

Thesis: An electronic voting system based on blockchain technology

School: No.2 High School of East China Normal University, Zizhu

Name: WenboZhu

December 10, 2021

Abstract

Blockchain is the underlying technology of Bitcoin, which is essentially a consensus-based ledger technology with decentralized, tamper-evident, and anonymous features. This paper investigates the feasibility of applying blockchain technology to electronic voting systems. This paper first analyzes the shortcomings of the traditional voting system; studies the technical method of using blockchain technology to realize a high-security voting system, and designs a blockchain electronic voting system scheme, and implements a private blockchain simulation environment using python language; finally, a blockchain-based electronic voting software simulation prototype is implemented using Django and python. It is found that blockchain-based electronic voting system has greater advantages than traditional voting system in terms of privacy, verifiability, democracy, fairness and completeness, legalization, and is the better solution to ensure the fairness and security of the voting system.

Keywords: blockchain, electronic voting, asymmetric encryption, digital signature

1. Introduction

1.1. Background of the subject

Traditional data management systems usually use centralized databases, and such centralized systems are generally managed and maintained by a single organization or individual, and have absolute control over all data, so that other people do not have a complete understanding of the data update process, and data security trustworthiness is insufficient; they are also vulnerable to single-point attacks by attackers.

In recent years, with the widespread spread of digital currencies such as Bitcoin, the underlying bookkeeping technology used, the underlying blockchain technology, has also received a lot of attention from researchers and scholars. The essence of blockchain is an open and transparent database ledger that records all transactions. It is characterized by the security features of decentralization, immutability, and openness and transparency without a third-party intermediary. These technical features of blockchain solve the problems of high cost, low efficiency, and low security of the existing centralized system.

1.2. Significance and main content of the subject

1.2.1. Disadvantages of traditional electronic voting system

The traditional electronic voting system has the following drawbacks.

- 1) Insufficient data security, the centralized system is vulnerable to attackers' intrusion, tampering or even destroying voting results
- 2) Risk of leakage of voters' personal information
- 3) Poor disaster tolerance, risk of data loss and data corruption.
- 4) Voting data management authority is not transparent and there is a risk of favoritism and fraud
- 5) Voting results are completely controlled centrally, and voters cannot verify whether their voting results are correct.

The above problems have seriously affected the availability and credibility of the existing voting system.

1.2.2. Advantages of blockchain e-voting system

Blockchain technology itself is derived from Bitcoin, and its essence is a database ledger based on consensus mechanism, which has the characteristics of decentralization, traceability, tamper-evident and quasi-anonymity.

The electronic voting system based on blockchain technology has fair, transparent, verifiable, and tamper-evident voting data, which improves the traceability of the voting system and reduces the trust risk of the system. At the same time, it can effectively prevent illegitimate voters or malicious organizations from fraudulent ballots, disrupting the voting process and interfering with voting results. In addition, the anonymous algorithm protects the privacy of voters by allowing anyone to query and verify the voting results without compromising the openness and fairness of the voting process.

1.2.3. The main content of the subject

This paper investigates the feasibility of applying blockchain technology to electronic voting systems. This paper first analyzes the shortcomings of the traditional voting system; studies the technical method of using blockchain technology to realize a high-security voting system, and designs a blockchain electronic voting system scheme, and implements a private blockchain simulation environment using python language; finally, a blockchain-based electronic voting software simulation prototype is implemented using Django and python.

2. Research process

2.1. Security design of the voting system

The goal of an ideal e-voting system is to complete the voting through the public network and to satisfy the maximum demand with minimum cost. To this end, many e-voting system researchers have discussed the minimum requirements that an e-voting system should meet according to different voting situations, and formed different classifications, in general, an e-voting system should meet the following requirements.

1) Accuracy

Goal: Any invalid ballot will not be counted; no matter anyone deletes, modifies, or copies the ballot, it can be checked and processed by the system, so that it cannot disturb the normal voting.

The solution used in this paper: using secure hash algorithm, asymmetric cryptographic signature technology, and blockchain POW bookkeeping mechanism can achieve this goal.

2) Privacy

Goal: All ballots are kept secret, and no one can correspond the ballot to the voter to determine what a person voted for

The scheme used in this paper: using asymmetric encryption techniques, using public keys instead of direct identity information to represent the voting user can achieve this goal.

3) Verifiability

Goal: Voters can query their own clients to confirm whether their votes have been tampered with after submitting their ballots and before the system tallies the voting results.

Solution used in this paper: Public blockchain chain technology, which can be used by each user to independently query the voting information on the blockchain, retrieve their voting records, and verify their signatures, can achieve this goal.

4) Democracy

Goal: Only legitimate voters are allowed to vote, and only once

The solution used in this paper: Voting identity verification KYC verifies the legitimacy of the voter, and my private key generates products to ensure uniqueness, which can achieve this goal.

5) Legalization

Goal: Only authorized people can vote

The solution used in this article: Voting identity verification KYC verifies the legitimacy of the voter, which can achieve this goal.

6) Completeness

Goal: The certification authority should accept votes from any legitimate voter and all valid votes can be counted correctly.

The solution used in this paper: public blockchain technology, which can be used by each user to independently query the voting information on the blockchain; blockchain bookkeeping is tamper-proof and every valid vote is recorded, which can achieve this goal.

2.2. Architecture of blockchain voting system

The blockchain structure used in this paper is shown in Figure 2-1 below. Each block consists

of two parts: the block header and the block body, in which multiple transactions recorded in this block are stored, and the transaction data are stored according to the Merkle tree structure; the block header stores the hash of the previous block, random numbers, Merkle roots of the transaction data, etc.

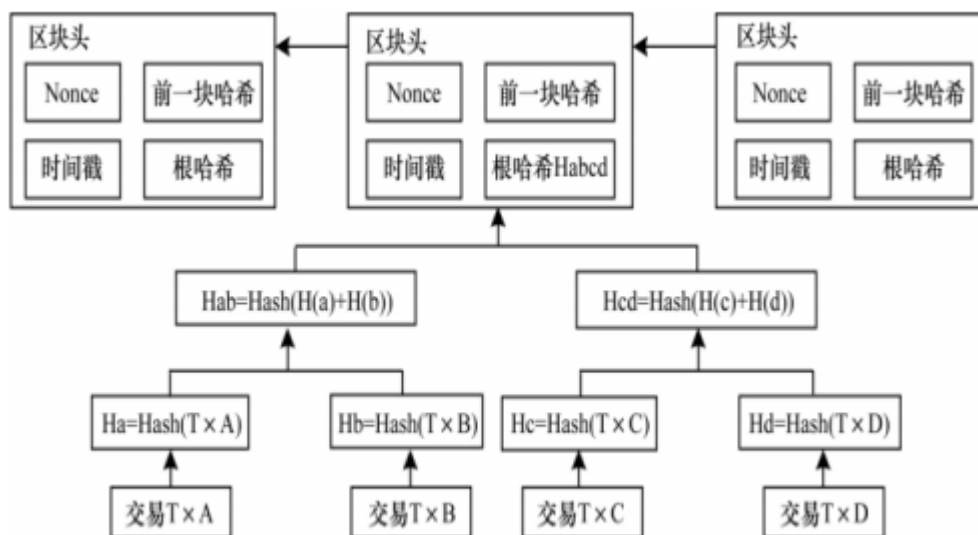


图 2-1

The block header structure of the blockchain voting system in this paper is designed as follows
Table 2-1

Table2-1

| Fields | type | Meaning |
|-----------|----------|----------------------------|
| prev_h | 32 chars | Previous block hash |
| merkle_h | 32 chars | Merkle tree root hash |
| h | 32 chars | This block hash |
| nonce | int | Random numbers for mining |
| timestamp | float | Block creation time |
| id | int | Block id for emulation use |

The transaction data of this blockchain voting system is designed as follows Table 2-2

Table2-2

| Fields | type | Meaning |
|-----------|----------|---|
| uid | 32 chars | Voter's id (public key) |
| vote_id | 32 chars | Voting project id |
| vote | int | Results of the selected voting candidates |
| timestamp | float | Voting time |
| id | int | Block id for emulation use |
| signature | 32 chars | Voting record signature |

2.3. Software architecture of the voting system

The software architecture of this blockchain voting system is shown in Figure 2-2 below:

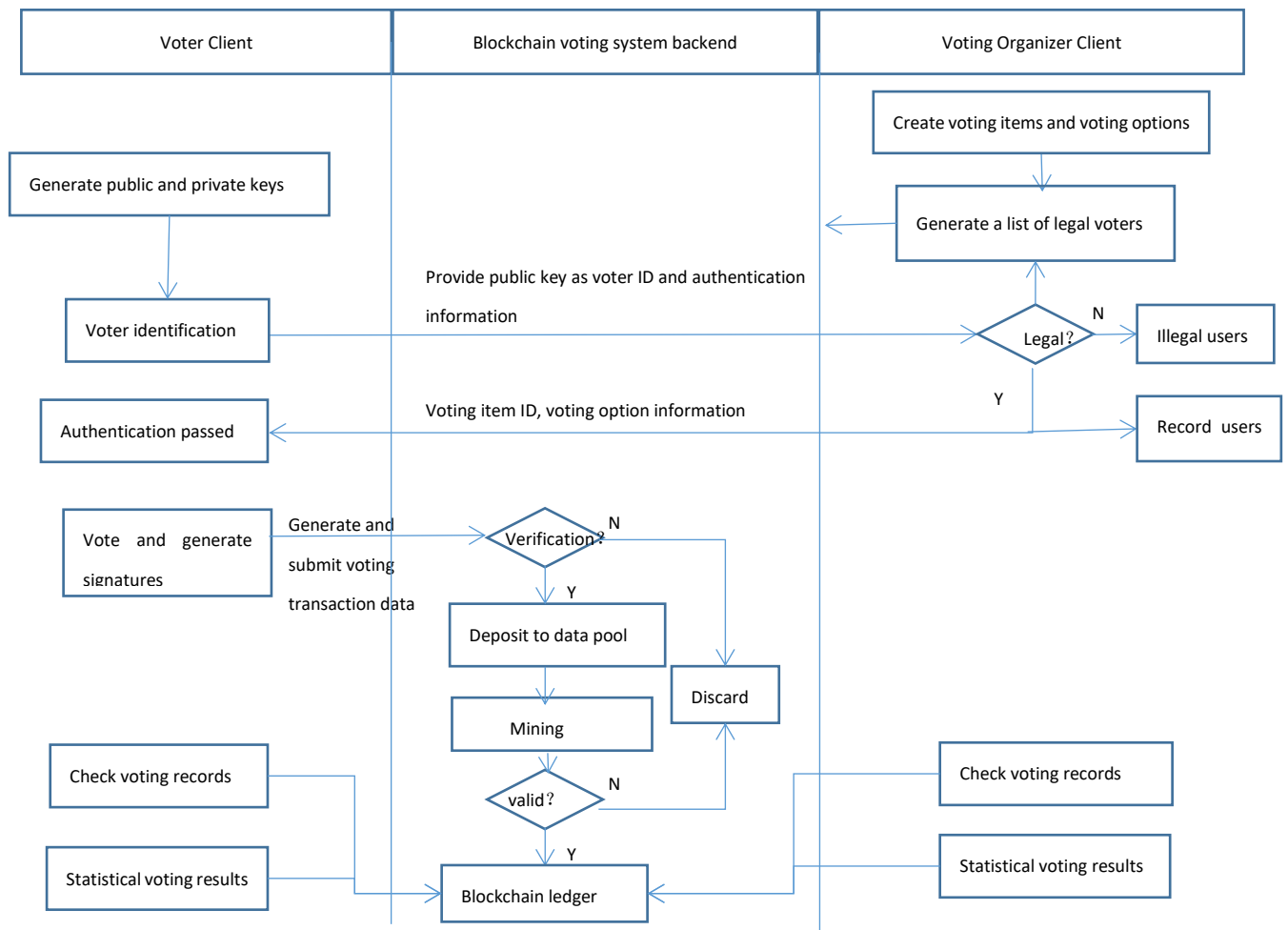


Figure 2-2

The overall scheme is roughly divided into three pieces, the first is the voter client, the voter is the person who votes, the electronic voting blockchain system is the voting backend, and the voting organizer is the person who initiates the voting project. The specific process is that the voting organizer first creates a voting project and voting options, and then generates a list of legitimate voters, who are qualified to vote. Before the voting begins, the voter can generate a pair of their own public and private keys through public and private key generation software. Then they can verify that they are eligible to vote through KYC. If the KYC is passed, the voting organizer will record the voter's public key; if the KYC is not passed, the voter will be determined to be an illegal user and will not be given the option to vote. Once the authentication is passed, the voter can vote and generate a signature, and perform subsequent signature verification. The specific signature verification method is that first the voter client calculates the hash value of the voter's voting record, and then encrypts this hash value with the voter's own private key to generate a signature cipher, which is saved in the voting transaction record. After decryption, the HASH value generated from the voting transaction record (voting ID, voting options, etc.) is compared with the decrypted value,

and if it is the same, it means the signature authentication is passed, if not, the voting transaction is considered illegal and discarded. Transactions that pass signature verification are stored in a data pool, packaged into blocks, and then it is time to mine and compete for bookkeeping rights. The bookkeeping also requires checking if the block is valid. How do you determine if a block is valid? The method is to calculate the root hash value of all the transaction records in the block according to the Merkle tree. If the hash value obtained from the calculation matches the hash value saved in the block header, the transaction data in the block has not been modified and the block is valid. In this way, a voting ledger is generated through the blockchain, and we can securely keep the voting records, query the voting records, and count the voting results.

3. Research results

3.1. Blockchain simulation environment

This subject uses python to implement a private chain simulation environment with block generation, mining, and synchronization functions. This simulation environment only needs to run on a stand-alone system and can be accessed through a local web interface.

3.1.1. Signature generation

The code of voting transaction and signature generation is shown in Figure 3-1; the signature mechanism is implemented by using the python system's asymmetric encryption algorithm package ECC and secure hash function package SHA3 256, the specific principle is to first construct the transaction record into the ballot structure, then use the private key to encrypt the hash value, generate the ciphertext signature value, and save it in the transaction record header.


```

voter_id = request.POST.get('voter-id-input')
vote = request.POST.get('vote-input')
private_key = request.POST.get('private-key-input')

# Create ballot as string vector
timestamp = datetime.datetime.now().timestamp()
ballot = "{}|{}|{}".format(voter_id, vote, timestamp)
print('\ncasted ballot: {}'.format(ballot))
signature = ''
try:
    # Create signature
    priv_key = ECC.import_key(private_key)
    h = SHA3_256.new(ballot.encode('utf-8'))
    signature = DSS.new(priv_key, 'fips-186-3').sign(h)
    print('\nsignature: {}'.format(signature.hex()))

```

Figure 3-1

3.1.2. Block generation and mining

The main flow of the code is shown in Figure 3-2.

First: generate the transaction records into a Merkle tree structure and construct the block header. Specifically: first read the list of transaction records to be generated into a block; add the hash value of each transaction record into the tree as a node of the Merkle tree, and update the node hash values two by two according to the requirements of the Merkle tree, and finally calculate a Merkle root hash value, which will be saved in the block header.

Then: start mining and compete for bookkeeping rights. The simulation system only simulates the local mining process; specifically: starting from 1, iterate through the random number nonce, each time calculate whether the hash value of the block header meets the difficulty factor puzzle (the system assumes that the difficulty factor is 0000 for the hash header), if not, then nonce+1, otherwise, record the nonce into the block header, and bookkeeping Success.

```

# Seal transactions into blocks
time_start = time.time()
number_of_blocks = settings.N_BLOCKS
prev_hash = '0' * 64
for i in range(1, number_of_blocks + 1):
    block_transactions = Vote.objects.filter(block_id=i).order_by('timestamp')
    root = MerkleTools()
    root.add_leaf([str(tx) for tx in block_transactions], True)
    root.make_tree()
    merkle_h = root.get_merkle_root()

    # Try to seal the block and generate valid hash
    nonce = 0
    timestamp = datetime.datetime.now().timestamp()
    while True:
        enc = ("{}{}{}{}".format(prev_hash, merkle_h, nonce, timestamp)).encode('utf-8')
        h = SHA3_256.new(enc).hexdigest()
        if h[:pcount] == puzzle:
            break
        nonce += 1

    # Create the block
    block = Block(id=i, prev_h=prev_hash, merkle_h=merkle_h, h=h, nonce=nonce, timestamp=timestamp)
    block.save()
    print('\nBlock {} is mined\n'.format(i))
    # Set this hash as prev hash
    prev_hash = h

time_end = time.time()
print('\nSuccessfully created {} blocks.\n'.format(number_of_blocks))

```

Figure 3-2

3.2. Voting Software Implementation

3.2.1. Voting UI

The voting UI designed in this paper, as shown in Figure 3-3, is divided into two parts. The left part is used for voting, giving it a name of block, and the right part is mainly for viewing the voting transaction records and voting results in the block chain.

一种基于区块链技术的电子投票系统



Created by GP for Prof. ABM. Modified by mendax1234.
13 Aug 2021.
This software is only a prototype to simulate a concept.

Figure 3-3




3.2.2. Example of voting

Figure 3-4 shows a specific example of voting implemented in this system, a vote for the most popular team for the 2020-2021 season of FRC, one of the more famous robotics competitions. The interface allows you to enter your public key id (representing the voter); at the bottom is the voting option selection, where you can choose one of the three voting items, that is, choose who you want to vote for. At the bottom is the possibility to paste your private key, which is used to generate a signature.

2020-2021赛季FRC最受欢迎战队

你的ID :

符合条件的队伍如下 :

| | | |
|--|--|--|
|  FRC Team 6940 (1) |  FRC Team 6941 (2) |  FRC Team 6907 (3) |
|--|--|--|

你选择的是 :

你的私钥 :

```
-----BEGIN PRIVATE KEY-----
MIIGHAgEAMBMGBYqGSM49AgEGCCqGSM49AwEHBG0wawIIBAQQgkScFYOR4uBsYXmd5
WHCY+NVm14uBz7QbYU532ropdsShRANCAAQnmv4Jpv6kOMmbUg25YmKv9M9y+omU
gQ80v+AVHLP2yXnk3yMQZyKyU5Kk0KQIQRvzv3H7QnHSmD085448Ptev
-----END PRIVATE KEY-----
```

* 请对你的公钥必须进行注册！

[提交投票结果](#)

Created by GP for Prof. ABM. Modified by mendax1234.
13 Aug 2021

Figure 3-4

3.2.3. voting blockchain

Figure 3-5 shows the block chain table implemented in this system, that is, the transactions are all calculated and stored in the block table; we can see that the block numbered 1 in the table has a hash value of 0 for its previous block, because it is a Genesis block. The hash of the block is a long string of numbers, and then there is a random number, the timestamp. The block numbered 2 with the previous block's hash value recorded in the block header is the hash value of the Genesis block, and so on. It is verified that the blockchain structure of this scheme is complete.

3.2.5. Security Verification

According to the security goals designed in this paper, the following verification tests were done

| Security goals | Verification method | Validation results | Research findings |
|----------------|---|--|-------------------|
| Accuracy | Modify voting record options | Signature verification failed, voting record discarded | High security |
| Privacy | View the content of voting transactions and find personally identifiable information about users. | No personally identifiable information | High security |
| Verifiability | Voters can check and examine their voting records. | You can query and check | High security |
| Democracy | Fraudulent voter voting | The signature does not pass, only the vote signed by the voter's own private key is valid. | High security |
| Legalization | Illegal user voting | Voting identity verification does not pass, can not vote | High security |
| Completeness | Voting data is available to all | Voting statistics are available to all | High security |

4. Research Summary

4.1. Blockchain is feasible to be applied to voting system

Through the design and implementation of the simulation prototype, it is verified that the blockchain technology is feasible and functionally correct to be applied to the voting system.

4.2. Blockchain voting system with higher security.

The proposed blockchain voting system scheme achieves the security objectives of the voting system, including Privacy, Verifiability, Democracy, Fairness, Completeness, Legalization.

4.3. Promotion of application value

In the future, this solution can be improved and deployed in public chain systems, such as Ether, which can achieve higher security and have high application value.

References

- [1] JiulinXi,Haohu 《Blockchain-based voting system design》 Communication Technology 2018.7
- [2] BingHe 《A secure e-voting based on chain ring signature》 Xiamen University
- [3] JunZou, HainingZhang, YiTang,LeiLi 《Blockchain Technology Guide》 Machinery Industry Press

Letter of Recommendation

Dear Admission Officer,

I am writing to recommend my student, Wenbo Zhu, for undergraduate study at your university. I am the instructor who participated in the cyber security workstation project study.

Wenbo Zhu has been a conscientious and courteous student during his studies at the workstation. He has chosen "Data Security" as his main research direction by learning about cyber security. The project is located in "Blockchain Technology for Electronic Voting System", which also reflects Wenbo Zhu's keen sense of new technology and emerging things, as well as his excellent thinking ability and innovation ability. He was also well prepared for the topic defense stage, and his presentation was fluent, and he was able to answer the questions from the judges at the topic defense site calmly and logically.

Wenbo Zhu showed great self-learning skills during the research and study sessions. He was able to advance his research by actively researching literature and leveraging resources from Open-Source communities, perfectly exemplifying the phrase "standing on the shoulders of giants."

Sincerely yours



Assistant Professor

School of Electronic Information and Electrical Engineering

Shanghai Jiao Tong University