Peer-to-Peer Networking and Applications (2021) 14:2801–2812
https://doi.org/10.1007/s12083-021-01100-x

AMVchain: authority management mechanism on blockchain-based voting systems

Chenchen Li[1] · Jiang Xiao[1] · Xiaohai Dai[1] · Hai Jin[1]

Received: 27 October 2020 / Accepted: 10 February 2021 / Published online: 11 March 2021

Abstract

As blockchain technology booms, modern electronic voting system leverages blockchain as underlying storage model to make the voting process more transparent, and guarantee immutability of data. However, the transparent characteristic may disclose sensitive information of candidate for all system users have the same right to their information. Besides that, the pseudo-anonymity of blockchain will lead to the disclosure of voters' privacy and the third-parties such as registration institutions involved in voting process also have possibility of tampering data. To overcome these difficulties, we apply authority management mechanism into blockchain-based voting systems. In this paper, we put forward AMVchain, a fully decentralized and efficient blockchain-based voting system. AMVchain has a three-layer access control architecture, and on each layer, smart contracts are responsible for validation and granting permissions. Linkable ring signature is adopted in the process of voting to protect ballot-privacy. AMVchain also makes a tradeoff between efficiency and concurrency by introducing proxy nodes. The experiments results show that our system meets the basic requirements under the high concurrent users circumstance.

Keywords Authority management · Smart contract · Electronic voting · Blockchain · Linkable ring signature

1 Introduction

As Internet and cryptography technology develops, electronic voting, as a promising voting method, has gradually gained public attention. Electronic voting is defined as using electronic means to help implement voting and counting. Compared with traditional paper-based voting, electronic voting technology can speed up the calculation of votes, reduce staff costs, and to a certain extent, ensure the fairness of elections and protect the interests of voters. Thus, a valuable and successful electronic voting system must be secure, transparent, anonymous and efficient. However, most e-voting platforms, like Doodle, Polyas and Ballot, build their voting platforms based on a centralized server-client model. These platforms adopt central agencies to record and calculate votes. The crucial challenge for centralized voting systems is that voters cannot verify their ballots since their votes may be tampered by administers. Another severe obstacle is the risk of central servers downtime. So centralized voting systems are weak in terms of security, anonymity, and process transparency.

The advent of blockchain brought a new era of research to life on how to create trustful and decentralized e-voting systems. In 2009, after Satoshi [1] first proposed the concept of blockchain, researchers begin to explore the potential of aggregating blockchain technology into conventional e-voting systems. Blockchain is a data structure formed by the orderly linking of blocks containing transactions

This article is part of the Topical Collection: *Special Issue on Blockchain for Peer-to-Peer Computing*
Guest Editors: Keping Yu, Chunming Rong, Yang Cao, and Wenjuan Li

✉ Jiang Xiao
jiangxiao@hust.edu.cn

Chenchen Li
chenchenli@hust.edu.cn

Xiaohai Dai
daixh@hust.edu.cn

Hai Jin
hjin@hust.edu.cn

[1] National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Lab, Clusters and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China

information, and each block has a hash value of itself to prevent to be hampered. Its essence is a distributed ledger. In blockchain, the transaction information is open and transparent, and each node has a backup of the ledger. Different nodes in blockchain reach consensus throughout the network according to a consensus protocol. Therefore, the blockchain provides a possible solution for problems in electronic voting systems with its decentralized, transparent, and immutable characteristics.

However, there exist some challenges in blockchain-based voting system. In this paper, we summarize four main challenges.

1.1 Conflicts between privacy and transparency

Information stored in blockchain is transparent and can be accessed by anyone. It is an important feature of blockchain for some specific scenarios, such as finance field. But for voting system, it may be a disadvantage. Candidates' personal data involves sensitive information needs the confidentiality. Therefore, it is necessary to introduce authority management mechanism into such a transparent scenario.

1.2 Limitations of traditional certificate authority

Certificate Authority is widely used in voting system. Traditional voting systems adopt certificate authority to verify whether participants' identities are legal. However, once obtaining access certificates after verification, everyone has the same and indistinguishable access to data. This is not suitable for voting systems. Therefore, to design a more advanced Certificate Authority is inevitable.

1.3 Coexistence between concurrency and efficiency

Voting process usually has a time limit. During the time span, voters' terminals send data to blockchain systems continuously, which results in high concurrency. However, the complex consensus mechanism greatly limits performance of the voting system. If all voters participate in the consensus process as consensus nodes, the performance of the voting system will greatly slump.

1.4 Pseudo-anonymous for voters' ballots information

Blockchain is pseudo-anonymous, for participants can know the public key address of the voter. Attackers can analyze the signed data's addresses on the blockchain to track the user's digital asset flow. At the same time, attackers can also utilize data mining algorithms to achieve de-anonymity. As the blockchain grows, the more information is published, the easier for attackers to de-anonymize. So how to protect voters' ballots information becomes the last challenge.

Motivated by aforementioned challenges, we adopt smart-contract based authority management mechanism to provide a hierarchical access control framework in this paper. Different levels have different access control to data. Utilizing the characteristics of smart contracts to automate the assignment of permissions. In addition, to promote the performance, our system is developed based on consortium blockchain Fabric and we introduce proxy nodes at the same time to reduce the number of nodes participating in consensus.

Our contributions can be summarized as follows:

- We devise and implement AMVchain, an efficient and scalable voting protocol to accomplish transparent and decentralized voting with the power of blockchain and smart contracts.
- Proposing a smart contract-based access control machenism, which enables hierarchical authority management in different entities. Setting up supervisor roles and proxy nodes at the same time to facilitate and secure voting process, which makes a good tradeoff between concurrency and efficiency.
- We utilize linkable ring signature to encrypt voter' ballots, which cuts off the link between voters and ballots and guarantees voters' anonymity.
- Experimental results show that AMVchain can effectively handle voters' requests under relatively large-scale circumstance.

The rest of the paper is organized as follows. Section 2 introduces related work in blockchain-based voting systems and authority management. Section 3 presents an overview of AMVChain. Section 4 describes the three-tier access control framework in detail. In Section 5, we evaluate performance of the proposed system. Finally we draw conclusions in Section 6.

2 Related work

Although the voting system based on blockchain has the characteristics of transparency and immutability, these systems still require human intervention in the identity authentication and vote counting stages. Therefore, smart contracts are used to replace human operations. Simultaneously, participants' identities in the current voting system are complex, and a voting system without authority management will lead to certain risks. We present in this section various current research and categorize them into blockchain-based voting system and authority management mechanism. We then highlight the advantages of our proposed system compared to the others.

2.1 Blockchain-based voting system

With the popularity of cryptocurrency, many researchers begin to pay more attention to its underlying storage technology, blockchain. A lot of e-voting protocols were carried out combining with blockchain.

In 2015, Czepluch made the first attempt to evaluate the possibilities and strengths that blockchain technology as a service can offer in regards to decentralized trust-free voting systems [2]. At the same time, Z.Zhao designed a voting protocol using Bitcoin and zk-SNARKs with the properties of privacy, verifiability, and irrevocability [3]. This method added phase to distribute secret random numbers via zero-knowledge-proofs based on the lottery to protect individual voters' privacy. After that, Lee [4] proposed a four parties involved voting system and described it at a national level through examples. Agora [5], an end-to-end verifiable blockchain-based voting solution, was designed for governments and institutions to automate voting. Agora utilized Token to distinguish eligible voters for elections, and institutions distribute their tokens for each eligible voter. However, these schemes all relied on the trusted third parties (TTP) to supervise the voting process. The third-party may collude with candidates to tamper votes.

Besides, data stored in the blockchain is transparent and available to the public. This undermines the security and reliability of blockchain-based voting systems. To cope with this problem, smart contracts and encryption algorithms are adopted into blockchain-based voting protocols. Smart contracts were commonly utilized to replace the third party. A self-tallying protocol, the Open Vote Network (OVN), was realized by smart contracts [6]. This is the first implementation that the voting process does not rely on any trusted authority to tally election results. The voting scheme was implemented on Ethereum and totally distributed and automated. Votechain [7] utilized smart contracts to detect double voting and added transparency to the voting process. Ali Kaan Ko et al. [8] deployed a voting application as a smart contract on Ethereum and allowed voters to participate in voting via their valid EOAs. Nevertheless, this solution lacks a truly automated address verification protocol since the EOAs get their right to vote from a centralized authority to become eligible voters.

To protect voters' privacy and anonymity, researchers usually adopt digital signature schemes into these protocols. Digital signature schemes commonly adopted into blockchain-based voting systems can be divided into the following categories: public-key cryptography, blind signature, and homomorphic encryption. Voters' anonymity was protected in systems [9] and [10] by using public-key cryptography. Voting systems will generate a hash value that contains both voters' identification number and the hash of the previous ballot. But the disadvantages are that these systems led to a waste of computing resources, and voters cannot change their votes in case of their mistakes. Blind signature is less complicated and computationally intensive. Atusushi Fujioka et al. [11] utilized blind signature to encrypt ballot information to prohibit the leakage of intermediate election results and enabled a larger-scale voting. However, the blind signature requires anonymous channels. In some cases, it will fail to protect voters' privacy. For example, suppose the administrator knows the bitcoin address of the voter. In that case, he can get the identity of the voter by linking the address and message to the blockchain, thus destroying the anonymity of the voting system. Homomorphic encryption is also widely used in the electronic voting system. Hsiao et al. [12] combines homomorphic encryption and secret sharing schemes to realize a distributed electronic voting application without a trusted third party, which protects the anonymity of voters' identity, the privacy of data transmission, and the verifiability of voting. Also, [13] can verify the validity of votes and the correctness of election results without zero-knowledge proof based on LWE homomorphic encryption. However, the complexity of homomorphic encryption used in the aforementioned research is too high to apply on a large-scale.

Compared with the algorithms mentioned above, linkable ring signature has three advantages: Firstly, linkable ring signature can better guarantee anonymity among voters. Linkable ring signatures allow a participant to present a valid signature from a set of keys without revealing which key generated the signature. It can be seen as zero proof of knowledge of a key inside a set of keys. This allows a voter to remain anonymous among a set of participants. The larger the number of users involved in the signature, the more anonymous it becomes. Secondly, linkable ring signature can simplify the process and significantly improve the voting efficiency. This enables voting systems based on blockchain suitable for large scale use. Finally, given elections, people may regret their votes. We can use the linkability of linkable ring signature to cast new votes to reflect the real situation of voting better.

2.2 Authority management mechanism

Authority management is widely used to prevent unauthorized operations. In traditional centralized systems, the function of authority management is carried out by the third parties such as administrative organizations. These centralized entities may encounter single-point-of failures or hacking. While in decentralized systems such as IoT and blockchain, realizing authority management becomes a new research area.

Two standard approaches about access management in IoT were summarized in [14], respectively CoAP (Constrained Application Protocol) Management Interface (CoMI) and Lightweight Machine to Machine (LWM2M). As blockchain technology booms, many solutions have been proposed to realize decentralized authority management. Controlchain [15] is proposed to supple access control in IoT. It implements access control by storing data of different permissions into four different blockchains. Access control in wireless networking is a challenging problem. CCMMA [16] is designed to realize cross-layer access control in the Internet of Things. Grüner et al. [17] devises a quantifiable trust model based on blockchain to define trust levels, then realizes identity management according to it. Several solutions adopt smart contracts to automate the process of authority management. MedRec [18] is an application of authority management in medical scenario. It utilizes three different smart contracts to predefine permission rights of medical data. Also, it employs smart contracts defined authority management policies, such as access time, user identity to protect Biometrics information. To prevent the COVID-19 epidemic from worsening, [19] proposed a blockchain-based COVID19 medical research platform for CEMRs, which can provide efficient and privacy-preserving data sharing against COVID19.

Besides, a attributed-based authority management protocol was realized in [20]. In this protocol, users should obtain tokens by presenting their attributes to authorities, and enough tokens give them a privilege to do designated operations. Similarly, FairAccess [21] uses authorization tokens representing access right and entitlement. Only if requesters fulfill conditions defined in smart contracts, they will be delivered authorization tokens. Symmetric key encryption [22] is also adopted to realize permission assignment. It devises a key distribution scheme which only authorized people with keys to decrypt information content. DAOS was proposed based on identity-based encryption (IBE) technique to realize access control in data outsourcing service [23]. Besides, in the IoT and cloud storage, attribute-based encryption (ABE) [24] is a powerful cryptographic primitive for access control and fine-grained sharing on encrypted data. To address data storage and the semi-credibility nature of the IIoT cloud platforms, [25] proposed an attribute-based access control scheme that supports traceability and revocation for smart factories.

However, to the best of our knowledge, smart contracts introduced in blockchain-based voting systems mainly acts as substitutes for third parties to prevent external tampering. Few research utilizes smart contracts to realize access control in blockchain-based voting systems.

3 Overall design

According to the crucial challenges aforementioned that impede the traditional blockchain-based voting protocol, in this section, we devise a voting system called AMVchain based on consortium blockchain, combined with authority management mechanism to tackle these problems. Blockchain technology can guarantee the transparency and immutability of voting process. Authentication management is added in to supervise the entire voting process and improve voting efficiency at the same time, while linkable ring signature protects the privacy of system users. In this paper, we take student union election as an example.

3.1 Overview

The architecture of AMVchain is illustrated in Fig. 1. Its framework includes four layers: application layer, smart contract layer, consensus layer, and storage layer. The application layer is at the top of our system providing friendly graph user interface designed for voters and candidates to view or check voting status conveniently. The smart contract layer is composed of embedded programmatic contracts, which is driven by events, automatically executed and does not require human intervention. It provides some core functions such as registration and verifying validity of identity of participants in electronic voting. The functions of the consensus layer are to allow highly decentralized nodes to reach a consensus on the validity of block data. The Storage layer stores the data securely and permanently.

3.2 Application layer

The application layer is at the top of AMVchain, which aims to provide a user-friendly graphic interface for participants involved in our system. Different types of users have different functions. For voters and candidates, they can register in the websites by submitting their identity information. JavaScript files transfer foreground data to blockchain and invoke smart contracts. The system must satisfy some basic features, such as registering with email address and editing or uploading personal information. When sending an email, the framework will load the env configure properties file first, including the STMP server properties. And then, the system uses the Swift Mailer to make the connection with an STMP server and post the email to the target email address. Once smart contracts

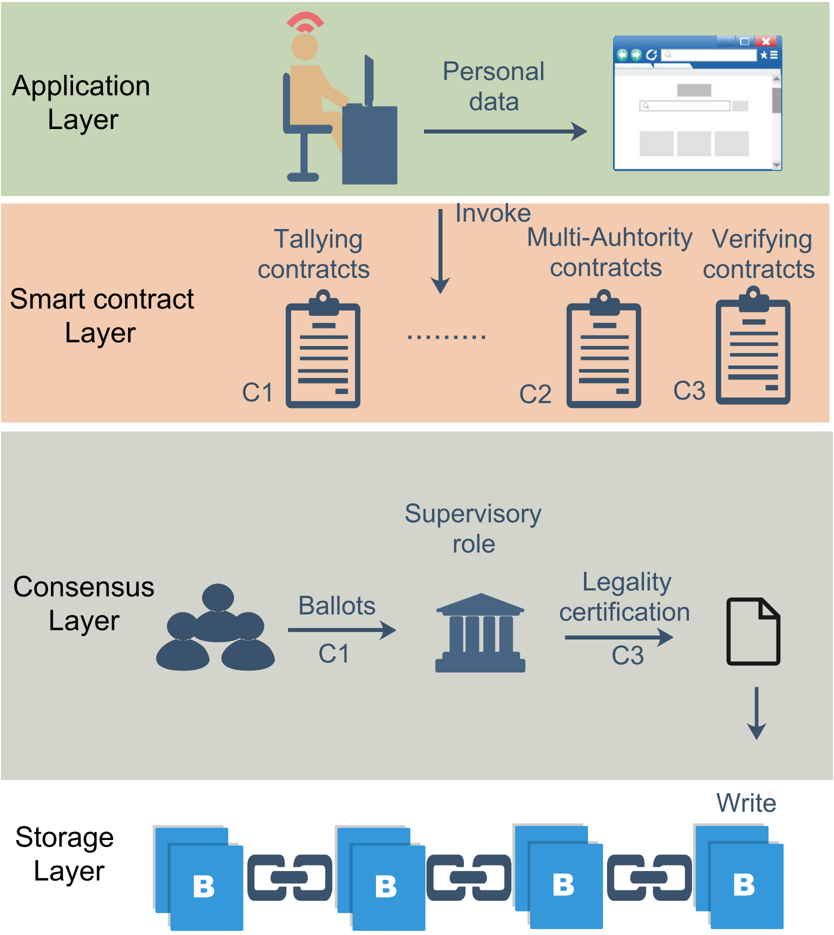
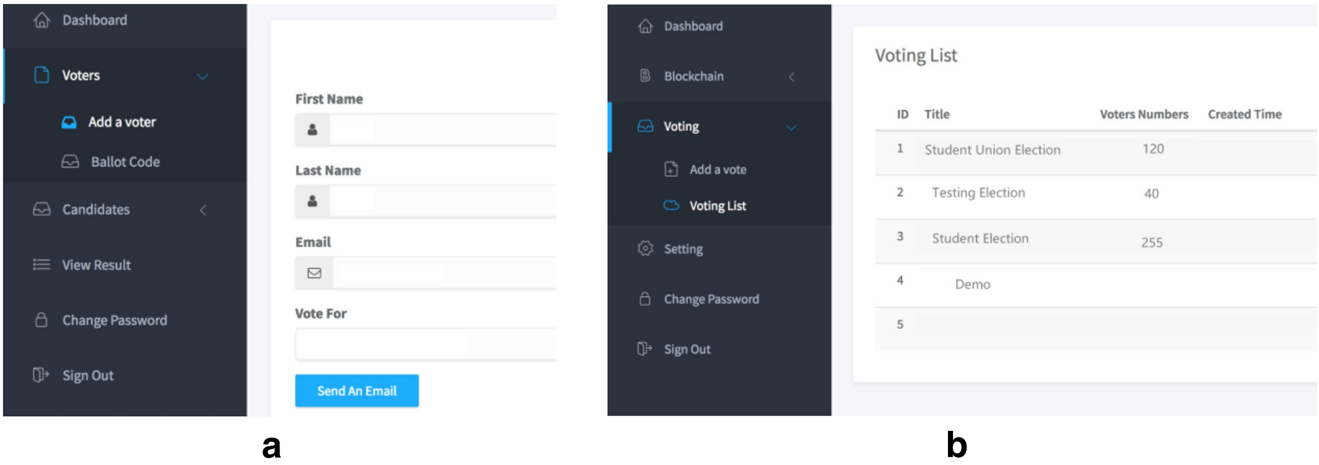


Fig. 1 The architecture of AMVchain

verify the validity of the information, they will be given unique voter id (Ui) and candidate id (Ci). The overview of voters and candidates is shown as Fig. 2a.

For Voting Initiator, functions and interfaces provided by this system are totally different. The dashboard presented to voting initiators should show the blockchain network and the voting information. Besides, the role of initiators needs to create a voting event with the title, description, the image and view the candidates of a voting event. So the voting list management interface is shown in Fig. 2b. Once he wants to start a poll, he can log in the website and deploy the chaincode on the consortium blockchain.



Voting List

ID	Title	Voters Numbers	Created Time
1	Student Union Election	120	
2	Testing Election	40	
3	Student Election	255	
4	Demo		
5			

Fig. 2 The overview of AMVchain

3.3 Smart contract layer

Smart contracts play an important role in our electronic voting protocol because they cannot be interfered by humans and will only execute according to pre-determined rules. So we choose smart contracts to realize some core functions such as user registration, checking the validity of users' identity and distributing the public key set for voters which will be utilized for generate ring signature. Besides, since smart contracts are stored on blockchain and cannot be modified once defined, we also adopt them to store voters' public key. At the same time, we utilize smart contracts to realize authority management. Not all users can access candidates' information. Only those who satisfy requirements pre-defined in smart contracts can obtain private keys to decrypt candidates' information. Authority management mechanism will be described in detail in Section 4. Web servers can easily call functions in smart contracts to process requests for they are publicly transparent in the consortium blockchain.

In proposed e-voting protocol, smart contracts are divided into management contracts *VoteManage*, authority management contract *VoteAuthority*, verification contracts *VoteVerify*, and counting contracts *VoteCount*. *Votemanage* is used to audit identity automatically. *VoteAuthority* is used to assign permissions to different roles. *VoteVerify* is responsible for verifying the signature of the vote while *VoteCount* is in charge of calculating ballots and announcing results.

3.4 Consensus layer

For the electronic voting system we proposed is suitable for a relatively enclosed environment, such as universities or research institutes, we employ hyperledger fabric, a consortium blockchain and *PBFT* as consensus algorithm. However, it is unrealistic for users to run full nodes on their terminal equipment for two reasons. Firstly, terminal equipment such as smart phones have limited resources, and full nodes need more powerful computing power. Secondly, *PBFT*'s time complexity is $O(n^2)$, it means the more nodes participating in consensus, the lower of system efficiency. To solve this problem, we leverage each institute as a proxy node. Specifically, the consensus process is executed by proxy nodes which have been authorized by CA. In this case, institutes package voting messages into block and broadcast it to other delegated nodes. Other nodes will verify the block. If the block is deemed as valid block, follower nodes will broadcast voting signature. Repeat this procedure until more than two-thirds of nodes' voting signature have been received. The consensus process is shown as Fig. 3.

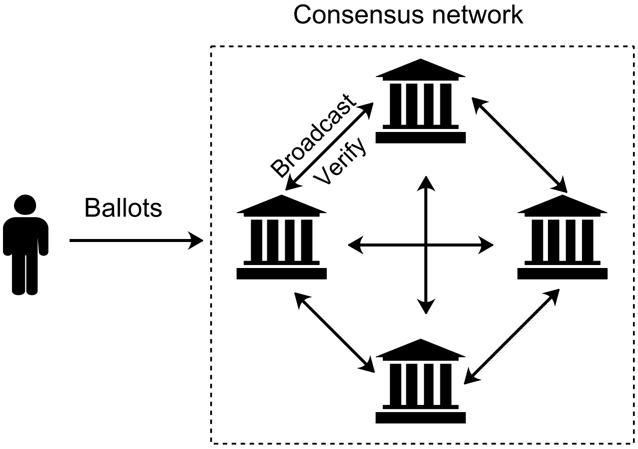


Fig. 3 The consensus process

3.5 Storage layer

Considering that the application scenarios of our electronic voting system are mostly relatively credible and closed environments such as student union election or scientific research institutions, we prefer consortium chain to public chain. In proposed protocol, Hyperledge Fabric is adopted as the underlying storage model. Important data in voting system such as ballots are stored in blockchain. In the example of the student union election, each institute runs a full blockchain node and has a copy of ballots to avoid likelihood of the single point failure. Ballots recorded in Fabric is immutable and transparent, this characteristic makes voters trust the voting protocol and rules out third parties like administers.

4 Smart contract-based authority management mechanism

In this section, we describe how to realize access control using proposed smart contract-based mechanism. Referring to Fig. 4, we take students union election as an example to enumerate the entities involved in. There exist five entities in our system: Voting Initiator, Students, Candidates, Proxy nodes, Supervisory roles. Different types of entities are endowed with varying degrees of authority.

Voting Initiator: Voting Initiator sets voting system parameters, such as start voting time T$start$ and end time Tend. Besides that, he has to generate system public key (spk) and private key (ssk) and deploy chaincode on Fabric.

Students: Students as voters are the main body of the system. Each student generates his own public key and private key. People who want to have access to voting system should send identity information to smart contracts to verify whether has eligible identity.

Candidates: Candidates are defined as Ci. The entity candidate can be considered as special voter for he also has the right to vote. The registration process is the same as above.

Proxy nodes: The role of proxy nodes is to collect and verify students' ballots. In the student union election scenario, Proxy nodes are different institutes. Institutes collect students' ballots and participate in consensus network.

Supervisory role: Verify that whether ballots submitted by institutes are legal and eventually write them into the blockchain. Supervisory role is Political Affairs Office in this case.

Certificate authority (CA) is used in traditional voting system to realize preliminary authority management. Certificate authority checks the validity of registration information and distributes admission certificates if applicants are qualified. However, for the data in the system, all participants who obtained the certificates have the same usage rights. This increases the risk of information leakage.

In the proposed mechanism, candidates as data owners, their personal information should be protected. It means only the one who meets the requirements can access to the information. The validation process is carried out by smart contracts. Candidates employ asymmetric encryption algorithm to encrypt their shared data. Public and private keys (PK, SK) are generated locally and the shared data is encrypted with the public keys. Private keys are stored in smart contracts. Once voters satisfy corresponding access control policies defined by candidates, the smart contracts will execute automatically to distribute private keys to voters. With private keys, students can access to candidates' information and generate their ballots. This implements the first level of access control, namely access control between candidates and voters. Related contracts are shown below.

```
func (t *VoteChaincode) verifyVoter(stub shim.
    ChaincodeStubInterface, args []string) peer.Response{
  var voteEventIterator shim.StateQueryIteratorInterface
  var error error
  if checkVoter(args) == 1 {
    voteEventIterator, error = stub.GetStateByPartialCompositeKey(
        voteCandidateTable, []string{args[0]})
    if error != nil {
      return shim.Error(”invalid voter identity!”)
    }
  }
  voteevent, error := getListResult(voteEventIterator)
  if error != nil {
    return shim.Error(”getListResult faile”)
  }
  fmt.Println(string(voteevent))
  return shim.Success(nil)
}
```

We fulfill the second-level of access control by introducing proxy nodes. In order to ensure that ballots in the blockchain are valid, and improve system throughput and efficiency, students don't have the right to participate in consensus and write data into the blockchain. When a student registers in the website, student Ui generates independent public and private key pair with local RSA tool (*pki*, *ski*), and logins in voting system to upload the public key *pki*. Smart contract receives the public key *pki* uploaded by the voter and generates a public key set *S*, while

$$S = (pk1, pk2, pk3, ..., pkn) \quad (1)$$

Besides that, smart contract calls the public key ring generation algorithm *RingGenerate* to generate the public key ring *PKR*, while

$$PKR = RingGenerate(S) \quad (2)$$

Students *Ui* choose their candidates *Ci*, and generates original ballot *votei*. Then he utilizes the system public key *spk* and calls the encryption algorithm *ENC* to produce encrypted ballot *Vi*, and *Vi* = *ENC(spk, votei)*. Voter *Ui* obtains the public key ring *PKR* from the smart contract, and uses the ring signature algorithm *RS* to generate a ring signature ballot *RingVotei*, and

$$RingVotei = (Vi, RS(Vi, ski, PKR)) \quad (3)$$

After these processes are completed, the ballot is encrypted and sent to proxy nodes, in our case, namely institutes.

Institutes as proxy nodes receive the ballots sent by students, and call the management contract to verify whether the voter *Ui* is in the qualified information list, and if not, discard these ballots. Then they obtain the public key ring *PKR* in the smart contract and verify whether the received *RingVotei* is a signature generated by a member in the ring. The verification is passed, the proxy nodes receive *RingVotei* and participate in consensus network, and send the voting data to the supervisory role. Related contracts are shown below.

```go
func (t *VoteChaincode) linkedVerifier(stub shim.
    ChaincodeStubInterface, args []string) peer.Response{
  id , error := strconv.Atoi(args[1])
  if error != nil {
    fmt.Println("Conversion of string to integer failed")
  }
  loadfuzhu(stub, args[0], id-1)

  w := witness(publicKeyOriginal, int(id)-1, publicRSA.N)
  result := linkedVerifier(w, w1[int(id)-1])
  fmt.Println(result)
  return shim.Success(nil)
}

func (t *VoteChaincode) signatureVerifier(stub shim.
    ChaincodeStubInterface, args []string) peer.Response{
  numvotor, error := strconv.Atoi(args[1])
  if error != nil {
    fmt.Println("fail to convert")
  }
  loadfuzhu(stub, args[0],numvotor-1)
  result := verify_signature(args[1], numvotor - 1, acc, T1[
      numvotor-1], tag1[numvotor-1])
  fmt.Println(result)
  return shim.Success(nil)
}
```

Supervisory role is the last layer in our hierarchical authority management framework. The identity authentication of the supervisory role is also automatically verified by the smart contract. In the case of the student union election, the voting initiator will issue a certificate to the supervisory role before voting. The certificate contains the hash value

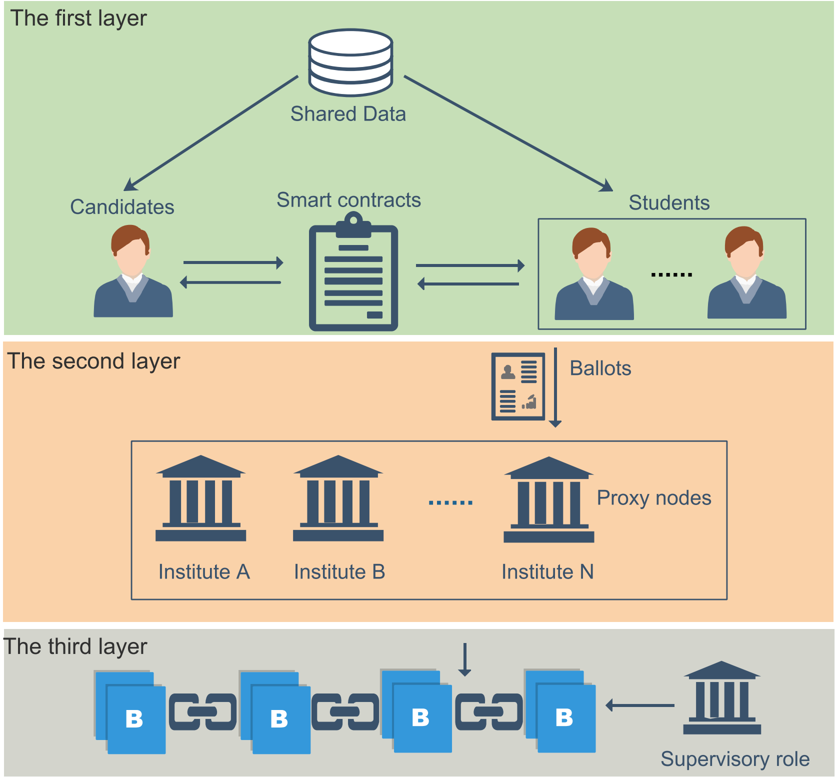


Fig. 4 The three-tier access cnotrol framework

of voting-related information. After entering the voting process, the supervisory role submits the certificate, and the smart contract verifies the hash value of the certificate to ensure the legality of the supervisory role. The supervisory role is independent of the voting process. The establishment of the supervisory role is to further ensure the legitimacy of the votes after the votes are collected. To avoid collusion attacks by proxy nodes during consensus, supervisory role (Political Affairs Office in this case) will call smart contracts to verify the ring signature of ballots which submitted by proxy nodes from the second layer. At the same time, in order to limit the power of the supervisory role, compared with other third-party agencies, the supervisory role only has the function to verify rather than changing the votes. If verification process fails, the vote event is aborted. After the verification is passed, the supervisory role writes the data into the blockchain and calls the ticket counting contract to count the votes.

```
func getListResult(resultsIterator shim.StateQueryIteratorInterface) ([]byte, error){

   defer resultsIterator.Close()
   var buffer bytes.Buffer
   bArrayMemberAlreadyWritten := false
   for resultsIterator.HasNext() {
      queryResponse, err := resultsIterator.Next()
      if err != nil {
         return nil, err
      }
      if bArrayMemberAlreadyWritten == true {
         buffer.WriteString(",")
      }

      buffer.WriteString(string(queryResponse.Value))
      bArrayMemberAlreadyWritten = true
   }
   return buffer.Bytes(), nil
}
```

```
func parseListResult(resultsIterator shim.StateQueryIteratorInterface) (int, string){
  votemaxticket := 0
  votemaxname   := ""
  defer resultsIterator.Close()
  var votecandidateint VoteCandidate

  for resultsIterator.HasNext() {
     queryResponse, err := resultsIterator.Next()
     if err != nil {
        break
     }
     err = json.Unmarshal(queryResponse.Value, &votecandidateint)
     if votemaxticket < votecandidateint.Candidateticket {
       votemaxticket = votecandidateint.Candidateticket
       votemaxname = votecandidateint.Candidatename
     }
  }
  return votemaxticket, votemaxname
}
```

During the period of the voting process, smart contracts play an important role. Firstly, smart contracts replace third-parties. All core functions such as identity verification and tallying ballots are carried out by them. Second, By establishing smart contracts between the candidates and the voters, it is guaranteed that only qualified voters can access the candidates' personal privacy, and achieving access control at the same time. In the end, introducing proxy nodes and supervisory roles, and limiting the usage rights of different roles to the blockchain through smart contracts to achieve the purpose of hierarchical management.

5 Evaluation

In this section, several experiments are conducted to test the performance of proposed algorithm and protocol. After analyzing the voting events, we conclude two crucial indices in our protocol: ring signature performance and tallying time. Therefore, we proposed following experiments to test system performance from these two aspects. In this section, some detailed information will be introduced.

5.1 System configuration

AMVchain is tested with four machines, and each machine runs a full node in Hyperledger Fabric. The machine has two 24-core Intel Xeon 8260 2.4GHz CPUs, with 128GB DRAM, and 7.2TB HDD. The operating system is CentOS 7.6.

5.2 Ring signature performance

We designed the process of the experiments according to the order of voting. First of all, we test the performance of the ring signature in voting phase. It can be seen from common sense that the smaller the number of voters, the more concise and efficient the signature process will be. The experiment sets up the signature size as 128bits. The result is shown in Fig. 5. As the user number ranges from 4 to 16384, the signing time vary from 27.13 ms to 34825.87 ms, while verifying time remains essentially unchanged. In other words, there exists a linear relationship between user number and signing time. Introduction of proxy nodes plays an important role under this circumstance, because it greatly reduces nodes involving in consensus. Besides that, considering application scenarios of our system are relatively less extensive like schools or research institutions, the experimental result shows that our system can meet the performance requirements.

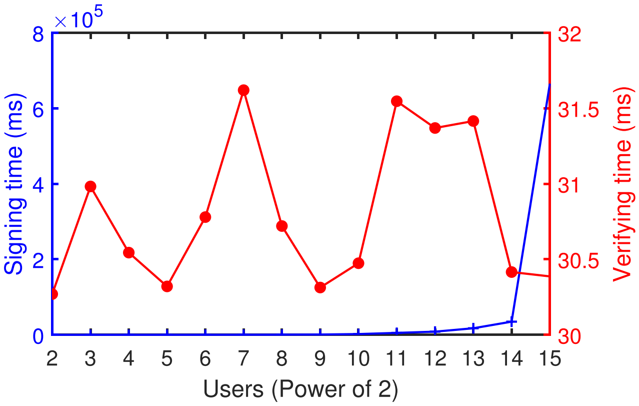


Fig. 5 The ring signature performance of AMVchain

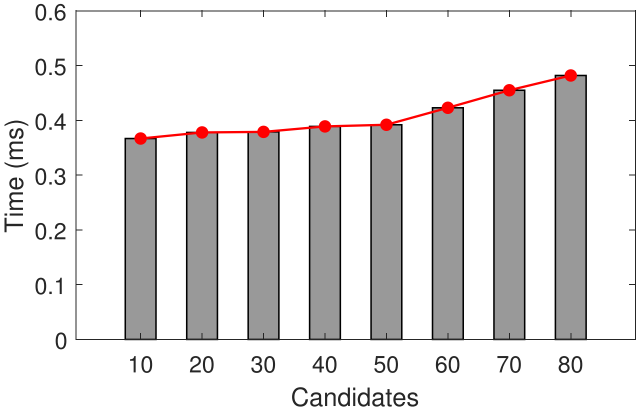


Fig. 6 The tallying performance of AMVchain

5.3 Tallying performance

In our system, tallying time is defined as the period between the voting deadline and the announcement of voting results. Tallying time is considered as the most important part of voting system because it is a process of interaction between voting system and users. System users don't want to spend a long time for feedback. Figure 6 depicts the relationship between candidates and tallying time. It can be concluded that as the number of candidates increases, the tallying time also raises, but the overall tallying time is in a reasonable range.

6 Conclusion

In this paper, we first analyze the shortcomings and challenges of existing blockchain-based voting system, and summarize relevant works for these drawbacks. Based on requirements of a qualified and efficient electronic voting system, we propose AMVchain, a blockchain-based voting system combined with hierarchical access cnotrol framework. Our voting protocol distinguishes from existing ones for we provide a three-tier authority management architecture and implement permission assignment through smart contracts automatically. By assigning different levels of authority to different entities in the system, the system protects the privacy of candidates and improves the concurrency and efficiency of the system. Furthermore, in order to cut off connections between ballots and voters, given the pseudo-anonymity of the blockchain, we introduced the linkable ring signature algorithm to encrypt the ballots. Experimental results show the feasibility and scalability of our system.

Acknowledgements This work is supported by Key-Area Research and Development Program of Guangdong Province under Grant NO. 2020B0101090005, and National Science Foundation of China under Grant No. 62072197.

References

1. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. http://bitcoin.org/bitcoin.pdf
2. Czepluch JS, Lollike NZ, Malone SO (2015) The use of block chain technology in different application domains. The IT University of Copenhagen, Copenhagen
3. Zhao Z, Chan T-HH (2015) How to vote privately using bitcoin. In: Proceedings of the international conference on information and communications security. Springer, pp 82–96
4. Lee K, James JI, Ejeta TG, Kim HJ (2016) Electronic voting service using block-chain. J Digit Forens Secur Law 11(2):8
5. Gailly N, Jovanovic P, Ford B, Lukasiewicz J, Gammar L (2018) Agora: bringing our voting systems into the 21st century
6. McCorry P, Shahandashti SF, Hao F (2017) A smart contract for boardroom voting with maximum voter privacy. In: Proceedings of the international conference on financial cryptography and data security. Springer, pp 357–375
7. Almeida RL, Ricci L, Camarinha-Matos LM (2019) votechain: community based scalable internet voting framework. In: Doctoral conference on computing electrical and industrial systems. Springer, pp 70–80
8. Yavuz E, Koç AK, Çabuk UC, Dalkılıç G (2018) Towards secure e-voting using ethereum blockchain. In: 2018 6th International symposium on digital forensic and security (ISDFS). IEEE, pp 1–7
9. Ayed AB (2017) A conceptual secure blockchain-based electronic voting system. Int J Netw Secur Appl 9(3):01–09
10. Hanifatunnisa R, Rahardjo B (2017) Blockchain based e-voting recording system design. In: Proceedings of the 11th international conference on telecommunication systems services and applications (TSSA). IEEE, pp 1–6
11. Fujioka A, Okamoto T, Ohta K (1992) A practical secret voting scheme for large scale elections. In: Proceedings of the international workshop on the theory and application of cryptographic techniques. Springer, pp 244–251
12. Hsiao J-H, Tso R, Chen C-M, Wu M-E (2017) Decentralized e-voting systems based on the blockchain technology. In: Advances in computer science and ubiquitous computing. Springer, pp 305–309
13. Chillotti I, Gama N, Georgieva M, Izabachène M (2016) A homomorphic lwe based e-voting scheme. In: Proceedings of the 7th post-quantum cryptography. Springer, pp 245–265
14. Novo O (2018) Scalable access management in iot using blockchain: a performance evaluation. IEEE Internet Things J 6(3):4694–4701
15. Pinno OJA, Gregio ARA, De Bona LC (2017) Controlchain: blockchain as a central enabler for access control authorizations in the iot. In: Proceedings of the GLOBECOM 2017–2017 IEEE global communications conference. IEEE, pp 1–6
16. Liu Y, Fan C, Liu H, Yang Q, Wu S (2018) Cross-layer cooperative multichannel medium access for internet of things. Peer-to-Peer Netw Appl 11(3):504–517
17. Grüner A, Mühle A, Gayvoronskaya T, Meinel C (2018) A quantifiable trust model for blockchain-based identity management. In: Proceedings of the 2018 IEEE international conference on Internet of Things (iThings) and IEEE Green computing and communications (GreenCom) and IEEE Cyber, physical and social computing (CPSCom) and IEEE Smart Data (SmartData). IEEE, pp 1475–1482
18. Azaria A, Ekblaw A, Vieira T, Lippman A (2016) Medrec: using blockchain for medical data access and permission management. In: Proceedings of the 2nd international conference on open and big data (OBD). IEEE, pp 25–30
19. Yu K, Tan L, Shang X, Huang J, Srivastava G, Chatterjee P (2020) Efficient and privacy-preserving medical research support platform against covid-19: a blockchain-based approach. IEEE Consumer Electron Mag
20. Guo H, Meamari E, Shen C-C (2019) Multi-authority attribute-based access control with smart contract. In: Proceedings of the 2019 international conference on blockchain technology, pp 6–11
21. Ouaddah A, Abou Elkalam A, Ait Ouahman A (2016) Fairaccess: a new blockchain-based access control framework for the internet of things. Secur Commun Netw 9(18):5943–5964
22. Huang J, Kong L, Chen G, Wu M-Y, Liu X, Zeng P (2019) Towards secure industrial iot: Blockchain system with credit-based consensus mechanism. IEEE Trans Ind Inf 15(6):3680–3689
23. Miao Y, Ma J, Liu X, Liu Z, Zhang J, Wei F (2018) Doas: efficient data owner authorized search over encrypted cloud data. Peer-to-Peer Netw Appl 11(3):349–360
24. Zheng H, Shao J, Wei G (2020) Attribute-based encryption with outsourced decryption in blockchain. Peer-to-Peer Netw Appl
25. Yu K, Tan L, Moayad A, Hekun Y, Jararweh Y (2021) Blockchain-enhanced data sharing with traceable and direct revocation in iiot. IEEE Trans Ind Inf

Chenchen Li received the B.S. degree from Zhongnan University of Economics and Law. He is currently working toward the PhD degree in the CGCL, Huazhong University of science and technology. His research interests include blockchain, data mining and finance security.

Jiang Xiao is currently an associate professor in School of Computer Science and Technology at Huazhong University of Science and Technology (HUST), Wuhan, China. Jiang received the BSc degree from HUST in 2009 and the PhD degree from Hong Kong University of Science and Technology (HKUST) in 2014. She has been engaged in research on blockchain, distributed computing, big data analysis and management, and wireless indoor localization.

Xiaohai Dai received the M.S degree in School of Computer Science and Technology from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2017. He is currently pursuing the Ph.D degree in School of Computer Science and Technology from HUST. His current research interests include blockchain and distributed system.

Hai Jin is a Cheung Kung Scholars Chair Professor of computer science and engineering at Huazhong University of Science and Technology (HUST) in China. He was awarded Excellent Youth Award from the National Science Foundation of China in 2001. Jin is a fellow of the IEEE and a member of the ACM. His research interests include computer architecture, virtualization technology, cluster computing and cloud computing, peer-to-peer computing, network storage, and network security.