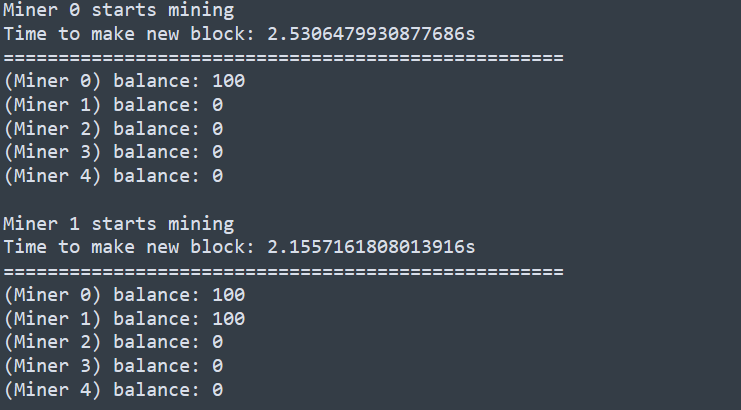
SUTDcoin Project Report

This project implements a simple blockchain with Account/Balance Model.

1. Mining and coin creation

The mining and coin creation are demonstrated by running minertest.py where Miners mine keeps mine new blocks and get 100 coins as a reward for each block. The time taken to mine a new block is between 2 to 5 senconds.



1. Fork resolution

Fork resolution is implemented in def resolve() in Blockchain.py. Each Blockchain object has a block dictionary which is updated whenever a new block is added. This dictionary is later used to find the longest chain in terms of blocks.

1. Transaction resending protection

Due to the randomness of a 32-bit nonce, it is hard to produce two identical transactions.

1. Payments between miners and SPV clients
   1. transaction validation (for miners and SPV clients)

In transaction.py, transactions are validated by checking the types and lengths of amount, comment, nonce and keys. In Miner.py, create\_transaction() and add\_transaction() validate and add the transaction to the pool of transactions before the transactions are put into blocks and broadcasted.

P2P network:

Run Miner.py and then spvclient.py.

Miner.py

SPVclient.py

Miner.py

SPVclient.py

Miner.py

SPVclient.py

I am sorry that due to time constraint and limited man power, some features and the two attacks are not implemented.

1. Differences between Bitcoin and my SUTDcoin

Bitcoin uses the UTXO Model but SUTDcoin uses the Account/Balance Model. Both models achieve the same goal of keeping track of account balances in a consensus system.

The benefits of the UTXO Model are:

* Scalability — Since it is possible to process multiple UTXOs at the same time, it enables parallel transactions and encourages scalability innovation.
* Privacy — Even Bitcoin is not a completely anonymous system, but UTXO provides a higher level of privacy, as long as the users use new addresses for each transaction. If there is a need for enhanced privacy, more complex schemes, such as ring signatures, can be considered.

The benefits of the Account/Balance Model are:

* Simplicity — Ethereum opted for a more intuitive model for the benefit of developers of complex smart contracts, especially those that require state information or involve multiple parties. An example is a smart contract that keeps track of states to perform different tasks based on them. UTXO’s stateless model would force transactions to include state information, and this unnecessarily complicates the design of the contracts.
* Efficiency — In addition to simplicity, the Account/Balance Model is more efficient, as each transaction only needs to validate that the sending account has enough balance to pay for the transaction.

One drawback for the Account/Balance Model is the exposure to double spending attacks. An incrementing nonce can be implemented to counteract this type of attack. In Ethereum, every account has a public viewable nonce and every time a transaction is made, the nonce is increased by one. This can prevent the same transaction being submitted more than once.