A Right Transfer Access Control Model of Internet of Things Based on Smart Contract

Jiuru Wang, Ping Gong, Haifeng Wang, Wenyin Zhang, Chongran Sun, and Bin Zhao

School of Information Science and Engineering, Linyi University, Linyi, Shandong 276000, China

Correspondence should be addressed to Jiuru Wang; wangjiuru@lyu.edu.cn

Received 12 November 2021; Accepted 10 March 2022; Published 2 May 2022

Copyright © 2022 Jiuru Wang et al. 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.

Sensor nodes play a crucial role in the promotion of development of Internet of Things (IoT). Through this transaction, RO defines access control policies in script form based on ABAC’s access control model to grant access right. The identity of all users in the model is identified by address. This paper builds a more flexible right transfer access control model by means of combining the Attribute-Based Access Control model (ABAC) and blockchain technology. Owing to the characteristics of ABAC’s attributes and right association, the massive problems of some sensor nodes can be solved. At the same time, for the sake of addressing the dynamic problems such as node access and right transfer, right transfer contract (TS) and access control contract (CS) are employed on the chain to ensure efficient and safe transmission of rights. To solve on-chain storage problems and ensure transparency of the operation, the idea of Rollup in Ethereum expansion is used to upload the final state of protocol policy and right exchange to the chain. Any user can know the policy and current right transfer status at any time. Finally, comparative and security analysis show that the model presented here can solve IoT devices’ massive and dynamic problems more effectively and it is more secure than the traditional models.

1. Introduction

With the development of IoT, the number and types of sensor nodes are both increasing. The traditional method of statically, mechanically binding users and rights for access control can no longer meet the essential needs of IoT. Sensor nodes also have the characteristics of dynamics. The dynamics here refer to the dynamics of access and include the mobility of nodes and the real-time change of subjects. This dynamic nature makes it impossible to predict all user information in advance, understand the user and right structure precisely, and present the corresponding relationship between a user and right in advance [1]. In recent years, the growth field of blockchain technology has been expanding from the financial field to IoT. When blockchain and IoT were combined, access control, as one of the critical technologies of IoT data protection, became the main combination field. Smart contract technology in blockchain automatically performs the right judgment and right transmission by setting the trigger conditions [2]. It should improve the security and effectiveness of access control by stipulating the operation and scope of an article or certain information (read, write, change) and avoiding the transmission of permission to illegal users. This paper takes the combination of access control technology and blockchain technology as the overall design policy, and it also proposes the following solutions to tackle the above problems:

1. Blockchain technology is applied to access control. Blockchain records the right owner and the final state of the right transfer [3]. Due to the tamper-proof characteristics of the blockchain, the authorization, revocation, transfer, and other operations are recorded on the blockchain and cannot be changed for audit.

2. The smart contract employed on the blockchain is used to transfer the access rights to resources from the current user to another user and prevent the intervention of the resource owner.
2. Related Work

Researchers have proposed new solutions to meet the security requirements of IoT. Currently, solutions for access control in IoT environments include the following: Yavari et al. proposed a lightweight but highly scalable data obfuscation technology to prevent unauthorized access and disclosure of sensitive information [4]. Liu et al. proposed a resource sharing access control model for multidomain Role-Based Access Control [5]. Jemel et al. proposed a decentralized access control mechanism to verify the legitimacy of users through blockchain nodes and added a time dimension to file sharing [6]. Ouchakoucht and others, using a machine learning algorithm, designed IoT access control model based on blockchain and machine learning [7]. Ouechtati et al. coupled ABAC model with the concept of trust and proposed a trust ABAC access control model for IoT environment [8]. CapBAC’s support for dynamics is mainly reflected in its distributed design, which makes it more suitable for the dynamic network topology in IoT. Wang et al. combined it with blockchain technology, designed the power token stored in the blockchain and the token management contract based on smart contract, and designed the token verification method based on the blockchain according to the decision-making mode [9].

Based on the comprehensive analysis, it appears that existing access control research results and schemes can be roughly divided into two categories: The first is the improvement of traditional access control models. Traditional access control models include the Discretionary Access Control model (DAC), the Mandatory Access Control model (MAC), the Role-Based Access Control model (RBAC), and a smaller fine-grained Attribute-Based Access Control model (ABAC). The second way is to combine access control technology with other technologies to create a new access control model. The most common of these is to incorporate blockchain technology into it. Both schemes have the following advantages:

1. The RBAC model assigns rights based on roles, and users get corresponding rights based on their roles in the system. The ABAC model can automatically obtain attributes and associate them with rights, allowing for fine-grained access control. CapBAC is a power-based distributed access control model, which can realize lightweight and dynamic access control and enhance the scalability of IoT [10].

2. With blockchain technology, all user authority granting and revocation records can be kept on the blockchain. A consensus can be reached across the whole network, making the operation more open, transparent, and traceable.

However, for the problem of right transfer in access control, the current research results still have the following deficiencies:

1. At present, these access control models are not mature and do not uniformly consider IoT’s massive, dynamic, and lightweight characteristics.

2. Right transfer lacks flexibility. In the process of right transfer, when the current owner of the right wants to transfer the right to other users, he/she must first withdraw his/her right from the resource owner and then reauthorize the new visitor. The steps are too cumbersome.

3. Blockchain storage is an increasing process, bringing a substantial burden on storage.

Therefore, it would be worthwhile to develop a model that takes into account the characteristics of massive, dynamic, and lightweight equipment, improves the capability of authority transmission, and reduces the storage pressure on the blockchain.

3. Relevant Technical Background

3.1. Blockchain and Related Technologies. As an essential part of the new generation of information technology, blockchains integrate distributed storage, point-to-point transmission, consensus mechanisms, encryption algorithms, and other technologies [11]. Through the characteristics of open and transparent operation, difficult data tampering, and traceable transaction, realize and manage the transactions in the chain [2] and solve the trust and security problems in cyberspace. Blockchain 1.0 is represented by bitcoin, which represents the application of virtual currency. Blockchain 2.0 refers to the smart contract. The combination of smart contracts and currency provides a broader application scenario for blockchain.

Blockchain-based smart contracts include transaction processing mechanisms and storage mechanisms. At the same time, a complete state machine is needed to accept and process smart contracts. Transactions contain the data to be sent. Events describe how the information was sent. The smart contract system automatically sends out preset data resources according to the trigger conditions contained in the event description information. Suppose the trigger conditions of one or more actions in the automatic state machine are met [12]. In that case, the state machine selects the contract action to be executed automatically according to the preset information. In short, the smart contract is just a system composed of a transaction processing module and a state machine. It does not generate a smart contract or modify the smart contract. It exists only to enable a group of complex digital commitments with trigger conditions to be correctly executed according to the participants’ will. The operation mechanism of the smart contract is shown in Figure 1.

3.2. Access Control. Access control refers to formulating related access control policies according to specific attributes of users to restrict users’ operations and effectively ensure the security of resources [13]. It is a gateway for network
security and resource protection [14]. Access control service comprises an access control model and policy description language. The model is policy-based, and various flexible attributes dynamically control the policy. In ABAC [15] model, a configuration file or DSL is usually used with the rule parsing engine, such as XACML (eXtensible Access Control Markup Language). In XACML architecture, there are five kinds of control nodes, as shown in Table 1.

A typical ABAC process is shown in Figure 2.

For the following reasons, this paper selects the ABAC model:

(1) Easy-to-manage attributes: ABAC can easily design attributes for different users because attributes are K-V, one of many tables that can control the correspondence between subjects, objects, and attributes [16].

(2) Fine-grained access control: ABAC can formulate access control policies for various object attributes, and role-based authorization and access control are also applicable. Roles can be used as an attribute of users in ABAC [17]. It can strictly control various conditions for visitors to obtain rights, accurately set the attribute right relationship, and realize the principle of minimum license.

(3) Meager cost of access control management: with the growth of users and resources, the number of RBAC rules increases exponentially, while ABAC rules increase linearly [18].

(4) Dynamic access control: in the ABAC access control model, resources, subjects, and objects are identified through attribute sets to separate policy management and right determination [19]. Therefore, the ABAC access control model has more robust flexibility and scalability than other control models and can meet the needs of different application scenarios.

3.3. Elliptic Curve Encryption Algorithm. ECC, ECDH, or ECDSA are commonly used for public key encryption. The first is the abbreviation of ECC, and the other two are improved algorithms based on ECC. These three crucial technologies build the modern web and IoT world, while ECC is more widely used in bitcoin and other digital currency encryption technologies [20]. ECC is an asymmetric public key cryptography algorithm [21]. This asymmetry means that the algorithm requires a pair of keys. One of them is used for encryption, called public key, and the other is used for decryption, called private key. A private key is a random number based on elliptic curve multiplication, signing the certificate as proof of holding authority. The private key generates the public key through ECDSA. The public key is obtained by elliptic curve transformation, and the corresponding transformation obtains the public key. The public key, as the intermediate bridge from the private key to the address, plays a vital role in verifying the transaction. It mainly includes the following:

(1) The public key generates the address to verify whether the transaction address is consistent with the address generated by the public key.

(2) The public key verifies the signature of the private key, which is used to verify whether the private key signature of the transaction is correct.

(3) The private and public keys appear in pairs. The public key can generate the corresponding unique address to confirm whether the transaction sent by the address uses the corresponding private key.

3.4. Rollup. In Ethereum, the gas of each block has an upper limit and regular average block time, so the calculation steps that can be done per unit time are limited. Therefore, when many transactions occur, especially with complex operations, Ethereum will be congested [22]. There are two ways to expand the blockchain’s capacity: the first is to make the blockchain itself have higher transaction processing capacity; the second is to change the method of using blockchain, such as Rollup.

Rollup is to roll up a pile of transactions into a rolled-up transaction. These rolled-up transactions and changed status (such as account information) after these transactions will be transferred to a ledger under the chain. This ledger is verified and maintained by some particular nodes, and the status of this ledger will be summarized and sent to Ethereum regularly. After receiving the rolled-up transaction, all nodes
only need to accept the execution results of this logic without executing each transaction [23], so the gas needed to execute the exchange is reduced. The rolled-up block is shown in Figure 3. A block can accommodate more transactions through rolling up transactions, and the corresponding TPS will be improved.

4. Access Control Model

4.1. Model Architecture. This paper proposes an access control model architecture that consists of four levels: the storage layer, the blockchain service layer, the API interface layer, and the application layer. The storage layer provides access to data. The data of the storage layer needs to be authenticated by the blockchain service layer. The data synchronization between nodes and an access control strategy formulation is realized in the blockchain service layer. The stable operation of the node network is guaranteed through an appropriate incentive mechanism and smart contract. At the same time, the storage layer provides support for the application layer, and the data generated by the application layer will be stored in the storage layer, as shown in Figure 4.

As for the first storage layer, to improve the TPS of the block, this model adopts the idea of Rollup, and only the access control policy and the final state of the transaction are stored in the chain. The access control policy is formulated by the resource owner when storing data. The second layer, blockchain service layer, can be subdivided into the network and smart contracts. The network layer covers P2P networking mechanisms, transaction consensus, and other modules, in which the transaction module can realize the synchronization and verification of block data between different nodes. The consensus layer includes consensus mechanisms such as Proof of Work (PoW) and Proof of Stake (PoS), which can select appropriate consensus strategies according to different application scenarios [24], reflecting the pluggable characteristics of the consensus mechanism. Smart contracts are used to write and deploy code on the chain. The primary function is to provide access control based on user attributes according to the code contained in the chain; that is, only visitors who meet specific attributes have the right to operate on the data, such as reading or writing data. The third layer, API layer, is used to provide the upper layer with interfaces such as online broadcasting and data query. Layer 4, application layer, provides users with a variety of applications, such as file transfer, access management, and query services.

4.2. Model Introduction. Through the deployment of a smart contract in conjunction with ABAC, this paper can realize dynamic right transmission. Resource owners define access rights through transactions, and all transactions representing right transfers are published on the blockchain. Therefore, any user can check at any time the user who currently has the right to perform a given operation on a given resource. As a database or policy retrieval point, the

| Node abbreviation | Node name                        | Node function                                      |
|-------------------|----------------------------------|----------------------------------------------------|
| PAP               | Policy administration point      | Managing access authorization policies             |
| PDP               | Policy decision point            | Evaluating whether the access request is legal     |
| PDP               | Policy enforcement point         | Sending the user’s request to access resources to    |
| PEP               | Policy enforcement point         | Implementing the access control policy according to |
| PIP               | Policy information point         | Providing access attribute information, including  |
| PIP               | Policy information point         | Resource information, access object information,    |
| PRP               | Policy retrieval point           | Saving the policy, usually a database or file      |

Table 1: XACML control node.

Figure 2: Flow chart of ABAC.
blockchain stores access control policies of resources and requesters in transactions. At the same time, as the index of the access control policy, the token represents the authorization of the transaction creator to its receiver. In the process of right transfer, CS forms a complete access control policy according to the index to judge whether the right can be transferred.

The identity and the proof of holding authority are established through public-private key pairs, addresses, and digital signatures. In this model, model entities include resource owner (RO) and requester (RE). All entities in this model use an address to identify interactions. Each entity has a wallet. In this model, the wallet as an Authorization Management Point (AMP) generates access control policies, and access rights are transferred through tokens [25]. The transactions involved in this model include GrantAccess, GetAccess, and DeliveryAccess. The overall framework of access control model is shown in Figure 5.

(1) GrantAccess: granting access right. Through this transaction, RO defines access control policies in script form based on ABAC’s access control model to grant access right to one or more resources identified by its address to requesters, also represented by the address. RE needs to obtain RO’s authorization to access RO’s resources. In this model, the token is used to represent the right transfer. RE sends a request to RO, indicating the address of the requested resource and the operation to be performed. Then, after receiving the request, RO encapsulates the access control policy in its output in the form of a locking script through GrantAccess and specifies the address of RE. The token is the signature of the transaction. The RO wallet broadcasts the transaction to the network. The network verifies whether the transaction is valid, and if it is valid, it will be stored in the blockchain. Otherwise, it will be rejected, and the sender will send a notification.

Entity: RE ⟽ RO
Input: access resource (address)
Operation
Output: script (access control policy encapsulation)

(2) GetAccess: obtaining access rights. RE proves to the network that he/she has completed the access through such transactions. RE accesses RO resources and can check whether the token is valid by checking the signature. At the same time, RE checks whether a
transaction is included in the blockchain and requested operation according to the access right of unlocking script defined in this transaction.

Entity: RE
Input: token
Output: verifying signature results

(3) DeliveryAccess: transferring access right. Through this transaction, the access resources’ rights can be transferred from the current user to another user without the intervention of the resource owner. The following section will describe the specific right transfer process in detail.

Entity: RE₁ (who already has access), RE₂
Input: token
Notice to the right deliverer
Output: right delivery message

4.3. Right Delivery Process. In the case of the Delivery Access transaction above, this paper primarily implements it using a smart contract to ensure the system’s efficiency and availability and enhance the model’s security in the process of right transfer. During access, the right transfer is required, mainly including the following two scenarios:

(1) The right owner A of the current resource R transfers all rights to B.

(2) The right owner A of the current resource R transfers some rights to B (A can only narrow the scope of right use by restriction rather than modify the right during token transfer).

In the existing research methods, all rights are granted by the resource owner. If the right owner A of the current resource R transfers the right to B, the access right can only be
returned first. Then, B applies to the resource owner after initiating the cancelation transaction. The method is too cumbersome and inefficient and cannot solve the massive problem of the current IoT sensor nodes' emotional problem.

This paper improves the ABAC access control model by using the right transfer smart contract (TS) and access control contract (CS) deployed on the blockchain to transfer the token representing authorization to another user, as shown in Figure 6.

The function of TS mainly includes confirming whether the message sender’s identity is legal and verifying whether the right can be transferred. The primary function of CS is to judge whether the right receiver meets the requirements of the access control policy. Two smart contracts are used for automatic authentication to avoid transferring rights to illegal users. The whole right transfer process can be divided into seven steps, as shown in Figure 7.

**Step 1.** The right owner A who has access to resource R negotiates with resource visitor B and decides to transfer the token of resource R to user B.

**Step 2.** The right owner sends the right transfer message generated from the negotiation result to TS in the blockchain network. The content of the right transfer message is to transfer the right of A to user B.

**Step 3.** User B sends a message confirming the receipt of the authorization token from TS.

**Step 4.** The effectiveness of the right transfer requires the consent of all three parties: A, B, and the access control contract of resources. Therefore, TS should send the right transfer message to CS.

**Step 5.** The access control policy in CS needs to decide whether to approve the transmission of the authorization token. Therefore, it is necessary to verify whether the attributes and constraints of the authorization token are legal, and then obtain the relevant information of B from the PIP to verify whether B is legal.

**Step 6.** CS makes decisions according to the collected information and then returns the decision results to TS.

**Step 7.** If the result is “allow,” TS will send the token to B and inform A; if it is “deny,” TS will send the rejection information to B and inform A.

5. Experimental Analysis and Verification

5.1. Comparative Analysis. Blockchain technology is evolving rapidly, and blockchain applications are expanding. The combination of blockchain technology with traditional access control models is an important application of blockchain in the IoT. Table 2 compares the characteristics of the combination of blockchain and different access control models, fully reflecting the advantages of the ABAC model. Table 3 mainly describes the application characteristics of smart contracts in access control.

5.2. Safety Analysis

5.2.1. Blockchain Security Analysis. The model proposed in this paper combines ABAC access control technology and blockchain technology. Therefore, the security of blockchain directly affects the security of this model. To solve the problem of blockchain security attacks, the primary attack method is used to attack the consensus mechanism. This paper verifies it through the verification model proposed in [30] combined with the workload verification mechanism. PoW is a kind of consensus mechanism. Whoever has more computing power in PoW will increase obtaining bookkeeping rights. In other words, when the attacker has more than half of the computing power of the whole network, the attacker can control the direction of the network. The competition between the attacking chain and the attacked chain can be expressed by random walk. The possibility of catching up with the attacked chain is equivalent to the gambler bankruptcy problem. For the above concepts, the following definitions are made:

- \( p \): the probability that the honest node finds the next block.
- \( q \): the probability that the attacking node finds the next block.
- \( q_z \): the probability that the attacking node can still find the next block when it lags behind \( z \) blocks.

Assuming that the time for honest nodes to generate a block is certain, the process of attacking nodes conforms to Poisson density distribution, and the expected value \( \lambda \) is

\[
\lambda = \frac{q}{p}
\]

Then, the probability that the attacking node successfully tampers with the blockchain is

\[
P = \sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \left\{ \begin{array}{ll}
\frac{q}{p} & k \leq z \\
1 & k > z
\end{array} \right\}.
\]

Simplification finishing is

\[
P = 1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left( 1 - \left( \frac{q}{p} \right)^{z-k} \right).
\]

Through calculation and analysis, the changing trend between the success probability of tampering with blocks and the number of attack blocks is shown in Figure 8.

From the trend chart of the probability of the attacker finding the next block, it can be seen that with the increase of \( z \) (the number of blocks in which the attacking node lags behind the honest node increases), the probability of the attacker catching up is lower. This ensures that not any node can unilaterally modify the transaction information. Otherwise, the cost of breaking the rule will be far greater than the cost of its regular operation, and the income realized by breaking the rule is not as significant as the income obtained by its regular operation. The implicit idea is the idea of game theory.
5.2.2 Model Security Analysis. Finite State Machines (FSMs) can be used to verify the security of a model. FSMs have limited state variables and state transition functions. To demonstrate the security of FSMs, we must first prove that the system’s initial state is safe. Secondly, it is necessary to explain that all state transition functions are also safe. At the same time,

![Diagram of improved access process](image)

**Figure 6: Improved access process.**

![Diagram of right transfer process](image)

**Figure 7: Right transfer process.**

| Model  | Distributed | Lightweight | Flexibility | Dynamics | Fine-granted |
|--------|-------------|-------------|-------------|----------|--------------|
| RBAC   | √           | √           | √           | √        | √            |
| ABAC   | √           | √           | √           | √        | √            |
| UCON   | √           | √           | √           | √        | √            |
| CapBAC | √           | √           | √           | √        | √            |

**Table 2: Comparison of different access control schemes.**

| Model     | Feature                                                                 |
|-----------|-------------------------------------------------------------------------|
| Ref [26, 27] | Access control, judgment, and registration through smart contracts       |
| Ref [28]   | Recording all information about data changes through smart contracts    |
| Ref [29]   | Realizing the cross-organizational utilization of roles through Ethereum’s smart contract technology |
| Ours       | Realizing authority transfer, combined with the ABAC model, through the smart contract |

**Table 3: Application of smart contract in access control.**

5.2.2 Model Security Analysis. Finite State Machines (FSMs) can be used to verify the security of a model. FSMs have limited state variables and state transition functions. To demonstrate the security of FSMs, we must first prove that the system’s initial state is safe. Secondly, it is necessary to explain that all state transition functions are also safe. At the same time,
The probability of the attacker finding the next block is 0.1
The probability of the attacker finding the next block is 0.3
The probability of the attacker finding the next block is 0.4
The probability of the attacker finding the next block is 0.5

Figure 8: Change trend chart of tampering with blocks.

Table 4: Model state transition table.

| Input set U | Output set T |
|-------------|--------------|
| $V_0$: initial state of system | $T_0$: send transaction and call request function. |
| $V_1$: GrantAccess | $T_1$: get the attributes of RO and RE. |
| $V_2$: GetAccess | $T_2$: emit the policy decision event. |
| $V_3$: PAP emits events | $T_3$: emit the request result event. |
| $V_4$: access | $T_4$: send a new access request. |
| $V_5$: DeliveryAccess | $T_5$: TS sends the information of right transfer message to CS. |
| $V_6$: PIP gets information | $T_6$: return the decision result to TS. |
| $V_7$: delivery token | $T_7$: send a new access request. |
| $V_8$: end state of system | $T_8$: end the whole process of access control. |

Figure 9: Schematic diagram of state transition.
time, meeting the above two conditions can ensure that the system will be in a safe state no matter which transition function is called.

The formal description is five-tuple $M = \{V, U, F, V_0, V_a\}$, where $V = \{V_0, V_1, \ldots, V_n\}$ is a finite state set. Under the state transition function, one state can be converted to another state, and the system can only be in a certain state at any time. $U$ is the system input and a collection of a series of attributes. $F$ is the state transition function, that is, under the drive of input from one state to another. The state transition functions of this model are $F_1: V_0 \times U_0 \rightarrow V_1; F_2: V_1 \times U_1 \rightarrow V_2; F_3: V_2 \times U_2 \rightarrow V_3; F_4: V_3 \times U_3 \rightarrow V_4; F_5: V_4 \times U_4 \rightarrow V_5; F_6: V_5 \times U_5 \rightarrow V_6; F_7: V_6 \times U_6 \rightarrow V_7; F_8: V_7 \times U_7 \rightarrow V_8$. $V_0$ is the initial state of the system and $V_a$ is the end state of the system. According to the status variables defined above, the settings of access control status are shown in Table 4.

In the initial state, there is no access request operation. The access control contract and right transfer contract are not called. The attribute set is empty, which meets the security requirements of FSM. Therefore, the initial state $V_0$ is safe. The state transition process is shown in Figure 9.

When RE initiates a request, the system state $V_1$ is secure. Blockchain is a shared database. The data or information stored in it has the characteristics of “unforgeability,” “traceability,” and “openness and transparency,” so the system state $V_2$ is safe. CS views the access authorization policy through the PAP node and judges whether there is an access right. CS automatically determines this process. When the PIP’s acquired properties and policy information match the access control conditions, access is allowed, satisfying the FSM definition of the security system. If not, the system security remains unchanged, so the state $V_3$ is safe. If RE wants to transfer the current permissions to other entities, it needs to make a decision through TS, so the state $V_4$ is secure. After verifying that the right can be transferred, complete the token transfer. Otherwise, the system will remain in a safe state. To sum up, the model meets the safety of initial state $V_0$ and model state transition function, so the whole model is safe.

6. Conclusions

Traditional access control models, including Discretionary Access Control model (DAC), Mandatory Access Control model (MAC), Access Control List (ACL), and Role-Based Access Control model (RBAC), have difficulty in meeting the requirements of the rapid development of IoT. Although Attribute-Based Access Control model (ABAC) can achieve fine-grained access control, it still lacks security and effectiveness. This paper takes the ABAC model as the overall design policy and improves it using blockchain technology. It also copes with the problem of massive terminal nodes of IoT and realizes the dynamic transmission of access rights so that the access rights can be directly transmitted from the current owner to others without the participation of resource owners, which enhances the efficiency of authority transmission. At present, the research has a single-use scenario. With the rapid development of the IoT, data has become the most crucial resource in the network. Data exchange and sharing among various organizations have become essential, but the complex and severe network security situation remains unchanged. Therefore, ensuring data sharing and access control among organizations is the work to be carried out in the future.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially funded by the Key Research and Development Program of Shandong Province (Nos. 2019GNC106027 and 2019JZZY010134) and the Natural Science Foundation of Shandong Province (Nos. ZR2020MF058 and ZR2020MF029).

References

[1] J. Shi and R. Li, “Survey of blockchain access control in Internet of things,” Journal of Software, vol. 30, no. 6, pp. 1632–1648, 2019.

[2] L. Ouyang, S. Wang, Y. Yuan, X. Ni, and F. Wang, “Development status of blockchain smart contract: architecture, application and development trend,” Journal of automation, vol. 45, no. 3, pp. 445–457, 2019.

[3] X. Han, Y. Yuan, and F. Wang, “Blockchain security: research status and prospects,” Journal of automation, vol. 45, no. 1, pp. 206–225, 2019.

[4] A. Yavari, A. S. Panah, D. Georgakopoulos, P. P. Jayaraman, and R. V. Schyndel, “Scalable role-based data disclosure control for the Internet of things,” in Proceedings of the IEEE 37th Int’l Conf. on Distributed Computing Systems, pp. 2226–2233, IEEE, Atlanta, GA, USA, June 2017.

[5] Q. Liu, H. Zhang, J. Wan, and X. Chen, “An Access Control Model for Resource Sharing Based on the Role-Based Access Control Intended for Multi-Domain Manufacturing Internet of Things,” IEEE Access, vol. 5, no. 99, pp. 7001–7011, 2017.

[6] M. Jemel and A. Serhrouchni, “Decentralized access control mechanism with temporal dimension based on blockchain,” in Proceedings of the 2017 IEEE 14th Int’l Conf. on E-Business Engineering (ICEBE), pp. 177–182, IEEE, Shanghai, China, November 2017.

[7] O. Aissam, E. S. H. Hamza, 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] H. Ouechtiati and N. B. Azzouna, "Trust-ARAC towards an access control system for the Internet of things," in Proc. Of the Int’l Conf. on Green, Pervasive, and Cloud Computing-Springer-Verlag, Berlin, Germany, 2017.

[9] A. Liu, X. Du, N. Wang, and S. Li, "Survey on information security techniques for blockchain technology," Ruan Jian Xue Bao/Journal of Software, vol. 29, no. 7, pp. 2092–2115, 2018.
[10] Y. Nakamura, Y. Zhang, M. Sasabe, and S. Kasahara, “Exploiting smart contracts for capability-based access control in the Internet of things,” Sensors, vol. 20, no. 6, pp. 33–41, 2020.

[11] Q. F. Shao, C. Q. Jin, and Z. Zhang, “Blockchain technology: architecture and progress,” Journal of Computer Science, vol. 41, no. 5, pp. 969–988, 2018.

[12] 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, 2018.

[13] S. Roy, A. K. Das, S. Chatterjee, N. Kumar, S. Chattopadhyay, and J. J. P. C. Rodrigues, “Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications,” IEEE Transactions on Industrial Informatics, vol. 15, no. 1, pp. 457–468, 2018.

[14] H. Li, M. Zhang, D. Feng, and Z. Hui, “Research on access control of big data,” Chinese Journal of Computers, vol. 40, no. 1, pp. 72–91, 2017.

[15] L. Fang, L. Yin, Y. Guo, and B. Fang, “Research review on Key Technologies of attribute based access control,” Journal of Computer Science, vol. 40, no. 7, pp. 1680–1698, 2017.

[16] K. Yang, Q. Han, H. Li, K. Zheng, Z. Su, and X. Shen, “An efficient and fine-grained big data access control scheme with privacy-preserving policy,” IEEE Internet of Things Journal, vol. 4, no. 2, pp. 563–571, 2017.

[17] S. Qiu, D. Wang, G. Xu, and S. Kumari, “Practical and provably secure three-factor Authentication protocol based on extended chaotic-maps for mobile lightweight devices,” IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 2, pp. 1338–1351, 2022.

[18] Q. Wang, D. Wang, C. Cheng, and D. He, “Quantum2FA: efficient quantum-resistant two-factor Authentication scheme for mobile devices,” IEEE Transactions on Dependable and Secure Computing, p. 1, 2021.

[19] Z. Li, D. Wang, and E. Morais, “Quantum-safe round-optimal password authentication for mobile devices,” IEEE Transactions on Dependable and Secure Computing, 2020.

[20] M. Zhao, B. Li, and W. Li, “Analysis and application research of ECC encryption algorithm,” Network security technology and application, vol. 7, no. 12, pp. 35-36, 2018.

[21] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, “PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs,” International Journal of Intelligent Systems, vol. 36, pp. 3596–3612, 2021.

[22] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, “‘PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs,” International Journal of Intelligent Systems, 2021.

[23] 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.

[24] N. Rifi, E. Rachkidi, N. Agoulmine, and N. C. Taher, “Towards using blockchain technology for IoT data access protection,” in Proceedings of the 2017 IEEE 17th Int’l Conf. on Ubiquitous Wireless Broadband (ICUWB), pp. 1–5, IEEE, Salamanca, Spain, September 2017.