Boston University **Spring 2023**

ECE Department

Computer Communication Networks (EC 541)

Final Project: Part B

1. Project Goals

The goal of the project is to reproduce results from the paper (posted on Blackboard):

> Eyal, Ittay, and Emin Gün Sirer "Majority is not Enough: Bitcoin Mining is Vulnerable," Financial Cryptography and Data Security, 2014.

Specifically, the project will reproduce the results of Fig. 2 that displays analytical and simulation results of selfish mining, as well as a comparison with honest mining. In addition, your project will compute and display confidence intervals for the simulation results.

2. Submission Guidelines

Files to submit on Blackboard:

- `Simulator_Selfish_Mining.py` (skeleton is available on Blackboard).
- A pdf file containing screen shots of the output of your programs for the cases $\gamma = 0$ and $\gamma = 1$ (see below an example for the case $\gamma = 0.5$).

50% of the grading will be given on correct execution of the Python file, 25% on the comments, and 25% on the pdf file with screenshots of the output. The place holder comments given in the program skeleton are not sufficient.

3. Project Details

The project goals are to perform a simulation of the DTMC describing selfish mining (see Fig. 1 in paper). In each iteration, for each given values of α and γ, the simulation will compute the total revenue of the selfish mining pool and the total revenue of the other (honest) nodes. We remind that for each type of miners, revenue is accumulated under certain state transitions, as shown in the lecture (see also Section 4.2 of the paper).

Once an iteration completes, the Relative pool revenue (R_{pool}) should be estimated as the total revenue of the selfish mining pool to the total revenue of all miners. If $R_{pool} > \alpha$, this indicates

that selfish mining is preferable over honest mining. Note that an analytical formula for R_{pool} is provided by Eq. (8) in the paper.

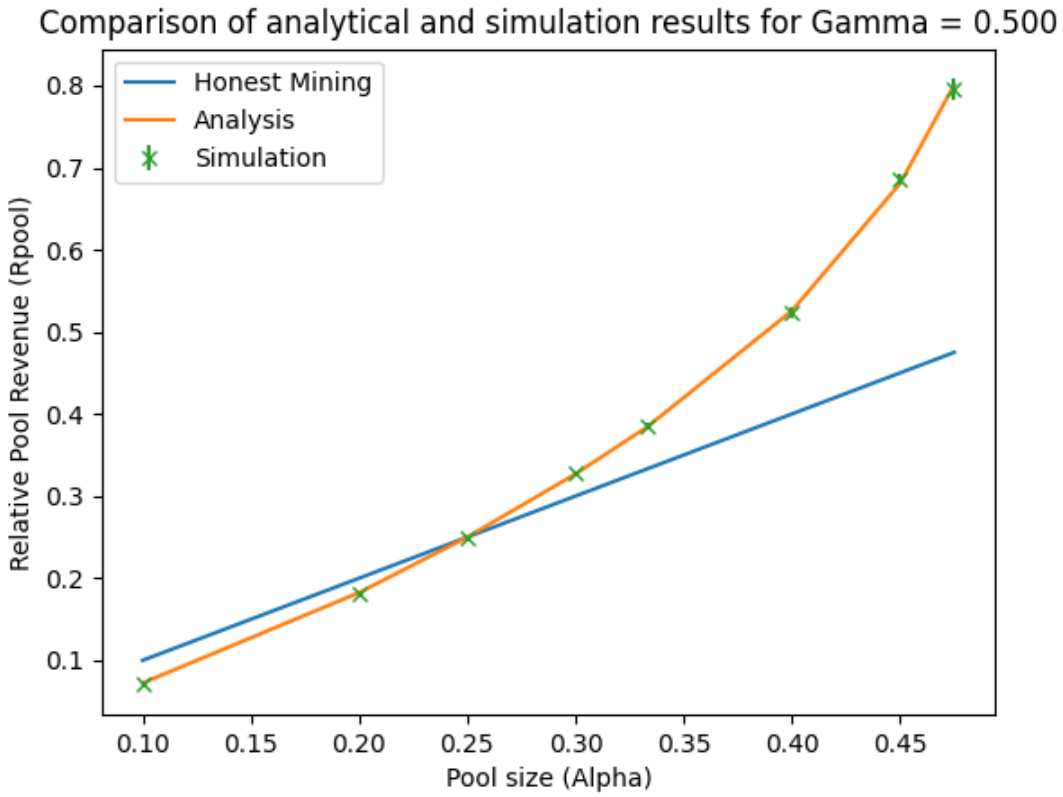
To help you with the project, a skeleton of the program `Simulator_Selfish_Mining.py` is provided to you on Blackboard. Do not import other Python packages than those included in the header.

An example of the output of the simulator for the case $\gamma = 0.5$ is shown below. Note that in this case, selfish mining becomes preferable over honest mining once $\alpha > 0.250$.

Statistical results for Gamma = 0.500
At Alpha 0.100:
Analytical Relative pool revenue is 0.072
Sample Mean Relative pool revenue is 0.073 with error 0.001.
At Alpha 0.200:
Analytical Relative pool revenue is 0.182
Sample Mean Relative pool revenue is 0.182 with error 0.002.
At Alpha 0.250:
Analytical Relative pool revenue is 0.250
Sample Mean Relative pool revenue is 0.250 with error 0.003.
At Alpha 0.300:
Analytical Relative pool revenue is 0.327
Sample Mean Relative pool revenue is 0.327 with error 0.003.
At Alpha 0.333:
Analytical Relative pool revenue is 0.385
Sample Mean Relative pool revenue is 0.385 with error 0.003.
At Alpha 0.400:
Analytical Relative pool revenue is 0.526
Sample Mean Relative pool revenue is 0.523 with error 0.004.
At Alpha 0.450:
Analytical Relative pool revenue is 0.681
Sample Mean Relative pool revenue is 0.679 with error 0.008.
At Alpha 0.475:

Analytical Relative pool revenue is 0.801

Sample Mean Relative pool revenue is 0.804 with error 0.012.



PDF submission: Your PDF submission should provide similar outputs as above, but for the cases $\gamma = 0$ and $\gamma = 1$.

4. Project Tips

Start with reproducing the analytical results (Eq. (8) in the paper).

For the simulation loop, at each iteration:

1. Use an integer variable `curr_state` to keep track of the current state of the DTMC.
2. Use a Boolean variable `flag` to distinguish between states 0 and 0'.

3. Use the variables `r_pool` and `r_other` to respectively maintain running sums of the revenue of the selfish pool and revenue of the other nodes.
4. At each step, toss a coin to decide on whether the selfish pool or the other nodes successfully mined a block.
5. When transitioning from state 0’ to state 0, if other nodes mined the block, you need to toss another coin to decide on whether that block was mined on top of a selfish block or on top of an honest block.
6. Note that neither selfish miners nor other miners collect revenue when transitioning from state 0 to state 1.
7. Note that other miners collect no revenue once the DTMC transitions to state 2 until the DTMC gets back to state 0.