Background: Cryptoeconomic signature aggregation

A cryptoeconomic aggregate signature (CAS) allows a node to make a claim of the form “either this sample of validators all signed off on what I am building, or I am burning my deposit”. This could be used as an ingredient to make forking in chains (scalable or otherwise) less likely and more expensive.

Here is how this would work. Suppose that you have a chain-based protocol, but creating a block requires adding a CAS from a sample of 200 validators, which sign off on their belief that the block is building on top of the current head at the time. If a CAS is later found to be fraudulent, then the block is still valid, but the creator gets penalized. If we trust that the majority of validators will refuse to sign blocks that do not build on what they think is the current head, then building a chain that reverts the current main chain would not be impossible, but it would

cost the validators an entire deposit for each block that they attempt to revert.

The scheme can also be extended as follows:

- Validators can be constrained with Casper FFG-style slashing conditions (NO_DOUBLE_VOTE and NO_SURROUND) preventing them from contradicting their previous votes.
- If signatures are included in a CAS in block N, then randomness revealed in block N+1 could select a few signatures from block N for which the proposer of block N+1 could reveal the Merkle paths, rewarding the signers.

If we can incentivize signers to publish their signatures most of the time, then we could have a notion of client-side finality, where clients could passively watch for signatures (which would also be required to be votes in a global Casper FFG cycle) and accept some block as justified if 2/3 of a global validator set votes for it (and finalized if it and its direct child are justified), though the chain would not have consensus on whether or not finality happened at any specific point in time.