Prerequisite: https://github.com/ethereum/research/wiki/A-note-on-data-availability-and-erasure-coding

Similar to how Schnorr NIZK proofs work, instead of the light client providing a set of random coordinates for chunks to the node sending the the block, the node sending the the block could generate the set of coordinates seeded from a precomputed challenge specific to each light client, e.g. H(blockHeader || clientID)

.

What's the security implication of letting miners possibly only release blocks that cause light clients to sample chunks that the miner wants? If a miner wanted to only fool a small set of light clients (the "only releasing individual bits of data as clients query for them" attack), then the miner might want to create a block such that those light clients don't sample chunks that contain provably bad transactions, if your fraud proof generation mechanism is fine-grained enough to generate fraud proofs from incomplete blocks. However, this is already very unlikely anyway with an interactive data availability proof mechanism, since a bad transaction can be hidden in 1 of the 4096 chunks for a 1MB block.