


Research Article

Access Control Model Scheme based on Policy Grading in Natural Language Processing Blockchain Environment

Jie Huang ^{1,2} and Dehua Wu^{1,2}

¹Hunan Provincial Engineering Research Center for Aircraft Maintenance, Changsha 410124, Hunan, China

²Changsha Aeronautical Vocational and Technical College, Changsha 410124, Hunan, China

Correspondence should be addressed to Jie Huang; huangjie918@163.com

Received 30 June 2022; Accepted 19 July 2022; Published 8 August 2022

Academic Editor: Imran Khan

Copyright © 2022 Jie Huang and Dehua Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to solve many problems such as secure storage of access policies and distrust of third parties in complex and dynamic big data environment, a hierarchical access control model under block chain environment (BP-ABAC) is proposed. Access control policies are stored in blockchain in the form of smart contracts, and access control policies are classified in contract design. Users can obtain the access permission of the corresponding policy set according to the rating evaluation. Access to a resource is obtained when the request attribute matches the policy in the policy set. The simulation results show that the model can grant corresponding access control permissions according to different users' access requests, improve the time efficiency and accuracy of the access control process, and improve the security and privacy of the storage of access policies and the interaction of data sharing.

1. Introduction

At present, all walks of life begin to transform to information technology and network, which brings an era of data sharing and data application of big data [1]. Big data has huge commercial value and has become a very important economic asset.

Access control mechanism is a kind of technology to maintain information data security, privacy information protection, and secure data sharing. It can restrict illegal access to key resources, prevent malicious users from entering the system [2], and prevent legitimate users from accessing and using system resources illegally [3], so as to protect the security of data storage and processing in the information system [4].

Attribute-based access control (ABAC) is an access control model based on the user subject, data resource, operation, and system environment. It makes full use of the attribute set owned by the subject requester to decide whether to grant access to the resource of the object.

Blockchain technology can be said to be a distributed shared ledger technology built on a variety of technologies. Using hash calculation and digital signature technology [5], block chain has a good, complete, and immutable information data record system. Compared with the centralized data management of the traditional access control model, blockchain access control adopts decentralized or weakly centralized data management [6].

In view of the shortcomings in the existing technology, this scheme based on ABAC model, combined with blockchain technology and policy grading, proposes an access control model (ABAC) based on blockchain and policy grading [4] (BP-ABAC) for policy grading in the blockchain environment [7]. This scheme combines access control policies with smart contracts and classifies policies in contracts [8]. Rank the data requester and obtain the access permission of the corresponding policy set according to the result of user rank evaluation [9]. Block as a decentralized, distributed policy storage system. Through the combination of the two, the scheme has good query efficiency, dynamic, and security [10].

2. Methodology

2.1. ABAC Model and Related Definitions. Attribute-based access control (ABAC) is not based on user identity, but by many entity attributes to carry out policy matching and policy decision, finally authorized to allow or deny user access control requests to resources. ABAC model has four main attributes, including subject attribute, object attribute, operation attribute, and environment attribute. The formal definition is as follows:

Definition 2.1: Basic element: the component of ABAC model is A quad (S, O, E, A) , in which the four letters S , O , E , and A , respectively, represent the meaning of subject attribute, object attribute, environment context attribute, and operation attribute.

Definition 2.2: Attribute access request (AAR): It means that when the access process occurs, the subject attribute (SA), supported by the environment attribute (EA), performs related operations on the resource object attribute (OA), namely: $AAR = \{sattr, oattr, eattr, pattr\}$.

Definition 2.3: Access control policy: the decision is made according to the preset access decision rules of the relevant attributes of the access request, and the decision results are mainly permit (refuse) unknown (unknown) status.

In the face of the present multifarious information system, there are substantial breakthroughs in the fine granularity of access control and the large-scale dynamic expansion of users. The idea of entity attribute is introduced to access control policy, model, and implementation mechanism. No matter subject, object, operation, and environment attributes are uniformly described, corresponding authorization, and access control constraints are established to ensure good flexibility and scalability.

2.2. General Framework and Workflow of BP-ABAC. The proposed access control framework of policy grading in the blockchain environment is shown in Figure 1. On the basis of the traditional ABAC model, it is combined with blockchain and smart contract, and on the premise of user level and policy level, it can control the access to data information resources. In the early stage, according to the collection of attributes and the integration of relations between attributes, related operations such as description, integration, and management of access control policies in blockchain transactions are carried out. This section describes how to publish, update, and revoke access control policies. At the same time, relevant policy sets and user-level permissions are published on the blockchain in the form of smart contracts. BP-ABAC frame diagram is shown in Figure 1:

The modules in Figure 1 are all implemented in the way of smart contract in blockchain. The functional modules are explained as follows:

- (1) Policy information point (PIP): obtain entity attributes of subject, resource, and object and upload access control rights of resources.

- (2) Policy administration point (PAP): manages and maintains the policies published by resource owners and the entire policy set.
- (3) Policy decision point (PDP): determine whether the subject has relevant access permissions according to the level of the subject and make authorization decisions according to the entity attributes of the subject, the access control policy of resources, and the current state of the system.
- (4) Policy enforcement point (PEP): responsible for receiving resource requester's access request and generating AAR in combination with entity attributes. Accept the policy decision point's decision on the resource requester and enforce the PDP's decision to permit or refuse access.

The steps of the BP-ABAC access control process are described as follows:

- (1) When the PEP module receives the access request from the resource requester, it analyzes and generates the AAR according to the entity attributes in the access request and the attribute information obtained from AA and sends the AAR to the PDP module
- (2) After receiving the AAR, the PDP module initiates policy information query through the smart contract and requests to judge whether the resource requester user is legitimate. If not, the access request is terminated.
- (3) If it is valid, obtain the level permission of the user as a request to obtain the attribute information of the resource requester from PIP module and a request to obtain the access control policy of the resource from PAP module. The access control policy set is matched based on the user level. If the policy set of the same level fails to match, the system obtains the policy set of the next level for policy matching.
- (4) THE PDP module compares the attribute of the access control policy of the resource with the attribute information of the resource requester and sends the decision result to the PEP module
- (5) The resource requester performs relevant authorization (permit or deny) operations on the data resources requested by the subject according to the decision result of PEP

2.3. BP-ABAC Smart Contract Design. Smart contract in access control under blockchain environment mainly consists of four parts, which are contract participant, contract resource set, automatic state machine, and contract transaction set, respectively. The data information is described in the form of events, and the data sent is the corresponding transaction. The entire transaction is saved, and its state is handled on the blockchain. When the transaction and related event data information are introduced into the smart contract, the resources in the contract will update the status, so as to trigger the state judgment mechanism of the smart

contract. If the automatic state machine meets the trigger conditions of some instructions, the state machine will select the corresponding contract instructions to execute according to the previously preset information.

2.3.1. Contract Participant Module. Add User() function: mainly adds the user's identity attributes to the smart contract and sends the user's information to the blockchain for preservation. For later, PIP contracts to invoke user properties.

The Delete User() function: revoking the access control permission of the user or deleting the user completely for some reason. The function is used to delete the corresponding user attributes in the blockchain.

2.3.2. Contract Resource Collection Module. Add Resource() function: the owner of the resource sends the access control permissions related to the shared data. Through this function, the access control policy is stored in the blockchain. When the smart contract is triggered, the PAP transmits the policy information needed to access the object resources. The function pseudocode is shown in Table 1.

Delete Resource() function: for some reason, the resource owner wants to take back the access control permission of the shared data. Through this function, the relevant data information in the blockchain can be deleted to complete the cancellation of the share permission policy.

2.3.3. Automatic State Machine Module. Automatic state machine can generate control signals according to the control protocol, so that it can carry out state transfer in accordance with the preset state, and then complete the control center of specific operations. The triggering process of the state machine is shown in Figure 2.

The Judge Level() function: first checks whether the access is valid. Second, the user level of the resource requester is determined based on entity attributes, and the state machine is triggered according to the result to query the policy or policy set matching the corresponding level.

The Policy Set() function provides attributes required for access control and sets access control policies for shared data. It provides attributes and policies for the following Compare Policy() function.

2.3.4. Contract Transaction Set Collection Module. Compare Policy() function: it is mainly responsible for comparing the entity attribute information of the data resource requester with the entity attribute information of the access control policy of the resource publisher. By judging the similarity of each attribute information of the data resource requester, it can judge whether the resource requester can obtain the access control permission of the object resource and perform related operations. The pseudocode of the Compare Policy() function is shown in Table 2.

Get Permission() function: grants access to the object resource to the resource requester and uploads the "transaction" to the blockchain, mainly based on the result

TABLE 1: AddResource() function.

INPUT: Resource Name, Resource Id, Action, Subject id
 OUTPUT: Add resource success/failed
 //Add or delete a policy resource
 (1) if input==null
 (2) return error
 (3) end if
 (4) err = ARlstub.Get State (Resource id)
 (5) if err! = nil
 (6) return resource not exist
 (7) end if
 //Store policies in the blockchain
 (8) Resource Bytes=json.Marshal (resource)
 (9) err = ARlstub.Put State (Resource Id, Resource Bytes)
 //Return the result of adding a policy
 (10) if err = nil
 (11) return Add resource success
 (12) else if
 (13) return Add resource failed
 (14) end if

returned by the Compare Policy() function in the PDP contract.

Execute Request() function: when the resource requester performs access operations (permit, refuse, unknown) on the resource object based on the result of the Get Permission() function in the PEP contract.

2.4. Design Hierarchical Policy. Each policy has different trust values for different entity attributes, which are influenced by access requests and system interactions. When an access request accesses an object resource properly or maliciously, the trust value of the corresponding policy is raised or lowered. When the trust value of a policy increases or decreases to a certain value, it is mapped to different trust levels. The trust value of each policy is calculated according to the initial trust value and the historical trust value, which is used as the mapping basis of the policy trust level.

2.4.1. Initial and Historical Trust Values. Each time when the policy owner issues the access control policy, the system will perform weighting calculation according to the correlation degree $R(s, e)$ between the main attribute and environmental security of the policy, the correlation degree $R(o, e)$ between the object attribute and environmental security, and the correlation degree $R(a, e)$ between the operation behavior and environmental security of the policy, where α, β, γ are the weight ratio of the attribute weighting value. Get the initial trust value of the policy.

$$\text{Current}_{T(u)} = \frac{\alpha R(s, e) + \beta R(o, e) + \gamma R(a, e)}{\alpha + \beta + \gamma} c. \quad (1)$$

The first historical trust value is generated when an access control policy is authorized for the first time. As the number of subsequent policy access authorization increases, the historical trust value of the policy changes according to the time slice.

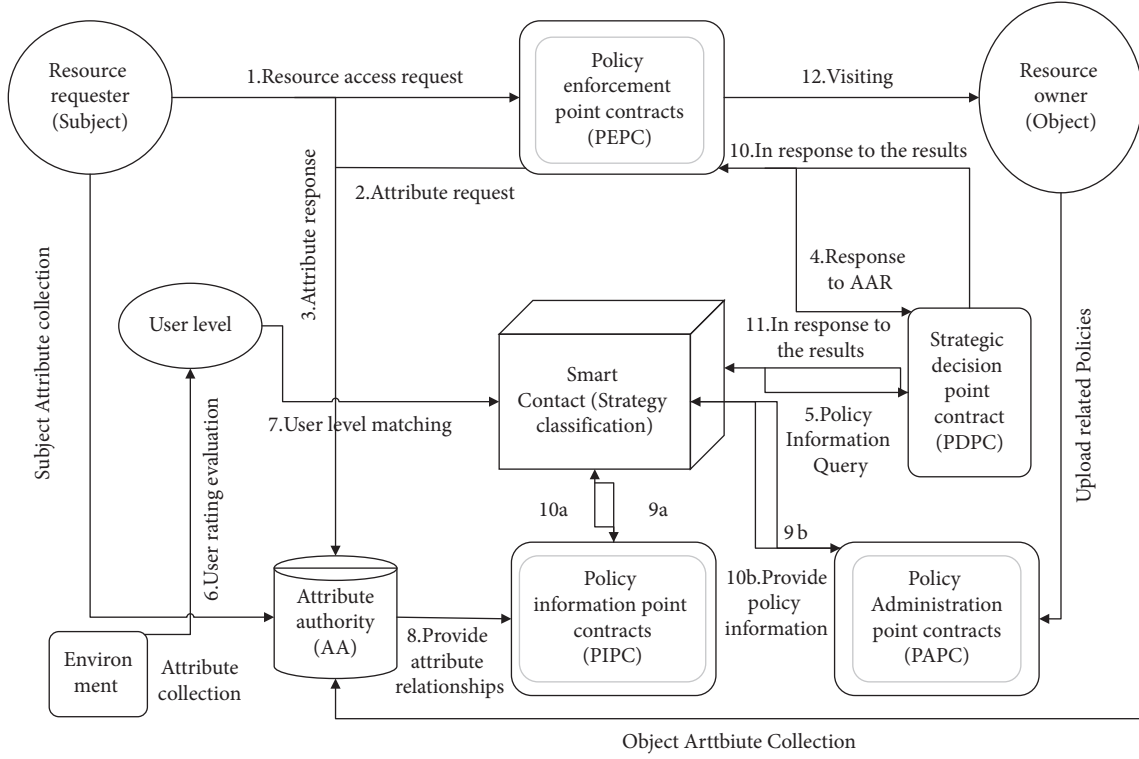


FIGURE 1: Block diagram of the policy grading access control model in blockchain environment.

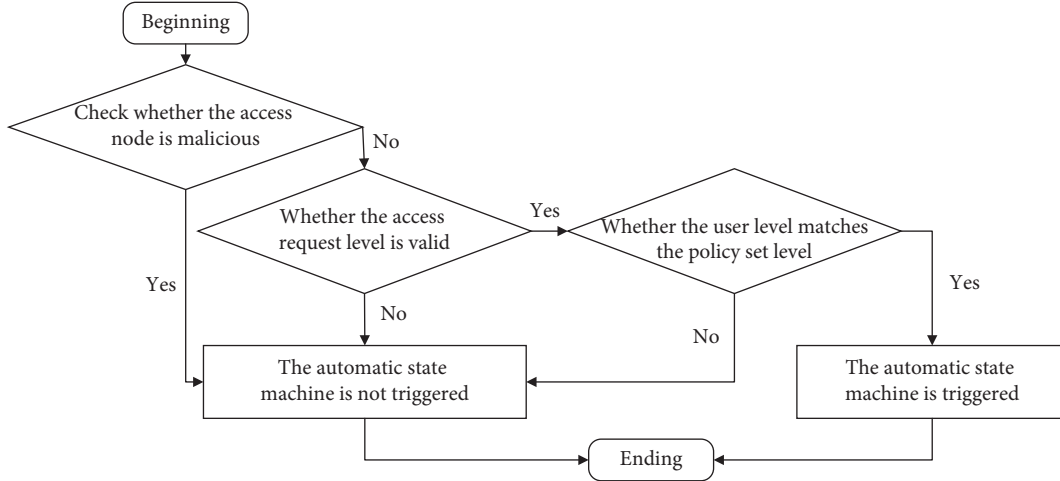


FIGURE 2: State machine trigger flowchart for smart contracts.

$$\text{History}_{T(u)} = \frac{\sum_{i=1}^n \text{current}_{T(u)} * W_t^T * t_i}{\sum_{i=1}^n t_i} \quad n > 0. \quad (2)$$

Here, W_t^T represents the weighted value of interaction behavior in the time slice t_i authorized by the policy. t_i indicates the slice of time that the access request and policy match until authorization is complete. If the access request is the first access, that is, when $n = 0$, there is no historical trust value $\text{History}_{T(u)} = 0$.

2.4.2. Final Trusted Values and Mapping. The final trust value can be obtained from the initial trust value and the historical trust value:

$$\text{Final}_{T(u)} = a * \text{Current}_{T(u)} + b * \text{History}_{T(u)}. \quad (3)$$

Here, a represents the weight of the initial trust value, b represents the weight of the historical trust value, and the two satisfy the relationship of $a + b = 1$. In the final trusted value calculation, $\text{Current}_{T(u)}$ has more reference value than

TABLE 2: ComparePolicy() function.

Algorithm 2: ComparePolicy()Function
INPUT: PolicySet()//Function to provide properties, policy parameters
OUTPUT: allow Access, Access Time

- (1) if input==null //whether attribute information is obtained
- (2) return error
- (3) end if
- (4) result Iterator = ARIstub.Get History For Key(Resource Id)
- (5) for resultIterator.next do //Get the attribute element in the sequence
- (6) query Set = resultIterator.Next()
- (7) Json.unmarshal(query Set.value, & plo)
- (8) policy_set = pol
- (9) end for
- (10) Allow Access = false
- (11) for $j=0; j < \text{len}(\text{policy_set}); i++$ do //Matches policy information
- (12) if input \in policy_set[j]
- (13) Allow Access = true
- (14) Access.Time = policy_set[j].Access time
- (15) break
- (16) end for
- (17) return AllowAccess, AccessTime //Return the policy comparison result

History $_{T(u)}$, so $a > b$. According to the policy value of Final $_{T(u)}$, the corresponding policy trust level can be obtained through mapping. The final mapping table of trust value and trust level is shown in Table 3.

According to the actual situation, the scheme temporarily divides the trust value into 5 intervals and the corresponding policy level into 5 trust levels. The access control policy can be classified into the above 5 categories according to the trust value calculation and different trust levels contain different policy sets. Resource visitors can obtain access authorization by matching corresponding policies according to the trust level. The user rating of resource visitors is similar to the policy trusted value calculation, so it will not be elaborated too much here.

Scenario assumption: when the access requester sends an access request, the system assigns an initial trust value Current $_{T(u)}$ based on the entity attribute of the request, and then starts to access the resource. Assuming Current $_{T(u)} = 1$, the weighted value W_t^T at $t1$ is 0.85 obtained through the interaction of historical behavior data of visitors, and the historical trusted value History $_{T(u)} = 0.85$ is calculated from Formula (2). Suppose the initial creditability weight $a = 0.66$, and the historical creditability weight $b = 0.34$.

Final $_{T(u)} = 0.949$ can be calculated by formula (3). As $0.949 \in [1.0, 0.9]$ can be obtained from the mapping table in Table 3, the request visitor matches the policy set with policy level 5 and performs policy traversal matching for N policies. If the AAR attribute matches an access control policy successfully, the AAR is granted the corresponding access permission. If malicious operations are carried out in the access process, the weighted value W_t^T will be reduced, thus affecting the historical trust value and weight value. In this way, malicious nodes can effectively prevent them from using the accumulated trust value of security operations to conduct serious malicious operations on access rights when they reach the maximum value or meet the trust value of corresponding operations. Therefore, the behavior control of

TABLE 3: The mapping table of trust value and trust level.

Reliability interval	Reliability level	Reliability intensity
[1.0-0.8]	5	Trustworthy
[0.8-0.6]	4	Reliable
[0.6-0.4]	3	Dependable
[0.4-0.2]	2	Undependable
[0.2-0.0]	1	Unreliable

node access process is strengthened, and the security of access control process is improved.

3. Results and Discussion

3.1. Simulation Experiment Analysis and Security Analysis

3.1.1. Simulation Experiment Analysis of Policy Query. In order to test the efficiency of access control policy retrieval based on block chain and policy hierarchy, this paper uses JAVA language to describe smart contract in Windows10 system with i7-8700 processor and 16G memory. The test is carried out according to the standard policy test package provided by XACML. The data set was used in HP LABS, and the experimental code was written based on cpABE-0.11 library. The authorization center is emulated by virtual servers. The access request AAR composed of all attributes forms an association attribute set: TT_AARA r Set. The association attribute set formed by the attributes in the access control policy is TT_policyA r Set. If the TT_AARA r Set matches the TT_policyA r Set property Set, Execute Request() executes the Permit authorization operation. If the TT_AARA r Set does not match the TT_policyA r Set property Set, If Refuse or Unknown is returned, it indicates that the access is rejected or the access request is incomplete and relevant authorization operations cannot be performed.

The traditional retrieval method, literature [11], literature [12], literature [13], and the retrieval method of this

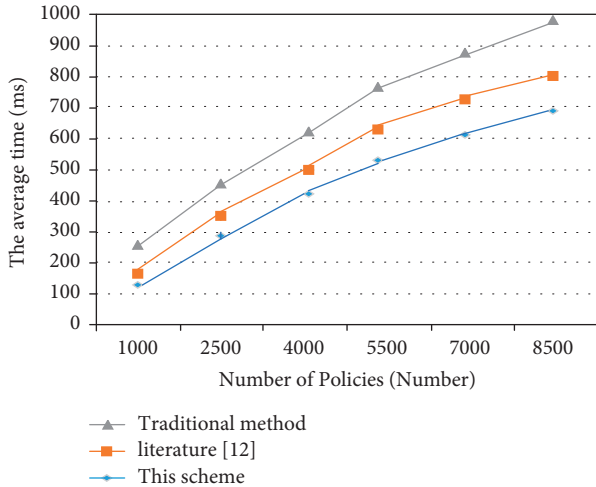


FIGURE 3: Policy query comparison diagram.

paper are tested under different strategy scales. The policy size is 1000, 2500, 4000, 5500, 7000, and 8500 samples in a total of 6 groups of single test set. The strategy scale experiment at each level was carried out 6 times, and the average value of the 6 experiments was taken. In the experimental results, different curves represent the query efficiency of different retrieval methods under different strategy scales. Comparison of policy query is shown in Figure 3.

According to the comparison in Figure 3, with the expansion of the policy scale, the query efficiency of traditional access control, literature [11], and this scheme tends to expand gradually. Through the average calculation of six values, the query efficiency of this scheme is improved by about 33.0739% compared with the traditional access control method. Compared with reference [11], this scheme improves the query time by 16.2400%.

In order to further verify the reliability of the scheme in policy query, this scheme is compared with the attribute decentralized access control model in reference [12] and the attribute security value based access control in reference [13]. The comparison results are shown in Figure 4.

As can be seen from the figure, compared with reference [13], this scheme improves access control by about 8.6948% and access time by 2.79% compared with reference [12]. Therefore, by comparing with several schemes with different characteristics, it can be seen that, in general, this scheme has obvious advantages over the above three methods in query efficiency.

3.1.2. Simulation Experiment Analysis of Strategy Decision.

With the expansion of strategy scale, the correctness of strategy decision is a problem worth paying attention to. As the scale of policies increases, the probability of conflicts between policies increases. In view of such problems, this paper has not introduced the policy conflict solution and will study and improve such problems in the subsequent part of the work. So, Figure 5 is the outcome of this scheme and the traditional access control success rate comparing, strategy by

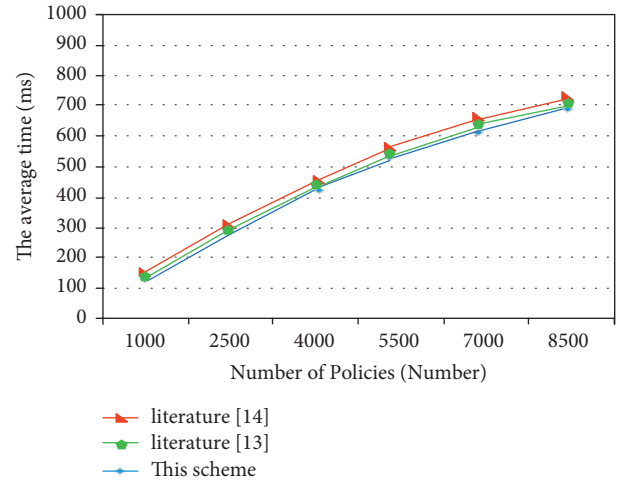


FIGURE 4: This scheme is compared with the attribute security value policy query.

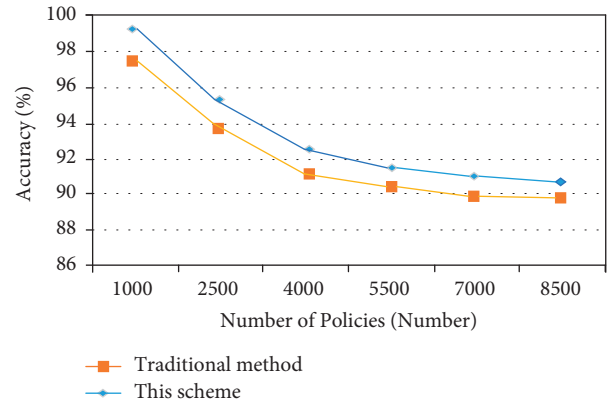


FIGURE 5: Strategy decision success rate comparison chart.

the preceding query result has to query, the success can be judged on the basis of strategy, according to the access control request is successful or not, and access to the success of subject to authorization due to expected result finally consider whether the results are in strict accordance with the access control policy enforcement. Get the result of judging the accuracy of the data. The comparison in the figure shows that the difference of accuracy between the two is in a controllable range. The comparison of success rate of strategy decision is shown in Figure 5.

3.2. Block Antiattack Analysis. The main challenge based on blockchain is that the consensus mechanism is threatened by security. In order to analyze the antiattack of blockchain itself, the proof of work mechanism of consensus mechanism is mainly used to analyze the security problems faced by blockchain, and the attack model proposed in literature [14] is analyzed. There is a competitive relationship between the trusted chain generated by trusted nodes and the attack chain generated by malicious nodes. This competition can be described by a binary tree “random walk” process. In contrast, when the trusted nodes produce a large number of

trusted chains, the trusted chain adds a block; otherwise, the malicious node adds a block to the attack chain. In order for the blockchain to be threatened, the length of the attack chain generated by the malicious node is greater than the trusted chain generated by the trusted node. The probability problem of malicious nodes chasing z blocks is similar to the gambler's bankruptcy problem. Therefore, the probability of success of malicious nodes chasing z blockchains is

$$q_z = \begin{cases} \left(\frac{q}{p}\right)^z, & p > q, \\ 1, & p \leq q. \end{cases} \quad (4)$$

In which p is the probability that the trusted node obtains the next accounting right, q is the probability that the malicious node obtains the next block accounting right, q_z is the probability that the malicious node successfully catches up with the difference of z blocks, and the higher the z value is, the lower the probability of success. Assuming that the expected average time for trusted nodes to generate a block, the potential block catch-up progress of malicious nodes is extremely consistent with the mathematical law of Poisson distribution, and its expected value is

$$\lambda = z * \frac{q}{p} \quad (5)$$

If the probability of success of malicious nodes attacking blockchain is needed, that is, the attack chain produced by malicious nodes exceeds the blockchain length of trusted nodes. It is necessary to know the Poisson distribution probability density of the block length generated by the malicious node and the probability that the malicious node can successfully trust the node's trust chain at this moment, and then multiply the two p_α to get

$$p_\alpha = \sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \begin{cases} \left(\frac{p}{q}\right)^{(z-k)}, & k \leq z, \\ 1, & k > z. \end{cases} \quad (6)$$

The size of the block difference between the malicious node and the trusted node can influence the probability of the malicious node successfully tampering with the block, and the relationship between the two is shown in Figure 6.

According to the figure analysis, when the block difference is set to a certain value, the probability of malicious nodes successfully tampering with the block will be significantly improved with the improvement of computing power. When q is less than 0.5, the block difference is inversely proportional to the probability of the malicious node tampering with the block. Only when the malicious node q is greater than or equal to 0.5, can it obtain the accounting right of the next block and grasp the overall trend of blockchain data. However, it is very expensive to control more than 50% of the computing power of blockchain, so it is difficult for attacks to succeed. Therefore, block chain can achieve good security in the process of access control.

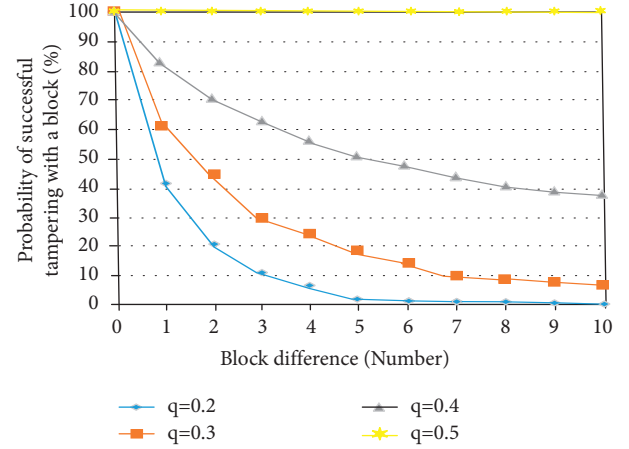


FIGURE 6: Probability of successful attack.

3.3. Security Analysis. The BP-ABAC model proposed in this scheme is based on attribute-based access control (ABAC), which makes the access control process more flexible and secure through the combination of blockchain and policy hierarchy. Compared with the traditional access control model, this model has the following advantages:

In terms of policy and access process: compared with traditional access control, this model does not need to establish a central database node to store policies, and the characteristics of blockchain make policies more tamper-proof. Access control policies issued by resource owners are voted and graded by all network nodes. During policy matching, you only need to query the corresponding tiered policy, which enhances the flexibility of authorization and ensures the consistency of policy updates and access records.

Constrained: each time the resource requester visits, the user is evaluated, and the current access control state will be changed only when the condition triggering the state set is met. By monitoring the status of the access control process, when the access control is complete, the policy is re-evaluated and graded. Compared with the traditional model, the constraint is increased, and the strengthening of the constraint can reduce the hidden trouble of illegal access.

Decision form: traditional access control models mostly use centralized third-party trust platform to make decisions, and the security of the entire access control process depends on the reliability of the third party. Uploading the access control policy to the blockchain can effectively curb the unauthorized operation of visitors. Attackers need to control more than half of the network nodes before they can have an impact on decision-making behaviors, avoiding the situation of incorrect authorization or paralysis of the access control process caused by the destruction of the central node.

This part configures the software and hardware tools and environment needed in the early stage of the experiment and gives a comprehensive and detailed introduction and related instructions to the whole experiment process. Simulation experiments are carried out on the efficiency and accuracy of

policy query of the BP-ABACA model in this paper. Based on different policy scales, six groups of experiments are selected, and the average value of the six times is taken to investigate the influence of human factors. In addition, in comparison with the traditional ABAC model, It is found that the result is better than the traditional access control model, and then compared with the prefix markup method and the access control based on attribute security value, the results also show that the proposed scheme has improved. In order to analyze the antiattack of blockchain itself, the security line analysis of blockchain is mainly based on the proof of work mechanism in the consensus mechanism. However, it is extremely expensive to master more than 50% of the block chain's computing power control. Finally, the security of the whole model is explained from the aspects of policy and access process, attribute, or policy constraint and decision form. Therefore, it is difficult for malicious node attacks to succeed, and the whole access control model has certain security.

4. Conclusion

ABAC has unique advantages in access control. In this era when privacy protection is becoming more and more important, access control system also needs to be further strengthened to adapt to the current environment. When the subject accesses the data share, the access identity and the rationality of the access authority need to be solved in a more in-depth authentication. Second, the legitimate habits of the access subject and the rationality of the access control policy need dynamic management and authentication mechanism. Not being able to stop losses immediately to a certain extent when data are leaked and the issue of retrospective responsibility division both need to be improved. With the development of block chain technology, its own consensus authentication mechanism can record the entire operation or sharing process of information resources, and the traceability of block chain ensures the security of information, which largely solves the credit problem.

This scheme proposes a block chain and policy hierarchical access control (BP-ABAC) model based on ABAC model. By using the decentralized and immutable characteristics of block chain, it gets rid of the limitation of traditional third-party trust mechanism and improves the reliability, security, and transparency of access control. According to the reference of smart contract, the whole access control authorization process is automated, which makes the access control model more flexible. Through trusted values, the user hierarchy and grading strategy makes the access control policy set and attribute request matching convenient fast, make the whole access control policy decisions and authorization process has a higher efficiency, ensure the requester legitimacy resources, prevent the excessive authorized and unauthorized access, to ensure data privacy and security in the process of access to a shared. It can effectively authorize subjects, realize data sharing between subjects and objects, and ensure the security of access control process.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by Scientific Research Project of Education Department of Hunan Province in 2021: "Research on cloud storage Security Architecture Technology based on block chain" (Research funder: Huang Jie, grant number: 21C1563).

References

- [1] H. Li, M. Zhang, and D. Feng, "Research on big data access control," *Chinese Journal of Computing Machines*, vol. 7, no. 1, pp. 74–93, 2017.
- [2] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-abe with efficient attribute revocation for cloud storage," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1767–1777, 2018.
- [3] W. Teng, G. Yang, Y. Xiang, T. Zhang, and D. Wang, "Attribute-based access control with constant-size ciphertext in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 617–627, 2017.
- [4] M. Jemel and A. Serhrouchni, "Decentralized access control mechanism with temporal dimension based on blockchain," in *Proceedings of the IEEE 14th International Conference on E-business Engineering*, pp. 177–182, IEEE Press, Shanghai, China, November 2017.
- [5] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "FairAccess: a new Blockchain-based access control framework for the Internet of Things," *Security and Communication Networks*, vol. 9, no. 18, pp. 5943–5964, 2016.
- [6] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Proc. Of the IFIP Int'l Conf. on Distributed Applications and Interoperable Systems*, pp. 206–220, Springer-Verlag, Cham, 2017.
- [7] A. Outchakoucht, H. E. Samaali, and J. P. Leroy, "Dynamic access control policy based on blockchain and machine learning for the internet of things," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 7, pp. 417–424, 2017.
- [8] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the Internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2019.
- [9] X. Fan, L. He, and X. Wang, "An approach to role management based on RBAC model," *Computer research Research and development*, vol. 45, no. 3, pp. 211–215, 2012.
- [10] S. Alansari, F. Paci, A. Margheri, and V. Sassone, "Privacy-preserving access control in cloud federations," in *Proceedings of the IEEE 10th International Conference on Cloud Computing (CLOUD)*, pp. 757–760, IEEE, Honolulu, HI, USA, June 2017.
- [11] J. Zou and Y. Zhang, "Strategy retrieval based on prefix label operation in ABAC," *Computer Engineering and Design*, vol. 36, no. 11, pp. 2943–2947, 2015.

- [12] M. X. chen, Z. H. U. J. Tao, and J. Shao, "A decentralized access control model based on attributes," *Computer Technology and Development Exhibition*, vol. 28, no. 9, pp. 118–122, 2018.
- [13] J. Chen, Y. Guan, and J. Liu, "Mandatory access control model based on attribute security value," *Computer Science*, vol. 44, no. S1, pp. 348–350, 2017.
- [14] W. Ding, G. Wang, A. Xu, H. Chen, and H. Chao, "Research on key technologies and information security issues of energy blockchain," *Chinese Journal of electrical engineering*, vol. 38, no. 4, pp. 1026–1034, 2018.