Profitable Censorship MEV

Note that herein lies Notes from @colludingnode presentation at Research day New York 2023. The highlight of this research is the exploration of censorship and liveness trade-offs for sovereign rollups with respect to the sequencer role: (i) based sequencing, (ii) centralized sequencing, (ii) and decentralized sequencing scenarios. This doesn't get into shared sequencing. These trade-offs are considered at the cost of leaking MEV to the base layer. My takeaway is that BFT voting combined with base layer fallback hostage blocks

which synchronize as DA layer blocks and act as a defacto secondary or dual proposer for the rollups (not conretely) – concretely hostage blocks serve the role of inclusion lists a topic of prominant research in the Ethereum community since late 2021. Overall this was a fantastic presentation and a key piece of research that will support forthcoming publication.

- Question: Is there a way for the validators of a DA layer to extract MEV from sovereign rollups by censoring, delaying or otherwise messing with the sequencers of the sovereign rollups

Sovereign Rollups

- Reach consensus by scraping blocks out of the DA layer, and applying a fork-choice rule
- Inherit safety from DA Layer

Sovereign rollups are a bit different than "settled rollups" as you see them in the Ethereum world as L2s. Rather than having a settlement contract, sovereign rollups have their own sovereign nodes which connect to some DA Layer such as Celestia and they scrape rollup blocks out of DA Layer Blocks and reach consensus by applying a fork-choice rule to blocks in the DA Layer.

Based Rollups (Pure Fork-Choice Rollups)

Let's start by talking about the simplest kind of sovereign rollup - Based rollups or pure Fork-Choice rollups. These are the simplest kind because they don't utilize any sequencer or sequencer scheme. Instead, the blocks are built by anyone permissionlessley out of transactions from the rollups mempool, sent to the DA Layer, then the nodes reach consensus by applying some kind of FCR.

Fork-choice rules can be expressive:

- First-ordered - differ to ordering in DA Layer
- Gas burned - select rollup block at certain height that burns the most gas, ultrasound money rollup
- AMM LP Returns - select the block that returns the most MEV to AMM LPs.

OG post from C-Node with Diagram of Gas based FCR along with follow up conversation.

- can see a bunch of blocks go to DA Layer blocks. The nodes of the rollup look at them then can construct a chain out of the valid block at the specified height that wins the FCR.
- There are DA Layers validators who ultimately get the final say over ordering and what goes in DA Layer blocks.
- The validators can decide they don't like the block that wins the FCR and maybe there is a block they like better because it is more profitable or gives them more control over the ordering.
- The real fcr at the end of the day is just highest bribe. This assumes DA layer is in its current form, minimally modified DA layer chain. Does not include any fancy censorship resistance schemes; threshold encryption schemes, or multiple concurrent block proposers like duality.
- There are DA Layers validators who ultimately get the final say over ordering and what goes in DA Layer blocks.
- The validators can decide they don't like the block that wins the FCR and maybe there is a block they like better because it is more profitable or gives them more control over the ordering.
- The real fcr at the end of the day is just highest bribe. This assumes DA layer is in its current form, minimally modified DA layer chain. Does not include any fancy censorship resistance schemes; threshold encryption schemes, or multiple concurrent block proposers like duality.

How do we not leak MEV to the base layer and come up with alternate sequencer schemes outside of based rollups?

Why?

- Rollups can keep their own MEV

- Provide faster than L1 UX with pre or soft confirmations

We will focus on keeping the MEV sovereign off the base layer and redirecting it to the rollups themselves.

The Simplest way to keep MEV sovereign and off the DA layer is to just use a centralized sequencer for the rollup

- you choose some privileged signer and reject all of the blocks from the rollups not signed by that signer and you accept the privileged sequencers blocks as canonical blocks for the rollup
- In the simplest scheme you trust the sequencer for censorship resistance and liveness. You get safety from the DA Layer.

There are ways from Ethereum land proposed for how a rollup with a centralized sequencer can inherent CR and liveness from the L1.

- Hostage blocks (rollkit team)- a basic scheme where user can send a transaction directly to the DA layer and send it past the sequencer. Every couple of sequencer blocks you hostage blocks which must include all of the transactions sent past the sequencer just to the DA Layer
- One way to implement this is to synchronize hostage blocks with sequencer blocks. Every n

sequencer blocks you have to have a hostage block come after that. This is like a forced TX inclusion scheme

- Base Layer Fallback - synchronize the hostage blocks with DA layer Blocks
- One way to implement this is to synchronize hostage blocks with sequencer blocks. Every n

sequencer blocks you have to have a hostage block come after that. This is like a forced TX inclusion scheme

- Base Layer Fallback - synchronize the hostage blocks with DA layer Blocks
- If you do a forced TX inclusion scheme you do get CR from the base layer but not liveness because if the sequencer stops making blocks the rollup halts
- BLF - even if the sequencer shuts down the rollup continues because there will always be hostage blocks synchronized by the DA layer. There become ways the DA layer can mess with the rollup by censoring/re-ordering etc.
- With the Forced TX inclusion scheme - if you simply accept the ordering of the user's transactions in the DA Layer as the execution order its trivial to understand how the DA layers get MEV from that as they control ordering.
- You could also include verifiable sequencing rules and get rid of that and reduce it to a problem of inclusion
- You could also include verifiable sequencing rules and get rid of that and reduce it to a problem of inclusion
- Also when you add a second way for users to get transactions sequenced you may wind up with a competitive dynamic where one of the ways which is controlled by DA validators cost less than going through the sequencers or vica versa, perhaps one is subsidized? Also sometimes one way is more expensive than the other. Celestia has a ton of capacity so should be cheaper to send just to the DA layer. However depending on the blob commitment schemes where two blobs cost more than a single blob there may still be cost saving benefits to user by batching it through the sequencer
- If you do the base layer fallback scheme all the previous strategies of getting MEV carry over and now they can censor/delay and front-run the sequencer entirely
- Quite a few MEV strategies now become viable when you choice to inherent liveness as well as censorship resistance from the base layer
- Quite a few MEV strategies now become viable when you choice to inherent liveness as well as censorship resistance from the base layer

Conjecture - If sovereign rollups inherit censorship resistance and or liveness from the base-layer, some form of MEV leaks down.

Do decentralized sequencers change anything?

If you replace the centralized sequencer with a decentralized scheme there are ways you can have better censorship resistance and liveness than a centralized scheme but fundamentally you cannot inherit those traits from the DA layer.

It is interesting to think about different decentralized sequencer schemes and all of the different ways the DA layer validators can mess with the different schemes and profit from doing so.

- BFT voting
- Stake-based FCR

- Round robin with Halts
- Round Robin with Skips

BFT voting

- Alternative to a centralized sequencer
- Cost to create a liveness failure is the cost of 1/3 of the rollup's tokens
- Can inherit liveness at the cost of some MEV
- Why be a rollup instead of an app chain?

BFT votings full on consensus on Layer 2. Sequencers for the rollups are a lot like validators for a blockchain. You may want to use something like tendermint for this. It is an alternative to a centralized sequencer. The cost of creating a liveness failure for this scheme is equal to the cost of getting one third of the stake. If that liveness is compromised that rollup halts unless you are using hostage blocks and you inherit those things from the DA layer and leak some MEV

One of the main selling points of sovereign rollups is to not need to pay for a validator set or go through the hassle of doing this. This is one major concession you make if you want to set up your rollup this way - it does require provisioning infrastructure, getting stakers, setting up validators and all that.

What do you even get? vs. being an app-chain or an app-rollup?

- you have to pay for infra and DA but what you get is trust minimized interoperability with other things on that DA layer and good light client support.

Stake-based fork-choice rules

- Like Gasper but uses Celestia as the finality gadget for some liveness-favoring fork-choice rule
- Stake-weighted "dice roll"
- Like based rollups, it's easily exploitable by Celestia validators

These are talked about a bit less. If you were to use Celestia as a finality gadget for some liveness favoring fork-choice rule like how Gasper works in Ethereum.

- You can have there always be some block proposer able to propose blocks.
- can do stake-weighted dice roll; use a VRF or RANDAO or random beacon based on proposer's stake to decide who wins the rollups block production slot
- The Celestia validators ultimately get the final say and can shoot down forks where they don't make profit, pick winners in this scheme, and gain quite a bit of control over MEV
- can do stake-weighted dice roll; use a VRF or RANDAO or random beacon based on proposer's stake to decide who wins the rollups block production slot
- The Celestia validators ultimately get the final say and can shoot down forks where they don't make profit, pick winners in this scheme, and gain quite a bit of control over MEV

Round Robin with Halts

- Rollup halts until the sequencer shows up to build a block for its slot
- MEV is 100% sovereign, but at a horrible cost
- If you don't want to do full BFT voting scheme you can take the round robin from tendermint and not have voting but still have rotation and a leader schedule.
- All of the leader selection schemes you see in BFT chains like Tendermint-based or Solana, have ways to recover from a validator missing their slot
- if you get rid of those recovery mechanisms and wait for them to show up to their slot - so you have the rollup stop until the scheduled proposer shows up to their slot you do not leak any MEV but at a horrible cost.
- One proposer can halt the chain by not showing up. The only way to recover is through a hard-fork. This is a pretty

bad trade-off but it keeps all of the MEV sovereign.

- One proposer can halt the chain by not showing up. The only way to recover is through a hard-fork. This is a pretty bad trade-off but it keeps all of the MEV sovereign.
- if you get rid of those recovery mechanisms and wait for them to show up to their slot - so you have the rollup stop until the scheduled proposer shows up to their slot you do not leak any MEV but at a horrible cost.
- One proposer can halt the chain by not showing up. The only way to recover is through a hard-fork. This is a pretty bad trade-off but it keeps all of the MEV sovereign.
- One proposer can halt the chain by not showing up. The only way to recover is through a hard-fork. This is a pretty bad trade-off but it keeps all of the MEV sovereign.
- All of the leader selection schemes you see in BFT chains like Tendermint-based or Solana, have ways to recover from a validator missing their slot
- if you get rid of those recovery mechanisms and wait for them to show up to their slot - so you have the rollup stop until the scheduled proposer shows up to their slot you do not leak any MEV but at a horrible cost.
- One proposer can halt the chain by not showing up. The only way to recover is through a hard-fork. This is a pretty bad trade-off but it keeps all of the MEV sovereign.
- One proposer can halt the chain by not showing up. The only way to recover is through a hard-fork. This is a pretty bad trade-off but it keeps all of the MEV sovereign.
- if you get rid of those recovery mechanisms and wait for them to show up to their slot - so you have the rollup stop until the scheduled proposer shows up to their slot you do not leak any MEV but at a horrible cost.
- One proposer can halt the chain by not showing up. The only way to recover is through a hard-fork. This is a pretty bad trade-off but it keeps all of the MEV sovereign.
- One proposer can halt the chain by not showing up. The only way to recover is through a hard-fork. This is a pretty bad trade-off but it keeps all of the MEV sovereign.

Round Robin with Skips

- “Fewest skips” turns out to be a type of stake-based fork-choice rule
- Missed slot recovery mechanism is exploitable by DA validators
- “Halting period” makes it slightly harder to extract MEV

Is there a way to skip people who miss their slot? You then have different forks being built where if someone misses their slot the next person builds at that height for them.

- Later they could show up to their slot and build at that height anyway.
- Then you would have two forks and ultimately the rollup nodes would have to apply a fork-choice rule to the different chain segments that landed on the DA layer.
- It once again becomes a stake based fork-choice rule that the Celestia validators can mess with. If you have a recovery mechanism for missed slots in a leader schedule that is something that DA validators can exploit potentially gain control over ordering and profit from it.
- One possible mitigation is what we call a halting period. If the proposer misses their slot you wait one DA block. That way if they miss their slot because they got censored at least a different DA proposer will be there for the recovery. The operator who censored may not profit then.
- If you want to inherit liveness from the DA layer you will leak some MEV
- It once again becomes a stake based fork-choice rule that the Celestia validators can mess with. If you have a recovery mechanism for missed slots in a leader schedule that is something that DA validators can exploit potentially gain control over ordering and profit from it.
- One possible mitigation is what we call a halting period. If the proposer misses their slot you wait one DA block. That way if they miss their slot because they got censored at least a different DA proposer will be there for the recovery. The operator who censored may not profit then.
- If you want to inherit liveness from the DA layer you will leak some MEV

Mitigations and Solutions to profitable censorship MEV

Mitigations

- Sovereign or shared decentralized sequencer networks
- Rollkit's "Domino" Scheme
- Let it Leak (PBS for DA Layer)

Solutions

- Improve DA layer censorship resistance
- Threshold encryption
- Shared builder (multiplicity)
- Use decentralized sequencer schemes with high economic security such as your own BFT consensus or a shared decentralized sequencer with its own token and high cost to create a liveness failure
- You could use a round robin scheme with different ways you can configure it with halts, halting period, or leak MEV
- If you are okay with that then we need to deal with MEV on the DA layer so it doesn't lead to gas wars or problems like this.

You can solve the problem entirely if you make the base layer more censorship resistant. If Threshold Encryption schemes work well, you can add these to the DA layer and have pay for blob transactions be private or least have guaranteed inclusion before ordering. Also maybe these shared builder schemes like multiplicity can allow all the validators of the DA layer participate instead of having one proposer monopolize the whole thing

Additional Notes

Hostage blocks eliminate the risks of centralized sequencer causing liveness failure for the rollup and offering a censorship resistance path, which could be further augmented with Ferveo at the base layer and/or multiplicity scheme.

Hostage Blocks

With a centralized sequencer you are trusting it for censorship resistance and liveness, you still retain safety properties of the base layer.

Hostage Blocks - user sends transaction to the da layer instead of to the sequencer. Every couple sequencer block you have a hostage block which has to include the txs sent past the sequencer to the DA layer.

2 different ways to implement

1. Force Transaction Inclusion (FTI) - synchronize hostage blocks with sequencer block
2. Every N sequencer blocks you have a hostage block come after that
3. If you do it this way it becomes like a forced transaction inclusion scheme
4. Every N sequencer blocks you have a hostage block come after that
5. If you do it this way it becomes like a forced transaction inclusion scheme
6. Base-layer Fallback (BLF) - Synchronize hostage blocks with DA blocks not the rollup blocks

Property

Base layer Fallback (BLF)

Forced Transaction Inclusion (FLI)

L1 Safety

L1 CR

L1 Liveness

Liveness is retained in BLF because even if Sequencer goes down hostage blocks will still reach the DA layer and they will

be included with DA layer blocks.