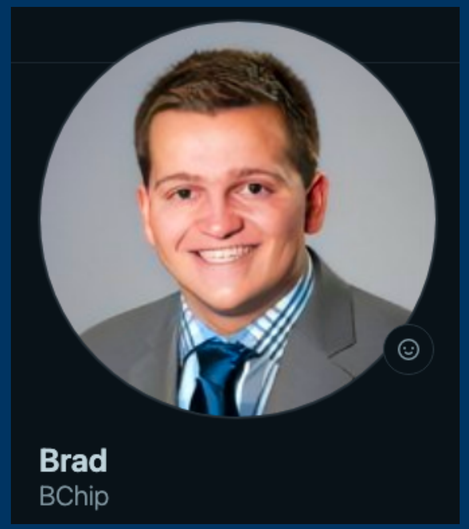
Building a Simple Blockchain in Python

Michigan Python Group - Meetup



Bradley Chippi

Getting Started

- We are going to build a simple proof-of-work system that mimics the simple functionality of a blockchain.
- We will be using SHA256 as our hashing algorithm of choice. The same hashing algorithm Bitcoin uses!

What is Proof-of-Work?

- Proof of work is a cryptographic proof in which one party proves to others that a certain amount of computational effort was performed.
- To verify that work, it must require a minimal amount of computational power.
- Bitcoin uses proof of work as the foundation for a consensus view of its decentralized network.

- Source: https://en.wikipedia.org/wiki/Proof_of_work

History

- Hashcash was one of the first systems to utilize proof-of-work.
- Hashcash’s purpose was to prevent/limit email spam and denial of service attacks.
- Bitcoin’s white-paper by Satoshi Nakamoto references the hashcash paper.

- Source: http://www.hashcash.org/papers/hashcash.pdf

SHA256 Hash

What is it?

- A hash function is used to map data of any size to a fixed size.
- A hash function should compute a hash of a defined range.
  - No matter how long the input is, the output of the hash should always be the same.
- The output should always be the same if the input is always the same.
- The output should not be reversible. Meaning, you cannot calculate or guess the input by looking at the output.
- Small changes to the input should be significant changes to the output.
- Source: https://en.wikipedia.org/wiki/Hash_function

Continued

- SHA256(“1”) = 6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
- SHA256(“I love python”) = cfb370f0419ff4e3a8b57f2a26ee861849936a64a4b8dbf7b2ad50894d9400fb
- SHA256(“The blockchain is so cool!”) = 7f65d25b60a7c0321a075d1f41b253a8d583009377c99c26dccb628e42f30e68

- Source: https://guggero.github.io/blockchain-demo/#!/hash

A Block in the Blockchain

- A blockchain has many blocks.
- A simple example of a block is:

```
class Block:

    def __init__(self, index, timestamp, data, hash="0", previous_hash=None):
        self.index = index
        self.timestamp = timestamp
        self.data = data
        self.previousHash = previous_hash
        self.nonce = 0
        self.hash = hash
```

- Source: https://guggero.github.io/blockchain-demo/#!/block

Continued

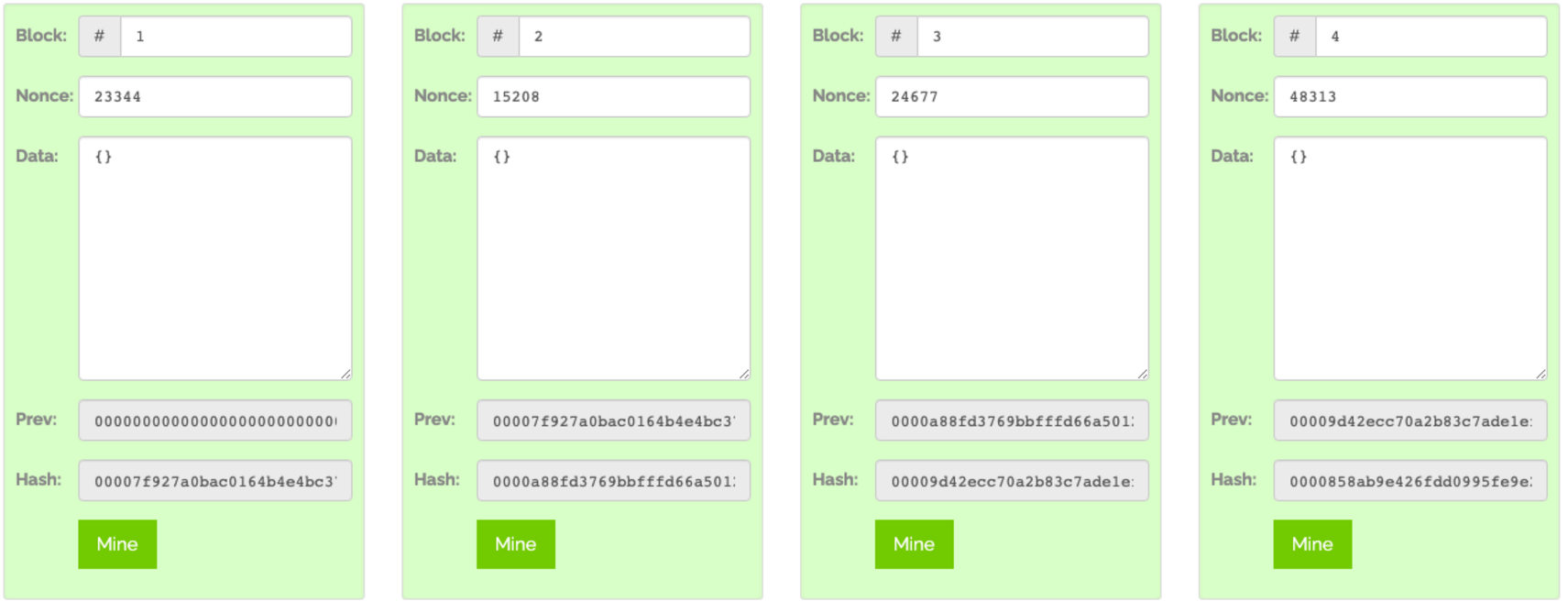
- A block is considered valid if its hash has a certain form.
- Bitcoin uses a certain number of zeros in the prefix of the hash for this.
- The more number of zeros in the prefix of the hash, the harder it is to find the nonce.
- Hence, the mining algorithm takes the block information and it keeps incrementing the nonce until it gets that “perfect/valid” hash. For example: 00002ba3f0e34acf1488c9c149ca1ebe2671e028857d13af22dc445b2f67dbff
- The first block in a blockchain is typically called the genesis block. Its previous hash is 0000000000000000000000000000000000000000000000000000000000000000

Continued

```
def hashBlock(self):
    data = (str(self.index) + str(self.data) + str(self.timestamp) +
            str(self.previousHash) + str(self.nonce)).encode('utf-8')
    return hasher.sha256(data).hexdigest()
```

Blockchain

- Source: https://guggero.github.io/blockchain-demo/#!/blockchain



How Many Leading Zeros are in the Hash for the Most Recent Blocks in the Bitcoin Blockchain?

- Go to: https://www.blockchain.com/explorer
- Look under Latest Blocks
- Click on the most recent one
- Look at the hash and count the number of zeros

Attacks on Blockchains?

Could We Add a Custom “Malicious” Block?

- You would have to control more than 51% of the network.

Is There Other Use Cases for Blockchain Technology?

- There are many use cases.
- The biggest selling point is immutability.
- In my opinion, the biggest use case is in supply chain applications.
  - Blockchain can offer immutable traceability.
  - A customer could see their purchase from end-to-end.
- Example: https://traceability.starbucks.com/#/
  - Built with Microsoft’s blockchain-as-a-service technologies.

Lets Code It!