COMM.NET.500 P2P Systems and Blockchain Technologies
Blockchain laboratory:
Remote instructions

Blockchain laboratory:
Remote instructions

1 Introduction

In this laboratory, you will set up an example Python implementation of a generic blockchain and try out some of its functions, such as mining and transactions.

You should follow the instructions given in this paper in order and prepare answers for any questions presented. Compile a report from the answers and submit it via the course Moodle page. Short answers with some key points are enough. More detailed information about the report submission can be found on the course Moodle page.

Whenever the instructions tell you to do something related to the PCs/nodes, it should be done on both PCs/nodes, unless stated otherwise.

If you have any questions or feedback regarding the instructions, please send an e-mail to the teaching assistant.

2 Initial preparations

In order to complete this lab, you need Wireshark, Python (at least 3.6) and some additional Python modules installed. Missing modules can be installed using pip from the command line: `pip install ed25519 Crypto pycryptodome sortedcontainers flask flask-cors`.

Open a terminal, navigate to the folder where the blockchain_client.py script is located and launch it with the command `python3 blockchain_client.py -p 8080`. The web GUI of the **client** is now accessible on port 8080. Open a browser and connect to `localhost:8080`. Create a wallet, and store the public and private keys in a text file.

Open a second terminal and navigate the the folder where blockchain.py is located. Start up the **miner**: `python3 blockchain.py -p 5000`. Open up the web GUI of the miner, which is now accessible on `localhost:5000`.

For the purposes of this exercise, the combination of this client and miner you just set up will be called the *first node*.

Next, you should set up the *second node*: either repeat the procedure with a second PC in your local network, or open up another two terminals and repeat the commands with *different* port numbers, e.g., 8081 and 5001.

3 Joining the blockchain

In this section, your task is to join the existing blockchain and examine the past transactions of the blockchain.

Launch **Wireshark** and start capturing on the *loopback* interface. If you are using two PCs, also capture traffic on the interface used for the local network at the same time. In the `Configure` tab of the miner web GUI, add the IP address and port of the other node, for example: `10.0.0.2:5000` or `localhost:5001`. Remember to repeat this for the other node.

4 Mining

As your wallet does not contain any currency at the moment, your task is to "mine" some currency. Go back to the `Mining` tab in the miner web GUI (top right corner) and click the *mine* button. Observe the terminal outputs, Wireshark output and the source code to find out what happens behind the scenes and then answer the following questions.

1. What mining actually is, i.e. what happens when a block is being mined?
2. What is the function of *nonce*?
3. When a block is considered to be valid and mined?
4. What do the other nodes do when they receive a new mined block from you?

5 Transactions

In this section, your task is to perform various transactions and observe in Wireshark and the source code what happens behind the scenes. Go to the `Make transaction` tab in the client web GUI. Fill in the requested information. Address means public key in this context. Try to make an invalid transaction and a valid transaction, which you then mine to the blockchain. It is enough to do this on one node.

1. How does the blockchain network know that it was really this person that sent out the transaction?
2. How does the blockchain network know whether you have enough currency to perform the transaction?
3. Can you somehow find out how much currency anyone else has?

Go back to the `Mining` tab and refresh the transactions in the blockchain by clicking the small blue button next to the *Transactions on the Blockchain* header.

In Wireshark, find and examine the blockchain data, and answer the following questions.

4. What kind of information is stored in the blockchain? (Hint: http://localhost:5000/chain)

5. How many blocks are now in the blockchain?

6. What transactions were made in the latest block which included transactions?

7. Could you edit a past transaction and then pass off the chain as a newer valid one? Why or why not?

6 Double spending

In this section, you will perform double spending. On one of the nodes, make two transactions which both use all of your available currency. Submit one of them to the other node (**Blockchain Node URL** in the confirmation dialog) and the other one to the node itself.

Check that the transactions are going to be added to the next block in the miner web GUI on each node by clicking the small blue button next to the *Transactions to be added to the next block* header. Mine the blocks on both of your nodes *at the same time* (*mine -> switch to other node -> mine* is okay). Examine the resulting blockchains on your nodes.

1. What happened? What is this situation generally called?

2. How will this situation be resolved? What will happen to the other chain at that point?

3. What should be done to avoid negative consequences from situations like this?