Hi,

I want to share a summary of a work I have done regarding a paper aboutFraud Proofs written by Mustafa, Alberto, and Vitalik.

The original idea was to verify the results presented in Table 1 by simulating the network instead of using mathematical closed-formulas (which is what they do in the paper). The simulation also explores other characteristics of the network.

I wrote an article to explain the logic and results of the simulation, future improvements, and some implementation performance notes while profiling the code.

The article tries to be self-explanatory, so reading the paper is not crucial (but recommended).

In particular, here I want to mention a particular part of the article. The comparison that the simulation does between the Standard and Enhanced model.

It is important to clarify what each model means in the implementation of the simulation:

- Standard Model: Each light-node decides individually for a unique random set of shares to pull. Each light-node ask for all their shares before letting the next do the same.
- Enhanced model: Each light-node decides individually for a unique random set of shares to pull. Each share pulling from the full node is done in rounds. In each round, a light-node is randomly elected to let him pull the next share.

The simulator runs a parameterizable number of iterations. Each simulation iteration is stopped when the full-node decides to reject replying to a share request. The full-node decides to reject a request if it shared all shares up to (k+1)^2 shares (worst case scenario for full-nodes).

I leave a plot that the simulation generates which show how the Standard and Enhanced model compares regarding the number of light-nodes that finishes pulling their ‘s’ shares from the full-node. Ideally, we want this number to be zero since this would mean that soundness is violated for some light-nodes.

[

image

653×624 28.6 KB

](https://ethresear.ch/uploads/default/original/2X/8/859f5564845076058ffe48e36be6e3619804047f.png)

If you enter the article you can see the same graph for (k=64 s=10) and (k=64, s=2).

The three configurations plot shape is similar, but we can appreciate how changing the value of ‘s’ impacts soundness. As we may expect, a bigger ‘s’ makes the total number of ‘finished’ light-nodes lower than a smaller ‘s’.

The code of the simulator is publicly available on GitHub.

Besides this comparison between model, the simulator also provides a way to estimate the number or light-nodes for a triplet (k, s, p) which, discussed with Alberto and Mustafa, is several of orders of magnitude faster than using Theorem 4. To be clear, estimation compared to exact calculation seems to be a good tradeoff here regarding time and precision.

Also, the article mention possible improvements and features that might be interesting to add to the simulator.

As I mentioned before, if you are curious and want to read more, the article has more details of this mini-project.

Thanks,

Ignacio

PD: Sorry for not facilitating more embedded images and links. Since I’m a new user, it seems I can’t post more than one/two links and embed more than one image.