A Blockchain-Based Gradebook System

Student: Harris Beg

Mentors: S. Sivagananam, V. Nandigam, K. Lin, S. Sakai

SDSC

UCSD

Abstract

With the advent of many large central organizations holding great amounts of data that are commonly subject to hacking, the role of a decentralized yet interconnected system becomes more apparent. While the use of blockchain in data authenticity has become increasingly popular, its presence is heavily lackluster in the scope of education, a system in which immutable and secure data is vital in products like gradebooks, transcripts, and digital classrooms. In light of this, a proof-of-concept blockchain network was built from the ground up with signed transactions and user authentication to demonstrate one possible application of a truly decentralized network.

Introduction

In a blockchain network, several users collaborate to add blocks of data to their own local copy of a chain of blocks. These blocks are linked through cryptographic hashes (as shown in Fig. 1) and are validated within the network. Because each registered instance of the blockchain network (called a "node") has a copy of the ledger of transactions, the network is deemed "decentralized."

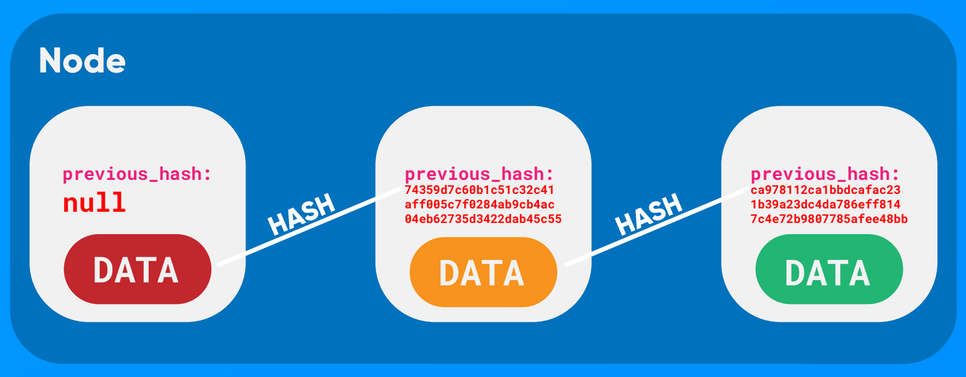
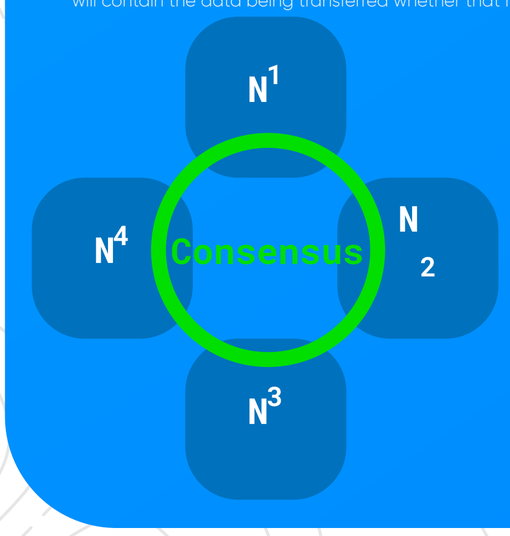


Fig. 1: A node in a blockchain network. The node contains a copy of the blockchain with several transactions encoding data. To prevent centralization, the data within each Nth block is verified through the hash of the (N+1)th block - this data, present in the diagram above, can be fetched as a transaction log [1]. Each transaction will contain the data being transferred whether that means Bitcoins, messages, or gradebook data.



Commonly, blockchain involves the participation of multiple peers. Because of this, consensus algorithms must be set up in the aforementioned networks to keep nodes engaged in addition to verifying that all nodes have the same set of information. In a blockchain network with a consensus protocol, every time a new transaction is requested by a peer, a majority of the peers in the network must agree that the new transaction is valid. Once the transaction is added locally, the hash of the transaction's corresponding block must be updated through a specially designed **consensus algorithm** [2]. If the transaction is successfully updated and verified, it will be appended to a block or instigate the creation of a new block.

Methods

Blockchain

To build a decentralized blockchain network for grades (called Blockbook), I utilized Flask for frontend components, routing, and debugging while using a custom-built Blockchain and Block class written in Python 3.6. While light, Blockbook does NOT reference a backend database and thus using a more complex framework such as Django was not deemed necessary.

Authentication

The authentication system built to privatize the blockchain network to only students and teachers uses the Google OAuth Web SDK.

Signing

PyNaCl was effectively implemented in Blockbook to sign the data transactions. For the sake of the demo, no private password was set, but this would be required in a real-world environment. By adding a signing system, nodes cannot act maliciously without having access to a valid private key provided by the Certificate Issuer.

Flask
web development, one drop at a time

Results

Data Structures

```python
class Block:
    def __init__(self, index, transactions, timestamp, previous_block, certificate):
        self.index = index
        self.transactions = transactions
        self.timestamp = timestamp
        self.certificate = str(certificate)
        self.previous_hash = self.get_hash(previous_block)

    def get_hash(self, block):
        if block is not None:
            return sha256(json.dumps(block.__dict__).encode()).hexdigest()
        return "0"
```

Fig. 2: The block class as written in Python. As shown, it contains the fields "index," "transactions," "certificate," and previous_hash. The transaction array holds the gradebook data while the other properties simply verify the block's validity.

To truly understand the application of blockchain to Blockbook, it is important to comprehend the underlying data structures that allow the gradebook data to be distributed and securely accessed. Blockchains are linked lists in which the pointers exist by accessing previous hash values. The data contained in each block is depicted in the code above.

Blockchain Workflow

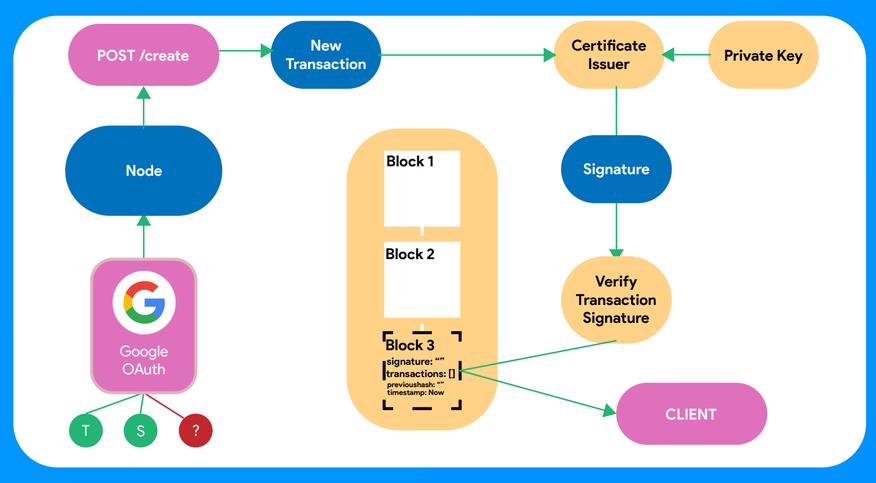


Fig. 3: The entire Blockbook system is shown with the magenta components representing client-side actions, blue representing a single node, and gold reprenting the network as a whole. While Blockbook only has one node as of the moment, its scalable architecture allows for multiple.

Blockchain Structure

```python
class Blockchain:
    chain: List[Block]
    def __init__(self):
        self.chain = []

    def add_block(self, transactions, timestamp):
        global certificate # Accesses global certificate instance
        if len(self.chain) == 0: # Checks if first block
            self.create_genesis_block(transactions)
        else:
            certificate.set_transactions(transactions)
            verify_key = nacl.signing.VerifyKey(certificate.public_key(),
                                                encoder=nacl.encoding.Base64Encoder)
            try: # Verified
                verify_key.verify(certificate.signature)
                block = Block(len(self.chain), transactions, timestamp,
                              previous_block=self.previous_block, certificate=certificate)
            except Exception: # Unverified
                return
            self.chain.append(block)

    def create_genesis_block(self, transactions):
        global certificate
        certificate.set_transactions(transactions)
        genesis = Block(0, transactions, time.time(), None, certificate=certificate)
        self.chain.append(genesis)  # No need to verify since it's the genesis block

    @property
    def previous_block(self):
        return self.chain[-1]

    def get_hash(self, i):
        # Used to retrieve current hash
        return self.chain[i + 1].previous_hash
```

To register each block with certificates, a CertificateIssuer class contains an inaccessible copy of a private key and assigns a public key based on the incoming transaction data. This is used to verify that the new transaction was not tampered with by an outside entity.

Note: As evident, the creation of a genesis block is enclosed in a separate method. This is because there is no previous hash.

Analysis/Future Work

- Truly immutable data
- A more advanced frontend (e.g. separate portal for students, can view overall grades)
- More data fields (e.g. comments, date assigned, point value)
- Use school authentication system instead of Google OAuth to make it truly privatized - currently, anyone with a Google account can make transactions
- Whitelist teachers DB: store the ID's of users who have been deemed teachers

References

1. @francesxu13gmail, Fenglian Xu, and Sasha @sashabakcht. "What Is Blockchain Technology? A Step-by-Step Guide For Beginners."
2. Blockgeeks, 26 July 2019, blockgeeks.com/guides/what-is-blockchain-technology/.
"Consensus Algorithms: The Root Of The Blockchain Technology."
3. 101 Blockchains, 16 May 2019, 101blockchains.com/consensus-algorithms-blockchain.
4. Hoffman, Chris. "What Is a Checksum (and Why Should You Care)?" How, How-To Geek, 29 Aug. 2018, www.howtogeek.com/363735/what-is-a-checksum-and-why-should-you-care/.
5. "The Ultimate Guide To Understanding The Basics of Blockchain and Cryptocurrencies." The Ultimate Guide To Understanding The Basics of Blockchain and Cryptocurrencies - By, hackernoon.com/the-ultimate-guide-to-understanding-blockchain-and-cryptocurrencies-f37cf4c0043.