I've written an experimental Go library, rsmt2d, that implements a two dimensional Reed-Solomon merkle tree data availability scheme. It has a reparation algorithm with the ability to repair squares in a byzantine setting, detecting byzantine rows and columns and inconsistencies between rows and columns, and allows for the generation of fraud proofs.

Feel free to use for experimentation or pick it apart. I plan to experiment with this as a data availability proof layer for fraud proof-supporting blockchain data structures.

I did discover some intricacies around designing a square repair algorithm: for example, it is important to verify that the original data of each row/column matches the extended data, even if the whole row/column is available, because otherwise other clients might reject the block if they receive different pieces of a row/column than you, causing a fork.

GitHub

musalbas/rsmt2d

Go implementation of two dimensional Reed-Solomon merkle tree data availability scheme - musalbas/rsmt2d