IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 16, 2020, accepted March 3, 2020, date of publication March 10, 2020, date of current version March 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979746

Token-Based Access Control

GUOHUA GAN, E. CHEN, ZHIYUAN ZHOU, AND YAN ZHU
School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

Corresponding author: Yan Zhu (zhuyan@ustb.edu.cn)

This work was supported in part by the National Key Technologies R&D Programs of China under Grant 2018YFB1402702 and in part by the National Natural Science Foundation of China under Grant 61972032.

ABSTRACT Traditional centralized access control has some shortcomings in robustness, trustworthiness and circulation. Blockchains have the advantages of fault tolerance and trust. Smart contracts have the characteristics of automatic execution and flexible expansion. Tokens can well record credential information and transfer easily. In this paper, blockchain, smart contract and token are integrated and applied to access control to solve the shortcomings of traditional access control. First, access control, blockchain, smart contract and token are briefly described. Second, this paper proposes a solution by giving the general data structure of access control token, elaborating the equivalence, split, merge and verification algorithms of access control token, and explaining the system architecture of token-based access control. Last, this paper uses a token-based access control simulation system to verify that token-based access control has certain comparative advantages in robustness, trustworthiness, circulation, concurrency and so on.

INDEX TERMS Access control, distributed control, distributed computing, blockchain, smart contract, token.

I. INTRODUCTION

With the continuous development of distributed applications, some shortcomings of traditional centralized access control (AC) have gradually been exposed. These problems or requirements include the following aspects.

(1) Robustness. The centralized access control system is in charge of the access privileges of the whole system, which is easy to become the bottleneck of the whole system. Once attacked or failed, the system will not provide services normally [1], [2]
(2) Trust. In the centralized access control system, the administrator has an absolute advantage over the user. For the pre-determined authority, the administrator has a larger authority to modify or cancel, resulting in the loss of authority to the user, causing the authority trust crisis [1], [2].
(3) Circulation. In distributed application networks, it is necessary to support the transfer of access control. For example, electronic tickets can be a kind of access control that can be transferred, and the business rights in work can also be a transfer of access control [3].

Considering the problems and requirements of access control in distributed application networks, many solutions have been proposed and practiced. Some papers [4]–[7] studied and combined distributed access control from the cross-domain and multidomain perspective. Another papers [8], [9] introduced rules into access control, which improve the flexibility and dissemination of distributed access control. One common feature of these methods is the use of traditional distributed technology and access control mechanisms to implement distributed access control, which provides a reference for the follow-up study of other forms of distributed access control technology.

Blockchain (BC) [10] technology, which was born in bitcoin [11], has been widely studied and applied in various fields. It has been widely recognized that blockchains have the advantages of decentralization, antimodification, fault tolerance and antirepudiation [11]–[13]. Many papers [14]–[17] had focused on the combination of blockchain technology and access control, which that can fully utilize the characteristics of blockchains, and can better solve the problems faced by distributed access control. Liu Ao-Di et al, aiming at the distributed data sharing scenario, used blockchain technology to realize access control of distributed databases, which enabled big data to serve society at a higher level and in a wider range [1]. SHIJin-Shan et al, aiming at the application scenario of the Internet of Things, used blockchain technology to realize the access control of the nodes of the Internet of Things, which can meet the needs of massive access control and dynamic expansion of access control [2]. Maesa DDF et al provided a general description of the method of generating, updating and transferring authority based on blockchain technology [3]. Compared with traditional centralized access

The associate editor coordinating the review of this manuscript and approving it for publication was Francesco Tedesco.

control, these studies and applications have made great progress and are more flexible and credible than general distributed access control. However, in a sense, these methods only regard blockchains as a storage carrier of access control, and support for the management and circulation of access control credentials can be further enhanced. Yuanyu Z et al proposed access control based on a smart contract (SC) to manage the access rights of IOT [18]. However, the mapping between smart contracts and actual applications needs to be better explained. In addition, the application of rules is insufficient, which makes the setting of access control not formal enough. Khaled A et al proposed access control to be recorded and transmitted in the form of a token, which improved the flexibility of transfer [19]. However, the token was only a traditional token, which runs between traditional distributed systems. The problems of security, concurrency and circulation are still not well solved.

This paper unifies blockchains, smart contracts and tokens, takes the blockchain as a trusted and safe storage and operation environment, and fully utilized the characteristics of automatic enforcement of smart contracts and efficient intelligent circulation of tokens and presents a token-based access control (TBAC). The solution of TBAC aims to achieve the following goals: 1) distributed and trusted access management mode, 2) flexible and efficient access transfer mode, and 3) a high-frequency fault-tolerant access execution mechanism.

Next, in Section 2, this paper describes the basic concepts and related theories of access control, blockchain, smart contract and token. In Section 3, the data structure and some logic algorithms of access control token(ACT) are given, and the system framework of TBAC is further elaborated. In Section 4, the TBAC simulation system is given and tested. Finally, it is summarized in Section 5.

II. FOUNDATION OF RESEARCH

A. ACCESS CONTROL

Access control technology is one of the core technologies of information security. Its function is to ensure that the correct subject (S) performs the correct operation (A) on the correct object under the correct environment (E). With the development of technology applications, discretionary access control (DAC), mandatory access control (MAC), role-based access control (RBAC) and attribute-based access control (ABAC) have appeared successively[20].

Attribute-based access control is one of the most important access control models. The model has the characteristics of flexibility, scalability and fine-grained [21], [22]. Whether in traditional centralized access control or distributed access control, attribute-based access control has been widely studied and applied. In this model, attributes are used to describe rules and policies. As the basis of logical judgment and execution of access control, XACML and other languages are used to formalize rules and policies so that all systems can understand and implement them in a unified way [23]–[25]. Referring to the research results of the previous reference, this paper formally describes access control as a five-tuple (S, E, O, A, R), shown in Formula 1.

$$\begin{cases} AC = (S, E, O, A, R) \\ R = f\ (rule_1, rule_2, \ldots, rule_m) \\ rule = (condition, action) \rightarrow result \\ result \in \{permit, deny\} \\ action \in A \\ condition = condition | condition \cap condition) \\ condition = (e.attribute \Delta value) | (e.attribute \Delta e.attribute) \\ e \in \{S, E, O\} \end{cases} \tag{1}$$

The explanation of Formula 1 is as follows.

(1) S, E, O and A represent the set of subject, environment, object and operation, respectively. R is a logical combination of one or more rules.

(2) Each rule is expressed as the result of executing an action under a certain condition. The value of the result can be permitted, denied or unknown. However, to ensure every decision has a definite result, conflict handling rules must be set beforehand; when set as permit first, regard unknowns as permits, and deny first when set as deny first.

(3) Conditions include single condition and compound condition. There are two descriptive forms of a single condition: one is the comparison of the attribute of an element with a specific value, and the other is the comparison of the attribute between elements, where elements belong to the subject, object and environment. Compound condition consists of two or more single conditions.

B. BLOCKCHAIN

Different references have different definitions and descriptions of blockchains, but their connotations are basically the same. This paper argues that the blockchain is a multiparticipation, distributed, multiple ledger technology or database technology. From the point of view of the software level, the blockchain can be divided into a data layer, network layer, consensus layer, incentive layer, contract layer and application layer [11], [26]. From the point of view of technology, the blockchain includes P2P network technology, distributed ledger technology, asymmetric encryption technology, consensus mechanism technology and smart contract technology [13]. From the point of view of the application mode or permission scope, blockchains can be public chains, alliance chains and private chains [27].

Blockchain technology is a multitechnology integration technology system. People enrich and enhance blockchain technology from various perspectives, such as lightning network technology to improve the performance of blockchains and cross-chain technology to realize the interaction between different blockchains and zero knowledge proof technology to protect privacy. All these technologies are introduced to increase the characteristics of blockchain, such as decentralization, traceability, tamper proof, high reliability and high availability [26].

With the continuous research and application practice of blockchain technology, the characteristics of blockchain will become more and more prominent. Compared with other distributed systems, the obvious advantages of blockchain will provide a more solid technical basis for building a secure and efficient access control system.

C. SMART CONTRACTS

Smart contract was originally proposed by Szabo [28]. A smart contract is a set of commitments defined in digital form, including agreements on which contract participants can implement these commitments. A set of commitments refers to rights and obligations agreed upon (often mutually) by contract participants. The original intention is to build smart contracts into physical entities to create flexible and controllable smart assets. However, due to the limitation of early computing conditions and the lack of application scenarios, smart contracts were a wide concern of researchers until the emergence of blockchain technology, smart contracts were restudied and applied.

With the deepening integration of blockchains and smart contracts, people's understanding of smart contracts has also been innovative. A smart contract is considered the key symbol of blockchain 2.0 [29]. As early as in bitcoin, its script has been the rudiment of smart contracts. Due to the limited expressive ability of scripting language, which only contains some basic arithmetic, logic and encryption operations, bitcoin scripts can only provide a prototype for blockchain programmability and open up ideas for the subsequent development of other blockchains. Ethereum contributes greatly to the development of smart contracts. It integrates and improves the concepts of scripts, competitive currency and meta-protocol, enabling developers to create arbitrary consensus-based, scalable, standardized, well-featured, easy-to-develop and collaborative applications. Hyperledger promotes the smart contract to another high level and uses the chaincode to refer to the smart contract, which has stronger expansibility and flexibility. Fabric, the star project of Hyperledger, has attracted the most attention for its contribution to smart contracts. Fabric divides chain codes into system chain codes and application chain codes. System chain codes are used to process system-level transactions, such as life cycle management and policy configuration. This approach is very advantageous for application development.

In general, we can define blockchain smart contracts as smart processors executed in accordance with rules written in a language or script considering the existing research on blockchain smart contracts [1], [29], [30], [31]. Figure 1 depicts the smart contracts model.

Figure 1 shows that smart contracts exist on blockchains, consisting of rules (R), states (ST), triggered by input (IN) data or events, giving output (OUT) data or execution instructions. To facilitate the academic research and application implementation of smart contracts, the model in Figure 1 is formalized, as shown in Formula 2. Formula 2 shows that the smart contract is described as a quaternion

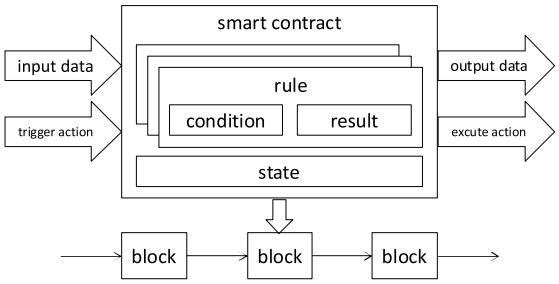


FIGURE 1. Smart contract model.

(IN, ST, R, OUT). Both IN and OUT can be data described by attributes of elements (e) or an operation instruction (action). ST represents the state of the current contract, which is described by attributes of elements. R represents the core logic of the contract, which is described by rules. The meaning is the same as R in the access control described earlier.

$$\begin{cases} SC = (IN, ST, OP, OUT) \\ IN = (in_1, in_2 \ldots in_n) \\ in = (e.attribute \Delta value) | action \\ ST = (state_1, state_2 .. state_n) \\ state = e.attribute \Delta value \\ OUT = (out_1, out_2 \ldots out_n) \\ out = (e.attribute \Delta value) action \\ OP = f\ (rule_1, rule_2 \ldots rule_n) \\ rule = (condition, action) \rightarrow result \\ result \in \{permit, deny\} \\ condition = condition | (condition \cap condition) \\ condition = (e.attribute \Delta value) | (e.attribure \Delta e.attribute) \\ e \in \{S, E, O\} \\ action \in A \end{cases} \tag{2}$$

The combination of smart contracts and blockchains greatly improves the programmability of blockchains and the trust of smart contracts and enables both to achieve important expansion ability, which opens another door for the application and development of DLT technology. Many applications that can only be achieved in centralized systems can also be achieved in blockchains and have characteristics that centralized systems do not have.

D. TOKENS

From ancient times to today, tokens have been used to refer to different objects and have played different roles in many application scenarios. In ancient military affairs, many kinds of physical tokens were often used to represent different functions, such as tokens representing the right to deploy troops, and tokens representing the right to pass through customs. In religious affairs, tokens are often used to make magic instruments, such as the swords used in sacrifice, the ruler used in lectures and so on. In modern science and technology, tokens are often used to refer to a data body in the information

network to control network transmission or system access, such as token rings and token buses. These different tokens have something in common, which is as a kind of evidence, representing a kind of rights and interests. In fact, there are many objects with this characteristic in modern times, such as labels, tickets, and eCards.

The diversity of token applications makes it difficult for tokens to form a unified definition in both natural language and formalization. This paper holds that tokens are recognized in a certain range, issued and circulated in accordance with a certain mechanism, and represent the evidence of certain rights and interests. Research on existing token systems, whether in the real world or in the network world, faces the problems of trust and dissemination. The question of trust refers to how to ensure that a token is trusted to a certain extent and not to give any chance to the relevant parties to use tokens to do evil, such as access control tokens, which authorizers may abuse, and users may forge. The issue of communication refers to how to promote tokens to be used flexibly by more people within the scope of the rules, such as token rings, and how to support more participants to join.

Chinese scholar Yan [32] creatively endowed tokens with a new environment on the basis of researching blockchains and smart contracts, thus promoting tokens to be better applied in digital society and real society, especially in economic fields. According to Meng Yan and others, although tokens are not necessarily related to blockchains, issuing and transferring tokens based on blockchains and smart contracts will help improve a token's security, trust, parallelism and dissemination. Based on blockchains and smart contracts, tokens can be improved and applied in many application scenarios, such as digital currency is a typical token, as well as shopping vouchers, electronic tickets, electronic invoices, and electronic cards. Assets issued through various ERC standards in ETF can also be considered different types of tokens. All these tokens can exist in a certain data structure, record corresponding information, execute corresponding processing logic, and obtain the established processing effect.

Tokens are an earlier concept than blockchains and smart contracts. With the help of distributed blockchains, tamper-proof and nonrepudiation, and automatic execution of smart contracts, tokens can play a more credible, more open and faster role. All the subsequent notices mentioned in this paper refer to the notices based on blockchains and smart contracts.

III. TBAC SOLUTION

Applying tokens to access control, using blockchain networks as storage, authentication and circulation environments, using smart contract issuance and application as a logical processing form, using tokens as data bodies, recording the rules and logic of access control, executing authentication processing of access control, and ensuring correct objects are accessed by correct subject correctly is an access control mode called token-based access control.

The access control system based on the TBAC mode needs to realize three basic processing functions: creation, transfer and revocation of access control tokens, and can verify and test access control tokens to ensure the correctness of access control tokens. In the transfer process of access control circulars, there may be the need to split or merge, which requires that the equivalence be guaranteed before and after to ensure the consistency of authority and verify the test. TBAC has no limitation on which consensus mechanism to adopt.

TABLE 1. Data structure of act.

```
<ACT id="hlscHOsdE027798744229jkhsl" name="ACTDEMO">
    <ACTT id="aslkdjg8796978ALSDHG98">
    <Issuer id="ssfdgKSD29se36slkdjfalg"/>
    <Fungible value="yes|no"/>
    <Divisible value="yes|no"/>
    <Transferable value="yes|no"/>
    <Rights>
        <Rule action="read" result="permit|deny" >
            <Condition attribute="age" compare=">" value="12"/>
        </Rule>
        <Rule action="write" result="permit|deny">
            <Condition attribute="age" compare=">" value="18"/>
            <Condition attribute="role" compare="=" value="manager"/>
        </Rule>
    </Rights>
</ACT>
```

A. ACCESS CONTROL TOKEN

This paper applies tokens that are based on blockchains and smart contracts to access control fields as the carrier of recording the logic of access control rules and are processed and executed by the blockchain-based access control processing engine. This token is called ACT in this paper.

Based on the data structure of many asset tokens and some requirements that access control may face, this paper presents the general data structure of ACT, as shown in Table 1. To simplify the formulation of uniform rules, the concept of the ACT template(ACTT) is proposed. All that ACTs referring to the same ACTT need to abide by the rules set in ACTT.

In Table 1, each element is described as follows.

(1) *ID* is a string used to identify the ACT, which can be calculated by hashing some source data according to some algorithms and has uniqueness in the whole chain.
(2) *Name* is used to set the name of the ACT, which is different from other ACTs mainly considering the easy identification of the application.
(3) *ACTT ID* is used to identify the IDs of ACTT applied by ACT. Similar to the ACT id, it can also be calculated by hashing some source data according to some algorithm, and it has uniqueness in the whole chain.
(4) *Issuer* represents the issuer of the ACT, which needs to record at least the issuer's ID information, which can be a string after a hash operation.There can be multiple IDs separated by commas to indicate that this ACT is commonly issued by multiple issuers.
(5) *Fungible* indicates whether the ACT can be replaced, and yes means that it can be replaced. For example,

in some scenic spots, two tickets can be replaced indiscriminately; and no means that it cannot be replaced. For example, each lottery is different, representing a specific right to honor prizes, which cannot be replaced by each other.

(6) *Divisible* indicates whether the ACT is separable, and yes denotes separable. For example, the total broadcast right of a TV play can be divided into one broadcast right for each episode, while no denotes nonseparable, such as train tickets. Once a train ticket is generated, it cannot be divided into multiple train tickets. Whether the ACT can be split depends on the specific application scenario.

(7) *Transferable* means whether the ACT is transferable; value yes means transferable, such as shopping vouchers, can be transferred from one holder to another holder, and value no means nontransferable, such as work cards, once generated and issued, they are nontransferable and can only be used by the original holder.

(8) *Rights* are an important element of access control communication, which are used to record access control rules. The definition description of rule is basically the same as the definition description of rule in AC and SC. For the combination of multiple rules and the combination of conditions within a rule, either an OR operation or an AND operation can be used. It can be proved that the AND relationship between rules can evolve into the AND relationship between conditions, and the OR relationship between conditions can evolve into the OR relationship between multiple rules on the basis of the consistency of action and result. Therefore, in this paper, to simplify, it is necessary to directly restrict multiple rules to only be OR relationship records. In the same rule, multiple conditions can only be AND relationship records, and the rules of the same action and result must be merged into one rule. According to the number of rules and conditions recorded in the rights, rights can be divided into three forms: single-condition single rule, multicondition single rule and multirule.

In addition to the ACTT name, there is no ACTT element in the data structure of ACTT; there is also a priority element, which is the same as the data structure of ACT. The value of priority is a permit for permission priority and denial for prohibition priority. In case of conflict between ACT and ACTT, the ACTT shall prevail.

B. EQUIVALENT JUDGEMENT OF THE ACT

The access control system must ensure that its managed permissions can be correctly set, transferred, determined and executed. ACT needs to ensure equivalent transformation and processing throughout its lifecycle. It can neither expand permissions nor reduce permissions. Considering the ACT data structure mentioned above, the algorithm of ACT equivalence determination is given as follows.

Algorithm 1 realizes the judgement of ACT equivalence, which embodies the following conditions.

Algorithm 1 Equivalent Judgement Algorithm of Act

```
INPUT: ACT1, ACT2
OUTPUT: result of comparison
Begin
Step1: compare base attribute
  If(ACT1.base attribute == ACT2.base attribute)
    goto step2
  Else
    Return ACT_IS_NOT_SAME_KIND
endStep
Step2: compare right
  Step2.1: compare ACT1 rule
    For(rule(i) in ACT1)
      Flag[i] = false
     If(exist rule in ACT2 && rule == rule(i))
       Flag[i] = true
    Endfor
    If(every flag[i] == true)
       goto step2.2
    Else
       Return ACT_IS_SAME_KIND_NOT_EQUAL
   endStep
  Step2.2: compare ACT2 rule
    For(rule(j) in ACT2)
      Flag[j] = false
     If(exist rule in ACT1 && rule == rule(j))
       Flag[j] = true
    Endfor
    If(every flag[j] == true)
     Return ACT_IS_EQUAL
   Else
    Return ACT_IS_SAME_KIND_NOT_EQUAL
  endStep
  endStep
End
```

(1) The basic attribute elements of ACT1 are the same as those of ACT2; that is, ACT1's ACTT, issuer, fungible, divisible, transferable and other elements are the same as those of ACT2.

(2) For each rule of ACT1, there is a rule equivalent to it in ACT2.

(3) For each rule of ACT2, there is a rule in ACT1 that is equivalent to it.

ACT1 and ACT2 are not equivalent in two ways. One is that the basic attribute elements are the same, but the rule is not equal. It is called the same kind of unequal ACT. The other is that the basic attribute elements are different. It is called different kinds of ACT. Different kinds of ACT are naturally not equal.

The equivalent judgement of ACT plays an important role in the separation, merger and circulation of the ACT. Specific application on this basis extends to support other aspects of equivalence judgement to ensure that the permission management of the TBAC system runs in a correctly controlled state.

Algorithm 2 Merge Algorithm of Act

```
INPUT: ACT2, ACT3
OUTPUT: ACT1
Begin
Step1: compare base attribute
If(ACT2.base attribute == ACT3.base attribute)
    ACT1.base attribute = ACT2.base attribute
    ACT1.Right = ACT2.Right
    goto step2
  Else
    Return null
endStep
Step2: merge right
   For(rule(i) in ACT3)
   If(exist rule in ACT1 && rule.action == rule(i).action
&& rule.result == rule(i).result )
      rule.addCondition(rule(i).getCondition())
   Else
      ACT1.addRule(rule(i))
   Endfor
endStep
Return ACT1
end
```

Algorithm 3 Splitting Act Algorithm

```
INPUT: ACT1, ACT2
OUTPUT: ACT3
Begin
Step1: compare base attribute
If(ACT1.base attribute == ACT2.base attribute)
     ACT1.base attribute = ACT2.base attribute
     ACT1.Right = ACT2.Right
      goto step2
   Else
     Return null
endStep
Step2: compare rule
   For(rule(i) in ACT2)
    If(exist rule in ACT1 && rule == rule(i))
     Continue
    Else
     Return null
   Endfor
   goto step3
endStep
Step3: split ACT
   ACT3.base attribute = ACT1.base attribute
    For(rule(j) int ACT1)
     If(exist rule in ACT2 && rule(j) == rule)
      ACT3.addRule(rule(j) - rule)
    Else
      ACT3.addRule(rule(j))
   EndFor
 endStep
 Return ACT3
end
```

C. SPLIT AND MERGE OF ACT

In practical applications, it is often necessary to split an ACT into two or more ACTs or to merge multiple ACTs into one ACT. For example, a user has both A and B roles at the same time. When deciding the user's rights, it is necessary to merge the A role's rights with the B role's rights to decide. When the permissions of a role C are changed to be assumed separately by role D and role E, the role permissions of C need to be split. Assuming that there are three access controls token ACT1, ACT2 and ACT3, ACT1 is split into ACT2 and ACT3, or ACT2 and ACT3 are merged into ACT1, then merge ACT Algorithm 2 as follows.

Algorithm 2 realizes the merging of two ACTs, which embodies the following characteristics.

(1) Only similar ACTs can be merged; that is, two ACTs have the same ACTT, issuer, fungible, divisible, transferable and other elements.
(2) After the merger, the rule on the right also maintains the OR relationship, and the condition in rule also maintains the AND relationship.
(3) Neither expands nor reduces the powers of the merged ACT to ensure its equivalence.

The splitting ACT Algorithm 3 is as follows.

Algorithm 3 achieves ACT3 after splitting ACT2 from ACT1, which embodies the following characteristics.

(1) Only the same kind of ACT can be split; that is, two ACTs have the same ACTT, issuer, fungible, divisible, and transferable.
(2) Only if the right of ACT1 contains the right of ACT2 can ACT2 be separated from ACT1 and ACT3 be obtained.
(3) It neither enlarges nor reduces the power of ACT after splitting to ensure its equivalence before and after splitting.

D. CREATION AND REVOCATION OF ACT

The creation and revocation determine the beginning and termination of the lifecycle of the ACT. The issuer field clearly records the issuer's identity, which establishes the basis for authenticating all subsequent operations of access control. Because blockchains are difficult to modify and delete, it is not advisable to attempt to revoke privileges by deleting blockchain data. Here, by means of the transfer of the ACT, the revocation of the ACT can be realized. In the TBAC system, one or more recycling addresses are set for each type of ACT. These addresses are owned by the issuer, and any ACTs transferred to the address are considered to be nonreusable ACT; thus, the revocation function of ACT is implemented.

E. TRANSFER OF ACT

Transfer of the ACT refers to the process of transferring the ACT from one owner to another. Transferable is used to determine whether an ACT can be transferred. There are

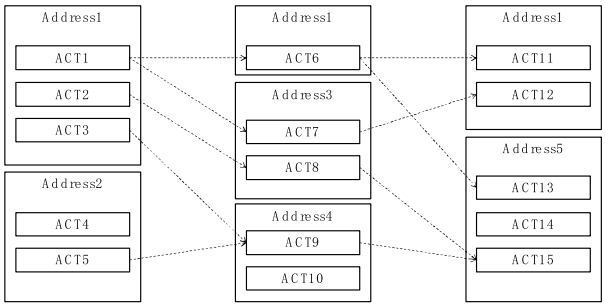


FIGURE 2. Transaction and transfer of the ACT.

three situations of transferring the ACT: (1) complete transfer, (2) split transfer, and (3) merger transfer. For 2 and 3, the principle of equivalence of transferring the ACT should be followed. Figure 2 depicts the principle of transaction transfer of the ACT.

The transfers of ACT2 to ACT8 and ACT7 to ACT12 belong to complete transfer, the transfers of ACT1 to ACT6 and ACT7 and ACT6 to ACT11 and ACT13 belong to split transfer, and the transfers of ACT3 and ACT5 to ACT9 and ACT8 and ACT9 to ACT15 belong to merger transfer. Since the transfer process is recorded on the blockchain, it is only necessary to calculate the ACTs in which a user's address has not transferred out when calculating the ACT. Using the above method to record the transfer process of access control notarization can clearly trace its origin. This method fully exploits the characteristics of the blockchain traceability and is conducive to tracking and monitoring the transfer of authority. In addition to the fact that the complete transfer does not need to create a new ACT, the split transfer and merge transfer need to create a new ACT according to the split and merge rules, The original ACT transfer directly to the recycling address for cancellation.

F. VERIFICATION OF ACT

Similar to general access control, the ACT also needs to verify and test the access control logic; otherwise, it will bring serious security threats to the application system. Hu Kail proposed a more general verification method, which is formalized into smart contracts, to verify whether contracts satisfy expectations by formal specification and formal verification and provide a solution for smart contract verification [33]. In this paper, the rule representation and computational processing logic consistent with access control and smart contracts are adopted. Therefore, for the verification test of the ACT, we can draw lessons from the verification test methods of access control and smart contracts. There are three aspects to verify and test the ACT. First, to verify whether the internal rules of the ACT are contradictory, and then to judge the contradictory rules according to the allowable or prohibited priority rules. Second, to verify the boundary value of the ACT and to test whether the ACT is correctly set to implement the access control in accordance with the boundary value. Third, the contradiction of ACT and ACTT also needs to be tested as to whether the ACT violates the rules of the ACTT.

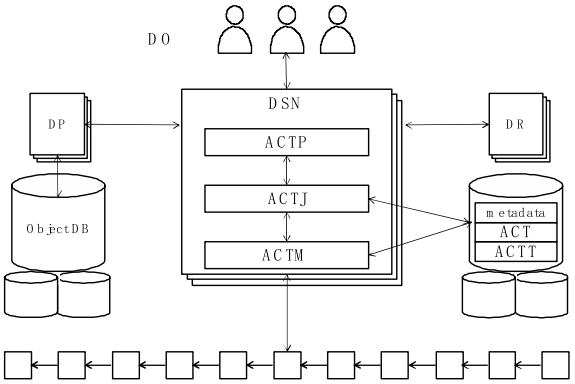


FIGURE 3. TBAC system architechure.

According to the need, we can also expand the validation test of other aspects of the ACT.

G. TBAC SYSTEM ARCHITECTURE

Distributed data sharing is a typical distributed application system. TBAC-based distributed data sharing technology institutions are shown in Figure 3.

In a TBAC-based distributed data sharing architecture, there are mainly the following objects.

(1) *Data Provider* (DP). As the supply side of a data sharing network, data providers can be data producers, managers, operators, etc. Data can be classified into various types, such as structured data, unstructured data, raw data and secondary processed data. With the 5G era approaching, increasing data are produced and stored in a distributed state in multiple object databases, which can be either a proprietary database or a large shared data center. To provide data services to more demanders, in this scheme, many data providers register their data on the blockchain through data service nodes to facilitate the searching and acquisition of demanders.

(2) *Data Requester* (DR). As the demand side of a data sharing network, the data demand side can be individuals, enterprises, institutions and so on. Data demanders often need to obtain data from multiple data providers. They can search directly from the blockchain through the data service nodes or register their own needs on the blockchain to facilitate data providers to supply data on demand. Data demanders can only request data access from data providers through data service nodes on the basis of ACT issued by the data providers and authorized by the data owner.

(3) *Data Owner*(DO). As the subject of data authentication in a data sharing network, data authentication mainly determines the subject of data ownership. Without the permission of the subject of data ownership, other roles of data cannot illegally process data. The ownership of data may be a single and exclusive entity or a multishared entity. The same entity may also have a large number of data distributed in different databases. Therefore, in this scheme, data authentication is also recorded on the blockchain to achieve distributed authorization.In reality,

DP and DO may be the same subject or separate subject. The difference lies in whether the data is managed and provided by DO itself or by DP on behalf of it. Both DP and DO may be independent or co publishers of ACT.

(4) *Data Service Node*(DSN). The data service node is a blockchain node that desplayed with TBAC program. The creation, revocation, transfer and process of the ACT can be realized by TBAC program and blockchain program together. With the connection of the data provider and requester, the goal of distributed authorization and distributed process of privilege can be achieved. Data service nodes can have many distributed deployments in the network, and data providers and data requesters can flexibly choose different data service nodes to provide or obtain data services.

(5) *Blockchain*. As a distributed data sharing data service bus (DSB: data service bus), the blockchain is not only the storage carrier of the ACT but also the execution environment of the ACT. Its characteristics will help to form an open access control mechanism and break the security bottleneck, concurrency bottleneck and flow bottleneck of traditional access control mechanism.

Distributed data sharing based on the TBAC is mainly realized through the DSN. On this node, not only the general blockchain node service functions but also the basic functions of access control are needed. These functions include management, judgment, and process of ACT.

(1) *ACT Management:* (ACTM) mainly realizes the functions of creation, revocation and transfer of ACT. To realize the management of the ACT more conveniently, in addition to establishing an ACT database, we also need to create and manage metadata databases and ACT template databases. Metadata are the basis for the correct understanding and unification of semantics of the whole system. Any licensed node application can create the basic information of management access control according to need through this module and store the corresponding key information and raw data in a hash on the blockchain. The ACT template is designed to facilitate the creation of ACT and unify the management of a similar ACT.Metadata, ACT, ACT template and other original data can be saved directly on the blockchain or in the distributed database off chain. Considering that too much data can not be saved on the chain, the distributed database under the chain is preferred.

(2) *ACT Judgment:* (ACTJ) obtains access requests from the ACT processor and calls relevant ACT information, metadata information and ACTT information according to access request, calculates logic according to access control rule, and finally gives access control decision results and returns them to ACT Processor.

(3) *ACT Processor:* (ACTP) is used to receive access requests from data requesters, call ACTJ to determine whether and how to start data access, and ultimately obtain permissible object data through the channels provided by data providers.

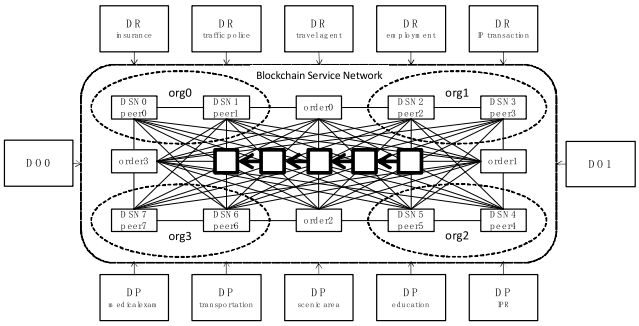


FIGURE 4. TBAC simulation system architecture.

TABLE 2. Deployment configration of BSN.

ORG	Domain Name	Hardware	Software
Order	Order0.tbactest.com	2Core, 8GRAM	Jdk1.8/Tomcat8
	Order1.tbactest.com	2Core, 8GRAM	Jdk1.8/Tomcat8
	Order2.tbactest.com	2Core, 8GRAM	Jdk1.8/Tomcat8
	Order3.tbactest.com	2Core, 8GRAM	Jdk1.8/Tomcat8
Org0	Peer0.org0.tbactest.com	2Core, 8GRAM	Jdk1.8/Tomcat8
	Peer1.org0.tbactest.com	2Core, 8GRAM	Jdk1.8/Tomcat8
Org1	Peer2.org1.tbactest.com	2Core, 8GRAM	Jdk1.8/Tomcat8
	Peer3.org1.tbactest.com	2Core, 8GRAM	Jdk1.8/Tomcat8
Org2	Peer4.org2.tbactest.com	2Core, 8GRAM	Jdk1.8/Tomcat8
	Peer5.org2.tbactest.com	2Core, 8GRAM	Jdk1.8/Tomcat8
Org3	Peer6.org3.tbactest.com	2Core, 8GRAM	Jdk1.8/Tomcat8
	Peer7.org3.tbactest.com	2Core, 8GRAM	Jdk1.8/Tomcat8

IV. TBAC EXPERIMENT SIMULATION

A. TBAC SIMULATION SYSTEM CONSTRUCTION

Many blockchain solutions can be used as the base chain of TBAC system, such as blockchains that modified from Bitcoin, Ethereumn or Fabric. Considering that access control itself is used in a certain range, Fabric is used as the base chain in this simulation system to build the TBAC simulation alliance chain system as shown in Figure 4.

This simulation system is composed of peripheral application system and internal blockchain service network (BSN). The application system includes 2 DOs, 5 DPs and 5 DRs. The blockchain service network is mainly composed of 4 order nodes and 8 peer nodes. Each 2 peer nodes constitute an organization, and each peer node is also a DSN. The deployment configuration of each node of the blockchain service network is shown in Table 2.

This simulation system contains several pairs of representative data providers and requesters, such as Medicalexam DP and Insurance DR on medical data, Transportation DP and Traffic police DR on traffic data, Scenic area DP and Travel agent DR on travel data, Education DP and Employment DR on education data, IPR DP and IP transaction DR on IPR data. As the owner of data, DO performs management operations on data, including data recorded on chain and ACT issue.

This simulation system deploys TBAC program together with peer node to form a DSN. Each DO, DP or DR can flexibly connect to any DSN to record data on chain. When the connected DSN cannot provide services, DO, DP or DR can automatically switch to other DSNs through the DSN SDK to ensure that DO, DP or DR can be continuously

connected to the blockchain. Set the endorsement policy as shown in Formula 3, which means that any two of the four organizations sign the endorsement successfully to reach a consensus, and the data can be packaged and recorded on chain.

$$\text{OutOf}(2, \text{'Org0.member'}, \text{'Org1.member'}, \text{'Org2.member'}, \text{'Org3.member'}) \quad (3)$$

The data shared in BSN can be structured data or unstructured data. In order to record, query and provide data services conveniently, unstructured data also need to be structured, such as author, date, label, summary and other information for each file. Special information for some special files may be needed, such as pixel Size of picture, playing time of audio and content description of a video, etc. Generally, the data access API is set through the action element in ACT, then the data can be accessed through the API.

In order to reduce the complexity and focus on the simulation, metadata, ACTT and ACT are saved on the blockchain directly. The default block size of Fabric is 64M, which can meet most of the data storage requirements. In the real environment, the original data can be saved off the chain, and the corresponding data index can be saved on the chain.

The system provides three basic functions: ACT issue, ACT transfer and ACT application.

(1) ACT issue is carried out through DO0 and DO1, where DO0 sets ACT1, ACT2 and ACT3 for medicalexam DP, transportation DP and scenic area DP according to the permit priority rule, and DO1 sets ACT4 and ACT5 for education DP and IPR DP according to the deny priority rule. After all ACTs are set up and recorded on chain, you can view them through the blockchain browser, and these ACTs are jointly approved by DO and DP and witnessed by other nodes on the blockchain.

(2) ACT transfer can be divided into three types: full transfer, split transfer and merge transfer. Transferring ACT1 from DO0 to insurance DR is the full transfer, transferring part of ACT2 from DO0 to traffic police DR to generate new ACT21, transferring part of ACT4 from DO1 to employment DR to generate new ACT41 is split transfer, merging ACT2 and ACT3 to DO1 to generate new ACT23, ACT4 and ACT5 to generate new ACT45 is merge transfer. All the ACTs that be transferred or generated can be viewed through the blockchain browser.

(3) ACT application is the target function of TBAC. Each DR requests data from the corresponding DP according to the ACT owned to it. After receiving the request, DP first verifies the ACT on which the request is based by verifying the blockchain. If the ACT does not exist, it will be rejected. If the ACT exists, it will start the access control decision and perform the access operation according to the decision result. In this system, insurance DR sends access request to medicalexam DP according to ACT1, traffic police DR sends access request to transportation DP according to ACT21, employment DR sends access request to education DP according to ACT41, DO1 can send access request to transportation DP or scenic area DP according to ACT23, DO0 can send access request to education DP or IPR DP according to ACT45.

TABLE 3. Robustness test methods and results.

Fault tolerance type	Test method	Test result
Fault tolerance of order node	Shut down one to three nodes of all the four order nodes	System can be running normally
Fault tolerance of peer node	Shut down all peer nodes of random two ORGs and one peer node of every other ORGs	System can be running normally

B. TBAC SIMULATION SYSTEM TEST

In the case that TBAC simulation system has the basic functions of ACT issue, ACT transfer and ACT application, its test mainly includes security test and performance test. Among them, the security test mainly includes: robustness test and trust test; the performance test mainly includes the ACT operation performance test and access control performance test.

Robustness test mainly refers to the test of fault tolerance, which refers to the test of whether the whole BSN can continue to operate normally and whether the TBAC simulation system can continue to function normally when some nodes fail. Because this simulation system uses Fabric as the base chain, its fault-tolerant ability mainly displays the fault tolerance of order node and peer node. The test method and test results are shown in Table 3.

The test results show that the simulation system has a good fault tolerance, and can guarantee the data access to continue to run normally in the worse network environment.

Trust test refers to the test of whether the ACT is unanimously recognized by the participants and cannot be tampered with, denied or misused. Non tampering means that once the ACT is issued, the relevant parties cannot tamper with it, so as to ensure the consistency of ACT. Non repudiation means that after the ACT is issued to the user, the relevant parties must approve it to ensure the authenticity of the ACT. What can't be misused is that ACT can only be used by the right subject to perform the right operation on the right object under the right conditions, so as to ensure the correct use of ACT. Because the simulation system is based on Fabric, so it inherits the characteristics of Fabric in tamper proof and non repudiation proof. There needs no test analysis here, this paper makes some tests of misusing ACT. Table 4 shows the misusing test method and results of ACT.

Whether ACT will be misused is directly related to whether the business logic coding of the three modules ACTM, ACTJ and ACTP is implemented correctly. By using a large number of test cases for testing, it can be proved that ACT can be used correctly as far as possible at the level of software engineering. Because ACT cannot be tampered with, repudiated or misused, all users of TBAC simulation system can believe this access control mechanism.

ACT operation performance test refers to the performance of the ACT issue, circulation and cancellation operation

TABLE 4. Trust test methods and results.

Misusing type	Test method	Test result
Misusing subject	Use wrong subject to request with appointed ACT	ACT judgement is failed and access is rejected
Misusing object	Request on wrong object with appointed ACT	ACT judgement is failed and access is rejected
Misusing condition	Request under wrong condition with appointed ACT	ACT judgement is failed and access is rejected
Misusing action	Request through wrong action with appointed ACT	ACT judgement is failed and access is rejected

TABLE 5. Act operation performance test methods and results.

Operation type	Test method	Test result
ACT issue	Issue 8, 40 or 80 ACTs at the same time	The TPSs are 310, 301, 294 respectively
ACT transfer	Transfer 8, 40 or 80 ACTs at the same time	The TPSs are 371, 368, 373 respectively
ACT revocation	Revocate 8, 40 or 80 ACTs at the same time	The TPSs are 365, 362, 369 respectively

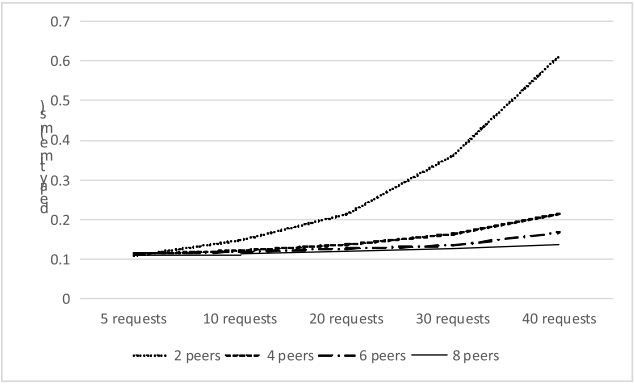


FIGURE 5. Delay time of concurrent access control.

information recorded on the chain. Because these operations of ACT include both business operation and chain record operation, this paper uses customized test tools to conduct simulation test by accessing 8 peer nodes in an balanced way. For each operation of ACT, TPS is recorded and calculated as shown in Table 5.

The test results show that the three operations of ACT maintain a higher TPS. Among them, the TPS of ACT issue is lower than that of ACT transfer and ACT revocation, which is related to the fact that ACT issue needs to prepare and write more data. ACT transfer and ACT revocation basically have the same TPS, because in TBAC, revocation actually performs the operation of transferring ACT to the recycling address.

Access control performance test refers to the performance of ACT judgement and access operation. Here we only test the performance of the ACT judgement that is from the request to the decision, because the specific access operation has a very big relationship with the business, and is not the focus of the simulation system. In this paper, a customized tool is used to run this test. Average access requests are launched by the 5 pairs of DP and DR and run on 2, 4, 6 and 8 peer nodes evenly. The test results are shown in Figure 5.

The test results show that with the increase of concurrent access numbers, the time required for access control decision is also increasing, and the more peer nodes are running, the less time required for access control decision. It can be concluded that multi nodes have better performance and support large-scale access control than few or single nodes.

V. CONCLUSION

This paper proposes a new access control mechanism TBAC based on the ACT, which inherits the characteristics of blockchains, smart contracts and ABAC. Compared with traditional centralized access control, TBAC has certain comparative advantages in security, credibility, circulation, concurrency, etc. The application of TBAC in distributed data sharing is conducive to forming a data service network with anti attack, high fault tolerance, anti tampering, anti repudiation, easy expansion and high concurrency, and better promoting data interconnection in a wider range.

In this paper, when discussing the storage of ACT on chain, it is also disclosed to the whole block chain network. Although other nodes cannot use the ACT, it is possible to obtain some information that the holders are not willing to disclose, such as their authority information, through the analysis of ACT. Therefore, the next work of this paper is how to ensure the ACT's information is not revealed, but it can be verified.

ACKNOWLEDGEMENT

This work was supported in part by the National Key Technologies R&D Programs of China under Grant 2018YFB1402702, in part by the National Natural Science Foundation of China under Grant 61972032.

REFERENCES

[1] L. Ao-Di, D. U. Xue-Hui, and W. Na, "Blockchain-based access control mechanism for big data," *J. Softw.*, pp. 2636–2654, Apr. 2019. [Online]. Available: http://kns.cnki.net/kcms/detail/11.2560.TP.20190409.1731.001.html

[2] S. Jinshan and R. Li, "Survey of blockchain access control in the Internet of Things," *J. Softw.*, vol. 30, no. 6, pp. 1632–1648, 2019. [Online]. Available: http://kns.cnki.net/kcms/detail/11.2560.TP.20190327.1640.006.html

[3] D. D. F. Maesa, P. Mori, and L. Ricci. (2017). *Blockchain Based Access Control*. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-319-59665-5_15

[4] Z. Xian, X. GuangLin, and H. Fan, "Summarisation of access control in distributed environment," *Microcomput. Appl.*, vol. 24, no. 1, pp. 4–7, 2005.

[5] A. Kapadia, J. Al-Muhtadi, and R. H. Campbell, "IRBAC 2000: Secure interoperability using dynamic role translation," in *Proc. Int. Conf. Internet Comput. (IC)*, Las Vegas, NV, USA, Jun. 2000, pp. 1–7.

[6] G. Denker, J. Millen, and Y. Miyake, "Cross-domain access control via PKI," in *Proc. 3rd Int. Workshop Policies Distrib. Syst. Netw.*, Jun. 2002, pp. 202–205.

[7] Z. ZhengDe and F. DengGuo, "Universal distributed access control decision middleware," *Comput. Eng. Appl.*, vol. 44, no. 1, pp. 17–20, 2008.

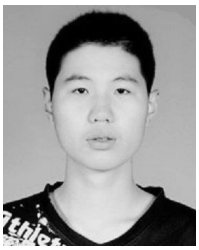
[8] G.-B. Liu, J. Shi, and J.-Y. You, "Access control in distributed system," *J. Comput. Reserch Develop.*, vol. 38, no. 6, pp. 735–740, 2001.

[9] J. Dong-Sheng and S. Yi-Yong, "Research on law-governed distributed access control model," *Jisuanji Yu Xiandaihua*, no. 3, pp. 89–91, 2006.

[10] Y. Yong and W. Feiyue, "Blockchain: The state of the art and future trends," *Acta Automatica Sinica*, vol. 42, no. 4, pp. 481–496, 2016.
[11] S. Nakamoto. (2009). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[12] H. Pu, Y. Ge, Z. Yan-feng, and B. Yu-bin, "Survey on blockchain technology and its application prospect," *Comput. Sci.*, vol. 44, no. 4, pp. 1–7, 2016.
[13] Z. Yan, G. GuoHua, D. Di, J. Feifei, C. Aiping, "Security architecture and key technologies of blockchain," *J. Inf. Secur. Res.*, vol. 2, no. 12, pp. 1090–1097, 2016.
[14] L. Xu, L. Chen, N. Shah, Z. Gao, Y. Lu, and W. Shi, "DL-BAC: Distributed ledger based access control for Web applications," in *Proc. 26th Int. Conf. World Wide Web Companion (WWW Companion)*, 2017, pp. 1445–1450.
[15] U. Ugobame Uchibeke, K. A. Schneider, S. Hosseinzadeh Kassani, and R. Deters, "Blockchain access control ecosystem for big data security," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2018, pp. 1373–1378.
[16] J. Paillisse, J. Subira, A. Lopez, A. Rodriguez-Natal, V. Ermagan, F. Maino, and A. Cabellos, "Distributed access control with blockchain," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.
[17] Y. Zhu, Y. Qin, Z. Zhou, X. Song, G. Liu, and W. C.-C. Chu, "Digital asset management with distributed permission over blockchain and attribute-based access control," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jul. 2018, pp. 193–200.
[18] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1594–1605, Apr. 2019.
[19] A. Khaled, M. F. Husain, L. Khan, K. W. Hamlen, and B. Thuraisingham, "A token-based access control system for RDF data in the clouds," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci.*, Nov. 2010, pp. 104–111.
[20] H. Dao-Jun, G. Jie, Z. Hao-Liang, and L. Lei, "Research development of access control model," *Computer Science*, vol. 37, no. 11, pp. 29–33, Nov. 2010.
[21] X.-F. Li, D.-G. Feng, Z.-W. Chen, and Z.-H. Fang, "Model for attribute based access control," *J. Commun.*, vol. 4, pp. 95–103, Apr. 2008.
[22] F.-H. Li, M. Su, G.-Z. Shi, and J.-F. Ma, "Research status and development trends of access control model," *Acta Electronica Sinica*, vol. 40, no. 4, pp. 805–813, Apr. 2012.
[23] W. Ying-Hong, H. Hao, and Z. Qing-Kai, "Techniques of distributed application access control policy refinement and policy conflict analysis," *Comput. Sci.*, vol. 41, no. 3, pp. 1–11, Mar. 2014.
[24] Q.-B. Liu, J. Shi, and J.-Y. You, "Access control in distributed systems," *J. Comput. Res. Develop.*, vol. 38, no. 6, pp. 735–740, Jun. 2001.
[25] K. Wu, "Research and application on ontology and rule based access control model," M.S. thesis, Fulfillment Requirement Prof. Degree, Chongqing Univ., Chongqing, China, 2018.
[26] Q. F. Shao, C. Q. Jin, Z. Zhang, W. N. Qian, and A. Zhou, "Blockchain: Architecture and research progress," *Chin. J. Comput.*, vol. 41, no. 5, pp. 969–988, Nov. 2018.
[27] X. Shen, Q.-Q. Pei, and X.-F. Liu, "Survey of block chain," *Chin. J. Netw. Inf. Secur.*, vol. 2, no. 11, pp. 11–20, Nov. 2016.
[28] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, pp. 1–21, 1997.
[29] M. Chunguang, A. Jing, B. Wei, and Y. Qi, "Smart contract in blockchain," *Netinfo Secur.*, vol. 18, no. 11, pp. 8–17, 2018.
[30] L. Ouyang, S. Wang, Y. Yuan, X. Ni, and F. Y. Wang, "Smart contracts: Architecture and research progresses," *ACTA Automatica Sinica*, vol. 45, no. 3, pp. 445–457, Mar. 2019.
[31] H. Kaifeng, Z. Shengli, and J. Shi, "The security research of blockchain smart contract," *J. Inf. Secur. Res.* vol. 5, no. 3, pp. 192–206, Mar. 2019.
[32] Myan. *Token is the Key Point of Next Generation Network*. Accessed: Dec. 4, 2017. [Online]. Available: https://blog.csdn.net/myan/article/details/78712506
[33] H. K. Xiaominl, "FormaI verification method of smart contract," *J. Inf. Secur. Res.*, pp. 1080–1089, Dec. 2016.

GUOHUA GAN received the master's degree from the School of Computer, Harbin Engineering University. He is currently pursuing the Ph.D. degree with the University of Science and Technology Beijing, after many years of work in the Internet of Things (IOT), communications, and cloud computing. His research interests include software engineer and blockchain. During his Ph.D. studies, he worked for a blockchain innovation enterprise as a consultant and participated in a number of blockchain research organizations.

E. CHEN received the B.S. degree from the School of Mathematics and Physics, University of Science and Technology Beijing, where she is currently pursuing the Ph.D. degree. Her research interests include attribute-based systems and lattice-based cryptography.



ZHIYUAN ZHOU received the B.E. degree from the School of Computer and Communication Engineering, University of Science and Technology Beijing, where he is currently a Graduate Student. His research interests include attribute-based access control and distributed ledger technology.

YAN ZHU received the Ph.D. degree in computer science from Harbin Engineering University, China, in 2005. He was an Associate Professor with Peking University, China, from 2007 to 2012. He was a Visiting Associate Professor with Arizona State University, from 2008 to 2009, and a Visiting Research Investigator of the University of Michigan–Dearborn, in 2012. He is currently a Professor with the University of Science and Technology, Beijing, China. His research interests include cryptography, secure computation, and network security.