Decentralized Sequencing Proposal: Irish Coffee

Proposed by: Espresso Systems team

This proposal is an adaptation of the Whisk-y proposal. It replaces the Single Secret Leader Election (SSLE) with a VRF, making it significantly more lightweight and efficient. The protocol is inspired by Chia/PoSAT but we are replacing the need for a VDF with Ethereum itself. That is, Ethereum serves as a timing and a beacon service. Future versions of Ethereum's beacon might actually rely on a VDF themselves. The core idea is that each user will evaluate a VRF based on their pre-registered key on a public beacon value (the Ethereum block header/beacon for example), and the user with the lowest output wins the ability to propose the next block. This has very similar properties to proof of work based blockchains like Bitcoin where multiple miners could find a block in the same time window but the probability of this happening is small. Unlike in PoW, the VRFs used in Irish Coffee can be evaluated efficiently, and there is almost no wasted work.

Prerequisites

Our proposal "Irish Coffee" requires a verifiable random function (VRF) and a random beacon:

- Verifiable random function (VRF): Evaluated on a public value this gives a private random value that is deterministically linked to a party's public key
- Random beacon: We assume that Ethereum provides a beacon that is accessible from the smart contract. The beacon is hard to predict and hard to bias. This can be the RANDAO beacon, a future VDF-based beacon, or HotShot's beacon.

Protocol

The protocol works as follows:

1. An Ethereum smart contract maintains a set of accounts eligible to participate as a sequencer. This may be a list of token holders who have staked more than a minimum threshold of tokens, similar to Whisk-y.
2. We denote k as the ratio between the mean time window for a sequencer (e.g., 10 mins) and the mean Ethereum block time (e.g., 12 secs).
3. Each party evaluates VRF on the random beacon, such as Randao or the HotShot Beacon.
4. For i=1 to \alpha \cdot k

(for a parameter alpha. In practice ~10\cdot k

)

1. Evaluate VRF(\textsf{beacon},i)

and check whether VRF(\textsf{beacon},i) \bmod (n\cdot k) = 0

.

1. If yes then you can publish a block at delta=current height + i.
2. If you have a winning VRF, i.e. a value for delta, then start producing a block
3. Collect transactions and generate a proof
4. Publish the block when the block number is equal to the VRF determined value
5. Smart contract verifies that the sequencer is eligible to participate, the validity of the VRF evaluation, and that the current block height >= VRF block height
6. There is a race condition if multiple parties are eligible to propose a block at the same index
7. If a valid Aztec block is published then stop participating in this instance and use the block's beacon to seed the next round.

VRF Specification

The Protocol only uses a single cryptographic primitive, a verifiable random function. A VRF can be constructed from the

BLS signature, or alternatively from discrete logarithms without pairing (see Section 2.4 https://eprint.iacr.org/2023/223.pdf . Both VRFs require mapping an element to a curve point. While it may not be prohibitive to implement, it is currently not supported as a pre-compile in Ethereum. An alternative way to get a VRF is through the SNARK itself. The protocol works as follows:

1. Prerequesites: A secure cryptographic, snark-friendly hash function H. Example: Rescue. A SNARK
2. Keygen()-> Generate a random value x

. Output y=H(x)

as the public key and x

as the private key.

1. VRF(x,\textsf{data})

: \sigma=H(x,data)

- a SNARK that proves that given \sigma,y,\textsf{data}

\sigma=H(x,\textsf{data}) \wedge y=H(x)

This can be implemented with a few 100 constraints inside a SNARK as it only involves a simple SNARK friendly hash.

Protocol Analysis

For a detailed analysis see the PoSat paper or also Ouroboros Praos

- The waiting time levels the playing field. Even weaker provers/sequencers can win as long as they have a low VRF value. This means that they need to wait for a shorter time
- Blocks will be published in irregular periods but with an average block time of 10 minutes.
- The distribution of blocks is similar to PoW protocols such as Bitcoin or such as Gasper.
- Two sequencers might have value 0 when computing VRF(\textsf{beacon},i) \bmod (n\cdot k) = 0

where i is the smallest index after the beacon. This leads to a race condition. The Ethereum consensus (i.e., which transaction is accepted first) decides which block is selected

- If the average block time is sufficiently long, e.g., 10 minutes, then close block races are unlikely (but not impossible).
- A few nodes assemble transactions and build a block but it will be a small constant fraction (careful analysis is necessary).
- The next leader is unpredictable even by the current leader (unless they can predict the beacon). It is therefore important to use a secure beacon!
- The protocol is simple and does not require interaction or additional data. The only cryptography is VRF.

(Experimental) Analysis

1. For k=50, the probability of that there are two sequencers with the same lowest VRF (race condition) is ~1.5%. That is the probability that you have the right to publish an Aztec block and someone else has a right to publish a block in the same Ethereum block is 1.5%.
2. For k=50, the probability that the second lowest sequencer value is within 10 blocks of the lowest sequencer value is ~18%. Or in other words the probability that when you find a block some other prover can publish an Aztec block within 10 ethereum blocks is ~18%.
3. The probability that no VRF value is less than 100 is 13.5%
4. The probability that no VRF value is less than 200 is 1.8%
5. The probability that no VRF value is less than 500 is 0.004%
6. The expected number of VRFs less than ck is c, i.e. the expected number of proposals less than 100 is 2, the expected number of proposals less than 200 is 4

Extensions

Privacy

It is possible to keep the sequencers'/provers' identities private through the use of zero-knowledge proofs. Instead of revealing the VRF the sequencer proves knowledge of an eligible VRF value.

Ensuring minimal parallel work

We can ensure that only about 10 provers attempt to produce blocks by reseeding the VRF if 10k=500 blocks have passed without a published proof. The idea is that between 0 and \alpha \cdot k

only \alpha

provers are expected to have a winning VRF ticket. The probability that no prover wins on the other hand, decays exponentially as e^{-\alpha}

.

Concretely for \alpha=10

, the probability of no prover winning is 0.004% or 1 in 22000 epochs. At this point we would re-seed the beacon. \alpha=10

means that no VRF over 500 will ever win and sequencers/provers only need to run the loop in step 4 of the protocol to 500.

Weighted Selection

The current proposal assumes that all provers/sequencers are equally weighted. This can be easily changed by assigning weights to the participants. Each prover would then check in step 4.a whether VRF(\textsf{beacon},i)=0 \bmod k\cdot \sum_{i=1}^n w_i /w_u

, where \sum_{i=1}^n w_i

is the total weight and w_u

is the users weight. Similarly, we can combine the proposal with Cookie Jar by weighting the provers based on bids.

Comparison with Whisk-y

The Whisk-y proposal is based on SSLE in order to elect a single sequencer/prover for a ten minute period. The high-level protocol can be described as follows:

1. Subselect a small set of sequencers using a random beacon
2. Shuffle these leaders continuously using Single Secret Leader Election (Whisk) such that each 10 minute period there exists a single leader
3. The leader collects a set of transactions, orders them and produces a rollup proof.
4. The leader publishes the transaction data and the proof on chain, revealing that they are a leader (or giving a zk-proof of leadership).

This has several advantages and disadvantages:

Advantages:

- There exists a single sequencer/prover for a period. This allows even smaller prover/sequencers to compete regardless of hardware requirements because they don't need to race/compete with other provers. This is important for competition and prover decentralization.
- The secret component of SSLE enables provers to hide their identity.
- The prover/sequencer window is a fixed 10 minutes, regardless of the prover's identity or hardware.

Disadvantages:

- The protocol is expensive, in particular the shuffling protocol requires data linear in the number of sub-selected provers, every 10 minutes.

- It leaks some information, such as which subset of the provers can get elected
- It's biasable: both using RANDAO as a beacon and using SSLE can enable parties to bias the outcome. Biasing the outcome can lead to one party getting elected more frequently.

Compared to Whisky, Irish Coffee still has the advantage of having a single prover over a long period (10 mins in expectation) and we ensure privacy of the provers until they are elected. In terms of efficiency, our protocol incurs little computation and communication—each prover is expected to compute one VRF for each Ethereum block and share the output only if it is a successful prover in a 10 minute window. Moreover, all provers can participate—downsizing the number of provers is not necessary as in the Whisk-y approach.

Requirements

Decentralization

Sequencer selection must be sufficiently Sybil resistant

Sequencer selection should not prioritize the best hardware or largest actors

Hardware requirements for sequencers must be similar to those of Ethereum validators

Liveness

Network participants must know in advance who the sequencer is for a given time slot

Achieving both this property and privacy at the same time is difficult as they contradict each other. Each leader has the ability to reveal themselves but that would give up on privacy. They can achieve privacy through zero-knowledge proofs.

A rollup should be created in every given slot to reduce network latency even in periods of low transaction activity

Censorship Resistance

Ensure the sequencer selection process is censorship resistant

Ensure transaction inclusion from a particular sequencer is censorship resistant

Privacy

Should allow sequencers the option of anonymity during selection and block submission