TLDR

: We build upon the idea of using proofs of sequential work (PoSW) for a clocked proposer sampling mechanism. By making a number of adjustments we construct a clocked random beacon that can be used as common infrastructure across all shards.

Construction

We use notaries as the global set of participants for the random beacon (as opposed to per-shard proposers). The random beacon is used recursively to select ranked notaries at every 5-second period. (Bootstrapping of the recursion is done with nothing-up-my-sleeve randomness near the genesis, specifically to fill the “seeder lookahead” defined below.)

At a given period we label the ranked notaries N_i

for i = 0, 1, ...

. Enforcement of ranking is done via PoSW to give better-ranked notaries an artificial latency edge over lower-ranked notaries. Specifically notary N_i

must produce a PoSW with difficulty factor D * i

where D

is adjusted to target 5-second periods. (Notice the first notary N_0

does not need to produce this PoSW, although there is another PoSW defined below for anti-grinding.)

To participate in the random beacon a notary must reveal a previously-committed hash preimage. (Commitment can be done for example with a hash onion H(H(H(...)))

.) This reveal process is a way to “reseed” the random beacon. (This is especially important because an ASICed attacker may produce PoSW faster than others.) Specifically the RNG could be the XOR of the fastest-revealed preimage at every period.

The above construction however has a grinding attack. Specifically a top-ranked notary may decide to not participate by not broadcasting the preimage after knowing before everyone else the next RNG output. To address this grinding attack we apply additional sequential work on the fastest-revealed preimage before XORing.

Specifically we apply 5*l

-seconds worth of additional sequential work, where l

is the seeder lookahead. That is, we use the RNG output that is l

periods old as the seed to select the ranked-notaries for the next period. This lookahead allows for the top-ranked notaries to start working on these l

-period-old preimages and provide a corresponding timely PoSW at the time of participation.

Setting l=1

or l=2

is possibly good enough in practice. (Formal modelling “à la Cardano” of the beacon and its attacks to choose good parameters can be done after the intuitive idea has been confirmed.) Notice the seeder lookahead does not have to equal the proposer lookahead.

Discussion

To abstract the clocked proposer sampling mechanism into a random beacon we’ve made the following changes:

1. Use notaries instead of proposers.
2. Use a recursive strategy instead of bootstrapping from blockhashes. This simplifies the need for synchronisation between period lengths and block times, allowing for shard-native periods “unchained” from the main chain.
3. Mitigate a grinding attack by applying a second round of sequential work.

The random beacon can be used as common infrastructure for all shards, and can be used for things other than proposer sampling. Note also that the proofs of work (see page 15) are ~200KB large so reuse across shards is a significant efficiency boost.

To harden the beacon we can also do timestamp comparisons in the context of an assumed global clock and honest majority. We can also apply penalties to notaries that do not participate.