distributed Computing Systems (ICDCS 2025)*, (Glasgow, UK), 2025. to appear.

[12] M. M. Alvarez, H. Arneson, B. Berger, L. Bousfield, C. Buckland, Y. Edelman, E. W. Felten, D. Goldman, R. Jordan, M. Kelkar, *et al.*, "Bold: Fast and cheap dispute resolution," *arXiv preprint arXiv:2404.10491*, 2024.

[13] D. Nehab, G. C. de Paula, and A. Teixeira, "Dave: a decentralized, secure, and lively fraud-proof algorithm," *arXiv preprint arXiv:2411.05463*, 2024.

[14] B. Berger, E. W. Felten, A. Mamageishvili, and B. Sudakov, "Economic censorship games in fraud proofs," in *Proceedings of the 26th ACM Conference on Economics and Computation (EC'25)*, 2025.

[15] S. Lee, "Hollow victory: How malicious proposers exploit validator incentives in optimistic rollup dispute games," in *Proceedings of the WTSC Workshop, in association with the International Conference on Financial Cryptography and Data Security 2025 (FC 2025)*, (Miyakojima, Japan), 2025.

[16] Optimism, "Tutorial: Run a node from source." `https://docs.optimism.io/operators/node-operators/tutorials/run-node-from-source`, 2025. Accessed: 2025-05-21.

[17] Base, "Base node github repository." `https://github.com/base/node`, 2025. Accessed: 2025-05-21.

[18] Arbitrum, "Run a full arbitrum node." `https://docs.arbitrum.io/run-arbitrum-node/run-full-node`, 2025. Accessed: 2025-05-21.

[19] Kroma, "About the first successful challenge on kroma mainnet," April 2024. Accessed: 2025-05-25.