LELEC2770 – Practical Sessions

Session 5: Cryptocurrencies

Theoretical question

1. In Okomato-based e-Cash, show how is it possible to recover the secret key if double spending happens.
2. Explain with a small drawing how double spending can occur in bitcoin and how you would protect from it as a merchant.
3. Explain what would happen if *md5* was used instead of *sha256*.

   The following research paper might help you. `https://eprint.iacr.org/2016/167.pdf`

Practical question

1. A new semi-decentralized blockchain has been created at the UCL. You are going to compete against each other in the mining process of the blocks to earn as much money as you can. The objective of the blockchain is to have a consensus over a dancing stickman:

```
 o  \ o / _ o        __|   \ /     |__       o _ \ o /  o
/|\   |    /\  ___\o   \o   |    o/   o/__   /\    |   /|\
/ \  / \   | \ /)  |   ( \ /o\  / )   |  (\  / |  / \  / \
```

   Where one of the move must be part of the block. The blockchain is then the dance chosen by the miners.

   A block should be composed of the following element:

   (a) parent_id: id of parent block

(b) hash_header: hash of parent block

(c) miner_name: your _unique_ miner name

(d) nonce: random value such that hash_header is valid

(e) dancemove: A chosen move by the miner (between 0 and 10)

The server acts as a centralised database and does some basic verification of the submitted block: you have the responsability to discard unvalid blocks/chains that your fellow camarades could have created. A block is valid only if it holds a correct PoW-hash of the previous block w.r.t the nonce and if the dancemove is a correct move.

a PoW (Proof-of-Work) is necessary to solve a block. The difficulty requested for this blockchain is to set the first 21 bits to 0 and to find the 4 following bits, known as a secret from the centralized database. The secret also depend of the index (within the chain) of the block. An other way to say this: the centralized database will only accept your block if after omitting the first 21 bits, the next 4 bits match a secret value known by the database and that depends of the index of the block. If you are wrong about the value of the 4 secret bits, the server is going to respond "higher" or "smaller" (see miner.py).

Each correct block to the main chain will be awarded some FabulousCoins to its miner. At the end of the game, the centralized database will enforce the main chain with the following policy:

- The chain must be valid
- It must be the first chain that reaches 40 blocks.

Hopefully, since everyone is super-interested in seeing a stickman dancing, miners can exchange their FabulousCoins against real $.

You will find interesting the file miner.py. Good luck !