

Hollow Victory: How Malicious Proposers Exploit Validator Incentives in Optimistic Rollup Dispute Games

No Author Given

No Institute Given

Abstract. Blockchain systems, such as Ethereum, are increasingly adopting layer-2 scaling solutions to improve transaction throughput and reduce fees. One popular layer-2 approach is the Optimistic Rollup, which relies security on a mechanism known as a dispute game for block proposals. In these systems, validators can challenge blocks that they believe contain errors, and a successful challenge results in the transfer of a portion of the proposer’s deposit as a reward. In this paper, we reveal a structural vulnerability in the mechanism: validators may not be awarded a proper profit despite winning a dispute challenge. We develop a formal game-theoretic model of the dispute game and analyze several scenarios, including cases where the proposer controls some validators and cases where a secondary auction mechanism is deployed to induce additional participation. Our analysis demonstrates that under current designs, the competitive pressure from validators may be insufficient to deter malicious behavior. To address this, we propose countermeasures such as an escrowed reward mechanism and a commit-reveal protocol. Our findings provide critical insights into enhancing the economic security of layer-2 scaling solutions in blockchain networks.

Keywords: Ethereum · Game Theory · Optimistic Rollup · Security.

1 Introduction

Blockchain technology, and Ethereum in particular, has witnessed rapid growth in recent years. However, scalability and high transaction fees remain critical challenges. To address these issues, various layer-2 scaling solutions have been proposed, among which *Optimistic Rollups* have gained significant attention. Optimistic Rollups enable off-chain computation while relying on an on-chain dispute resolution mechanism to ensure correctness. In these systems, block proposals are accepted optimistically, and validators are empowered to challenge any block they believe to be incorrect through a process known as a *dispute game*.

In a typical dispute game, a block proposer stakes a deposit and submits a block. Validators then monitor the block and, if they detect an error, submit a challenge. If the challenge is successful, a portion of the proposer’s deposit is confiscated and awarded to the challenger. Ideally, this mechanism is meant

to deter malicious behavior by imposing significant economic penalties on a proposer who submits an incorrect block. However, if the incentive mechanism is not carefully calibrated, a malicious proposer might be able to limit his losses to only a small challenge fee even when many validators participate in challenging the block. It results in undermining the intended economic deterrence.

In this paper, we model the dispute game using game-theoretic methods and analyze several attack scenarios. We consider a case where the malicious proposer controls a subset of validators as well as cases in which a secondary auction is deployed to induce additional validator competition to get the dispute winner prize. Our analysis reveals that, under current incentive structures, the liveness of the dispute game may be insufficient to motivate them, thereby leaving the system vulnerable to strategic exploitation.

Our contributions are as follows:

- We develop a game-theoretic model of the dispute game in Optimistic Rollups, capturing the interactions between a block proposer and validators under various challenge scenarios.
- We identify a critical vulnerability in current dispute game designs, where a malicious proposer can not only limit his losses but also reduce the validator’s dispute winning prize nearly zero.
- We propose countermeasures, including an escrowed (deferred) reward mechanism and a commit-reveal protocol to rebalance the incentive structure and enhance the economic security of Optimistic Rollups.

The rest of the paper is organized as follows: Section 2 reviews related work on dispute games and incentive mechanisms in layer-2 solutions. In Section 3, we present our formal model of the Optimistic Rollup dispute game. Section 4 details the strategies a malicious proposer may employ and introduces our proposed auction mechanism. Section 5 provides a game-theoretic analysis of the auction mechanism, including conditions under which validators are incentivized to participate. Section 6 proposes two countermeasures to solve this structural vulnerability. Finally, Section 7 discusses potential solutions and future research directions, and concludes the paper.

2 Related Works

A variety of recent research has focused on understanding the economics and security of layer-2 scaling solutions. In this section, we review them particularly in the context of optimistic rollups.

Economic and Incentive 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.* [14] propose a framework for *accountable safety* in rollups, which emphasizes the need for designs that hold participants economically accountable for misbehavior. Li [8] 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 [10] analyze incentive schemes for rollup validators, discussing strategies under the attention test.

Data Availability Cost Optimization. Given that Optimistic Rollups are designed as a layer-2 scaling solution, minimizing the costs associated with data availability is a critical concern. Several studies have tackled this problem from different angles. Palakkal *et al.* [12] provide a systematization of compression techniques in rollups, outlining methods inefficiency in some rollups’ practice. Mamageishvili and Felten [9] propose efficient rollup batch posting strategies on the base layer using call data, while Crapis *et al.* [4] offer an in-depth analysis of EIP-4844 economics and rollup strategies including blob cost sharing. Complementary empirical investigations by Heimbach and Milionis [5], Huang *et al.* [6], and Park *et al.* [13] further document the inefficiencies in current DA cost structures and examine their impact on rollup transaction dynamics and consensus security. Lee [7] investigates blob sharing as a potential remedy for the dilemma faced by small rollups in the post-EIP-4844 era.

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. For example, BoLD [1] and Dave [11] propose protocols that optimize the dispute game to achieve low delays and cost-efficient on-chain verification, while Berger *et al.* [3] analyze economic censorship dynamics to guarantee that disputes are resolved even under adversarial conditions. In contrast to these works—which emphasize maintaining the smooth execution of dispute games through liveness—the present study concentrates on a different vulnerability: even when dispute games proceed smoothly, the existing incentive structure may fail to adequately reward honest validators, thereby allowing a malicious block proposer to strategically minimize his costs.

3 Dispute Game Model for Optimistic Rollups

In this section, we present a formal model of the dispute game used in optimistic rollups. We define the game in the style of a game-theoretic construct, denoted by $\mathcal{G}$, and specify the participants, assumptions, rules, and reward mechanisms.

3.1 Participants and Deposits

We consider two types of participants in the dispute game:

- **Proposer:** Denoted by P , the proposer is responsible for submitting a proposed block.

- **Validators:** Denoted by the set $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$, the validators (or challengers) participate in the dispute game to verify the correctness of the state transition.

Each participant is required to hold a minimum deposit to obtain participation rights:

- The proposer holds a deposit D_P .
- Each validator v_i holds a deposit D_V , where (for simplicity) we assume the same minimum deposit D_V for all validators.

3.2 Rules of the Dispute Game

The dispute game $\mathcal{G}$ is triggered when a proposer P submits a block that is subject to challenge by one or more validators. The key rules are as follows:

1. **Verification of State Transitions:** A dispute is resolved by verifying the correctness of the state transition. This is achieved either:
 - (a) via the opponent’s timeout, or
 - (b) through the execution of a bisection protocol on a dispute game contract, wherein the virtual machine (VM) executes the disputed state transition steps.
2. **Collateral for Dispute:** Upon initiation of the dispute game, both parties are required to post an additional collateral deposit D_g , where

$$0 < D_g < D_{\max}.$$

This additional deposit is intended to further incentivize honest participation in the dispute resolution process.

3.3 Reward and Penalty Policy

The reward policy of the dispute game is designed to penalize a malicious proposer and reward the challenger(s) as follows:

- If the proposer P loses the dispute game, the challenger(s) may claim a portion of the deposits. Specifically, the winning party is entitled to claim:

$$\alpha \cdot D + D_g,$$

where D represents the deposit of the losing party (either D_P or D_V , as appropriate) and α is a reward parameter satisfying. In the most of real world rollups, most of systems do not specify this parameter except for Arbitrum. An Arbitrum, one of the leading optimistic rollups, document mentions only 1% of the confiscated asset will be certainly rewarded to the dispute game winner [2]. Still, its DAO can reward more of the confiscated asset to the winner.

$$0 < \alpha \leq 1.$$

- The deposits that are not forfeited are allocated to a communal fund (e.g., managed by a DAO) to support the overall system.

3.4 Additional Assumptions

For the purpose of our analysis, we assume:

1. **Sequential and Parallel Occurrence:** The dispute game can occur in multiple rounds sequentially; furthermore, different instances of the dispute game may run in parallel.
2. **Timeouts:** Timeouts are set generously so that the proposer cannot deliberately force a timeout by delaying responses. The creation time of each dispute game, denoted by t_i , is sufficiently long to preclude strategic timeouts.

This model captures the strategic interaction between the proposer and validators under the assumptions of minimum deposits and additional collateral requirements. In our subsequent analysis, we use game-theoretic techniques to examine the incentive imbalances that may arise, particularly focusing on how the profit for a validator as a dispute game winner can be limited nearly to zero.

4 Strategy of Malicious Block Proposer

In this section, we illustrate the various strategies that a malicious proposer may employ to minimize his cost in the dispute game. We consider three scenarios and then describe an auction mechanism that the malicious proposer can trigger to force additional validator participation. In our analysis, a malicious proposer is denoted by P and the set of validators by $\mathcal{V}$.

4.1 Scenario Descriptions

We consider the following three scenarios. The first scenario assumes the malicious proposer controls at least one validator. On the other hand, the left two scenarios do not assume the proposer-controlled validator. The difference between the scenario 2 and 3 is that the number of submitted challenges before the malicious proposer responds to any challenge. The scenario 3's feasibility will be analyzed in the next section.

Scenario 1 (Proposer-Controlled Validator Accounts) In this baseline scenario, the malicious proposer P controls a subset $\mathcal{V}_P \subset \mathcal{V}$ of validator accounts. When a block B is proposed and a valid challenge is submitted by some validator $v_i \in \mathcal{V} \setminus \mathcal{V}_P$, P can immediately counter by initiating a dispute game through one of his controlled validators $v_j \in \mathcal{V}_P$. Under the reward mechanism (where a winning party receives $\alpha \cdot D + D_g$ and $\alpha \in (0, 1]$), the net cost for P is limited to:

$$\text{Cost}_P = (1 - \alpha) \cdot D,$$

since the controlled validator wins the challenge and the attacker absorbs only the shortfall.

Scenario 2 (Multiple Challenges Exist) Suppose that multiple validators submit valid challenges against B . In this situation, P may reduce his effective cost by deploying an auction contract. The auction mechanism is designed to order the challenges: validators who have not yet challenged are induced to bid. Since a validator’s alternative is to obtain a reward of zero (if they do not participate), the auction forces competitive bidding. In equilibrium, validators will bid amounts that reflect their full valuation, roughly given by

$$v = \alpha \cdot D + D_g.$$

Thus, if P loses the resulting auction-based challenge, his cost will be at least the winning bid, which in a competitive setting is driven upward by the participation of many validators.

Scenario 3 (Single Challenge Initially Submitted) In the case where only a single challenge is initially submitted, P may still deploy the auction contract to attract additional challenges. Even though one challenge exists, the auction mechanism creates a framework in which validators are incentivized to bid—since non-participation yields zero reward. The competitive pressure in the auction will then increase the cost borne by P if a challenger wins the dispute game. In both Scenarios 2 and 3, the key idea is that by forcing additional validator participation via an auction, P ’s effective cost is raised beyond the minimal challenge fee.

4.2 Auction Contract Construction

To explain the scenario in which multiple or a single initial challenge is submitted (Scenarios 2 and 3), we propose an auction-based mechanism that integrates directly with the dispute game. The auction contract is designed to foster competition between validators, thereby minimizing the malicious proposer’s loss. In essence, if a valid challenge has already been submitted, the malicious proposer can trigger this auction contract to invite further challenges. As an on-chain smart contract, the auction contract ensures that the auction winner (validator) will win the dispute game first. Otherwise, the auction operator’s (malicious proposer) deposit in the auction contract will be confiscated and given to the auction winner.

Our design leverages only minimal EVM functionalities—such as reading block timestamps/number, basic list manipulations, and conditional checks—which makes it well-suited for implementation in a standard smart contract environment. The pseudocode (presented in Algorithm 1) outlines the essential steps:

We now describe the detailed steps of our proposed auction mechanism, which is designed to force additional validator participation and increase the economic cost for a malicious block proposer. The mechanism proceeds as follows:

Auction Initialization: When at least one valid challenge is detected, the auction contract is deployed. At this point, the malicious block proposer transfers, as a deposit, an amount corresponding to the reward promised by the

Algorithm 1 Dispute Game Auction Contract

```

1: Input: disputeId, auction duration  $T_{\text{auction}}$ 
2: Initialize:
3:   auctionStart  $\leftarrow$  current timestamp (or block number)
4:   Bids  $\leftarrow \emptyset$ 
5:   Malicious proposer deposits  $E$  (equal to the promised reward)
6: procedure SUBMITBID(validator, bidAmount)
7:   if current time  $<$  auctionStart +  $T_{\text{auction}}$  then
8:     Verify dispute game instance exists for disputeId
9:     Append (validator, bidAmount) to Bids
10:  end if
11: end procedure
12: procedure FINALIZEAUCTION
13:  if current time  $\geq$  auctionStart +  $T_{\text{auction}}$  then
14:    Let  $b_{\text{max}}$  and  $b_{\text{second}}$  be the highest and second-highest bids in Bids
15:    Winner  $\leftarrow$  validator associated with  $b_{\text{max}}$ 
16:    Grant Winner the exclusive right to initiate the dispute game at cost  $b_{\text{second}}$ 
17:  end if
18: end procedure
19: procedure RESOLVEDISPUTE(outcome)
20:  if outcome = first win for the winning validator then
21:    Refund excess deposit to Winner
22:  else
23:    Transfer  $b_{\text{second}}$  from Winner's deposit to malicious proposer
24:  end if
25: end procedure

```

dispute game (i.e., the forfeitable funds). This initialization triggers the fixed bidding period during which the auction will accept bids.

Bid Submission: Validators who have not yet submitted a challenge are invited to participate in the auction. Prior to bidding, each validator must initiate a dispute game for the challenged block. This step is essential as it enables the auction contract to verify the existence of a corresponding dispute game contract and ensures that all contractual conditions are met. If a validator's valuation (which reflects the full reward) exceeds the current highest bid, they submit their bid along with the necessary deposit.

Auction Finalization: Once a pre-defined number of blocks have passed (or the auction duration has elapsed), the auction period terminates. At this point, the auction contract finalizes the bidding process using a Vickrey (second-price) auction mechanism, whereby the highest bidder wins but pays the amount of the second-highest bid. The winning validator is granted the exclusive right to initiate the dispute game challenge.

Dispute Resolution and Cost Recovery: The dispute game then proceeds, and the system monitors all submitted challenges. Once it is verified—by querying the status of all dispute games—that the winning validator's dispute game is the first to conclude, the auction contract triggers the cost-recovery pro-

cess. Specifically, the winning validator's deposit is used to transfer the winning bid amount to the malicious block proposer, while any remaining difference (if the winning bid is less than the full deposit) is refunded to the winning validator.

In summary, this auction contract serves as a critical component of our overall strategy. It transforms the challenge submission process into a competitive auction, where the equilibrium bid is driven by each validator's valuation.

5 Analysis on the Dispute Game Auction

In our dispute game auction, each validator faces a fixed participation cost $c > 0$ (accounting for gas fees, contract invocation, etc.). The reward available if a validator wins a dispute is given by

$$R = \alpha D + D_g,$$

where D is the forfeitable deposit, D_g is the additional collateral, and $\alpha \in (0, 1]$ is a reward parameter. In practice, since the protocol is uniform, validators' valuations are very similar. To capture this slight heterogeneity, we assume that each validator's valuation is drawn from the interval $[R - \mu, R]$, where $\mu > 0$ is small. Under these assumptions, the potential surplus a validator can obtain by winning the auction comes solely from the dispersion μ .

5.1 Expected Payoffs and Participation Conditions in a Second-Price Auction

Assume that n validators participate in a second-price (Vickrey) auction. Since the valuations are nearly identical, each validator's true valuation is $v_i \in [R - \mu, R]$. By a change of variable, let

$$X_i = v_i - (R - \mu),$$

so that X_i is uniformly distributed on $[0, \mu]$. It is well known that in a second-price auction with n independent draws from $U[0, \mu]$, the expected maximum is

$$E[X_{(1)}] = \frac{n}{n+1}\mu,$$

and the expected second-highest value is

$$E[X_{(2)}] = \frac{n-1}{n+1}\mu.$$

Thus, the expected surplus (i.e., the difference between the highest and second-highest bid) is

$$E[X_{(1)} - X_{(2)}] = \frac{\mu}{n+1}.$$

Since each validator pays a fixed cost c upon participation regardless of winning or losing, the net gain for a winning validator is

$$\pi_{\text{win}} = \frac{\mu}{n+1} - c.$$

A validator who loses simply incurs a loss of c . Therefore, a validator's decision to participate (as opposed to abstaining and receiving a payoff of zero) is individually rational if the expected surplus exceeds the cost:

$$\frac{\mu}{n+1} - c > 0 \iff \mu > c(n+1).$$

We formalize this result as follows.

Theorem 1. *Let n risk-neutral, symmetric validators have valuations drawn independently from the interval $[R - \mu, R]$, with $\mu > 0$ representing the small dispersion in valuations. In a second-price auction in which each validator incurs a fixed participation cost $c > 0$, the expected surplus for the winning validator is*

$$E[\Delta] = \frac{\mu}{n+1}.$$

Thus, a validator has a positive incentive to participate if and only if

$$\mu > c(n+1).$$

Proof. Define $X_i = v_i - (R - \mu)$ so that $X_i \sim U[0, \mu]$. In a second-price auction with n bidders, the expected highest value is $E[X_{(1)}] = \frac{n}{n+1}\mu$ and the expected second-highest value is $E[X_{(2)}] = \frac{n-1}{n+1}\mu$. The expected surplus for the winner is therefore

$$E[X_{(1)} - X_{(2)}] = \frac{n}{n+1}\mu - \frac{n-1}{n+1}\mu = \frac{\mu}{n+1}.$$

Because each participating validator pays the fixed cost c , a validator who wins obtains a net payoff of $\frac{\mu}{n+1} - c$ while one who loses obtains $-c$. Thus, participation is beneficial compared to abstention (which yields 0) if and only if

$$\frac{\mu}{n+1} - c > 0 \iff \mu > c(n+1).$$

□

Next, consider the situation where k validators have already participated in the auction. For an additional validator joining, the auction becomes one among $k+1$ bidders. In this case, the condition for participation is modified to

$$\frac{\mu}{k+2} > c \iff \mu > c(k+2).$$

This corollary illustrates that as more validators join, the incremental benefit for an additional validator decreases, but participation remains attractive as long as the dispersion μ is sufficiently large relative to the total cost incurred by all bidders.

Corollary 1. *If k validators have already participated in the auction, an additional validator will have a positive incentive to join if and only if*

$$\mu > c(k+2).$$

5.2 Application to the Secondary Auction in a Single-Challenge Scenario

In realistic deployments of optimistic rollups, it is often observed that only a single dispute challenge is initially submitted. Without further intervention, additional validators may refrain from joining because the early mover appears to secure the reward, leaving later participants with a guaranteed loss of the fixed cost c . To overcome this free-riding behavior, we propose the deployment of a secondary auction contract that is triggered once the first challenge is registered. This contract compels additional validators to decide whether to participate in the auction.

Under the secondary auction, an additional validator now considers joining an auction with k current participants. As shown above, the participation condition is given by

$$\mu > c(k + 2).$$

For instance, if initially only one challenge exists (i.e., $k = 1$), then an extra validator will join if and only if

$$\mu > 3c.$$

If this condition is satisfied, the secondary auction effectively transforms the environment into one with multiple bidders, thereby enabling competitive bidding. As more validators join, the competitive pressure increases, and the equilibrium outcome will impose a cost on the malicious proposer that approximates the winning bid. In turn, the malicious proposer must incur a cost that reflects the aggregate competitive surplus, rather than merely a minimal challenge fee.

In practice, even if the valuations are nearly identical (i.e., all validators have valuations in $[R - \mu, R]$ with a small μ), the auction mechanism’s ability to induce additional participation depends critically on the relation between μ and the fixed cost c . If μ is sufficiently large relative to c (specifically, if $\mu > c(k + 2)$ for the given number k of current participants), then additional validators are incentivized to join the auction. This, in turn, increases the cost imposed on the malicious proposer when the final dispute is resolved.

In summary, our analysis under a second-price auction framework with a slight heterogeneity in valuations modeled by the interval $[R - \mu, R]$ demonstrates that validators have a positive incentive to participate if the dispersion μ exceeds the product of the participation cost and the number of competitors plus one. Moreover, even in scenarios with an initial single challenge, a secondary auction contract can induce further participation under the condition $\mu > c(k + 2)$, thereby ensuring that the malicious proposer faces a significantly higher economic burden. This mechanism is designed to operate efficiently within minimal EVM environments, relying only on basic arithmetic operations and timestamp checks, which makes it both practical and robust in deployment.

6 Potential Solutions

In this section, we propose two solutions to mitigate the structural vulnerability which deincestivizes optimistic rollup validators.

6.1 Escrowed Reward Mechanism

The first approach addresses the vulnerability arising from early challenge exploitation by modifying the reward distribution mechanism. Rather than immediately allocating the prize to the first valid challenge—as many existing protocols do—this mechanism locks the reward in escrow as soon as a valid dispute challenge is raised. As shown in Algorithm 2, the reward remains locked until *all* challenges pertaining to the block are finalized. This delay in reward allocation prevents a malicious proposer from benefiting by simply finalizing the belatedly triggered challenge first, ensuring that the final reward distribution reflects the order and promptness of validators.

Algorithm 2 Escrowed Reward Mechanism for Dispute

```

1: Input: disputeStart, disputeDuration
2: Initialize:
3:   rewardsLocked  $\leftarrow$  false, rewardPool  $\leftarrow$  0
4:   challengerDeposits  $\leftarrow$  empty mapping
5: procedure INITIATECHALLENGE(challenger, depositAmount)
6:   if currentTime < disputeStart + disputeDuration then
7:     challengerDeposits[challenger]  $\leftarrow$  challengerDeposits[challenger] + depositAmount
8:     if rewardsLocked is false then
9:       rewardsLocked  $\leftarrow$  true
10:      rewardPool  $\leftarrow$  current contract balance
11:    end if
12:  end if
13: end procedure
14: procedure FINALIZEDISPUTE
15:   if currentTime  $\geq$  disputeStart + disputeDuration then
16:     winningChallenger  $\leftarrow$  SELECTWINNINGCHALLENGER
17:     Transfer rewardPool to winningChallenger
18:     rewardsLocked  $\leftarrow$  false
19:   end if
20: end procedure
21: procedure SELECTWINNINGCHALLENGER  $\triangleright$  Determine winning challenger based on predefined criteria
22:   return chosen challenger address
23: end procedure

```

Mechanism Details:

- **Escrow of Rewards:** Regardless of the order in which challenges are initiated, the entire reward (i.e., the forfeited deposits) is held in escrow until the dispute resolution period for the block ends.

- **Final Distribution:** Once the dispute window closes, the reward is distributed to the winning challenger. If multiple challenges are valid, the reward is apportioned or allocated according to a predefined rule.

Trade-offs. This mechanism effectively prevents a malicious proposer from preempting the dispute process by quickly triggering a challenge using controlled validator accounts. However, a potential downside is the introduction of Miner Extractable Value (MEV) opportunities. An attacker might monitor the mempool for early challenge transactions and submit parallel challenges via MEV-boost techniques to capture a larger portion of the reward. Mitigating this risk may require validators to use private mempools or other techniques to hide their challenge submissions.

6.2 Commit-Reveal Protocol for Challenge Decisions

The second approach presented in Algorithm 3 introduces a commit-reveal scheme that obscures validators' intentions to challenge a block. Instead of immediately broadcasting a challenge, validators submit a commitment during a designated epoch and then reveal their decision after a short delay.

Algorithm 3 Commit-Reveal Dispute Mechanism

```

1: Data: Mapping Commits ( $\text{validator} \rightarrow (\text{commitHash}, \text{revealed}, \text{decision}, \text{blockNumber})$ ),
   commitDeadline, revealDeadline
2: procedure COMMITCHALLENGE(validator, commitHash)
3:   if currentTime < commitDeadline then
4:     Commits[validator]  $\leftarrow$  (commitHash, false, _, _)
5:   end if
6: end procedure
7: procedure REVEALCHALLENGE(validator, decision, blockNumber, nonce)
8:   if currentTime  $\geq$  commitDeadline and currentTime < revealDeadline then
9:     if hash(decision, blockNumber, nonce, validator) = Commits[validator].commitHash then
10:      Update Commits[validator] to (_, true, decision, blockNumber)
11:    end if
12:   end if
13: end procedure
14: procedure PROCESSCHALLENGES
15:   if currentTime  $\geq$  revealDeadline then
16:     Process valid challenges and distribute rewards.
17:   end if
18: end procedure

```

Mechanism Details:

- **Commit Phase:** In each epoch, validators submit a hash of their decision (e.g., `challenge` or `not challenge`), the block number, and a nonce, all signed with their private key. This commit prevents others from knowing the validator’s decision in advance.
- **Reveal Phase:** After the commit phase, validators reveal their decision and associated data. The smart contract verifies that the reveal matches the commitment.
- This process prevents a “follow-the-leader” attack where a malicious actor could monitor challenge submissions and mimic or preempt them.

Trade-offs. By using a commit-reveal protocol, the challenge decision of each validator is concealed until all commitments are made. This prevents any validator from strategically adjusting their decision based on others’ actions (i.e., challenge follow-up attacks). In addition, if multiple valid challenges exist in the same block, the reward can be distributed fairly. However, this approach introduces extra overhead in the form of additional Layer-1 transactions for both commit and reveal phases. To mitigate increased transaction fees, one might consider extending the epoch duration, aggregating commits off-chain, or leveraging data blob solutions.

In summary, both proposed solutions aim to eliminate the possibility for a malicious proposer to exploit the current dispute game incentives: The **Escrowed Reward Mechanism** ensures that rewards are only distributed after the dispute game for a given block has fully resolved, thereby removing the advantage of being the first to challenge. However, it may introduce MEV risks. The **Commit-Reveal Protocol** obscures validators’ decisions during the dispute phase, preventing strategic follow-up challenges. The main trade-off here is the increased overhead due to additional transactions.

7 Discussion and Concluding Remarks

In this work, we have presented an analysis of dispute games in Optimistic Rollups, focusing on a critical structural weakness in the incentive mechanism. Our study reveals that even when dispute games proceed smoothly with liveness, the current reward distribution may leave honest validators with negligible net gains. Consequently, a malicious block proposer can exploit this flaw which minimizing his losses to a small challenge fee optimistically while honest validators, who are expected to safeguard the system, may not receive adequate compensation. To address this problem, we proposed two countermeasures: an escrowed reward mechanism that defers the allocation of rewards until all challenges are finalized, and a commit-reveal protocol that obscures validator decisions to prevent follow-the-leader attacks. These mechanisms are designed to be both efficient and compatible with minimal EVM or smart contract functionalities, thereby offering practical solutions for enhancing the economic security of layer-2 scaling systems.

While our theoretical analysis provides critical insights, further empirical evaluation under realistic network conditions is needed. Future work should incorporate dynamic validator behavior, transaction fee variability, and potential MEV impacts to refine these mechanisms. Ultimately, our findings underscore the importance of rethinking incentive structures in dispute games to guarantee that honest validators are fairly compensated, thereby bolstering the overall security and integrity of optimistic rollup systems.

References

1. Alvarez, M.M., Arneson, H., Berger, B., Bousfield, L., Buckland, C., Edelman, Y., Felten, E.W., Goldman, D., Jordan, R., Kelkar, M., et al.: Bold: Fast and cheap dispute resolution. arXiv preprint arXiv:2404.10491 (2024)
2. Arbitrum: Bold economics of disputes. <https://docs.arbitrum.io/how-arbitrum-works/bold/bold-economics-of-disputes> (2025), accessed: 2025-03-07
3. Berger, B., Felten, E.W., Mamageishvili, A., Sudakov, B.: Economic censorship games in fraud proofs (2025), presented at Financial Cryptography 2025, to appear
4. Crapis, D., Felten, E.W., Mamageishvili, A.: Eip-4844 economics and rollup strategies. In: Budurushi, J., Kulyk, O., Allen, S., Diamandis, T., Klages-Mundt, A., Bracciali, A., Goodell, G., Matsuo, S. (eds.) Financial Cryptography and Data Security. FC 2024 International Workshops. pp. 135–149. Springer Nature Switzerland, Cham (2025)
5. Heimbach, L., Milionis, J.: 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
6. Huang, Y., Wang, S., Huang, Y., Tang, J.: Two sides of the same coin: Large-scale measurements of builder and rollup after eip-4844. arXiv preprint arXiv:2411.03892 (2024)
7. Lee, S.: 180 days after eip-4844: Will blob sharing solve dilemma for small rollups? arXiv preprint arXiv:2410.04111 (2024)
8. Li, J.: On the security of optimistic blockchain mechanisms. Available at SSRN 4499357 (2023)
9. Mamageishvili, A., Felten, E.W.: Efficient rollup batch posting strategy on base layer. In: International Conference on Financial Cryptography and Data Security. pp. 355–366. Springer (2023)
10. Mamageishvili, A., Felten, E.W.: Incentive schemes for rollup validators. In: The International Conference on Mathematical Research for Blockchain Economy. pp. 48–61. Springer (2023)
11. Nehab, D., de Paula, G.C., Teixeira, A.: Dave: a decentralized, secure, and lively fraud-proof algorithm. arXiv preprint arXiv:2411.05463 (2024)
12. Palakkal, R., Gorzny, J., Derka, M.: Sok: Compression in rollups. In: 2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). pp. 712–728 (2024). <https://doi.org/10.1109/ICBC59979.2024.10634469>
13. Park, S., Mun, B., Lee, S., Jeong, W., Lee, J., Eom, H., Jang, H.: Impact of eip-4844 on ethereum: Consensus security, ethereum usage, rollup transaction dynamics, and blob gas fee markets. arXiv preprint arXiv:2405.03183 (2024)
14. Tas, E.N., Adler, J., Al-Bassam, M., Khoffi, I., Tse, D., Vaziri, N.: Accountable safety for rollups. arXiv preprint arXiv:2210.15017 (2022)