Motivation

Data availability is not objectively attributable as everyone has to sample independently to be sure if data is available or to download the whole block.

The motivation for this short post is to see how strong you can make an oracle that attests to data availability that lives in the state machine of another blockchain. In particular, we want to see how we can provide the best guarantee for relaying Celestia's availability.

Previous Work

One proposal by @musalbas was Towards on-chain non-interactive data availability proofs. The main problem the protocol was trying to solve was the selective disclosure attack and make the smart contract receive data chunks through a randomly generated onion circuit. This required on-chain private randomness, which is non-viable as of this writing.

A follow-up proposal by @adlerjohn was On-Chain Non-Interactive Data Availability Proofs, which, in essence, has the same conclusion as this post.

Should the majority of main chain block producers attack the main chain to cause a safety violation, social governance can be used to mitigate this.

This post explores what happens if you base the entire construction on deriving Celestia's social consensus. Suppose a chain (receiving chain) receives an attestation on the availability of Celestia. What happens if we can assume that the validators of the receiving chain will be socially slashed when relaying the wrong attestation?

Construction

When you naively relay Celestia's data root, you have a ⅔

honest majority assumption on its availability. This is because on-chain light clients cannot sample (since they're on-chain) and don´t download the whole block (since they're light clients). We can at least verify the correctness of consensus, which in this case is CometBFT. Celestia validators will be socially slashed in case of a data-withholding attack, but this does not prevent the attack itself. The question is now if you can punish the validators of the receiving chain that accept a data root that is not available.

Let's take the simplest approach: We want to relay the availability to a second CometBFT chain. Each validator of the receiving chain votes whether data is available for a given data root; if more than ⅔

of validators think that the data is available, we deem it available. Let's look at all the possible cases.

If it is available but the validator votes 'not available,' then the attack will only delay the interaction with the DataRoot, resulting in a liveness failure but not the theft of funds. Votes are additive, as availability is a unidirectional property. Something that is deemed available now is not expected to be unavailable later. Therefore, you can always vote available later and unavailable only after a certain timeout during which you failed to sample.

If validators are voting available even though it is not, then the most damage can be done here. How can we disincentivize the validators lying about it? This can be punished through social slashing derived from the social slashing of Celestia validators.

After a data withholding attack, Celestia's validators will be slashed. After the slashing of Celestia validators, the validators of the receiving chain will be socially slashed when dishonest. This way, the validators are incentivized to be honest about the availability. The assumption here is that Celestia's social consensus can influence the social consensus of the receiving chain, so this works for native, and Celestia aligned integrations much better than for something like Ethereum. Ethereum validators would have to fork on a data withholding attack, and all full nodes would have to embed Celestia light nodes in the first place. The nodes of the receiving chain are required to run light nodes to enforce Celestia`s social consensus.

The selective disclosure attack can be a severe problem, as validator IP addresses are not that hard to find. One mitigation could be to run multiple light nodes not connected to the validator's IP, increasing the number of samples in the network as a bonus. You can even run a Celestia full node or, in the future, Celestia upgrades using some form of mixnet to prevent the selective disclosure attack by design.

Conclusion:

If a separate chain receives the DataRoot, then derived social consensus can increase the guarantee of correct availability by potentially slashing the dishonest validators of the receiving chain. Validators are now incentivized to be honest about voting on availability because users of the receiving chain run light nodes of Celestia.