Author: @SilentCicero

Reviewers: @pixelcircuits, @Mikerah, @brapse. Special thanks to L2Beat and Justin Drake.

Crost post from here.

Optimistic Rollups ("ORs") have become a de facto scaling technique for Ethereum. ORs enable Ethereum grade security so long as a single honest minority is validating and can submit fraud proofs back to Ethereum. However, ORs come with a fairly significant short-coming, a long finalisation window (conventionally 7 days) due to potential mass censorship attacks on Ethereum [1]. In this post, we will explore a censorship-resistant check-in based model for ORs which aims to safely reduce that window down to less than a few Ethereum blocks while retaining only an honest minority assumption.

Definitions:

- State Transition Function: the mechanism which honest actors use to progress chain state into future.
- Validator: an actor running validation software containing the state transition function.
- Honest Minority: at least one actor running validation software and correctly responding to events.
- Censorship: the prevention of transactions from state transitioning a blockchain.
- Fraud: a state transition outside the state transition function (including bugs or faults).

Censorship Assumptions:

- For the purposes of this post, we must assume an adversary to be an extremely well funded entity that has billions of dollars at their disposal and that 2/3 of Ethereum validators can be entirely co-opted for at least a moderate window of time (several hours to many days) to exclude specific or all transactions (either by filling up Ethereum blocks or taking over a large portion of the validator set) [1-7].
- We also assume intentional censorship is extremely hard to detect and likely impossible to prove with on-chain mechanisms.

[

1

695×592 34.3 KB

](https://ethresear.ch/uploads/default/original/2X/c/ce931fb05198139169b564fcdc3f2e007b550354.jpeg)

Prior Art: Fraud Proving:

- Fraud proving games of the past either require a single (e.g. Fuel V1) or multi-round challenge (e.g. Arbitrum) process, whereby a single honest validator may submit a fraud proof after witnessing an invalid state transition on a rollup [8].
- The fraud proving window is typically set to 7 days.
- The 7 day window is set to ensure that an adversary, even under the event of mass censorship, would either run out of money or be forked away by Ethereum social consensus.
- The longer window also accommodates for time to path major bugs in the state transition function or rollup software.
- We believe even 24 hours may be long enough to thwart this attack, but the 7 day mark provides enough assurances that the attack will be unprofitable on Ethereum [12].

[

2

665×602 18.6 KB

](https://ethresear.ch/uploads/default/original/2X/3/395962922defe7f081c1d5f61baac7b420f43de6.png)

Prior Art: Closed Committees:

- Some rollups have opted for a security committee model, such that, a closed group of honest actors would effectively become the final say in state transitions, whereby an honest majority may be able to spot fraud and reverse state transitions.
- Closed committees are typically not permission-less and don't allow non-committee members to reverse invalid state transitions or defend valid transitions.

Prior Art: Lowering the Fraud Proving Window (Forking and Sliding Windows):

- Ed Felton proposed an idea of a forked block censorship oracle, where missed blocks are a measure of censorship and thus a window could be extended if missed blocks are detected in the previous epochs. While this is a fine measure of forking censorship, it does not accommodate for the case that block producers could all censor out of protocol (e.g. mempool or p2p layer) preventing a fraud proving transaction from ever being processed for a long period of time, potentially up to many days [9].
- Ayelet Lotem, Sarah Azouvi, Patrick McCorry and Aviv Zohar proposed a sliding window challenge based on a fee oracle which detects if Ethereum is in high demand and extends the window accordingly, thus allowing a shorter window if the fee oracle reports only moderate activity and an extended window under high demand.

The shortcoming here is that it doesn't accommodate for large scale censorship (forking) or for when the block space is extremely busy at a norm, thus making it harder to determine an extension of the window [10].

- 3Sigma proposed a governance adjusted approach to reducing the finality window using a dynamic challenge window, including rational for a safe minimum window. The shortcoming here is that adding a governance mechanism of this nature would change our base assumption around honest minority in addition to being subject to censorship itself past safe minimum window (as a majority within the governance body would be required to potentially alter the finality window period transactionally) [12]

Transacting without Transactions:

- We must assume any censoring party can entirely prevent a fraud proving challenge or any transaction to be submitted and for an extended period of time (close to 7 days on the higher end).
- Under mass censorship, the only mechanism an honest party has is to do nothing.
- This means that a mechanism would need to leverage the idea of "no transaction submission" in order to counter sophisticated forms of censorship on Ethereum.
- While most fraud proving games have used a transactional state transition to signal fraud, they have not used the idea of no state transition as evidence of censorship or fraud.
- With this idea, we define what we call a "check-in" model.

Check-in based Dynamic Challenge Window:

- Instead of having honest minority validators only submit challenges when an invalid state transition is witnessed during a fixed challenge window (conventionally 7 days past last valid transition), we flip the model on its head, and say that all validators must check-in periodically and if they miss a check-in transaction, this is treated as either censorship or offline inactivity and the challenge window dynamically expands accordingly up until a maximum safe window (e.g. 7 days).
- In a check-in model, validators submit transactions every epoch, say every 64 Ethereum blocks. The check in would be as follows:
- True, there has been no invalid transitions witnessed over the past epoch
- False, there has been an invalid transition over the last 64 blocks
- No-report, my node is down or I am being censored
- True, there has been no invalid transitions witnessed over the past epoch
- False, there has been an invalid transition over the last 64 blocks
- No-report, my node is down or I am being censored
- In the happy path (true), all validators check-in with true, and the state is finalised within 64 blocks.
- In the fraud path (false), all or one validators check-in with a false, and the fraud proving window is opened up in order for a challenge game to commence, the state is finalised once the challenge game is concluded.
- In the no-report path, we assume one of the honest minorities is being censored (even if inactivity is the cause) and thus we keep extending the window every epoch up until the maximum window, in which case the state is finalised either after a check-in before the max window or after the max window has passed.
- The check-in model enables honest minorities to vote without making a transaction, using a dead man switch style mechanism of "no-report".
- A well funded adversary now has no mechanism in which mass censorship will produce the outcome they are looking

for as the system passively reverts back to the safety of long finalisation windows to thwart any censorship attack.

- This empowers a single honest minority to thwart potentially a multi-billion dollar censorship attack on Ethereum, just by doing nothing at all.
- Unlike closed committees, the check-in model is permission-less and thus anyone can enter or leave at will and participate as a validator at will.

[

3

895×676 49 KB

](https://ethresear.ch/uploads/default/original/2X/3/38c357a019d585a01027be7da5a90ff7bc887310.png)

Grieving Vector:

- Any registered validator of the group may grief the rollup, in the worst case, reverting the rollup back to a 7 day challenge window.
- This means that even if the entire group is submitting periodically, a single party could prevent finalisation for up to 7 days.
- Note, this grieving vector outcome alreadys exists with many fraud proving systems today, where finalisation would be typically set to a fixed 7 days anyway.

Reducing or Punishing Grieving:

- Bonding: each validator can be bonded, such that if their node goes offline, they can be punished by slashing their bond after 7 days have elapsed past their last check in. Similarly, if they don't submit a check-in and then do not come back online or come back online but have no fraud to report or an invalid fraud challenge, they can be punished.
- Slashing: the first choice would be to burn either a portion or all their bond depending on the circumstances. A possible option would be to recommend burning half their bond, then distributing the remaining bond to honest participants who are checked in, this acts as a disincentive for validators to be inactive and hold up the finality window. The negative consequence would be that this would incentize validators to censor others, so the game theory would need to be worked out there.
- Holding: if one of the validators decides to not submit, all other honest participating parties can hold off on submitting a check-in past the last honest check in (the one beyond the lagging validator), either until the lagging validator comes back online or is slashed by the 7 day challenge window. Reducing cost for active validators during grieving.
- Time-based penalties: we can consider multiple time based response mechanisms, such that there are different slashing punishments the longer the lagging validator is inactive. Such that, there is more and more incentive to come back online with a check in.
- MPC: to reduce honest inactivity, validators could use MPC based setups over different nodes to ensure that even if a minority of their nodes go offline, that their validator continues to publish check-ins. This setup is commonly used in proof of stake validator blockchains.
- Stake-weighting: the bonding process could be such that lower bonded validators could be slashed earlier, where higher bonded validators can extend the window longer. The risk here is that smaller honest minorities are treated worse in the slashing game, but they would also have less bond at stake. An example would be if you're bonded 10 ETH, then you can extend the finality window up until 7 days, but if you're bonded 1 ETH you can only extend it to 3 days (but this should only be lowered to a safe lower bound such as 24 hours). If multiple honest minorities go offline, then you can add the stake of all lagging minorities together which would result in a larger window than any single lagging minority.

Costs:

- Check-in's would be a relatively simple transaction. Likely only the base cost + a single state modification and a few minor checks. Despite this, having all validators check-in and do so very frequently, would result in potentially high costs for honest validation.
- If a validator were to check-in every hour, with an average transaction costing an average of ~3 USD on Ethereum [11], that would be 8760 check-ins a year costing around $26,280 USD, if we have multiple validators, say 5, this could cost closer to $131,400 per year in Ethereum L1 fees if the mechanism is naively constructed.
- While certain batching techniques could be applied, this is still a very high cost penalty per rollup without batching or aggregation.

Reducing Check-in Cost:

- Zk-Aggregation: this could be used to aggregate the check-ins of many roll ups into a single check-in proof, drastically reducing the check-in cost for all participating rollups in the happy path case. Of course, in some unhappy paths, you would fallback to normal more expensive check-ins.
- MPC Multi-Party Signatures: MPC over ECDSA signatures could reduce cost significantly for honest validators when posting check-ins to Ethereum, such that they could both ensure better uptime for nodes going down and signature batching between honest and lively validators.
- Validators could split submission costs, such that 26,280k per year could be split between potentially 100s of honest validators. If we take the cost of an hourly check-in $26,280 USD and divide it by 100, that’s an annual cost of 262.8 per validator, a much lighter on-chain cost.
- MPC in the highest secure setting should require an N of N or as close to N of N construction, such that any honest minority can prevent a valid group signature. In which case, the group can reform and remove the lagging validator, likely with some additional bonding and punishment.
- Validators could split submission costs, such that 26,280k per year could be split between potentially 100s of honest validators. If we take the cost of an hourly check-in $26,280 USD and divide it by 100, that’s an annual cost of 262.8 per validator, a much lighter on-chain cost.
- MPC in the highest secure setting should require an N of N or as close to N of N construction, such that any honest minority can prevent a valid group signature. In which case, the group can reform and remove the lagging validator, likely with some additional bonding and punishment.
- Block Submission Check-ins: Check-in submissions could potentially be included within block submission, such that the necessary state elements are changed during by honest builders. This would also significantly reduce submission cost, depending on the implementation.
- Expanding the check-in window length, if the window is a little longer, say 4 hours, this would reduce the cost 4x from initial 1 hour estimates.
- Request based check-ins: the reporting window could also be less frequent and only when enough withdrawals are requested. If an honest minority who is staked fails to report, the window extends until the 7 day mark.
- Passive Validation: validators may still chose to validate without a check-in which would be effectively free of on-chain cost, this would mean that they are subject to the very small submission window, but so long as Ethereum remains not under a censorship attack, even a 1 hour window should be enough to get a regular submission in. This does not cover the most adversarial cases, but does cover some happy path cases of validation. Rollups could choose a mixture of check-in based and passive validation nodes to ensure security, both from censorship and from invalid state transitions.
- Offshoring to Based Rollup for Censorship Resistance: many roll ups could also off-shore the security assumptions to a third-party rollup, such that so long as one honest minority in that rollup attempts to submit the request transactions and cannot, the window is extended, whereby many other rollups can then look to this censorship rollup for their own finality window. If any rollup is being censored, all rollups finalisation is extended. Validators of that rollup would have a very high bonding and slashing scheme and again, validators would need to have a safe transport mechanism and mempool to ensure they can receive all request fraud proving transactions quickly.

[

4

954×697 46 KB

](https://ethresear.ch/uploads/default/original/2X/2/2909fbbda891bc7b09f770c4716d7d7d6d1a4122.png)

Rewards Systems and Staking:

- Honest validators could be rewarded similar to any staking model, whereby random stake weighted rewards could be applied overtime.
- Transaction fees could be used to pay the additional security budget for honest minorities during block production.
- Newly minted rollup tokens could be dispersed for those who validate and are active to incentivize honest validation.

Downsides of Short Check-in Windows:

- Short check-in windows do come with a potential downside, such that if there is slashing applied to a very time sensitive window, it could negatively impact honest validators if they are censored.

- Censoring, in this case, can be used against validators if the penalty is overly sensitive and the minimum slashing period set too short without any recourse mechanisms.
- A possible solution to this would be based around the already existing prior art we have, such as using the base fee, detection of forked blocks, a governance mechanism such as a token DAO or committee, or a third-party shared sequencer to help better calibrate the slashing penalties during more time sensitive windows, namely for offline or inactive validators.
- These mechanisms in this case would only be used to calibrate penalties and not the window itself, ensuring that users have enough time / warning to withdraw if penalties are unreasonable for validators. If they are applied poorly, honest validators could make their case to the community and people would have enough time to withdraw in this event if nothing can be done.
- In the event a rollup applies too harsh a penalty/requirement without appropriate recourse, validators would be disincentivised to validate the rollup which would be a negative outcome for the rollup itself. This follows a similar concept to any blockchain in regard to operational cost and penalties, so Rollups will want to set these carefully.

Bugs:

- There are many reasons to have a longer challenge window outside of censorship, one of these are bugs.
- If there is a bug in the state transition function, the only possibility would be for an honest actor to identify this and proceed with no-report.
- This bug surface area also exists in zk-rollups as well and with the same potential level of consequence.
- Watchtower based validators could be implemented to reduce mass withdrawal scenarios.
- An added benefit of the check-in model is that, if a validator panics during a state transition function, the validator could be designed to not report, and thus allow up to 7 days for developers to fix the issue.
- A hyper aware validator may be able to hit pause in time, thus giving a security council the 7 days to upgrade state transition function.

Bribery and Collusion:

- In the case of the check-in model, an adversary would likely attempt to bribe or collude with validators. Which is why it's important that for each rollup at least one well aligned honest validator is operating, however, unlike large honest majority validator set style mechanisms, only one honest validator is needed for proper state transition enforcement.
- Bribery exists within operating ORs today. A well funded adversary could collude with all the known rollup validators and have them pretend to validate until it is too late for anyone to react (post 7 day finalisation). In conventional 7 day window designs, it may be the case that someone could notice this and spin up their own nodes or intervene with a security council thus countering this form of attack.
- Another notable benefit conferred from a 7 day window non-checkin model is that it is harder to see who is validating on chain as there are no check-ins, and thus hard to collude on exactly all the parties involved. There would also be more time to react or notice invalid state transitions and apply a patch or spin up an honest node to defend the chain. However, understanding how much that really helps the situation is hard to quantify. In all ORs, the main security assumption of a single honest party is still required and this still holds true in the check-in model or longer conventional ORU designs.

Universal Interoperation between Optimistic Rollups using Zk Aggregation for Check-ins:

- If we maintain that for each rollup there is one honest minority, that all validators must check-in in the happy path and many roll ups check-into a single universal aggregated zk-proof creation process, we can assert the state of finalisation for different rollups is true across rollups based upon the complete last check in and validator set, enabling better cross-rollup interoperation for ORs.
- This enables faster bridging between ORs in the happy path cases.
- In the unhappy path, bridging or finalisation acceptance may still take up to 7 days for certain lagging ORs.
- A note that zk-aggregation of check-ins would be significantly less proving computation than proving rollup execution for all rollups and likely far cheaper than posting all votes on Ethereum.

Conclusion:

- In this post we conclude that a check-in based challenge design enables an honest minority to thwart mass censorship attacks, reducing the overall finality time of Optimistic Rollups significantly from 7 days to only a few blocks (depending on the security budget) while still defending against all censorship attacks.

- We believe this dead man switch mechanism might also have applications elsewhere where time based operations are required that may be subject to censorship attacks, such as lending CDPs, time sensitive voting, or broadly time sensitive decentralised applications.
- We also believe that this no-report mechanism is useful in events of panicking state transition functions, watchtowers, bugs or mass withdrawals and improved cross-chain interoperation for ORs while giving projects and developers enough time for an emergency upgrade if need be.

Explain Like I'm 5 (EL15):

1. Censorship can stop validators signalling fraud in Optimistic Rollups (ORs).
2. To counteract censorship, ORs have a 7-day period where challenges can be made. This long window makes it unprofitable to censor fraud signals on Ethereum.
3. However, validators in ORs can also regularly signal that everything is correct.
4. If the only thing that can be censored is the signalling of correctness, we can also assume no signalling means censorship or fraud. A short challenge window can then increase passively until the challenge is submitted, up to a safe maximum of 7 days.
5. This ensures validators have enough time to overcome any censorship attacks while keeping the challenge and finality window short in the happy path.
6. With these measures, the time it takes for transactions in ORs to be considered final can now be safely reduced to just a few Ethereum blocks.

References:

[1] Why is the Optimistic Rollup challenge period 7 days?

[2] Non-attributable censorship attack on fraud-proof-based Layer2 protocols

[3] Fighting censorship attacks on smart contracts

[4] Nearly-zero cost attack scenario on Optimistic Rollup - Layer 2 - Ethereum Research

[5] Ethereum is Inherently Secure Against Censorship

[6] Responding to 51% attacks in Casper FFG - Proof-of-Stake - Ethereum Research

[7] Breaking BFT: Quantifying the Cost to Attack Bitcoin and Ethereum

[8] What's up with Rollup

[9] Reducing challenge times in rollups - Layer 2 - Ethereum Research

[10] Sliding Window Challenge Process for Congestion Detection

[11] YCharts Ethereum Average Gas Prices

[12] Challenging Periods Reimagined: Road to dynamic challenging periods