An Access Control Model and Its Application in Blockchain

Xiangwu Ding
School of Computer Science and Technology
Donghua University
Shanghai, China

Jianming Yang
School of Computer Science and Technology
Donghua University
Shanghai, China
yangjm20@outlook.com

***Abstract*—Access control technology is an important information security mechanism. At present, most of the database systems and enterprise information systems are role-based access control technologies, this rights management system has been running stably. However, due to the simple role access control, its flexibility and control granularity sometimes can't meet the requirements of actual access control. This paper proposes a secure access control model ARBACV1 based on RBACV1 combined with ABAC model, which is more flexible than RBACV1 and can perform fine-grained access control. The open transparency of data in the blockchain has caused people's high attention to data privacy protection issues[1]. A complete access control mechanism has not been provided in the Ethereum blockchain. To this end, according to the blockchain architecture, the proposed access control model ARBACV1 is applied to the blockchain through smart contracts, and the access of the blockchain users is controlled securely, and the code is written in Solidity language in Ethereum[2]. ARBACV1-based access control is implemented in blockchain.**

Keywords-component; role access control, attribute access control, ARBACV1, blockchain, smart contract

I. INTRODUCTION

In the 1990s, the role-based access control model has been studied continuously, since 1996, after Sandhu et al. proposed the RBAC model[1], scholars proposed a series of improved models based on this model, which makes the management users have more convenient permissions, reduce the complexity of system rights management. More than 20 years later, the role-based access control model still occupies more than half of the access control system.

With the continuous development of big data and cloud computing, early role-based access control systems are becoming larger and larger, and the amount of data increases, and the maintenance cost of access control systems increases rapidly. When the user puts forward new permission requirements, the original role-based access control system is difficult to maintain, and the simple role access control, its flexibility and control fine grain can't meet the actual needs. With the deepening of the application of blockchain, although the blockchain can solve some security problems in the traditional system, all transaction data are transparent in all nodes. If the blockchain is applied to the management of medical data[2], credit data, personal identity and other data, the privacy data in these applications will be exposed to the public's vision, and the application of the blockchain system to these scenarios will face enormous challenges. Each participating node can analyze the transaction records, obtain the user's transaction data and private data, which will bring serious privacy leakage risks. Therefore, it is necessary to control the access in the blockchain. This paper implements the encrypted storage of sensitive information, at the same time, we introduce the access control model ARBACV1 into the blockchain to realize the rights management mechanism and combine the blocks. The characteristics of the data in the chain are difficult to tamper, and the rights management log is recorded in the blockchain for review and forensics. The paper contributions are listed as follows.

1. A security access control model ARBACV1 based on RBAV1 combined with ABAC model is proposed. The model can first dynamically add access rights according to the needs of special users, Secondly, it can filter the role permissions according to attribute information, which provides more flexibility and more granular access control for the system. Among them, the improvement of the RBAC model is implemented and combined with the ABAC model, fine-grained control of permissions to the assignment of roles based on the RBAC model.

2. Through the smart contract as a bridge, the proposed access control model (ARBACV1) is integrated into the Ethereum blockchain[3], and the smart contract is designed to implement functions such as role addition, role removal, role judgment, role authorization, role revocation, and attribute control. This section can be used as an access control for any smart contract that integrates into the specific application using the public part of Ethereum blockchain.

II. RELATED WORK

In 1996, Sandhu et al. proposed a role-based access control model, the basic idea of the model is to bind permissions to roles and then assign roles to users. Once proposed, RBAC has been widely adopted because it facilitates the management of rights, also has been a research hotspot of secure access control. However, with the rise of Web services, role-based access control does not have fine-grained control permissions. In 2005, Eric et al. proposed an attribute-based access control model for Web services[4]. Based on the topic, object and environment

978-1-7281-3681-3/19/$31.00 ©2019 IEEE
DOI 10.1109/CISCE.2019.00044

attributes, the model described the ABAC model from the authorization system and compared it with the role-based access control, the ABAC model can solve the RBAC model can't be the defect of fine-grained access control, but the strategy mechanism of ABAC is quite complicated, which increases the complexity of the system and makes it difficult to land. In 2019, he proposed the RBAC model based on attributes and trust in the cloud computing environment[5]. The model uses the ciphertext policy attribute based encryption (CP-ABE) idea and trust evaluation method to embed a trust threshold for the role. The access structure, only when the user attribute set matches the role access structure, the user can get the role and the corresponding permissions.

In the past two years, access control issues in blockchains have been a hot topic of research. In 2017, Mayssa et al. proposed a Timely CP-ABE access control model in the time-division distributed access control mechanism based on blockchain[6], firstly, the distributed access control mechanism was introduced, secondly, CP-ABE added time dimension to shared files. To solve the problem of shared data security access control in the blockchain. In 2018, Yan et al. proposed a new transaction-based access control (TBAC) platform that integrates standard attribute-based access control (ABAC) models and blockchain systems[7], proposing four types of transactions and bitcoin encryption. Scripts to describe topic registration, object hosting and publishing, access request and authorization processes to ensure secure transactions of dynamic policies in the blockchain.

The above research shows that the combination of RBAC and ABAC model can solve the problem that RBAC model can't achieve fine-grained access control, however, as the user's demand increases, the size of the privilege is also very large, and it is impossible to propose a privilege to open up a role. This problem, the improved RBAC model, combined with the ABAC model, proposed an ARBACV1 model to achieve permission filtering, rather than role filtering, which simplifies the complexity of the ARBACV1 model. Secondly, the access control problem in the blockchain is based on the access control designed in the specific scenario of the blockchain, instead of the general access control framework. Therefore, the general access control framework ARBACV1 model proposed in this paper is applied to the blockchain. It can solve the problem of integrating the blockchain permission system in any scenario and solve the problem of privacy data leakage in the blockchain system.

III. ARBACV1 MODE

A. RBAC and its improved model

RBAC[8][9][10] is a widely used access control model. The basic idea is to bind permissions to roles and assign roles to users to facilitate the management of permissions.

This model rights management system has been running stably, but when the user asks for such a requirement: Can you set the authority A for this enterprise, and do not set permissions for that enterprise? Similar to the essence behind this demand: the same role, you can set a different set of permissions, this is contrary to the design principles of RBAC. In RABC, usually a class of roles is given the same permissions. Faced with this need, we can continue to use RBAC design principles to achieve this type of demand, so create new roles to meet such needs. If the user only makes such a request for two special permissions, then it is okay to use this solution. However, today in the big data environment, this type of requirement involves up to N permissions (not to be ruled out later), then the number of combinations of permission switches is Y:

$$Y = \frac{N\ (\mathrm{N}-1)}{2}\{N \geq 2 \text{ and integer}\}$$

If each combination of permissions is to create a new role, then as the number of permissions increases, the combination of permissions will increase by a multiple, as shown in Figure 1, which is quite troublesome to manage.

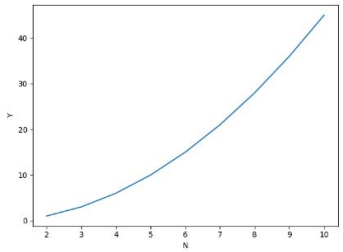

Figure 1. Privilege portfolio growth trend

In response to this type of demand, we propose an improved version of RBACV1 based on RBAC, which separates such permissions, then associates such permissions directly with the user, as shown in Figure 2.

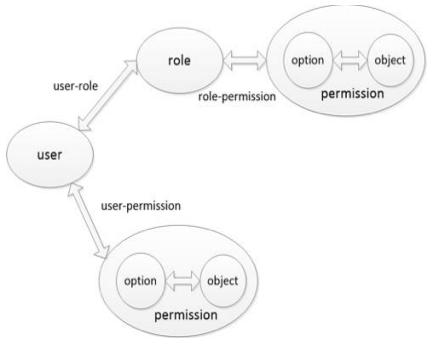

Figure 2. RBACV1 mode

Under this kind of thinking, rights management is divided into two categories: determine the permissions based on the role, determine the permissions based on which user.

B. ARBACV1 MODE

As you can see from Section 2.1, the model first starts from the user body of the access, classifies them according to the rights management requirements, generates different roles and sets corresponding operation access rights. In this way, only the role of the subject and the permissions of the user are considered, but the security of the system cannot be improved more finely. In this model, the role is fixed and cannot change with changes in the environment, subject and other attributes. Consider

introducing Attributes-based access control(ABAC) to an entity, After the entity's(subject, object, resources, environment) attributes are formalized, the user can filter the unsafe operation rights according to their attributes, their status, and security level. The specific definition of the ABAC[11][12][13] model is given below:

$$S = \{a\ set\ of\ subject\ or\ user\}$$
$$O = \{\text{a set of object or resource}\}$$
$$E = \{a\ set\ of\ environment\}$$
$$SA = \{SA_k \mid 1 \le k \le K\}$$
$$OA = \{OA_m \mid 1 \le m \le M\}$$
$$EA = \{EA_n \mid 1 \le n \le N\}$$
$$attr(s) \subseteq SA_1 \times SA_2 \times \ldots \times SA_K$$
$$attr(o) \subseteq OA_1 \times OA_2 \times \ldots \times OA_M$$
$$attr(e) \subseteq EA_1 \times EA_2 \times \ldots \times EA_N$$

A set of attributes that represent subjects, objects, and environments. In general, whether a subject s can access an object in a specific environment is determined by a Boolean function of the subject, the object, and the environment attribute.

$$request(s, o, e) \leftarrow \text{function}(\text{attr(s)}, \text{attr(o)}, \text{attr(e)})$$

In complex cloud computing or other suitable scenarios, the combination of the two features can achieve more fine-grained, more dynamic, and more secure access control, as shown in Figure 3.

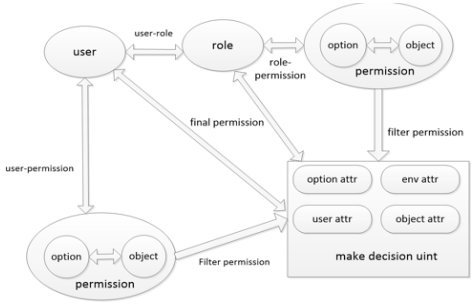

Figure 3. ARBACV1 mode

The model is based on RBACV1 and integrates with ABAC. We call the integrated model ARBACV1 (Attribute-Role-Based Access Control V1), filtering out the mismatched operation rights by the main attributes of the subject, object, operation, and environment.

According to the illustration, the determination of the user's final authority can be divided into 2 steps. First, the role determines the user's initialization permissions; secondly, the unmatched permissions are filtered from the initial permissions according to the main attributes of the subject, object, operation, and environment. Among them, the former is mainly based on user needs analysis, creating users, roles, permissions, user role sets, role permission sets, and user permission sets; the latter needs to determine the corresponding thresholds according to the four major entity attributes to filter the corresponding permissions.

IV. Application of ARBACV1 in Ethereum Blockchain

With the increasing popularity of blockchain applications, privacy data is exposed to the public's view due to its open and transparent nature. In this section, it is necessary to introduce access control technology into the blockchain system. This chapter will be simple. Introduce the basic concept of blockchain, and then explain in detail the process of introducing the new access control model ARBACV1 proposed in the previous chapter into the blockchain.

A. Blockchain

A blockchain is a special structure consisting of blocks and chains (data structures linked by hash pointers). Each of these blocks has a hash pointer to the previous block. This hash pointer is generated by hashing all relevant data in the previous block. If any value in the blockchain is as small as one bit change, the hash value of the block and the hash pointer of the subsequent block will change. Therefore, this chain structure can achieve the purpose of data tampering and traceability.

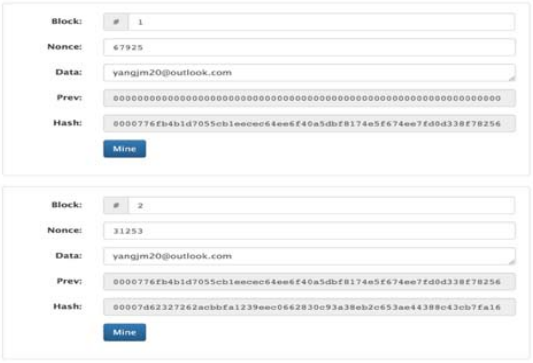

Figure 4. Blockchain

Figure 4 shows the basic information of the underlying data structure of the blockchain, where two blocks are linked together by hash values to form a blockchain. Each block has a block number, a random number, transaction data, a hash of the previous block, and a hash value of the current block. In addition, the block with block number 1 is the first block of the entire blockchain, commonly known as the genesis block. The hash pointer of the previous block is empty (all 0), and Block with block number 2, the hash pointer of the previous block is exactly the hash value of the previous block, and in this way, the blockchain arranged in time series is finally formed.

B. Blockchain integrated access control application architecture

In this section, the designed ARBACV1 access control model is applied to the blockchain, the access control model script that can be executed in the blockchain is realized mainly through the unique programming paradigm(Solidity) of the smart contract, the access control of smart contract is deployed

in the blockchain by the administrator. The application architecture is shown in Figure 5. In this application architecture, there are the following types of entities.

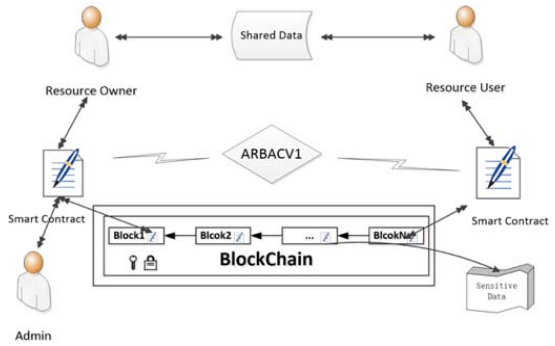

Figure 5. System access control architecture

Admin: Contract creator, responsible for contract creation, role assignment, management resource owner and visitor information registration.

Resource Owner: Resource Owner is the owner of the shared data (eg, individuals, organizations, companies, etc.), Resource Owner will upload the shared data by encrypted to the shared data service center, sensitive private data can be directly uploaded to the blockchain, set the appropriate access policies for them and store them in the blockchain..

Resource User: Resource User is a resource visitor. If want to access the data shared by the Resource Owner, they must initiate the corresponding permission request through the smart contract.

Shared Data: The data shared by the Resource Owner, which can be accessed by authorized users.

Sensitive Data: The Resource Owner stores sensitive data in the blockchain with a higher level of security.

ARBACV1: The access control framework proposed in this paper is integrated into the blockchain through smart contracts[14].

According to the above architecture, the execution process is as shown in Figure 6.

（1） This step is to prepare the process before starting the process. Admin deploys the prepared access control smart contract script to the Ethereum blockchain and implements the addition, modification, deletion, and assignment and cancellation of role permissions. Resource Owner And the Resource User completes the information registration work in the blockchain and gives it the relevant role (Admin unified management).

（2） The Resource Owner can upload the data encryption to the data server sharing center, and set the corresponding access policy for the data to be stored in the blockchain, and the log event of the uploaded data is stored in the blockchain.

（3） The Resource User initiates a request to access the shared data to the blockchain through the smart contract (carrying the relevant role and attribute information).

（4） After receiving the request, the smart contract automatically executes the ARBACV1 access control smart contract (execution role judgment, attribute decision, etc.).

（5） The result of the ARBACV1 smart contract script execution returns information to the Resource User.

（6） The Resource User receives the data returned and parses the returned data. The data is parsed, if it is allowed, the data is decrypted according to the returned key and the access log event is stored in the blockchain. Otherwise, the access is denied.

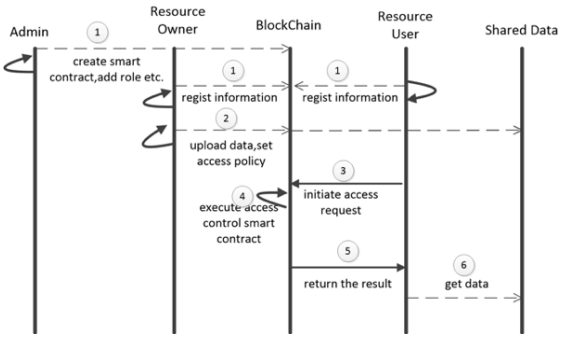

Figure 6. access control flow

C. *Case Realization*

Role.sol assigns the user the underlying implementation of the role. Owner.sol provides the most basic authorization function. School.sol inherits OwnShip.sol and imports Roles.sol to implement access control based on ARABACV1.

Below we take the most common events in life as an example to code access control between school-teacher-student. The requirements are as follows:

At the beginning of the school, the students in the school need to pay the school a tuition fee equal to 8,000 eth (fine-grained attribute control) to be registered successfully, and the tuition can only be transferred to the student's school; between the teacher and the student, only the student can pay, the teachers can only rate students and only administrators can assign roles(Role-Base control). The entities involved are as follows: School, Student, and Teacher. Implementation access to https://github.com/yangjm20/ARBACV1.git, as shown in Figure 7.

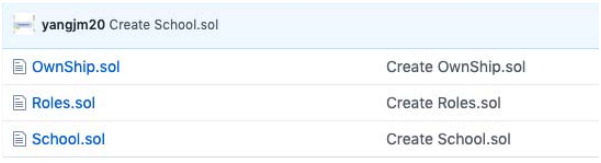

Figure 7. Smart Contract Case

School.sol access control smart contracts provide users with authentication, authorization and auditing capabilities. First, addTeacherRoles and addStudentRoles function are provided to assign users for the role of teacher or student and can only be added by the administrator (fine-grained attribute access control). Secondly, tuition and scoring are provided by payFees, gradeStudent function, by role judgment, payFees only Can be called by the student and the tuition fee must be equal to 8000 eth (fine-grained attribute access control), gradeStudent can only be called by the teacher. Finally, using the blockchain design, each transaction in the smart contract is permanently stored in the blockchain, which is convenient for people to trace.

V. SUMMARY

In this paper, an ARBACV1 access control model is proposed and applied in the blockchain to solve the existing data security problems in the open blockchain environment. By improving the existing role access control model to achieve the assignment of role permissions and the allocation of special user rights, combined with the widely used ABAC model, the access control operation is more flexible and finer. Finally, through the unique smart contract script in the blockchain, it is more flexible and better to integrate the proposed access control framework into the blockchain. The value of this work can be used as a general access control scheme in the blockchain system, developers do not need to pay attention to access control framework, they only pay attention to the development of business requirements.

ACKNOWLEDGMENT

The work is supported by the Special Fund of Shanghai Municipal Commission of Economy and Informatization (Grant No.201801027), Shanghai Science and Technology Innovation Action Plan(Grant No.16JC1400803), Shanghai Informatization Development Fund Project(Grant No. XX-XXFZ-05-16-0139).

REFERENCES

[1] Sandhu RS, Coyne EJ, Feinstein HL,Youman CE. “Role-Based Access Control Models,” Computer, 1996, 29(2):38-47. [doi: 10.1109/2.485845]

[2] MEI Ying . “Block chain method research for secure storage of medical records,[J]” Journal of Jiangxi Normal University(Natural Science),2017,41(5):484-490.

[3] Ethereum White Paper. “A next-generation smart con- tract and decentralized application platform [Online],” available: https://github.com/ethereum/wiki/wiki/White- Paper, November 12, 2015

[4] Yuan E, Tong J. “Attributed Based Access Control (ABAC) for Web Services,” In: IEEE International Conference on Web Services. IEEE, 2005. [doi:10.1109/ICWS.2005.25]

[5] Bo Yu, XianQing Tai, ZhiJie Ma. “Research on RBAC Model Based on Attribute and Trust in Cloud Computing Environment [J],” Computer Engineering and Applications | Comput Eng Appl,2019:1901-0361

[6] Mayssa Jemel, Ahmed Serhrouchni, “Decentralized Access Control Mechanism with Temporal Dimension Based on Blockchain,” in 14th International Conference on e-Business Engineering. IEEE, 2017, pp.177-182.

[7] Yan Zhu1∗, Yao Qin1, Zhiyuan Zhou1, Xiaoxu Song1, Guowei Liu2, William Cheng-Chung Chu3, “Digital Asset Management with Distributed Permission over Blockchain and Attribute-based Access Control,” in Services Computing, International Conference on. IEEE, 2018,pp.193-200.

[8] Ferraiolo DF, Kuhn DR. “Role-Based Access Controls,” Computer, 1992, 4(3):554-563. [doi: 10.1007/978-1-4419-5906-5_829]

[9] Moyer M J, Abamad M. “Generalized Role-based Access Control,” In: Proceedings 21st International Conference on Distributed Computing Systems. IEEE, 2001: 391-398.[doi:10.1109/ICDSC.2001.918969]

[10] Bertino E, Bonatti P A, Ferrari E. “TRBAC: A Temporal Role-based Access Control Model,” ACM Transactions on Information and System Security (TISSEC), New York, ACM, 2001, 4(3): 191-233.[doi:10.1145/501978.501979]

[11] Hemdi M, Deters R. “Using REST Based Protocol to enable ABAC within IoT Systems,” In: Information Technology, Electronics and Mobile Communication Conference. IEEE, 2016. 1-7. [doi:10.1109/IEMCON.2016.7746297]

[12] Han Q, Li J. “An Authorization Management Approach in the Internet of Things,” Journal of Information & Computational Science, 2012, 9(6):1705-1713.

[13] Wu J, Dong M, Ota K, Pei B. “A Fine-Grained Cross-Domain Access Control Mechanism for Social Internet of Things,” In: Ubiquitous Intelligence and Computing, IEEE, 2014. 666-671. [doi:10.1109/UIC-ATC-ScalCom.2014.140]

[14] Sambit Nayak,Nanjangud C Narendra,Anshu .Saranyu: “Using Smart Contracts and Blockchain for Cloud Tenant Management[C],” 2018 IEEE 11th International Conference on Cloud Computing. IEEE,2018,pp,857-861