Practice 2: **Blockchain**

PTI - G12

Josep Martínez

Adrià Martínez

Index

1. Environment

For this lab session we will be using ***Visual Studio Code*** and ***PyCharm*** as our IDE, ***Python*** as the practice programming language and ***Ubuntu*** (Linux) as the host OS. For the server, as the guide says, we will use ***Flask*** and ***curl*** for the requests.

2. Blockchain functions

This section is dedicated to detail how we have implemented each function of the blockchain proposed by the practice guide.

2.1 Node listing

We used the internal variable blockchain to obtain the desired list of nodes, which were accessible thanks to the nodes attribute. It was implemented with the following code.

```python
@app.route('/nodes/list', methods=['GET'])
def nodes_list():
  response ={}
  if len(blockchain.nodes) == 0:
    print("The node list is empty")

  else:
    response = {
      'total_nodes': list(blockchain.nodes)
    }
  return jsonify(response), 200
```

With that we can list the different blockchain nodes, validating first if the list of nodes is empty.

2.2 Chain validation

For this endpoint, the procedure is very simple. There is only one call to the pre-implemented function *valid_chain()* in the Blockchain python class.

```python
@app.route('/validate', methods=['GET'])
def validate():
  response = {}
  if blockchain.valid_chain(blockchain.chain):
    response = {
      "message":"Valid blockchain."
    }
```

```
    else:
      response = {
        "message":"Invalid blockchain."
      }
    return jsonify(response), 200
```

As we can see, a message is returned depending on the chain validation.

2.3 Chain manipulation

Manipulating a chain involves modifying a field of any block. Consequently, the hash value of this particular block will differ from the original one. It's important to note that every block contains the hash of its predecessor, so modifying information within any block will invalidate subsequent blocks and thus the entire chain.
In order to achieve this manipulation, altering the proof value of the second block of the chain (where the first block has a hash value of 1), will be enough.

```python
@app.route('/nodes/manipulate', methods=['POST'])
def manipulate():
    response = {}

    # Replacing the proof of the second block by a random value,
    # should invalidate the blockchain.
    if len(blockchain.chain) < 2:
        response = {
        "message":"Not enough blocks stored in the blockchain."
    }
    else:
        blockchain.chain[1]['proof'] = 100
        response = {
            "message":"Blockchain manipulated successfully."
        }
    return jsonify(response), 200
```

As the code shows, if a chain has only the initial block, we cannot manipulate it.

3. Additional tasks

3.1 Alternative to PoW

An alternative to Proof-of-Work is Proof-of-Stake. But let's first talk about PoW.

Proof of Work (PoW) and Proof of Stake (PoS) are the most common consensus mechanisms. They are adopted by major cryptocurrencies to secure their network.

Proof of Work is used in Bitcoin to validate transactions and secure the network. Apart from other things, PoW prevents double-spending. The blockchain is secured by participants called miners, who use computational power to compete for the right to confirm new blocks and update the blockchain. A successful miner will be rewarded in BTC by the network. As of December 2021, a miner can get a block reward of 6.25 BTC plus transaction fees by successfully mining a Bitcoin block.

3.1.1 Proof of Stake

The major difference between PoW and PoS is the way they determine who gets to validate a block of transactions. Proof of Stake is the most popular alternative to Proof of Work. It's a consensus mechanism that aims to improve on some limitations of PoW, such as scalability issues and energy consumption. In PoS, participants are called validators. They don't need to use powerful hardware to compete for the chance to validate a block. Instead, they need to stake (lock) the native cryptocurrency of the blockchain.

	Proof of Work (PoW)	Proof of Stake (PoS)
Who can mine/ validate blocks?	The higher the computational power, the higher the probability of mining a block.	The more coins staked, the more likely you get to validate a new block
How is a block mined/ validated?	Miners compete to solve complex mathematical puzzles using their computational resources.	Typically, the algorithm determines the winner randomly, taking into account the amount of coins staked.
Mining equipment	Professional mining hardware, such as ASIC, CPU, and GPU	Any computers or mobile devices with an internet connection
How are rewards distributed?	The first person to mine the block receives a block reward	Validators can receive a share of the transaction fees collected from the block they validated
How is the network secured	The greater the hash, the more secure the network	Staking locks crypto on the blockchain to secure the network

To be eligible to validate a block, participants need to lock a certain amount of coins in a specific smart contract on the blockchain. This process is known as staking. The PoS protocol will then assign a participant to validate the next block. Depending on the network, this selection can be done randomly or according to their holdings (stake). The selected validator can receive transaction fees from the block they validated as rewards. Typically, the more coins they lock up, the higher the chance to be selected.

3.2 Restrict transactions

In our implementation of a blockchain solution, everybody could write a new transaction. In order to protect the execution of the transaction method, there is an existing solution, called permissioned Blockchain.

A permissioned blockchain is a technology that restricts access to its network and data unless ensuring that the user have permission or credentials to join the network. This controlled access offers greater security, confidentiality, and control, making it suitable for enterprises and organizations with specific requirements.

3.2.1 Characteristics of permissioned blockchain

The main characteristics of permissioned blockchain are:

- Access Control: Access can be controlled through digital certificates, cryptographic keys, or other authentication methods.
- Network Governance: They are typically governed by a group of trusted entities. These users collectively decide on network rules, consensus mechanisms, and governance. Hence, these blockchains have a more centralized governance structure.
- Consensus Mechanisms: Such blockchains use consensus mechanisms to validate transactions and maintain the ledger's integrity. They often employ more efficient consensus mechanisms. As a result, these mechanisms enable faster transaction validation and lower energy consumption.
- Privacy and Confidentiality: Enhanced privacy and confidentiality are significant characteristics of permissioned blockchain. Users have greater control over who can access their data and transactions. Thus, it is crucial for industries where sensitive information must be kept confidential.
- Smart Contracts: Many such blockchains support smart contracts, self-executing code that automates processes and agreements when predefined conditions are met. Smart contracts can streamline complex business processes, reduce the need for intermediaries, and enhance operational efficiency.

3.3 Blochain's state

Our exercise does not store the state of the amounts A and B have at any time of the transaction history. Wallets are the answer to this issue.

3.3.1 Wallets

E-wallets allow individuals to store cryptocurrencies and other digital assets. In the case of Blockchain Wallet, users can manage their balances of various cryptocurrencies such as the well-known Bitcoin and Ether as well as stellar, Tether, and Paxos Standard.

Wallet security is an important consideration for users, as a compromised account may result in users losing control of their assets. Blockchain Wallet has several levels of security to protect user funds from any possible attacker, including the company itself.

How blockchain wallets work?
Bitcoin wallets and other crypto wallets are based on public-key cryptography. The essential elements of a wallet are a public key, a private key, and an address.

It all starts with the private key, a 256-bit binary number that is generally represented as 64 numbers and letters like so: D88C 5E31 8005 A994 C378D 9021 66E9 04E2 69CA 3860 8DBB E274 884F 3010 F004 C08C. The private key is essentially the password you use to gain access to your data and resources on the blockchain.

You can think of the public key as your account number. It, too, is a long series of numbers and letters. It is used to encrypt information that is intended for you before the information is posted on the blockchain. Only your private key can decrypt information that is encrypted with your public key.

3.3.2 Smart contracts

We find it necessary to mention smart contracts. Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's

involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met.

They can be used to maintain users' balances and regulate asset transfers.Each user interacts with the smart contract to conduct transactions, and the contract keeps the state of users' balances updated.

3.4 Data privacy

In our exercise, the data of the transactions being made is not private, being anyone capable of knowing the movement of data being made. Although the manipulation of the data is secured, it would also be interesting to keep private the data being shared because, once sent, it will stay in the blockchain forever.
Among all the existing techniques/technologies, we had chosen Zero-Knowledge Proof as our approach to this problem.

3.4.1 Zero-Knowledge Proof (ZKP)

ZKPs are a cryptographic method used to prove knowledge about a piece of data, without revealing the data itself. They work by having the verifier ask the prover to perform a series of actions that can only be performed as needed if the prover knows the information inside.
ZKPs can either be interactive (prover convinces a specific verifier but needs to repeat the process for each individual verifier) or non-interactive (where a prover generates a proof that can be verified by anyone using the same proof).
The fundamentals characteristics of this technique are:

- Completeness: If a statement is true, then an honest verifier can be convinced by an honest prover that they possess knowledge about the correct input.
- Soundness: If a statement is false, then no dishonest prover can unilaterally convince an honest verifier that they possess knowledge about the correct input.
- Zero-knowledge: If the state is true, then the verifier learns nothing more from the prover other than the statement is true.

4. Conclusions

The conclusions from this laboratory experiment are as follows. Initially, we successfully created and comprehended a basic blockchain structure that was provided. Later on, we developed endpoints to meet specified requirements, such as validators for the integrity of the nodes and listing existing nodes.

Through this process, we gained an understanding of the important role blockchain plays in securing transactions, particularly in the realm of cryptocurrencies. The utilization of pertinent hash functions emerged as a critical element in maintaining data integrity within the blockchain.

Having achieved these milestones, we went deeper into the exploration of a basic blockchain's limitations. Our focus extended to understanding and addressing challenges inherent in blockchain technology, leading us to learn concepts such as Proof of Stake, Proof of Work, Wallets, and Zero-Knowledge Proofs. This exploration not only help us understand better our theoretical knowledge but also helped us with insights into practical approaches to overcoming challenges within the blockchain landscape.

5. References

E-wallets:
https://www.investopedia.com/terms/b/blockchain-wallet.asp
https://kriptomat.io/blockchain/what-is-a-blockchain-wallet/

Smart contracts:
https://www.ibm.com/topics/smart-contracts

Proof of Work - Proof of Stake:
https://academy.binance.com/en/articles/proof-of-work-vs-proof-of-stake

Zhang, X., Ye, C. A novel privacy protection of permissioned blockchains with conditionally anonymous ring signature. Cluster Comput 25, 1221–1235 (2022).

Zero-Knowledge Proof: https://chain.link/education/zero-knowledge-proof-zkp