applied sciences

MDPI

Article

An Internet of Things Access Control Scheme Based on Permissioned Blockchain and Edge Computing

Lihua Zhang [1,*], Boping Li [1], Haodong Fang [2], Ganzhe Zhang [2] and Chunhui Liu [1]

[1] School of Software, East China Jiaotong University, Nanchang 330013, China
[2] School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang 330013, China
* Correspondence: lhzhang@ecjtu.edu.cn

Abstract: In the IoT (Internet of Things) environment, the existing access control schemes for device resources have some problems, such as poor scalability, high latency, security, and dynamics. Combining the advantages of the permissioned blockchain and edge computing, an access control scheme for the Internet of Things based on the permissioned blockchain and edge computing is proposed. By authenticating the user's identity at the edge, the user's identity is reliable and the response time is improved. In the ABAC (Attribute Based Access Control) model, the blockchain is regarded as a trusted entity, and the access control policy is written into a smart contract and deployed on the blockchain for calling. Most of the existing consensus algorithms have the problems of low throughput and scalability. A Kraft (Kademlia–Raft) consensus algorithm is introduced to solve the above issues. Security analysis and experimental results show that the scheme can achieve fine-grained, dynamic access control, has high throughput and low latency, and ensures security and reliability.

Keywords: Internet of Things (IoT); blockchain; access control; smart contract; edge computing

check for updates

Citation: Zhang, L.; Li, B.; Fang, H.; Zhang, G.; Liu, C. An Internet of Things Access Control Scheme Based on Permissioned Blockchain and Edge Computing. *Appl. Sci.* **2023**, *13*, 4167. https://doi.org/10.3390/app13074167

Academic Editor: Gianluca Lax

Received: 1 March 2023
Revised: 21 March 2023
Accepted: 23 March 2023
Published: 24 March 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As soon as the concept of IoT was put forward [1], it aroused widespread concern. The IoT generates a large amount of data daily, which is stored in the database, and which ultimately serves more market players, such as users and suppliers. It contains a large amount of personal information, and if this private information is leaked, it will bring huge losses to the users [2]. As one of the cornerstone applied sciences of information protection, access control ensures that information can solely be accessed through customers with corresponding permissions [3]. Access manipulation refers to the technology where the machine limits the get admission and the use of sources in accordance with the user's identification and authority. The foremost purpose is to restrict the get entry of the situation to the object, so as to ensure the use of records and assets inside the legal scope.

Traditional IoT access control schemes are mainly based on the access control models, including RBAC [4] (Role Based Access Control), ABAC [5] (Attribute Based Access Control), UCON [6,7] (Usage Control), and CapBAC [8] (Capability-Based Access Control). RBAC, ABAC, and UCON all require a centralized server to complete the authorization decision. CapBAC has realized lightweightlightweight dispensation to get admission in order to manage in the IoT environment [9], and it helps with dynamics and scalability.

Because the abovementioned access control model does not further verify the legitimacy of visitors, it will lead to information leakage, and the identity and behavior process of the responsible person cannot be effectively traced, so the spread of losses cannot be stopped in time [10]. The blockchain is a decentralized distributed database, which can use the consensus authentication process to record the running process of information, and effectively solve the traceability problem of the behavior process in the blockchain system. It is particularly important to design an access control scheme based on the blockchain

in which only specific users have the right to access any kind of data. Because of a large number of resource-constrained devices in the IoT, edge computing can be combined to achieve low latency. Edge computing refers to an open platform that transfers computing and communication resources from the cloud to the edge nodes of the network in order to provide faster services and computing responses for edge users [11]. Because of the characteristics of the IoT, this paper uses the permissioned blockchain. The permissioned blockchain refers to that of each node participating in the blockchain device to which it is licensed, and the unauthorized node cannot be accessed on the machine [12].

The resource owner first publishes the access control policy of resources in the blockchain. In order to access data, the resource requester will request access to blockchain data by using smart contract API and whether access permission is granted by the access control policy running in the blockchain. The role of the blockchain in access control is not only to store access control policies and permission transaction information, but also to provide functions such as automatically executed access control policies and grant permissions.

The main contributions of this paper are as follows:

(1) A scheme of access control in the IoT environment is proposed, which combines edge computing and license chain, and realizes access control with low delay, fine granularity. and dynamic performance.
(2) The solution uses blockchain networks as an interactive medium, smart contracts as the driving force, and the KRaft consensus algorithm as the execution standard to reasonably improve the existing attribute-based access control model, achieving policy implementation goals such as automatic decision-making and authenticity, credibility, transparency, traceability, and auditability.
(3) Edge computing is introduced to ensure relatively low computing and communication costs, because the edge side has lower latency and higher availability than the cloud side network, and data will not be attacked during transmission.
(4) The security analysis of the scheme is carried out, and the performance of the scheme is compared with other schemes through experimental simulation. Experiments show that the scheme has several strengths, such as throughput and low latency with security and reliability.

2. Related Work

Access control is a protection manipulation mechanism to ensure that licensed customers can acquire the required assets whilst denying unauthorized customers access to any resources. Researchers have proposed many extraordinary access control models.

Reference [13] uses the blockchain to solve the problem of cross-organization access control in RBAC, and realizes the cross-organization authentication of user roles. The model proposed in reference [14] automatically revokes and reconfigures user roles by observing user access behavior. The model is used to provide "role-assignment-service" in the cloud to optimize the cost of built-in roles. RBAC uses a central server to process the access requests of all users, and usually assigns the same permissions to a class of roles, which reduces the complexity of permission granting and the overhead of management, but it is lacking a dynamic. There will also be the role explosion, which is not suitable for the implementation of security policies in complex and fuzzy IoT scenarios. At the same time, a large number of information, such as users and roles as well as roles and permissions are calculated and saved, which brings greater challenges to the limited computing and storage resources of Internet of Things devices.

ABAC is more suitable for complex scenarios than RBAC, can provide more fine-grained access control, and has high flexibility. In reference [15], the policy language of attribute rules and the method of solving the policy of conflict and redundancy are proposed, which simplifies the complexity of traditional ABAC. In reference [16], the attribute-based access control mechanism is used to verify the user's identity, focusing on how to achieve data security access for low-power IoT devices. Like RBAC, ABAC has a trusted central server.

CapBAC has achieved lightweight, distributed, dynamic, and scalable properties in the IoT environment [17], which is considered as a promising solution for the IoT system. A decentralized IoT access control method based on the blockchain has been proposed [18]. Each participant (i.e., person and IoT device) in the IoT system is identified by at least one unique did, and the access control of a specific IoT device service is determined by the ownership/capability token and credential.

Although CapBAC's dispensed diagram avoids the single factor of failure brought on by using the use of centralized servers when CapBAC implements dispensed gain admission to manipulating selections on light-weight gadgets in the IoT, lightweightlightweight units can now not guarantee their personal security, and it is feasible for attackers to threaten the protection of get admission to manipulate through susceptible safety IoT devices. Therefore, disbursed CapBAC can no longer clear up the trouble of IoT access control to manipulate it in an untrusted environment.

In this paper, we combine ABAC model with the blockchain; take advantage of the characteristics of the blockchain, such as decentralization, tamper proof and traceability; treat the blockchain as a trusted entity; and write the access control policy as a smart contract deployed on the blockchain. This can not only solve the problem of untrusted access control in the IoT, but can also achieve fine-grained and dynamic access control. At the same time, the introduction of edge computing, a new computing mode, can reduce the delay as well as improve the response time and the quality of service.

3. Scheme Framework

3.1. ABAC Basic Information

ABAC model based totally on the subject, object, permission and environment [19] takes the subject attribute and object attribute as key factors to get entry to manage the decision and judge whether or not to supply the right of entry to the permission through the subject attribute set. The core concept of ABAC is attribute, which includes subject attribute, object attribute, permission attribute, and environment attribute. Subject attribute is the entity that can access and operate the protected resources, such as user, process, etc. Object attribute is the entity that can be accessed and operated, such as document, picture, and video data resources. The permission attribute refers to various operations on object resources, such as reading, writing, modifying, executing, etc. The environment attribute refers to the environment, time, context information, etc. in the process of the principal accessing resources. The basic concepts involved in the ABAC model are summarized in Table 1.

Table 1. Notation and Description involved in ABAC model.

Notation	Description
OAR	Original Access Request
AAR	Attribute based Access Request
PDP	Policy Decision Point—the fundamental feature is to take a look at whether or not the foremost in AAR meets the coverage rules, and ship the outcomes lower back to PEP for execution
PAP	Policy Authority Point—the access control policy for PDP decision-making is provided, in which the decision rules, basis and related constraints of PDP are defined
PEP	Policy Enforcement Point—the essential characteristic is to set up AAR, ship AAR to PDP, and execute the choice end result of PDP
PIP	Policy Information Point—used to manipulate the key attribute facts of the system used for get right of entry to control judgment

3.2. Blockchain

Blockchain is a new distributed computing and storage paradigm integrating a variety of existing technologies [20], which has the characteristics of decentralization, non-tampering, and traceability. According to the degree of decentralization, it can be divided into the permissionless blockchain and the permissioned blockchain. In permissionless

blockchain, it has no control over who can join the network. All nodes in the permissionless blockchain take part in the wide consensus main block verification. Due to the nature of the permissionless blockchain network, the consensus protocol ought to be extra stringent to keep away from any collusion strife between nodes. Bitcoin is a common example of the permissionless blockchain community that operates using the workload proof consistency approach. The permissioned blockchain is a semi-open blockchain, which is controlled and supervised. Only the node that has been granted permission can access it, while the node without permission cannot. Due to the regulatory nature of the network, compared with the permissionless blockchain, the consensus algorithm is simpler and less intensive.

According to distinct blockchain applications, the records amassed in the block can be of specific fact types. The hash function converts a block and its contents into a special constant-size output that is restrained to that block [21]. Each current block has to refer to the hash price of the preceding block. This will result in an orderly association of blocks to create a blockchain, as proven in Figure 1. Changing the hash value of any block will invalidate subsequent blocks due to the fact the hash values of preceding blocks have to be changed. This characteristic enhances the safety of the block and protects it from any variability and tampering. The blockchain permits the layout of a complete and strong point-to-point network, while eliminating any single-factor errors in the setup.

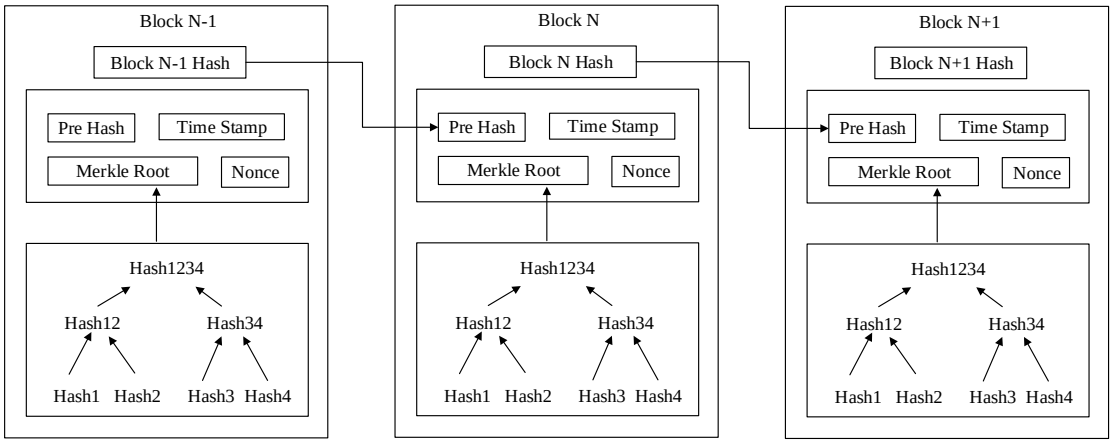


Figure 1. Process of secure storage.

3.3. Bilinear Mapping

Let G_1,G_2,G_T be three multiplicative cyclic groups, whose order is a prime q, g_1, and g_2,g_3 be generators of three cyclic groups. Let $e : G_1 \times G_2 \to G_T$ be a map with the following properties:

(1) Bilinearity: $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $g_1 \in G_1$, $g_2 \in G_2$, $a, b \in Z_q$.
(2) Non-degeneracy: There exists $g_1 \in G_1$, $g_2 \in G_2$, such that $e(g_1, g_2) \neq 1_{G_T}$. That is, for any element in G_1,G_2, the identity element in G_T is not obtained by this mapping.
(3) Computability: There is an efficient algorithm to compute $e(g_1, g_2)$ for all $g_1 \in G_1$, $g_2 \in G_2$.

3.4. Decision Bilinear Diffie Hellman Hypothesis

Let g be generator of group G, given random value $x, y, z \in Z_p$ and a bilinear mapping $e : G_1 \times G_1 \to G_2$, while G_2 is a group, as well as $L \in G_2$. The DBDH assumption is that no probabilistic polynomial-time algorithm Q can distinguish with more than a negligible advantage between tuple $(g, g^x, g^y, g^z, e(g, g)^{xyz})$ and tuple (g, g^x, g^y, g^z, L), while the advantage δ of algorithm Q is as follows:

$$\delta = |\Pr[Q(g, g^x, g^y, g^z, e(g, g)) = 0] - \Pr[Q(g, g^x, g^y, g^z, L) = 0]| \quad (1)$$

4. System Model

Figure 2 illustrates the system model of the scenario. As the resource requester, the user is the subject of access control, and the data perceived and stored by the IoT devices is the object resource. The proposed model mainly includes PEP, PDP, PAP, and PIP.

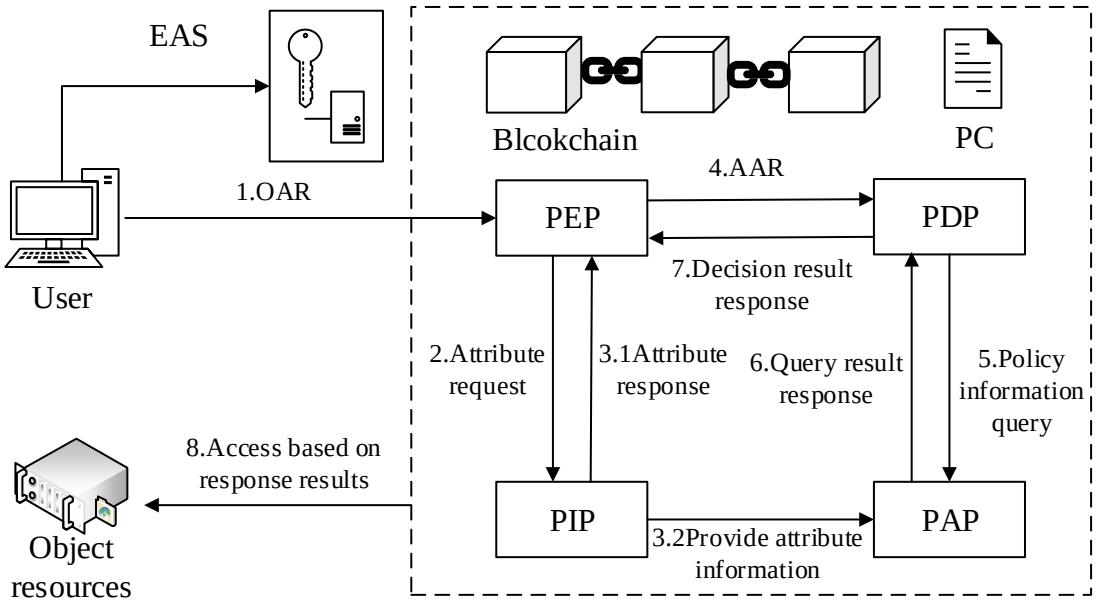


Figure 2. System model.

Firstly, the system should be initialized and executed by the edge server to generate the public parameters required by the system to implement the corresponding encryption function. Secondly, users and various terminal devices can use their own ID to register on the network. Users then need to log in for authentication. Only after passing the authentication can they make access requests. The user first requests access to the device service from the object resource. After the access request is passed, the device can be operated. This stage is completed by calling the smart contract on the blockchain. The smart contract defines policies and constraints related to privileges and services.

When different application devices make access requests to system devices (IoT devices), the challenge first locates the proxy device and makes the request, and the PEP is responsible for receiving and processing these requests, before it interacts with different practical modules to acquire useful resource access.

The PDP and PAP features are described in detail in Table 1. The key statistics of access control in the main are in the attribute records managed by the means of PIP, and the attribute records are stored on the blockchain to facilitate the well-timed call of the policy contract (PC), while, at the identical time, using the traits of the blockchain to efficaciously forestall malicious nodes from tampering with the attribute statistics.

Due to the susceptible storage and computing potential of most IoT units and the wide variety of IoT devices, the IoT gateway has the ability of both data caching and local computing. The IoT gateway is used as a proxy device to manage IoT devices. The ABAC model integrated into the blockchain retains the advantages of the traditional ABAC model, such as fine-grained access control, and adds some new advantages, such as the blockchain used to file the distribution of attributes, fending off the single factor of failure and also fact tampering.

4.1. User Registration

Each user has its own identity ID. It is necessary to register the user's identity ID first, generate the user's unique identity DUID, and then store it on the blockchain. To ensure

the legitimacy of users, users must register on the Edge Authentication Server (EAS), as shown in Figure 3. The specific process is as follows:

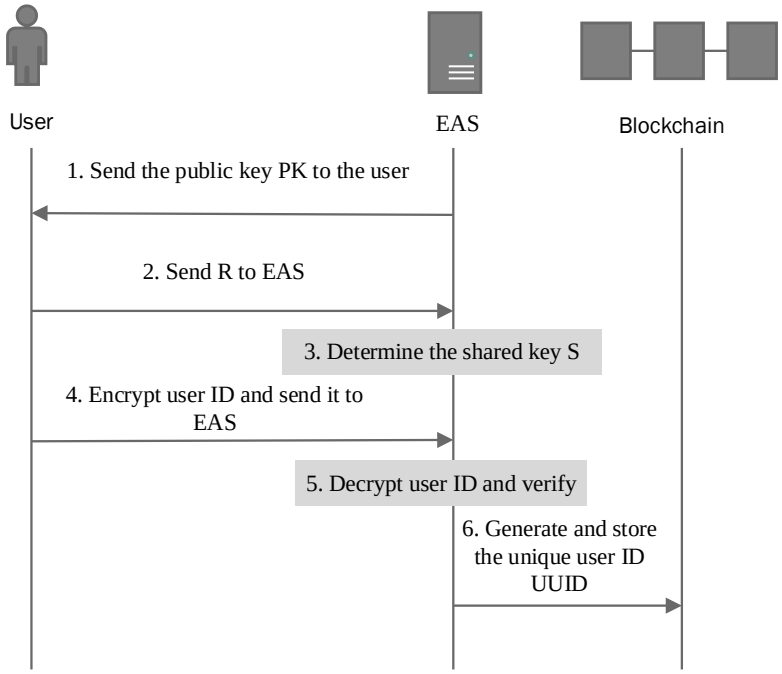


Figure 3. User Register.

(1) EAS generates a random number r_i, public key $PK_i = r_i \cdot G$, and sends the public key PK_i to the user.
(2) The user generates a random number d_i through the mobile device, calculates $R_i = d_i \cdot G$ and $S_i = d_i \cdot PK_i$ sends R_i to EAS.
(3) EAS can determine the shared key:

$$S_i = d_i \cdot PK_i = d_i \cdot (r_i \cdot G) = r_i \cdot (d_i \cdot G) = r_i \cdot R_i \quad (2)$$

(4) The user selects an identity ID_i, calculates $H(ID_i)$ with the hash function SHA256, encrypts $H(ID_i)$ with the shared key S_i, gets $Enc(S_i, H(ID_i))$, and sends it to EAS.
(5) EAS receives the encrypted message, decrypts $Dec(S_i, H(ID_i))$ with the shared key S_i, and queries whether the user ID_i is in the member list of the legal user.
(6) If it exists, a unique user ID $UUID_i$ is generated for it and stored on the blockchain. If it does not exist, it indicates that the user is illegal and disconnected.

At the registration stage, it uses a random number, a hash algorithm, and an elliptic curve integrated encryption algorithm to both encrypt and decrypt the terminal equipment and user information, and stores them in a decentralized and tamper-proof blockchain network.

4.2. User Authentication

Users need to be authenticated by the edge authentication server (EAS) before they initiate the access request. Only after the authentication is successful can they access it. The detailed certification process is as follows:

(1) The user inputs him/her through the mobile device, and then the mobile device connects to the edge server and enters the blockchain network.
(2) The edge server verifies whether there is a corresponding $UUID_i$ through the decoding function $f(UUID_i) = ID_i$. If so, the user passes the authentication. Otherwise, the session ends.

(3) The edge server verifies whether there is a corresponding one through the decoding function. If so, the user passes the authentication. Otherwise, the session ends.
(4) The user authentication process is recorded in the blockchain and cannot be tampered with.

The user logs in directly with his public and private keys as the username and password during the next authentication. Because the user ID and the user's unique identifier UUID are one-to-one mappings, the illegal connection of malicious users can be avoided, and the common network attacks in wireless networks, such as man-in-the-middle attacks, can be effectively avoided.

Through the edge end rather than the cloud center to authenticate the terminal equipment, the authentication of the terminal equipment is localized, which reduces the network burden of the cloud center and improves the efficiency of authentication.

4.3. Smart Contract System

Smart contract is a virtual account with the contract code and storage space in the blockchain. To create a smart contract, it is necessary to write, compile, and deploy a smart contract [22]. Both the advent and execution of smart contracts depend on blockchain protocols, which can guarantee the execution of smart contracts. Access control policies are then turned into smart contracts and they are installed on the blockchain.

There are three sorts of smart contracts involved in this scheme—an administration contract (MC) is used to control different contracts in the system, a permission decision contract (PDC) is shown in the Algorithm 1, which is responsible for obtaining attribute information from the PIP and making final access control decisions, and a massive variety of coverage contracts (PCs) are used to put into effect precise attributes based totally on access control policies.

MC is accountable for the administration of PDC and PCs in the system. The principal ABI are:

getCheckContract(): The accountability of this ABI is to attain the tackle of the PDC.

getContract(): The accountability of this ABI is to reap the tackle of PC.

According to the acquired subattribute and objattribute, PCs judge whether the subject has access to the object, and the result is "true" or "false".

Algorithm 1 Permission decision contract

Input: AAR
Output: Policy decision result, Permit or Deny
1: Permit_Result_Set = null
2: Deny_Result_Set = null
3: result = PolicyDecide(AAR)
4: **if** (result = Permit) **then**
5: Permit_Result_Set.add()
6: **else**
7: Deny_Result_Set.add()
8: **end**
9: **if** (Permit_Result_Set $\neq$ null && Deny_Result_Set == null) **then**
10: **return** Permit
11: **else**
12: **return** Deny
13: **end**

4.4. Access Control Process

The scheme simulates the entire access request method of the user device (subject), inquiring for access to the sources of the IoT device (object), and the particular access control manner is proven in Figure 4.

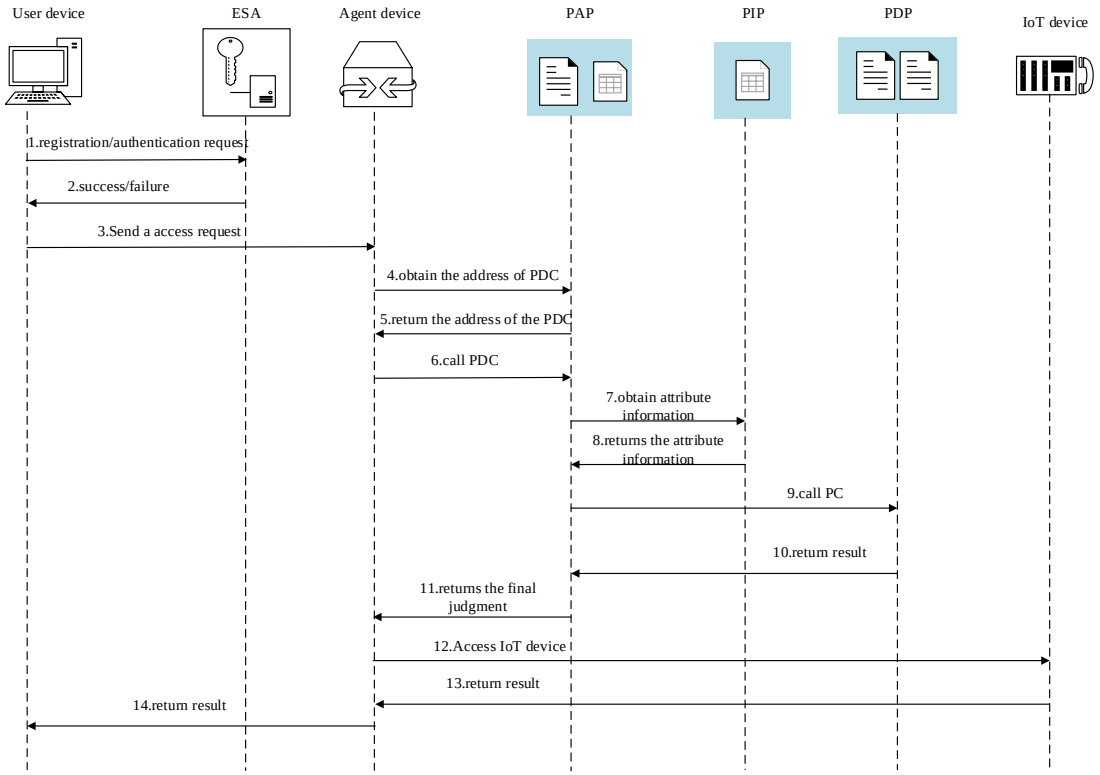


Figure 4. Access control process.

Step 1: The subject initiates a session creation communication request to ESA, provides its own identity ID, and sends a registration/authentication request.

Step 2: ESA returns the result of authentication to the principal.

Step 3: After the authentication end result is verified, the guest agent device will acquire the statistics data sent through the subject.

Step 4: After receiving the access request from the subject, the agent device obtains the address of PDC.

Step 5: The address of the PDC will be sent to the proxy device by the Management Contract (MC) on the PAP.

Step 6: The proxy device decides access based on key information by calling the PDC on the PAP.

Step 7: After the PDC receives the access decision task, it obtains the device attribute information in the PIP.

Step 8: PIP returns attribute information to the PDC.

Step 9: The PDC makes access control judgments based on the relevant information obtained.

Step 10: The PC determines the access control and feeds the result back to the PDC as "true".

Step 11: The PDC makes a final judgment on the feedback results and returns them to the proxy device.

Step 12: After waiting for the access authentication to be finally passed, the proxy device starts accessing the guest resources.

Step 13: The proxy device performs operations on the guest resource and receives the returned operation results.

Step 14: The result of the operation is returned to the principal.

5. KRaft Consensus Algorithm

Consensus algorithm is the key technology in the blockchain [23]. Because of the admission mechanism of the license chain, all nodes in the blockchain network are honest

and credible nodes, and there is no Byzantine fault tolerance problem. Only unexpected scenarios such as network interruption, disk failure, and machine downtime should be considered [24]. The Kraft consensus algorithm is based on an improvement of the Raft algorithm, using Kademolia's efficient log replication logic to solve the parallel and leader overload problems, improving the efficiency and scalability of the blockchain. In this paper, the deployment of smart contract access control policy, access authorization, access authorization logging, and access logging are all recorded into the object blockchain and log blockchain through the KRaft consensus algorithm.

The KRaft consensus algorithm is introduced to solve the problems of low throughput and scalability of the existing consensus algorithm and improve the scalability of the license chain. The following explains how to log access into the blockchain using the KRaft algorithm.

The KRaft consensus algorithm [25] consists of two essential processes: Leader election and log replication. During the device initialization phase, all nodes act as candidate nodes and ship election requests to all different nodes. When a Candidate node is supported by greater than half of the different nodes, it will be chosen as the Leader node. If no candidate node meets the election criteria, the election will take location once more after a positive quantity of built-in clock. For Leader node failure or downtime due to exterior reasons, the state of affairs is equal as it is throughout the re-election transition phase. Because all different nodes end up receiving records from the Leader node, they enter the re-election segment early. The leadership election method is proven in Figure 5.

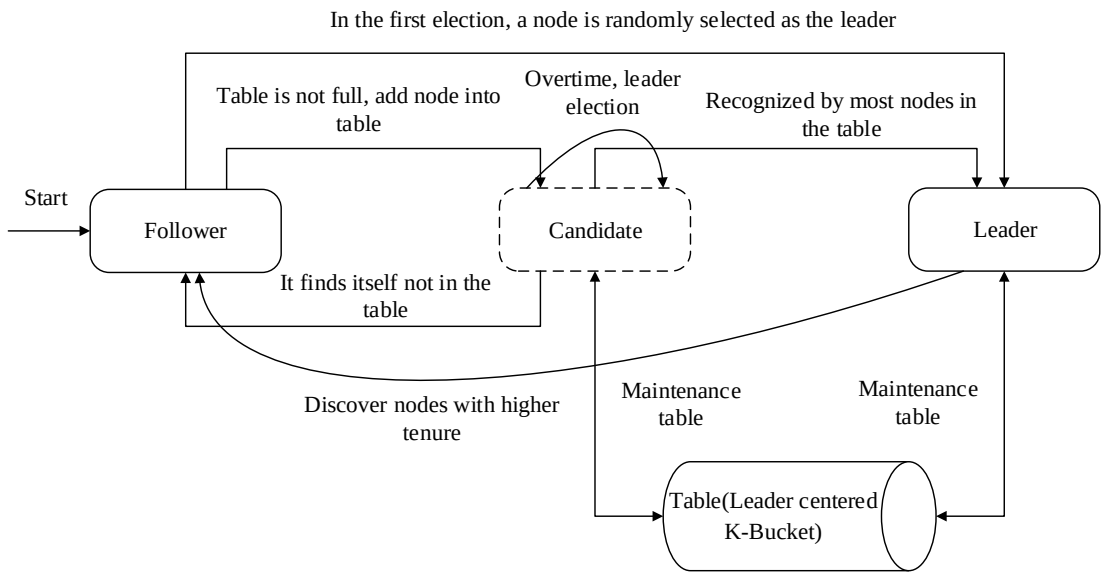


Figure 5. Leader election process.

The core consensus process of the KRaft algorithm is the log replication process, as shown in Figure 6. It can be briefly described as follows: (1) The client initiates an access logging request to the Leader node. (2) The Leader node adds an entry for the request and broadcasts it to the Follower node. (3) The Follower node stores the entries locally and informs the Leader node. (4) After receiving more than half of the replies, the Leader node confirms the request and returns it to the client. The confirmation message is sent to the Follower node in the next broadcast.

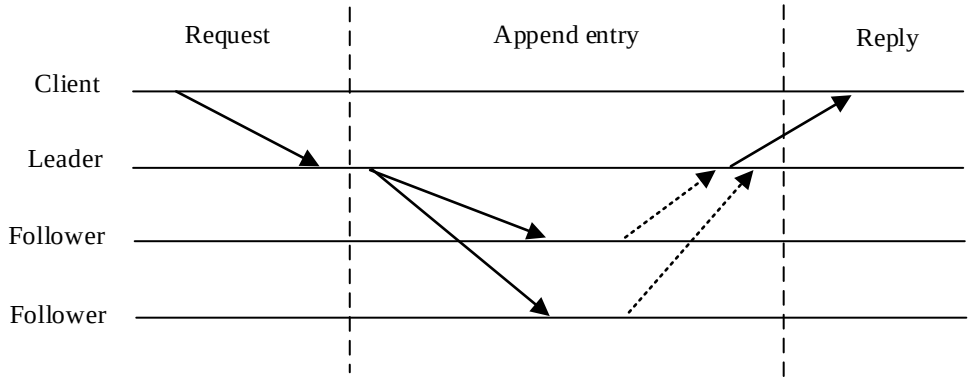


Figure 6. Log replication process.

The Leader node sends extra log entries repeatedly to make sure that the log entries stored in the Follower node are consistent with itself and attain a consensus. Log entries will also no longer be synchronized immediately after comments from the Followers node to the Leader node. We should wait for the next round of log entries furnished by means of the Leader node to incorporate confirmation that the Leader node has synchronized the previous round of entries, and then synchronize the entries.

In the process of access logging, the Leader node packages the collected access logs into blocks. Once a consensus is reached, the block is added to the blockchain.

6. Security and Performance Analysis

6.1. Scheme Comparison

The scheme proposed in this paper is decentralized due to the introduction of the permissioned blockchain. Therefore, it can overcome the single point of failure problem and provide the main requirements of IoT access control, and because of the admission mechanism of the permissioned blockchain, it can provide higher security. Table 2 shows the comparison between the existing access control scheme and the scheme in this paper in realizing different functions.

Table 2. Function comparison.

Schemes	Decentralization	Flexibility	Traceability	Fine Grained	Scalability
Refs. [8,9]	×	√	×	×	commonly
Refs. [10,11]	×	√	×	√	commonly
Ref. [13]	√	√	×	√	commonly
This scheme	√	√	√	√	good

6.2. Security Analysis

(1) Confidentiality. It specifies the access control rights of different users to IoT data, and only authorized users can obtain data and perform related operations. Specifically, confidentiality requires the system to set corresponding authentication rules, access control, and review mechanisms. In this paper, the system architecture first authenticates the user at the edge, which eliminates the risk of malicious users attacking the management system from the outside. It presents a lightweight interface for the end user. The user does not need to join the blockchain and can access the data by interacting with the agent device.

Theorem 1. *If there is an adversary A who can win the security game with a non-negligible advantage ε in polynomial time, another adversary in polynomial time can be constructed to solve the difficult problem of dbdh with the advantage of* $\varepsilon/2$.

Proof. Select plaintext attack model, which is carried out through the interactive game between Challenger B and Adversary A, with the following steps:

(a) Setup. First, Challenger B runs the global initialization algorithm to generate the global parameter GP of the system, and then Adversary A specifies the set of destroyed AA. The AA initialization algorithm is implemented in each attribute center AA to generate public-private key pairs. For the destroyed AAS, the adversary can get the public key and private key; for normal AAS, the adversary can only get the public key.

(b) Phase 1. Adversary A asks Challenger B for the private key with attribute set $S = (y_1, y_2, \ldots, y_{q1})$ and the conversion key, but the attribute set cannot satisfy the access structure (A, ρ) to be challenged by Adversary A, and Challenger B runs the key generation algorithm to generate the private key and conversion key corresponding to attribute set s for adversary A.

(c) Challenge. Adversary A sends two equal length messages M_0, M_1 to Challenger B. When Challenger B receives the message, he tosses a fair coin randomly, the message is encrypted according to the result $b \in \{0, 1\}$, and then encrypted under the access structure (A, ρ) to generate the encrypted ciphertext CT. Challenger B then sends the challenge ciphertext to Adversary A.

(d) Phase 2. Repeat phase 1 and continue to ask Challenger B about the private key and the conversion key. The attribute set of the private key and the conversion key cannot meet the challenge of access structure (A, ρ).

(e) Guess. Guess $b' \in \{0, 1\}$ about ciphertext b output by adversary A. If $b = 0$, adversary A's advantage is ε, that $P_r[b' = b] = \frac{1}{2} + \varepsilon$. If $b = 1$, that $P_r[b' = b] = \frac{1}{2}$, So, the challenger's advantage in the DBDH game is:

$$Adv_A = P_r[b' = b] - \frac{1}{2} = \frac{1}{2}(\frac{1}{2} + \varepsilon + \frac{1}{2}) - \frac{1}{2} = \frac{\varepsilon}{2} \tag{3}$$

□

Therefore, if Adversary A overcomes our scheme with an advantage that cannot be ignored, Challenger B can solve the difficult problem of DBDH with the advantage of $\varepsilon/2$.

(2) Credibility. Based on the characteristics of the blockchain, it ensures that the contract is transparent, tamperproof, and traceable, and also ensures that the users and data owners execute the contract honestly. At the same time, the introduction of a third-party authentication entity is omitted to ensure that users can obtain accurate access results. That is, when the user initiates an access request, the smart contract will be triggered to operate. Based on the characteristics of the blockchain, accurate results can be guaranteed.

(3) Privacy. The privacy issues in the blockchain system can be divided into account privacy and transaction privacy, corresponding to the user's identity privacy and data privacy, respectively. In this data management architecture, the user's identity information, physical address, and IP address are not associated with the public information (such as the user's public key and account address) in the system, and no node can infer the user's real identity information from the public information in the system. The system runs an automatic trigger mechanism based on Smart contract, which reduces the risk of privacy leakage caused by human intervention. Therefore, user identity privacy has been well protected.

(4) Anti-conspiracy attack. In order to ensure correct access control, the scheme must be able to resist conspiracy attacks. Sometimes, these devices may collude with each other in an interest-driven attempt to verify valuable data with other devices, and none of which independently have a satisfactory set of attributes. However, in our scheme, subjects are accessing the IoT equipment resources through proxy devices, and the agent devices act as the implementation point of the strategy and interact with Smart contracts, which can resist the collusive attack among the devices of the IoT.

6.3. Performance Analysis

In order to test the performance of the access control mechanism proposed in this paper, an experimental environment of the alliance chain is built in this section. The experimental environment is shown in Table 3.

Table 3. Experimental environment.

Software/Hardware	Parameter
Operating system	Ubuntu Linux 18.04 LTS
CPU	AMD Ryzen5 3600X 3.6 GHz
Writing language	Solidity
Internal storage	8 GB
Raspberry Pi	3B+

The proposed system architecture is built using Ethereum as a platform for testing and evaluation. In the experiment, Ethereum nodes are divided into full nodes and light nodes. High-performance devices become full nodes, such as edge servers and gateways. The devices with ordinary performance become light nodes, such as sensors. The gateway is implemented by running the SDN controller in an independent virtual machine hosted on a Linux server. Raspberry Pi is configured as Raspbian 2019-09-26 as an edge server. Smart contracts are written in a stable format using Solidity v.0.4.20 which was proposed by Gavin Wood in August 2014. Distributed applications are deployed and compiled using the Truffle development suite.

In the access control mechanism based on the blockchain, access logging and access control policy deployment belong to the right blockchain. The access control policy is evaluated as executing the Smart contract on the blockchain and writing the log to the blockchain. The log is viewed as reading the data of the blockchain. The Ethereum Smart contract does not run directly on the node, but runs in the Ethereum virtual machine. The contract is stored on the blockchain of the Ethereum, and it is compiled into the bytecode of the Ethereum virtual machine, which runs through the virtual machine.

The performance of the access control mechanism based on the blockchain is evaluated by testing the speed of the reading/writing blockchain. The read-write speed of the blockchain is not only related to the hardware environment, but also to the number of nodes in the blockchain network. In this paper, we configure a sort node, with 5 and 10 peer nodes. Through simulation, we continuously send transactions to the blockchain at different speeds, including the reading blockchain and the writing blockchain. We test the speed of the read-write blockchain under different conditions.

(1) The speed of the reading/writing blockchain when five peer nodes are configured (as shown in Figure 7).

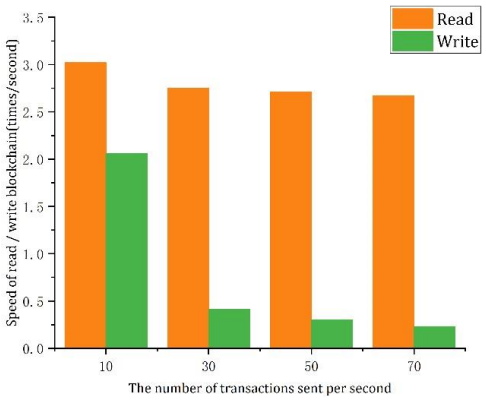


Figure 7. Read/write blockchain speed for 5 peer nodes.

(2) The speed of the reading/writing blockchain when ten peer nodes are configured (as shown in Figure 8).

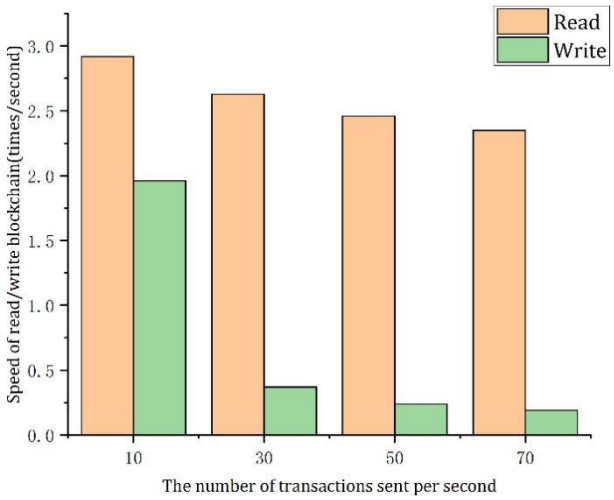


Figure 8. Read/write blockchain speed for 10 peer nodes.

As can be seen from Figures 7 and 8, whether there are 5 or 10 peer nodes, the speed of reading into the blockchain is faster than that of writing into the blockchain. The consensus mechanism of the blockchain determines the writing and the verification speed of permission transactions on the chain. Because the KRaft consensus algorithm used in this paper can provide high throughput and low latency, the speed of the reading/writing blockchain is faster than the general one.

The total time from sending the access request to the final response can be used as a measure of the system computing overhead. The access control model in cloud computing is based on the traditional access control model. It is improved on the traditional access control model to make it more suitable for the environment of cloud computing. The experiment compares this scheme with the cloud-based access control scheme. By changing the size of user data requests, we can observe the request processing time of the cloud-based scheme and this scheme. Processing time includes all the time from sending the access request to obtaining the final result. The request processing time for both schemes is shown in Figure 9 (the coordinate value is obtained by taking the logarithms of the bottom 10 as the actual value).

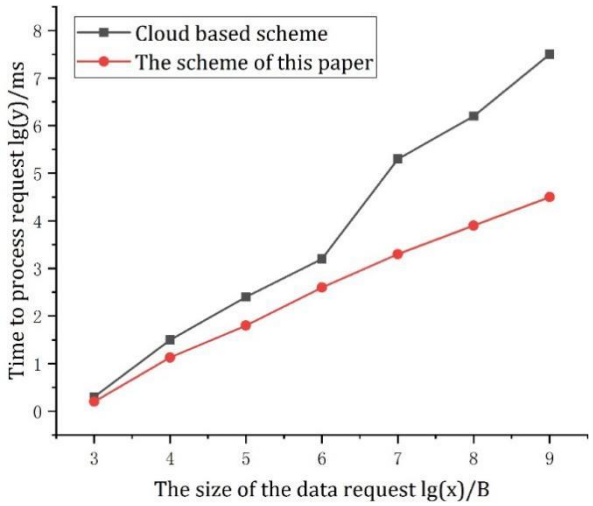


Figure 9. The processing time of the two schemes.

Figure 9 shows that when the data request size is set to less than 106 B, the processing time of the two schemes is very close and short. However, as the size of the data request increases, the processing time will rapidly widen the gap, which can be nearly three orders of magnitude. Obviously, the access control scheme in this paper can process the user's data request for the object resource more quickly with less computational overhead.

In order to verify the effectiveness of the access control mechanism, the following methods are used to test and compare the effectiveness of the access control policy, RBAC policy, ABAC policy, and CapBAC policy under different policies. The test content is the efficiency of policy adjudication in the access control mechanism. The policy decision delay is calculated based on the average response delay of all requests. In each simulation, the test set sample constructs 100 different access requests for 10 user IDs, each with five attribute values, and each request has five random sending opportunities. The experiment simulated the algorithm using the same dataset: when fully operational, the simulation results averaged 10 experiments, with 100 connection requests per experiment. The confidence interval of the 95% confidence interval was derived using the independent replication method. There are five requests for test base samples, corresponding to 100, 200, 300, 400, and 500 single policies. The experimental results are shown in Figure 10.

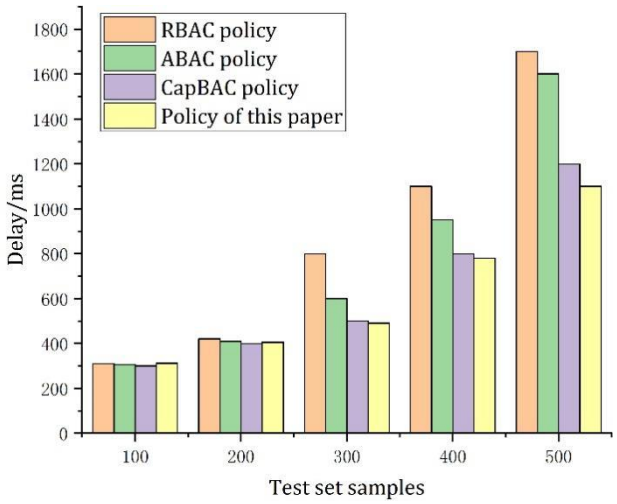


Figure 10. Delay comparison of four policies.

From Figure 10, it can be seen that the adjudication latency of all policies grows with the increase in the test set samples. When the number of the tests set sample is small, the delay of this access control policy, RBAC policy, ABAC policy, and CapBAC policy decision is not significant; when the number of samples is 100, the decision delay of each policy is basically controlled at about 300 ms; and when the number of samples is 200, the decision delay of each policy is basically controlled at about 400 ms, the reason being that when the samples are small, the system throughput (i.e., the number of transactions requests per second) using the above four different strategies can meet the normal processing conditions of a small number of samples, and there is no difference in delay.

With the increase in the number of test set samples, the decision delay of the four policies gradually shows a gap. When the number of samples in the test set increases to 500, the latency of the CapBAC strategy and the strategy in this article has a slower growth trend compared to the RBAC strategy and the ABAC strategy, where the RBAC policy decision latency increases from 300 ms to 1700 ms, while the strategy decision latency in this article only increases to 1100 ms, which is lower than the CapBAC policy decision latency of 1200 ms. The reason is that the throughput of four different policies mechanisms cannot satisfy the decision requests with a large number of concurrent policies in the same period due to the increasing number of samples, which leads to the queuing of policy decisions, and the difference in throughput makes the performance of the four policies

gradually widen the gap. However, due to the P2P network architecture of the blockchain, each node can make decisions on policy transaction requests, and process more policy requests concurrently. Smart contracts are used to implement complex algorithms to make decisions on policies quickly.

7. Conclusions

This paper proposes a physical network access control scheme combining the permissioned blockchain and edge computing. In this scheme, users first authenticate at the edge end, and based on ABAC model, a Smart contract is used to achieve flexible, scalable, and fine-grained access control, triggering the set contract content in order to complete the automatic operation of trusted access control of devices. Moreover, the KRaft consensus algorithm can achieve a low latency access request response, and it can bring good scalability. A security evaluation suggests that the system scheme is secure, and simulation consequences exhibit that the scheme is advantageous in implementing strict and fine-grained access control in the IoT. Using a Smart contract to obtain access control in the IoT will be a future research direction.

Author Contributions: Conceptualization, L.Z.; methodology, L.Z.; software, C.L. and G.Z.; investigation, L.Z.; resources, L.Z.; data curation, C.L. and G.Z.; writing—original draft preparation, B.L.; writing—review and editing, L.Z., C.L., B.L. and H.F.; supervision, L.Z.; project administration, L.Z.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key of National Natural Science Foundation of China (grant number U21A20463) and the National Natural Science Foundation of China (grant numbers 62261023 and 72262014).

Data Availability Statement: Some or all data, models, or code generated or used during the study are available from the corresponding author by request.

Acknowledgments: The authors would like to acknowledge the East China Jiaotong University for the lab facilities and necessary technical support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ali, M.S.; Vecchio, M.; Pincheira, M.; Dolui, K.; Antonelli, F.; Rehmani, M.H. Applications of Blockchains in the Internet of Things: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1676–1717. [CrossRef]
2. Du, R.; Liu, Y.; Tian, J. An Access Control Method Using Smart Contract for Internet of Things. *J. Comput. Res. Dev.* **2019**, *56*, 2287–2298.
3. Cai, F.; Zhu, N.; He, J.; Mu, P.; Li, W.; Yu, Y. Survey of access control models and technologies for cloud computing. *Clust. Comput.* **2019**, *22*, S6111–S6122. [CrossRef]
4. Yavari, A.; Panah, A.S.; Georgakopoulos, D.; Jayaraman, P.P.; Schyndel, R.V. Scalable role-based data disclosure control for the Internet of things. In Proceedings of the IEEE 37th International Conference on Distributed Computing Systems, Atlanta, GA, USA, 5–8 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2226–2233. [CrossRef]
5. Ouechtati, H.; Azzouna, N.B. Trust-ABAC towards an access control system for the Internet of things. In Proceedings of the International Conference on Green, Pervasive, and Cloud Computing, Cetara, Italy, 11–14 May 2017; Springer: Cham, Switzerland, 2017; pp. 75–89. [CrossRef]
6. Zhang, G.; Gong, W. The Research of Access Control Based on UCON in the Internet of Things. *J. Softw.* **2011**, *6*, 724–731. [CrossRef]
7. Zhang, G.; Gong, W. The research of access control in the application of VANET based on UCON. *Procedia Eng.* **2012**, *29*, 4091–4095. [CrossRef]
8. Shen, H.B.; Liu, S.B. A context-aware capability-based access control framework for the Internet of things. *J. Wuhan Univ. Nat. Sci. Ed.* **2014**, *60*, 424–428, (In Chinese with English Abstract). [CrossRef]
9. Shi, J.; Li, R. Survey of blockchain access control in Internet of Things. *J. Softw.* **2019**, *30*, 1632–1648.
10. Rongna, X.; Hui, L.; Guozhen, S.; Yunchuan, G.; Ming, Z.; Xiuze, D. Blockchain-based access control mechanism for data traceability. *J. Commun.* **2020**, *41*, 82–93.
11. Bonnah, E.; Shiguang, J. DecChain: A decentralized security approach in Edge Computing based on Blockchain. *Futur. Gener. Comput. Syst.* **2020**, *113*, 363–379. [CrossRef]

12. Altarawneh, A.; Sun, F.; Brooks, R.R.; Hambolu, O.; Yu, L.; Skjellum, A. Availability analysis of a permissioned blockchain with a lightweight consensus protocol. *Comput. Secur.* **2021**, *102*, 102098. [CrossRef]
13. Cruz, J.P.; Kaji, Y.; Yanai, N. RBAC-SC: Role-Based Access Control Using Smart Contract. *IEEE Access* **2018**, *6*, 12240–12251. [CrossRef]
14. Rajesh, R.K. Role recommender-RBAC: Optimizing user-role assignments in RBAC. *Comput. Commun.* **2021**, *166*, 140–153.
15. Xie, R.; Li, H.; Shi, G. Attribute-based lightweight reconfigurable access control policy. *J. Commun.* **2020**, *41*, 112–122.
16. Wang, S.; Zou, S. Blockchain and capability based access control mechanism in multi-domain IoT. *J. Appl. Sci.* **2021**, *39*, 55–69.
17. Nakamura, S.; Enokido, T.; Takizawa, M. Information Flow Control Based on the CapBAC (Capability-Based Access Control) Model in the IoT. *Int. J. Mob. Comput. Multimedia Commun.* **2019**, *10*, 13–25. [CrossRef]
18. Liu, Y.; Lu, Q.; Chen, S.; Qu, Q.; O'Connor, H.; Choo, K.-K.R.; Zhang, H. Capability-based IoT access control using blockchain. *Digit. Commun. Networks* **2020**, *7*, 463–469. [CrossRef]
19. Men, R. Research on access control method of Digital Archives based on blockchain. *J. Phys. Conf. Ser.* **2020**, *1550*, 062021. [CrossRef]
20. Zeng, S.Q.; Huo, R.; Huang, T.; Liu, J.; Wang, S.; Feng, W. Survey of blockchain: Principle, progress and application. *J. Commun.* **2020**, *41*, 134–151.
21. Gai, K.; Wu, Y.; Zhu, L.; Xu, L.; Zhang, Y. Permissioned Blockchain and Edge Computing Empowered Privacy-Preserving Smart Grid Networks. *IEEE Internet Things J.* **2019**, *6*, 7992–8004. [CrossRef]
22. Zhang, J.H.; Cui, B.; Li, R.; Jin, J.S. Access Control System of Internet of Things Based on Smart Contract. *Comput. Eng.* **2021**, *47*, 21–31. [CrossRef]
23. Min, X.P.; Li, Q.Z.; Kong, L.J.; Zhang, S.D.; Zheng, Y.Q.; Xiao, Z. Permissioned blockchain Dynamic Consensus mechanism based multi-centers. *Chin. J. Comput.* **2018**, *41*, 1005–1020.
24. Huang, D.; Li, L.; Chen, B.; Wang, B. RBFT: A new Byzantine fault-tolerant consensus mechanism based on Raft cluster. *J. Commun.* **2021**, *42*, 210–220.
25. Wang, R.; Zhang, L.; Xu, Q.; Zhou, H. K-Bucket based Raft-like consensus algorithm for Permissioned blockchain. In Proceedings of the 2019 IEEE 25th International Conference on Parallel and Distributed Systems, Tianjin, China, 4–6 December 2019; IEEE Press: Piscataway, NJ, USA, 2019; pp. 996–999.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.