Financial applications of blockchains and distributed ledgers

SESSION OF OCTOBER 27

REQUIREMENTS

- .NET Core
- Visual Studio Code
- NBitcoin and QBitNinja.Client packages

OPTIONAL

- Bitcoin Core full node

Remember that in an ideal situation, you would install a Bitcoin full node on your computer/server and use it to interact with the blockchain directly rather than trusting a third-party API. An intermediary approach would be to use a so called SPV wallet – I let the interested reader document himself about it.

REFERENCES

Programming the Blockchain in C# by Nicolas Dorier: https://programmingblockchain.gitbooks.io/programmingblockchain/.

You may use it to help you implement the exercises below.

REPORT

You are expected to hand in your source code as well as a report explaining each of your results, justifying your choices, and documenting your experiments with explicit links to the blockchain. For instance, your report should contain the unique IDs of the transactions that you generate and broadcast from your source code.

Deadline for the report is December 15, 2017.

PROJECT

Mobile wallet on testnet

1. Install CoPay wallet on your phones
2. Create a testnet wallet on CoPay
3. Charge it with testnet bitcoins from an online faucet

Wallet management in C#

1. Generate a private key with cryptosecure random number generator
2. Compute corresponding public key
3. Deduce Bitcoin address for mainnet and testnet
4. Store the private key and addresses in a text file
5. Build the QR code associated to the testnet address
6. Send testnet bitcoins from CoPay to your testnet address

A little bit of vanity and proof-of-work

1. Generate a *vanity address* starting with the first 4 (or 5?) letters of your first name
2. Store the private key and address in a text file
3. Explain how you solved this problem and how it is related to a proof-of-work algorithm
4. How could you scale your algorithm to find the solution faster? Does that apply to mining?

Interact with QBitNinja blockchain indexer in C#

1. Choose a transaction at random on https://blockchain.info
2. Download all information available for this transaction from your code
3. Download all information relative to your CoPay address from your code
4. Print in the console the balance and the list of transactions related to this address
5. Print the list of *unspent transaction outputs*, also known as *UTXO* or *coins*
6. Explain the relationship between the balance and the UTXO set

Build your first bitcoin transaction in C#

1. Explain the structure of the transaction to send bitcoins from your vanity address to your CoPay wallet
2. Build the raw transaction
3. Sign the inputs of the raw transaction
4. Broadcast the signed transaction to the testnet network using QBitNinja
5. Verify that the transaction gets confirmed by the network
6. Implement a script writing in the console the number of confirmations of a transaction as soon as it changes

Investigate the blockchain

1. Implement a script to identify a list of addresses and transactions that credibly belong to Satoshi Nakamoto, the creator of Bitcoin
2. Using social engineering, try to estimate Satoshi's wealth and justify your conclusion
3. Identify Satoshi's hidden message in block 1 and explain how it is encoded

RESEARCH

- Explain in your own words how proof-of-work works, what purpose it serves, its limitations, and possible alternatives or improvements
- Explain in details the main technical differences between Bitcoin and Ethereum, and describe a few concrete applications that Ethereum may achieve and Bitcoin may not
- Explain in your own words the notion of oracle

Finally, imagine an industrial application where Bitcoin, Ethereum or another sort of distributed ledger might bring value, reduce costs, improve efficiency or increase security. Present the case as if you wanted to convince investors of the market opportunity, and motivate your design decisions.