In this post we present an optimised version of the proposal commitment scheme from this post. When combined with this challenge scheme it addresses the proposer withholding attack.

Construction

During his assigned period p

the eligible validator gathers all the proposals p_1, ..., p_n

he has received from proposers. The proposals p_i

are hashed and the hashes H(p_i)

are Merklelised in a Merkle tree with root r

where the leaves H(p_i)

are ordered lexicographically. The validator signs [p, r]

with a signature s

to form a commitment C = [p, r, s]

. The commitment C

is then broadcasted alongside the leaves H(p_i)

to the proposers.

A commitment C

is valid if:

1. The period p

is valid

1. The signature s

is valid

We now introduce two slashing rules to the VMC as follows:

1. Improper ordering

: If a valid commitment C

alongside two leaves H(p_j), H(p_k)

matching r

are presented to the VMC for which the lexicographic ordering is not respected then the validator is slashed.

1. Non-membership

: If a valid commitment C

alongside two leaves H(p_j), H(p_k)

matching r

are presented to the VMC for which H(p_j), H(p_k)

are immediately adjacent in the Merkle tree and H(p_j) < H(\widetilde{p}) < H(p_k)

where \widetilde{p}

is the proposal submitted to the VMC then the validator is slashed. The “left” and “right” edge cases also need to be handled.

Discussion

The scheme allows proposers to safely disclose their collation body to the validator once they have received a valid commitment C

alongside the leaves H(p_i)

which match r

.

This scheme has been optimised compared to the original scheme because, in the default case, there is no overhead to the VMC. In particular, there is no need for an extra field or extra data in the collation headers. Notice also that the scheme imposes no storage overhead in the VMC.