TLDR

: We present a simple secret single-leader election with a per-block overhead of one SNARK plus 32 bytes. (Related posts here, here, and here.) Thanks to Kobi Gurkan for helpful discussions.

Construction

Let g

be an elliptic curve generator. Let v_1, ..., v_k

be a list of validators to secretly shuffle for block proposals. Every validator v_i

has a permanent public key pk_i

as part of their validator record where pk_i = g^{sk_i}

for some secret key sk_i

. Let n

be a nonce, e.g. an election period counter.

To participate in the election a validator v

broadcasts (e.g. over Tor) an ephemeral public key epk

and a corresponding SNARK proof p

with:

- public inputs

: pk_1, ..., pk_k, n, epk

- private inputs

: sk, i

- statement

: pk_i = g^{sk}

and epk = g^{H(sk, n)}

After sufficient time for the ephemeral public keys and proofs to be included onchain, the received ephemeral public keys are shuffled using onchain randomness into an ordered list. A validator v

then signs the block at position j

using the ephemeral secret key esk = H(sk, n)

corresponding to the ephemeral public key at position j

in the ordered list.

Onchain overhead

For concreteness we assume Groth16, the BLS12-381 pairing-friendly curve, and we let g

be a Jubjub generator. The onchain overhead is as follows (where 32 bytes is the size of a compressed Jubjub point):

- state

: 32 bytes per validator for the permanent public keys

- computation

: ~3ms per block for the SNARK verification

- data

: 32 bytes per block for the ephemeral public keys plus 127 bytes for the SNARK data

Offchain overhead

Notice the circuit complexity is dominated by the single hash computation H(sk, n)

. If H

is a SNARK-friendly hash function (e.g. MiMC) we expect the proving time to be ~0.1s on a commodity laptop. If H

is SHA256 the proving is ~1s. In the context of Eth2 shard persistent committees, this proving time is marginal relative to the preparation time validators have prior to elections.

To benefit from a universal trusted setup one can use Sonic instead of Groth16. The main drawback is increased prover time by ~30x. (There are rumours of a “Sonic 2.0” which may bring improvement to prover time. And in general the SNARK stack —proof systems, circuits, prover algorithms, hardware—is rapidly improving.)

Finally, there is overhead from using a private network such as Tor to disseminate the ephemeral public keys and proofs. The concrete overhead is small because the corresponding messages are tiny.