Fernet on the rocks - Sequencer-Prover Consolidation Proposal

Summary

This proposal has one goal: unify the role of sequencers and provers in Aztec, maximizing protocol simplicity. Specifically in the context of Fernet (a random leader election), I outline the most simple version of block production possible. The sequencer can only propose blocks that they have the ability to prove themselves. Their “proving period” begins immediately after the proposal phase, and must end within the specified window. That’s about it, folks!

TLDR; It is entirely on the sequencer to propose and prove their own blocks.

Comparisons

This proposal is basically Fernet, and saying that Fernet is good enough to get (small but sufficiently sized) blocks produced. This is why it’s called “fernet on the rocks”.

This proposal is most similar to Sidecar however it effectively removes the reveal phase, and proving commitment phases entirely. This is a nice simplification that makes the protocol easier to reason about and build. It also ensures that the protocol is not fundamentally dependent on third party proving marketplaces, or an out of protocol RFQ/quote/auction mechanism. However, it is likely that less well resourced sequencers in fernet on the rocks still subcontract out of protocol to others that have larger resources, if it economically makes sense to do so.

It is a significant departure to cooperative proving network and stake based proving network because it does not try and define a proving network that could have larger throughput capacity. This is a much, much simpler design at the basic cost of throughput capacity, &/or potential decentralization. ie. the network could only be as decentralized as the sequencer/staker set, and fernet’s random leader election, which could be a hundred to thousands randomly selected per day… which in my personal opinion is still quite good!

Details

[

image

1231×396 73.7 KB

](https://europe1.discourse-cdn.com/business20/uploads/aztec/original/1X/641731b586132cbfcc568505b944a2589527e53f.png)

Proposal phase

As defined within Fernet.

Estimated duration: 3-5 Ethereum blocks

Reveal phase

This proposal suggests removing Fernet’s suggested reveal phase. An honest sequencer wouldn’t withhold themselves or propose a block they are not sufficiently confident in proving.

Proving phase

The highest ranking sequencer now has ~8 minutes to prove their block (which they now know they are the leading candidate and therefore it’s worth their time/effort/compute) and submit it to L1 for verification. In this model, it is likely that significantly large sequencers operate their own proving hardware/infrastructure, and smaller (less well resourced) sequencers subcontract these proofs out of protocol to third party marketplaces, such as nil or gevulot.

Estimated duration: 40 Ethereum blocks (8 min)

Slashing considerations

A sequencer becomes “elected” if they a) publish a valid block proposal during the proposal phase, and b) have the highest ranking VRF output/score. We know that they are ‘live’ and not subject to internet connectivity issues, or otherwise, due to

their participation in the proposal phase. In the event that an “elected sequencer’s” proposal does not end up proven and finalized by the end of the proving phase, this proposal suggests we slash 50% of their stake (assuming fixed deposits) and burn their stake permanently, for stalling the network’s liveness and likely causing a block reorg. This means that there’d be “two strikes, and you’re out”, effectively. Perhaps too aggressive.

The cost of attacking/censoring

These are tunable parameters of the system, figures provided as reference

144 sequencers per day (1 per 10 minute slot).

32 eth (equivalent) staked deposits of a token to become a sequencer.

This has a current value (at the time of writing) of $56,724.48.

If we slash 50% of your stake, that’d be ~$28,362.24.

If you got maximally lucky or sophisticated (eg dominant of a staking pool), the cost to censor the network per day could become as low as ~$4,084,162.56 or ~2,304 eth (of the token being burnt). It is not a fully repeatable attack, in the sense that 50% of the stake deposit gets burnt each time. Surely the attacker could continue purchasing tokens and adding new sequencers, but eventually they would drive the token price up significantly beyond the value of the attack, or an attacker would simply run out of stakable tokens available for purchase.

Outstanding questions

1. Should this change Fernet to a a multi-leader (candidate) election?
2. There are ongoing conversations around whether to restrict Fernet to a single leader at the end of the proposal window (i.e. the sequencer with the highest random number generated, that was published during the proposal window) or allow multiple (i.e. the top 3 or N VRF outputs) to act as redundant leaders to ensure liveness.

3. In this proposal, it may make sense to nominate N candidates, and the highest scoring proven block wins. This would mean only N computers are potentially working on “redundant” or work that may not end up being canonical, and therefore potentially only N “uncle rewards”… Which could be acceptable for the increased liveness guarantees.

4. In this proposal, it may make sense to nominate N candidates, and the highest scoring proven block wins. This would mean only N computers are potentially working on “redundant” or work that may not end up being canonical, and therefore potentially only N “uncle rewards”… Which could be acceptable for the increased liveness guarantees.

5. There are ongoing conversations around whether to restrict Fernet to a single leader at the end of the proposal window (i.e. the sequencer with the highest random number generated, that was published during the proposal window) or allow multiple (i.e. the top 3 or N VRF outputs) to act as redundant leaders to ensure liveness.

6. In this proposal, it may make sense to nominate N candidates, and the highest scoring proven block wins. This would mean only N computers are potentially working on “redundant” or work that may not end up being canonical, and therefore potentially only N “uncle rewards”… Which could be acceptable for the increased liveness guarantees.

7. In this proposal, it may make sense to nominate N candidates, and the highest scoring proven block wins. This would mean only N computers are potentially working on “redundant” or work that may not end up being canonical, and therefore potentially only N “uncle rewards”… Which could be acceptable for the increased liveness guarantees.

8. What is the expected, or target network throughput?

9. This proposal’s primary downside is the inability to propose and prove larger blocks, that could potentially be proven by a large distributed or federated proving network, or a large and decentralized proving marketplace. Which could significantly limit the network’s throughput/capacity. Again, notably, this protocol doesn’t prevent sequencers from leveraging third party proving marketplaces via subcontracting (none of them can…), but in this case none of the actors have “enshrined economic guarantees”.

10. This begs the question of expected average, required, and minimum hardware + networking targets…

11. This proposal’s primary downside is the inability to propose and prove larger blocks, that could potentially be proven by a large distributed or federated proving network, or a large and decentralized proving marketplace. Which could significantly limit the network’s throughput/capacity. Again, notably, this protocol doesn’t prevent sequencers from leveraging third party proving marketplaces via subcontracting (none of them can…), but in this case none of the actors have “enshrined economic guarantees”.

12. This begs the question of expected average, required, and minimum hardware + networking targets…

13. Does it make sense to run the VRF in advance of the proposal phase? eg 24 hrs?

14. The notion of “proving your own block proposals” leads me to believe they’d want to start proving as quickly as they could, so they could build as big of a block as possible, in order to get the most fees/MEV possible. Maybe it makes sense to try and make sure they can run the VRF at least 24 hours in advance (based on yesterday’s RANDAO values?) to generate their probability of winning the (future) election?

15. The notion of “proving your own block proposals” leads me to believe they’d want to start proving as quickly as they could, so they could build as big of a block as possible, in order to get the most fees/MEV possible. Maybe it makes sense to try and make sure they can run the VRF at least 24 hours in advance (based on yesterday’s RANDAO values?) to generate their probability of winning the (future) election?