TLDR

: We present a 1-bit custody bond scheme which is friendly to BLS aggregation.

Construction

Let V

be a 32-ETH collateralised validator that has published H(s)

onchain where s

is a 32-byte secret. Given a piece of data D

the validator V

can compute the corresponding “custody bit” b

as follows:

1. Partition D

into 32-byte chunks

1. XOR every 32-byte chunk with s

and H(D)

1. Merkleise the XORed chunks to get a root r
2. Let b

be the least significant bit of r

The signed message [H(D), b]

is a non-outsourceable 1-bit custody bond assuming the following constraints (enforced with slashing conditions):

1. Secrecy

: The secret s

must be kept secret for some time, e.g. at least 15 days after publishing the proof of custody.

1. Expiry

: The secret s

must be revealed eventually, e.g. within 30 days after publishing the proof of custody.

1. Consistency

: The custody bit b

must derive correctly from s

and D

(enforced with a TrueBit-like challenge game).

Notice that a 1-bit custody bond has 16 ETH of cryptoeconomic security. Indeed, choosing b

better than at random requires custody of D

(and of s

, for non-outsourceability).

We now add custody bonds for BLS aggregated votes. Let V_1, ..., V_{1024}

be a committee of 1024 validators voting on H(D)

. Each validator is invited to make a “bonded vote” [H(D), b] which can either be a “0-vote” where b = 0

or a “1-vote” where b = 1

. The 0-votes and 1-votes are aggregated into a single 96-byte signature, and the 1024-bit string is replaced with a 1024-trit string to account for the three possibilities per validator (no vote, 0-vote or 1-vote).

Discussion

The added costs of custody bonds in aggregated votes are marginal:

- Every validator consumes 1 trit (~1.58 bit) instead of 1 bit, adding 75 bytes of overhead per aggregated committee vote.
- Signature verification requires 3 pairings instead of 2, adding <2.7ms of verification time per aggregated committee vote.

Notice also that we can augment the cryptoeconomic security of custody bonds close to 32 ETH. The reason is that the same piece of data D

can be included in several custody bonds for added security. (Having overlapping custody bonds works especially well for shard attestations when the committee of attesters is infrequently shuffled.)

For example, two 1-bit custody bonds [H(D_1), b_1]

and [H(D_2), b_2]

where D

is contained in both D_1

and D_2

yields a 2-bit custody bond for D

with security 24 ETH. (The reason we XOR chunks with H(D)

in the construction above is to avoid validators caching the Merkleisation of D

across custody bonds thanks to the unpredictability of H(D)

. We could also use beacon outputs instead of H(D)

.)