Whisk-y is a sequencer selection protocol proposal tailored for Aztec, based on George Kadianakis' "Whisk: A Practical Shuffle-based SSLE Protocol for Ethereum" published on ethresearch in January 2022. This proposal unabashedly adapts the Whisk protocol to suit Aztec's requirements. Its primary goal is to offer a reasonably fair and cryptographically secure method for selecting a lead sequencer from a permissionless set for a specific time duration. Key benefits of Whisk include its ability to provide some privacy, randomization preventing network centralization around superior hardware, and leveraging the Ethereum mainnet for coordinating sequencers, ensuring optimal censorship resistance. However, some trade-offs involve potential loss of capital efficiency and increased complexity in the protocol's implementation.

Details

As the whisk proposal eloquently stated…

"The beacon chain first randomly picks a set of election candidates. Then and for an entire day, block proposers continuously shuffle that candidate list thousands of times. After the shuffling is finished, we use the final order of the candidate list to determine the future block proposers for the following day. Due to all that shuffling, only the candidates themselves know which position they have in the final shuffled set."

I propose that we employ a verifiable randomness function (VRF) to select 1,440 candidates daily from the entire permissionless set of potential sequencers. This subset, called candidates, is then verifiably shuffled, eventually resulting in an ordered list. The protocol takes the first 144 candidates from this list as sequencers for the next 24 hours, with each granted a 10-minute window to act as the "lead sequencer" and produce rollups. Note that the specific parameters serve as a baseline and are open to discussion or change. For example, the number of candidates could be 2-10x higher, or different durations could be considered, such as having 720 sequencers per day with 2-minute leadership periods. The proposal assumes access to a reliable VRF but does not recommend a specific one (Whisk uses RANDAO, but other promising solutions are being explored).

How it works

[

whisky-overview-diagram

1801×1013 72 KB

](https://europe1.discourse-cdn.com/business20/uploads/aztec/original/1X/b021ebb7ea5d0a0fa244f6a2bcef86b7c09c8e41.png)

1. Bootstrapping

Users can join the sequencer set by staking a minimum amount of a specific token on the Ethereum mainnet and adding a randomized "tracker," initially a dummy tracker that can be updated. To ensure sufficient decentralization from Day 1, the protocol waits for a minimum number of trackers before proceeding to step 2. Once a specific threshold, e.g., 10,000 staking commitments, is reached, a VRF is used to finalize the bootstrapping phase by reducing the sequencer list to 144 candidates for Day 1. The 144 number of sequencers correlates to each getting a 10-minute window to act as a leader and produce a rollup, which is then posted to Ethereum (via step 5).

1. Candidate selection

Using the VRF, a randomly chosen subset of sequencer candidates is selected from the list of trackers for a specific duration, e.g. every 24 hours. This ensures a (relatively) equal chance for all staked participants to be chosen, without prioritizing those with better hardware (but admittedly giving those with more stake a larger chance, as is true for almost all PoS systems). In Aztec's case, 1,440 trackers could be selected as candidates for the next day. With this number, there's a 10% chance of becoming one of the 144 trackers who will act as a leader, assuming a 10-minute "leadership period". These specific parameters can be adjusted to account for factors such as rollup production time, minimum staking requirements, and desired potential anonymity set. It would be interesting to consider not requiring any minimum window for rollup production, and go through a variable number of sequencers from the ordered list.

1. Shuffling

Trackers corresponding to the candidates are continuously shuffled by the current sequencer for a specific duration, such as every 24 hours. Proofs of shuffle are posted on the Ethereum mainnet for each leadership slot. While the exact shuffling technique remains open for discussion, options include the Feistelshuffle algorithm from Whisk (although they may have moved on to another method? (ref)). As for the party responsible for shuffling, the simplest solution could be assigning this task to the current sequencer, in addition to their other responsibilities outlined in the RFP.

1. Leader selection

The final order of the shuffled commitments determines the order of sequencers, or "sequencer leaders," for the upcoming 24-hour period.

1. Rollup production

The chosen sequencers perform the following tasks, taken directly from the RFP, during it's time as leader:

1. Select pending transactions from the mempool
2. Order transactions into a block
3. Verify all private transaction proofs and execute all public transactions to check their validity
4. Compute the ROLLUP_BLOCK_REQUEST_DATA, which includes private state database updates, public state updates, and KZG blob commitments (EIP-4844)
5. Compute state updates for messages between L2 & L1
6. Broadcast the ROLLUP_BLOCK_REQUEST_DATA to the prover network via the proof pool for parallelizable computation

I currently think that this step requires the sequencer to reveal their place in line, or at least, attesting to their right to produce this block via their tracker in line. At this point, they may no longer be secret, depending on reveal mechanism, but they have committed to the transactions within a block/rollup already. In theory, there could be a separate entity responsible for the following steps. If so, the "proposer" should be incentivized to build the block that ends up being worked on and put into the rollup.

1. Build a rollup proof from completed proofs in the proof pool
2. Tag the pending block with an upgrade signal to facilitate forks
3. Publish completed block with proofs to Ethereum as an ETH transaction.

If a sequencer produces a valid rollup, processed by Ethereum, they should be incentivized to do so. It is worth exploring (if?) slashing could be implemented here. We could also in theory have a different entity aggregate proofs and publish the rollup (steps 5.g through 5.i), to give some semblance of proposer-prover-aggregator (?) separation.

Beyond those defined in the RFP, this proposal adds the additional requirements of:

- Revealing their tracker to verify they are the current leader (during step 6)
- Shuffling the list of candidates (throughout it's time as a leader)
- Publishing a proof of shuffle alongside the rollup on L1 (during step 9)

Feasibility

Implementing Whisk-y into Aztec over the next 6 months appears very feasible, given that it is based on a protocol that's been out there/discussed for over a year (which is a relatively long time in this field/design space). However, potential challenges may arise in terms of integration with the zk-zk-rollup architecture, and/or ensuring the desired levels of privacy when coordinating with the proving network. The available references, such as a rust implementation, is a big +1 in my book, when compared to net new solutions.

Comparisons

In comparison to Whisk, this protocol reduces validator count and candidate selection from 16k trackers and 8k candidates down to a much smaller number of 1,440 candidates and only 144 leaders/sequencers per day. But otherwise it's basically trying to be the same protocol (unabashedly). While a smaller number and potentially more centralized than Ethereum, which will have worse privacy implications, this aligns to the expectation that sequencers will get a 10 minute window to act as a sequencer, due to potential proof generation and execution times. Depending on benchmarks, we could adjust / tune these parameters.

Compared to Sequencer Selection: PBS with a federated prover network, another proposal to this RFP, it is quite different. This other proposal is almost entirely market based, and uses cryptoeconomic incentives to ensure the rollup will be incentivized to continue moving forward, in contrast to the more traditionally cryptographic guarantees that this proposal provides. Additionally, unlike Joe's proposal, it does not make any suggestions on how to coordinate steps e-f with the proving network. This will come in a future RFP/proposal, I'd imagine.

Pros

1. This is a randomized election, and therefore does not prioritize best hardware, nor largest bidders.
2. Well understood and documented compared to alternatives - with available references, specs, & rust reference

implementation, etc.

Cons

1. Could be better designed to ensure liveness & rollup production (10 minutes is a long time if someone misses their chance or is acting nefariously…)
2. Somewhat complicated
3. Capitally inefficient as it requires some % of tokens to be actively staked, doesn't create or establish a market to evaluate what block space ought to be worth, etc.

Questions

1. What are we using for randomness?

1.a. Current thought: RANDAO

1. What quantity of tokens need to be staked on L1 to ensure sybil resistance?

2.a. Current thought: answer is cheap enough that average folks can participate, but not cheap enough someone could acquire enough to take the network offline for a significant amount of time.

1. Do we slash sequencers who do not submit a valid rollup during their time as leader?

3.a. While this proposal does not yet address slashing, it seems easy enough to imagine that if a sequencer fails to produce a rollup within their time period, we slash some of the stake.

1. What size candidate set should we choose?

4.a. I think 1,440 is a good number to start with and then we can tune this parameter later.

1. What is the right way to do the bootstrapping event?

5.a. Current thought: rely on basic randomness via RANDAO. Or alternatively, coordinate a "shuffling ceremony" alongside what other trusted setups may be required to kickstart the network.

Disclaimer

This is a first draft with outstanding questions. I hope to put it out as early as possible within the RFP process to solicit general feedback and provide an alternative example to Joe's proposal. Please let me know if you have questions, see issues, etc! They likely exist.