Programming Assignment in IN5420/IN9420

We have developed a simulator to simulate the use of selfish mining strategies in Bitcoin. The simulator implements the selfish mining strategy described in [1] along with the basic building blocks of Bitcoin such as consensus and communication. For a similar concept, see reference [2].

The assignment:

1. Understand the code:
    1. consensus protocol implementation in `consensus/customized/consensus.go`
    2. pow implementation in `consensus/customized/pow.go`
    3. network dissemination in `communication/customized/network.go`
2. Experiment with different parameters in the configuration file and observe the behavior of selfish mining.
    1. Parameters of interest are *number of nodes* (2 to 1000), *mining power of the selfish node*, *mining powers of the honest nodes*, *block interarrival times*, and *propagation delay*.
    2. While this is a less influential parameter, you can also vary the *number of blocks* (500 to 10000).
    3. For a large number of nodes, it is advisable to keep the logging level at `WARN` or `ERROR`. However, for 2-3 nodes and a smallish number of blocks, one can set the logging level to `INFO` or even `DEBUG`, in order to understand what is going on.
    4. Output metrics - number of blocks produced vs expected.
3. Present the results in graphical plots.
4. Identify when the selfish mining strategy is useful for a miner based on the analysis.
5. Understand the differences between the implementation of selfish mining in the paper and `consensus/customized/consensus.go`
6. Optional: change the strategy parameters, or even make a small adjustment in the code, run the simulator and analyze the results.

The deliverables:

A document (pdf) including the following:

1. An overview of the differences between the code and the paper [1].
2. Visual plots presenting the experimental results.
3. Analysis of the results and the resulting conclusions regarding selfish mining.

Submission Deadline: **April 15, 2025 before 24:00**

How to submit: email to the teacher (email address: romanvi@ifi.uio.no)

How to get the simulator?

The TA will send it to your email address. Please do not distribute it publicly.

How to run the simulator?

Run the `main()` method

Usage:

```
go run main.go [configFile]
```

Args:

configFile: The path to the configuration file which contains the configuration for the simulation. If not present, a default configuration file "`config.toml`" is used.

The configuration parameters are documented in `config.toml`.

Running the above code will generate an `output` directory containing subdirectories named "`blocks`" and "`measurements`." Additionally, the directory will include the files `selfish_index.sql` and `results.html`.

Note: the `results.html` file serves as a basic summary of all simulations, extracted from the simulations index within the `selfish_index.sql` file. It includes a table displaying the parameters and a link to access the raw results in JSON format.

References:

[1] Eyal, I., & Sirer, E. G. (2018). Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, *61*(7), 95-102.

[2] Nayak, K., Kumar, S., Miller, A., & Shi, E. (2016, March). Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. *In 2016 IEEE European Symposium on Security and Privacy (EuroS&P) (pp. 305-320)*. IEEE.