Siddhesh Gupte – UFID 2975 - 1118

Sagar Prabhu – UFID 8915 - 4951

**What has been implemented:**

1. A transcription can be made by a wallet

The wallet takes the input as a string of “Receiver Amount Tip”. The user also enters which transaction outputs he wants to refer to as the input of this transaction. The wallet will then get the required blocks from the full node using a **Bloom Filter**. The bloom filter is used to make the set membership lookup quick as well as obfuscating the public keys of the wallet. The obtained blocks are then stored in the local cache for the wallet. The inputs are then verified and a transaction is created. To this transaction the change address is added and the overall hash of the transaction is set. This transaction is then signed with the private key and then sent to the full node.

2. A transaction can be added to a full node

The incoming transaction is checked for validity i.e the hash for the transaction is calculated to see whether it matches the given txid. Then the transaction is checked to confirm that it is not a double spend – this is done by checking all uncommitted transactions + all blocks’ transaction inputs to check if input is equal to the inputs given here. The signature of the transaction is then checked and the transaction is gossiped to 8 other nodes.

3. A transaction can be gossiped while avoiding double spends

The transaction is gossiped continuously to 8 neighbours. However, to make sure that double spends do not happen the following conditions are checked.

```
Don't propagate if
    # 1. This transaction already exists in uncommitted
    # 2. Signature isn't valid
    # 3. Transaction isn't valid for it's hash
    # 4. Inputs of the transaction aren't valid
    # 5. Is a double spend wrt already seen transactions - take the input and
check uncommitted + all blockchain transactions inputs
```

4. A block can be mined

The full node can mine a block and then gossip the block to 8 other random neighbours. The proof of work has been kept simple and tries to find a hash with at least one zero at the beginning of the hash. The transactions are removed from uncommitted transactions and a coinbase transaction is added which gives the miner 25 bitcoins + sum of fees of the transactions taken in the block. A **Merkel tree** is made and the merkel root is also added to the block.

5. A block can be gossiped

The full node can then send the block to 8 other random neighbours. At every step the block will be checked and verified.

```
# VERIFY BLOCK
    # 1. Is this a new block?
    # 2. Is the block valid?
    # 3. Are all the transactions in this block valid? (Check double spends
also here?)
    # 4. Builds on the current known longest chain? (to avoid forks)
```

**How to run the tests:**

The tests can be executed by executing the command “mix test” in the terminal.

**Features implemented for the bonus part:**

1. **Bloom filters** are sent from wallets to the full nodes for getting blocks and then these blocks are cached thus, the public keys of the wallet are obfuscated and set membership is done efficiently. SPV is fully functional.
2. **Merkel Trees** are used for checking if a transaction exists in a block and is not put there after the fact by a malicious party. The merkel root is stored in the blocks and is also used for calculating the hash and nonce.
3. **Distributed protocol** (gossip) is working with the transactions as well as blocks being gossiped. The transactions being gossiped are also gossiped in such a way that **double spends are avoided**.