The Journal of Supercomputing (2024) 80:1396–1425
https://doi.org/10.1007/s11227-023-05517-4

Smart contracts attribute-based access control model for security & privacy of IoT system using blockchain and edge computing

Chen Zhonghua[1,2] · **S. B. Goyal**[2] · **Anand Singh Rajawat**[3]

Accepted: 22 June 2023 / Published online: 18 July 2023

Abstract

In order to solve the problem of data security and management between IoT edge nodes and massive heterogeneous devices, combined with the wide application of blockchain technology in distributed system data security management, a blockchain-based Internet of Things access control model (SC-ABAC) is proposed by combining smart contracts and attribute-based access control. The traditional consensus algorithm PoW (Proof of Work) and SC-ABAC access control management process are optimized. By quantitative analysis, the time to call contracts in the query process increases linearly, the time of the policy addition and judgment process is constant, and the energy consumption of the optimized consensus mechanism is smaller than that of the PoW unit. This model provides decentralized, fine-grained, and dynamic access control management in IoT environments, enabling distributed systems to reach consensus faster and ensure data consistency.

Keywords Blockchain · Internet of things · Smart contract · SC-ABAC

✉ S. B. Goyal
sb.goyal@city.edu.my

Chen Zhonghua
chenzhonghua0106@163.com

Anand Singh Rajawat
anandsingh.rajawat@sandipuniversity.edu.in

[1] BaiCheng Normal University, Baicheng 13700, Jilin, China

[2] Faculty of Information Technology, City University, 46100 Petaling Jaya, Malaysia

[3] School of Computer Science & Engineering, Sandip University, Nashik, Maharashtra, India

1 Introduction

Industry 4.0 is a term that was made up to describe how quickly technological, industrial, and social patterns and processes are changing in the twenty-first century. One trend that has come up during this stage of industrial growth is the mixing of technologies that blur the lines between the physical, digital, and biological worlds. Artificial intelligence, gene editing, and advanced robots are all examples of these kinds of technologies. Modern smart technology, M2M communication, and the Internet of Things are making it possible to automate more and more of the traditional production and industrial processes. This is causing big changes in the global production and supply network. When these technologies are used together, there are more automation, better communication and self-monitoring, and smart equipment that can evaluate and diagnose problems on its own.

With the increasing popularity of IoT technology, access to large devices creates a heterogeneous environment, which generates large amounts of data for processing in a short period and at the same time is transmitted to the data center through the network. Network latency is inevitable [1]. Edge computing provides a new idea for solving the problem of high delays in the process of huge data transmission of IoT devices [2]. The existence of edge nodes in IoT systems increases system computing power and reduces latency but also brings new security risks. When the newly accessed edge node is malicious, it can decrypt and obtain the user privacy data in the data center by computing in parallel with other edge nodes. Therefore, the access control problem for IoT systems with edge nodes has become a new research hotspot.

Blockchain technology has been increasingly used in distributed IoT systems to provide security and privacy [3]. Complex attribute management, using too many resources, delay problems caused by blockchain, security risks caused by edge computing, and the need for effective consensus methods are just some of the problems it must solve. It ensures data are tamper evident while guaranteeing distributed data storage, which makes it particularly suitable for storing and protecting important private data, reducing the risk of leaking users' personal private data or losing large amounts of data due to attacks on IoT devices in data centers. The introduction of smart contracts in Blockchain 3.0 makes it possible to solve the access control problem through blockchain technology with new possibilities [4].

Much of the current research to address IoT security issues under edge computing integrated with blockchain is based on privacy security and data security aspects, and no further research has been done on the access control issue of IoT edge nodes [5–8]. For example, (Nyamtiga 2018) only address the problem of data anonymity and integrity in IoT systems with integrated edge computing and use blockchain technology to achieve secure storage of IoT data. [9] face the problem of large-scale data transmission and storage under industrial IoT, and edge computing processes the source data and then blockchain to achieve secure storage and management of data.

Currently, Internet of Things (IoT) data access and control research is only done in the context of blockchain technology. (Jinshan Shi et al. 2020) proposed

a blockchain-based IoT access control framework; (Siyuan Wang et al. 2021) proposed a blockchain-based power token ring to address IoT transgression access; [10] proposed a blockchain-based role access control model to address IoT transgression access. (Siyuan Wang, 2020) offered a power token based on the blockchain. Researchers [11] suggested an IoT data management architecture based on blockchain and edge computing to keep data safe. Their job was to combine edge computing with blockchain technology.

Building on the already established architecture for traditional integrated blockchain and edge computing, a new access control model for Internet of Things (IoT) systems with edge computing has been presented [3] as a possible solution to the problems mentioned above.

Xue et al. (2022) proposed a general architecture for the integration of blockchain and edge computing systems and studied how blockchain can be used to benefit edge computing and how edge computing can be used to benefit blockchain.

The above researchers have ignored the potential risks of edge nodes, assuming that the existing edge nodes are safe and reliable, but ignoring the expansion of the scale of the Internet of Things, edge nodes should be confirmed by a trusted mechanism when accessing the Internet of Things, and also ignoring the powerful computing power brought by edge computing, that is, under the current new paradigm of the Internet of Things system integrating edge computing. The role of edge nodes in the access control process is not considered.

To solve these problems, based on the existing traditional integrated blockchain and edge computing architecture, an access control model for the Internet of Things system containing edge computing is proposed. Specifically, the three-layer access control architecture of the Internet of Things is designed on the basis of the three-layer architecture of the Internet of Things. On the basis of the attribute-based access control mechanism, the smart contract in the blockchain is responsible for access control decisions, while the edge node is responsible for encryption, decryption, and data transmission, and the "SC-ABAC" access control model is proposed. To achieve effective access control and management of IoT. Finally, the experimental results show that the time consumed for continuous block access increases linearly with the number of accesses, the CPU utilization is stable, and the security is good.

Three contributions of this paper for a study on a Smart Contracts Attribute-Based Access Control Model for Security & Privacy of IoT System using blockchain and edge computing include:

1. First, design and use a smart contract-based access control mechanism. This method makes IoT devices and infrastructure safe. The approach would determine access based on roles, responsibilities, and IoT system links.
2. Integrate blockchain for security and privacy: The second purpose may be to integrate blockchain technology into attribute-based access control architecture. Blockchain may record access transactions securely and transparently. Blockchain may improve IoT security and privacy, according to the study.
3. Edge computing speeds data processing. Third, edge computing could speed up IoT data processing. Edge computing accelerates data processing near the source.

Edge computing and blockchain-based access control should be considered for safe, fast, and reliable IoT systems.

This research work is organized in the following manner: Section 2 presents a Preliminaries. In Section 3, we outline the method employed and describe the design of the IoT Access Control Architecture that integrates edge computing. Section 4 details the methodology used to identify the applicability of blockchain to our use case. Section 5 is dedicated to presenting the results of our study and a discussion of these findings. In Section 6, we engage in a discussion on the related work, comparing it with our findings. Finally, Section 7 offers the conclusion and outlines directions for future research.

2 Preliminaries

2.1 Basic concepts of access control policies

In an IoT system, access control is the authentication and control of users' legitimate use of resources, which literally means controlling access to certain resources in accordance with the relevant authorization, to prevent unauthorized access or improper use of certain illegal users. Users ensure that the entire system resource can be used rationally and properly. The IoT application system is a multi-user and multi-task working environment that opens the door to illicit use of system resources, so it is imperative to take effective security precautions for the computer and its network system to prevent unauthorized users from entering. Illegal use of system resources by the system and legitimate users. This requires the use of access control systems [12].

2.2 Types of access control

When access control is set up right, it can stop hackers from getting sensitive information and stop authorized users from misusing system resources. Autonomous access control and mandatory access control are two methods that are often used in traditional security settings. Access control methods like object-based access control, task-based access control, role-based access control, and attribute-based access control have come about because of the rise of distributed application environments.

1. DAC (discretionary access control).

This means that people own the things (files, databases, etc.) they make and can choose who has access to them.

2. MAC (mandatory access control).

It refers to the unified binding control by the system of user-created objects (specifically set up by the system security authority), to determine by defining rules which users can access which objects for which operating system type, regardless of the creator user, whose object creation cannot have the right to access the object.

3. OBAC (object-based access control).

Object-based access control systems are made up of control policies and control rules, which are the main building blocks. When using a model based on controlled objects, rights are given to specific people, groups, or roles, and the access control list is linked to either the controlled object or its properties.

4. TABC (task-based access control).

Task-based access control builds security models and puts security measures based on the tasks being done in place. It also lets security be managed in real time in a flexible and adaptable way. Task-based access control can be used to solve security problems at both the application level and the network level.

5. RBAC (role-based access control).

It is to assign the right of access permission to certain roles, and the user gains access to ownership of the role by playing various roles. This is because, in many practical applications, the user does not own the object information resources that can be accessed (this information belongs to the enterprise or company). Access control is determined by the role each user plays in the department and consists of three main features: user, roles, and permissions, as shown in Fig 1.

The principle of RBAC access is shown in Fig. 2, where the mapping of roles and permissions is the core of RBAC, and a group of roles is assigned a scope to obtain the corresponding permissions during actual use. RBAC consists of five major elements: user table U, role table R, permission P, permission configuration (R, P), and user authorization (U, R).

ABAC mainly includes three entities: subject, resource, and environment, as shown in Fig. 3. Subject attributes include static attributes such as role, identity, age, etc. Resource attributes mainly include metadata describing resources, and environment attributes include contextual attributes describing the access process.

The ABAC can update the access control decision according to the modification of the related entity attributes, thus providing a more sophisticated and flexible access control method, and the access flow is shown in Fig 4.

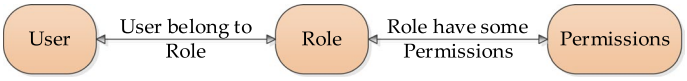

Fig. 1 Attributes of RBAC

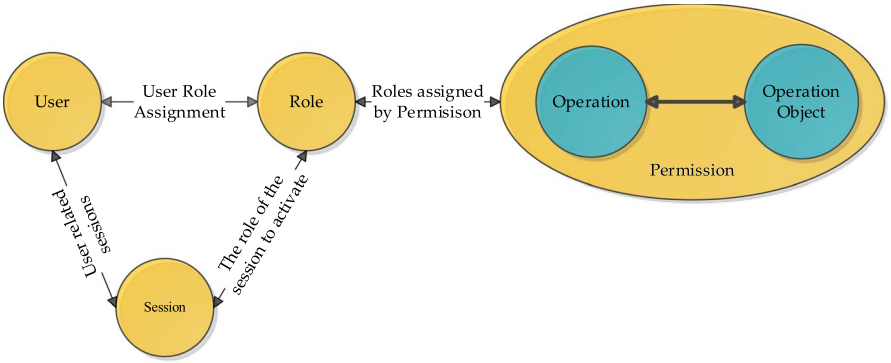

Fig. 2 Principle of RBAC

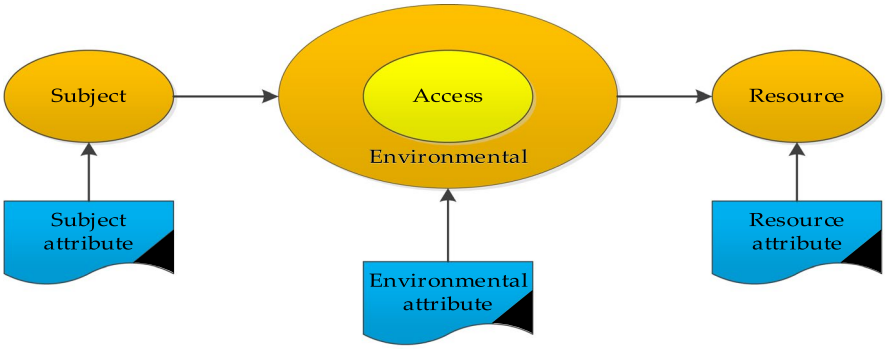

Fig. 3 Subject of ABAC

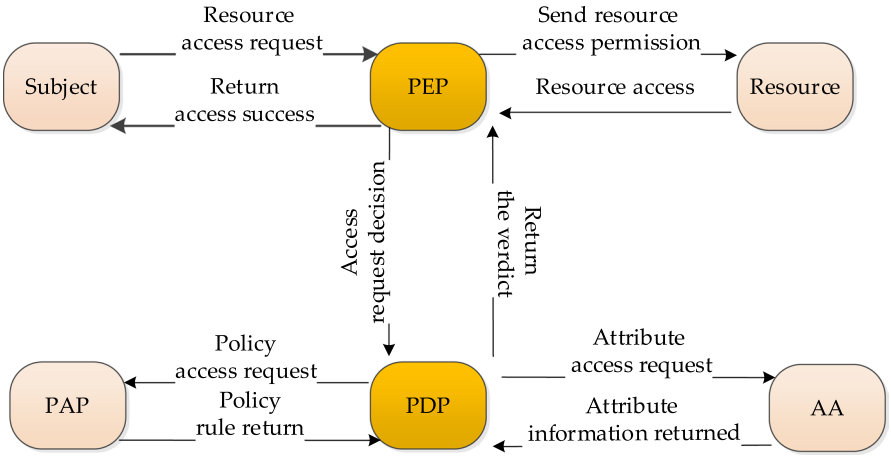

Fig. 4 Access process of ABAC

A policy enforcement point, or PEP, is a device that is physically connected to a protected resource and blocks access requests, starts the decision-making process, and applies the right results by reading information to allow or deny access.

PAP (policy administration point): It is the component responsible for the management of the access control policy.

AA, which stands for "Attribute Authority," is a part that controls who has access to what subject, resource, or environmental quality and how that quality can be changed.

A policy decision point, or PDP, is a central location that gets policies, access requests, and information about attribute values.

2.3 Blockchain and IoT: an overview

Blockchain technology brings about a big change in the way things are done. In just a few short years, reliable human systems that are run from one place will be replaced by networks that are spread out. The process of change has already begun. Between 2016 and 2022, the market for IoT technology is expected to grow at a compound annual growth rate (CAGR) of 32.4%. In 2015, there were 10 billion devices that could connect to the internet. By 2020, there will be 34 billion. The distributed ledger, which is what the blockchain is, will be the system's backbone.

The trust protocol that blockchain technology [13] offers is starting to become clearer, as is the fact that blockchain technology is real. Because of this, financial experts and governments around the world are thinking about how it could be used in large-scale solutions and what benefits it could bring if it were to be used. Even though there are signs of progress, there are still problems.

For example, a few questions come to mind when I think about all the different ways it can be used. One of these problems is figuring out how to manage hybrid cloud and edge IoT Blockchain AI systems in a way that is both effective and safe. This is a big problem, especially since only 30% of companies think they are ready to deal with the challenges that security holes in the Internet of Things pose. Attacks on the Internet of Things could lead to some parts of the building, like the lights and door locks, being taken over. In addition to stealing data, hackers may also be able to find out who certain people are and try to harm them. When it comes to providing services, there is a chance that medical devices like pacemakers could be attacked as part of an attack on the Internet of Things attacks on the Internet of Things (IoT) [14]. The problem of how to protect devices that are connected to the Internet of Things might be solved by blockchain technology. In the future, both may be put into a single place, but first, there are some technological problems that need to be solved. Using this would be a good idea because blockchain makes it possible to build a very high level of trust by using a decentralized database. This deep sense of self-confidence can be found everywhere.is not taken into account. Edge computing also gives a lot of powerful computer power, which is also Even though the default presence of edge nodes is safe and secure and should be confirmed by trusted mechanisms as the IoT grows, the role of edge nodes in the access control process is not

taken into account in the current paradigm of the IoT system with integrated edge computing. This is because the role of edge nodes is not taken into account.

2.4 Related work discussions

The literature [15] integrates edge computing and blockchain technology to solve the data security problem in IoT architecture, but only the conceptual design of the solution is given without experiments to support it. The literature [7] also integrates edge computing and blockchain technology and uses blockchain technology and bloom filter [16] to solve the authentication of devices and access control of data in industrial IoT, but again, no experimental results are given in the experimental part. The literature [10, 15, 17–20] only considered the use of blockchain technology in their research on IoT access control, including the proposed access control framework, the use of access tokens or the design of better access control policies, but did not consider the indispensability of edge nodes in the current scale of IoT [14, 21–25]. Therefore, the existence of edge nodes and the integration of edge computing with blockchain were not considered in the access control research on IoT impact on existing IoT access control research. For relevant comparative content, see Table 1.

3 Method

The methodology includes three main steps: IoT Access Control Architecture Design for Integrated Edge Computing; flow chart of SC-ABAC-based smart contract implementation; processing control flow chart of specific device access data. The details are shown in Fig 5.

3.1 IoT access control architecture design for integrated edge computing

This article talks about a proposed access control model for IoT systems that use edge computing. This model is based on the common architecture for integrating blockchain and edge computing that is used right now. We suggest the "SC-ABAC" access control model and design a three-layer access control architecture for IoT based on the three-layer IoT architecture. With the attribute-based access control mechanism, the smart contract in the blockchain decides who can get in and who can't. The edge nodes are in charge of encryption, decryption, and data transmission. Our goal is to get access control and management of IoT to work well. The IoT access control model that uses edge computing is made up of three modules: a module for managing IoT devices, a module for controlling who can use those devices, and a module for managing the data that those devices collect. Figure 6 shows the finished plans for the model.

Incorporating a module for managing IoT devices is the first step in designing an IoT access control framework for integrated edge computing. This part is in charge of making sure that Internet of Things devices are registered, authenticated, and set up/monitored. Then, the part that controls how Internet of Things devices can be used is set up. This makes use of the Blockchain's Attribute-Based Access Control

Table 1 Comparison of related research work

Author	Algorithm	Advantages	Limitations
[26]	No	The cloud layer, the edge layer, and the device layer made up the framework	Conceptual designs that only give solutions without experiments to support them
[7]	Yes	Adapting to the IoT necessitates a method that combines access control and identity management, which is where blockchain technology and edge computing come in	No experimental results are given in the experimental section
[27]	Yes	Proposed access control framework	Failure to take into account the edge node's indispensability at the scale of the Internet
[28]	Yes	Access control policy using access tokens	In access control studies for IoT The presence of edge nodes is not taken into account
[19]	Yes	Design Better access control policies	Research on access control of existing IoT without involving edge computing and blockchain integration
[11]	Yes	Access control is used in the access control list approach	There are problems such as low access control efforts

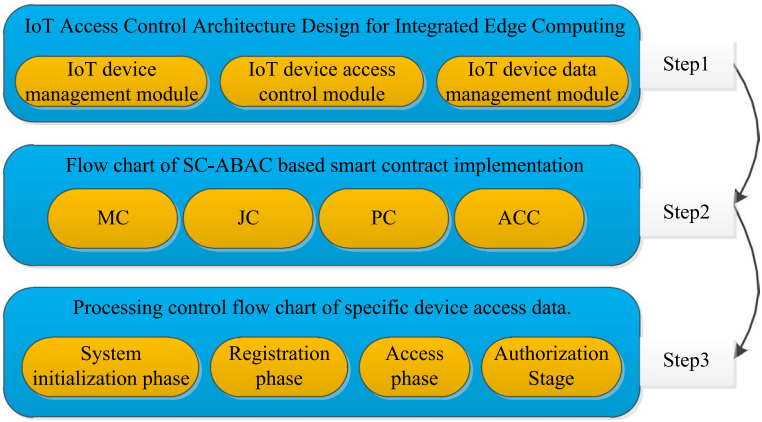

Fig. 5 System process flow of methodology

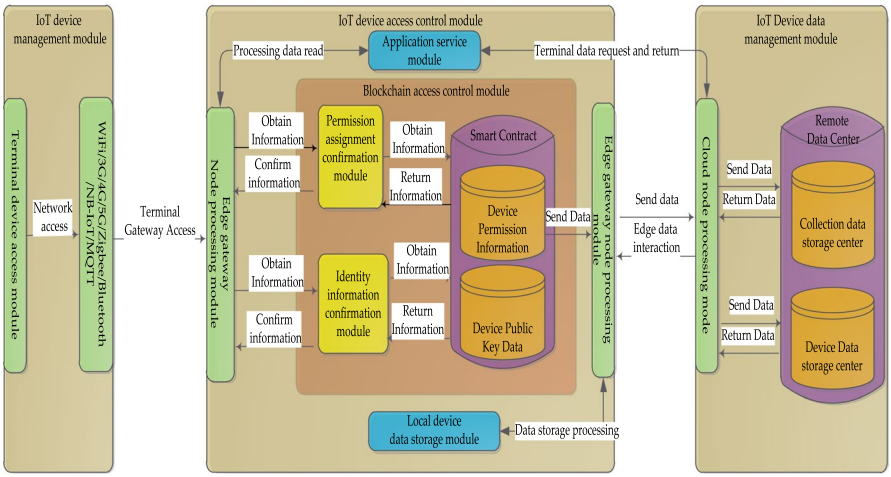

Fig. 6 IoT Access Control Architecture Design for Integrated Edge Computing

Model for Smart Contracts. IoT devices' features are put into smart contracts, which are contracts whose terms are written straight into code and which run themselves. They are a safe and foolproof way to allow, check, and enforce the negotiation or performance of an agreement. As IoT devices talk to each other, the access control module uses these smart contracts to allow or deny entry based on qualities and conditions that have already been set. The last part is the data handling part of the Internet of Things. Data from IoT devices can be saved, processed, and retrieved in a safe way. This part makes it easier to process data at the network's "edge," which is the part that is directly closest to where the data came from. Together, the changes to security and privacy, as well as the decreases in delay and bandwidth use, make the system run at its best.

3.1.1 IoT device management module

The IoT device management module manages the access of various IoT heterogeneous devices, pre-processes the collected data by parsing different protocols in the network, packages them into a unified format and applies them to the IoT device access control module for access to the system, and then sends the data to the edge gateway nodes.

3.1.2 IoT device access control module

The IoT device access control module integrates edge computing and blockchain and is the core module of the framework used to understand the fine-grained access control management of devices.

The edge gateway node processing module is a device with certain computing power and is in the middle layer of the network (e.g., smartphones, smart gateways, etc.). The main functions include:

1. encrypting device information;
2. decrypting data information;
3. verifying device privileges;
4. dynamically changing device privileges;
5. pre-processing data;
6. data forwarding and acquisition.

The device can only obtain or send the corresponding data through the edge gateway node based on its own authority after the identity confirmation of the edge gateway node.

The blockchain access control module is an integral part of the access control module, providing a variety of access control mechanisms for IoT systems on a variety of occasions through programmable smart contract. It supports dynamic power allocation of devices to achieve micro-access control of data by devices. The main functions include:

1. access control decision execution;
2. device identity confirmation;
3. device permission assignment.

The local device data storage module is used to interact with edge gateway nodes, store pre-processed data, and send them to the application service module for users to view in real time when needed.

The application service module is the upper layer data transmission interface for users to visualize the data. By acquiring local data and final processed data from edge gateway nodes and cloud terminal nodes, the data accuracy is improved while ensuring timeliness.

3.1.3 IoT device data management module

The IoT device data management module stores:

1. pre-processed data from the edge gateway module after processing by the cloud terminal for viewing when necessary.
2. device information in the blockchain module is broadcast by the edge gateway module to reduce the frame risk.

3.2 Smart contract-based attribute-based access control model

The attribute-based access control model (ABAC) has an access structure that includes a subject, object, operation, and environment [26]. It determines whether the subject has access to the appropriate features in the corresponding environment in the operation of the object by which the subject is given access.

3.2.1 SC-ABAC

In the IoT environment, since attributes are inherent to each subject, object, operation, and environment, associating attributes of each device and resource with access rights makes the ABAC model suitable for managing simple devices and extensive data in the IoT. However, for dynamic access management of heterogeneous devices, ABAC uses an attribute discovery mechanism and does not assign precise and appropriate permissions to the accessed devices by { attribute, permission }. This limits the normal access and real-time processing of data by heterogeneous devices, which poses a challenge for scenarios where edge nodes exist.

Therefore, in this paper, we propose an ABAC-based access control model "SC-ABAC" for dynamic access to heterogeneous devices, in which ABAC combines smart contract with ABAC to allocate resources and nodes by permissions and ensure proper implementation of the corresponding operation on smart contract.

In SC-ABAC, subject and object are treated as the same object, because each device in IoT can be an object providing resources to other devices or accessing resources from other devices as a subject. Each IoT device needs to register information with the blockchain through the edge server when it first accesses and obtains the corresponding authority through its inherent attributes; for the security of the device itself, the authority is bound with the public key obtained based on Elliptic Curve Cryptography (ECC) and encrypted with the private key during subsequent access and data transmission. The judgment of attributes is implemented based on smart contract, and each access request and judgment result are stored in the blockchain for synchronization.

3.2.2 Smart contract design based on SC-ABAC

Smart contract is a man-written program script that directly controls the data within the blockchain and is implemented by multiple users within the blockchain to control the behavior of transactions even without a third party. Smart contract

may use calls to a custom function to execute related commands: for example, requesting access to a specific node or returning results.

Based on SC-ABAC, this paper designs a Manager Contract (MC) { address, attribute, permission }, Judgment Contract (JC) { number, subject, subject permission, resource, resource permission, result, time}, Punish Contract (PC) {number, JC result, result, time }, and Access Control Contract (PC) {number, JC result, result, time}.

(1) MC is used to manage the related policies, the subject is the access initiator, replaced by the corresponding MAC address, the attributes are the inherent characteristics of the object, the permission is the actionable behavior described by the value, and the permission is raised to read, write, manage, etc. in order. Among the custom functions are:

ManageAdd(): used to add the permission information of an object.
ManageUpdate(): used to update the permission information of an object.
ManageDelete(): used to delete the permission information of an object.
QueryData(Hash): used to obtain information about an object by Hash.

(2) JC is used to determine the application of an object to operate on a resource, by comparing the subject permissions and resource permissions to derive the results, including allowing all operations, read and write, readable and illegal access. Among the custom functions are:

JudgeFromMC() is used to get the permission information of the object from MC.
JudgeToPC() is used to send the result of the judgment to the PC.
JudgeToACC() is used to send the result of the judgment to the PC.

(3) The PC is used to process the results sent by JC, such as implementing a penalty mechanism for devices that access the overstepped data or providing a reward mechanism for devices that access and send data normally to assign dynamic permissions. Among the custom functions are:

PublishToACC() is used to send penalty results to ACC.

(4) ACC is used to realize the final access control of devices and resources, return the judgment result to the subject through JC result, and operate the function of the subject and resources to MC through PC result.

ACCAnswer() returns JC result to the subject to complete the access control of resources;

ACCSetMC() calls the function in MC to modify the object permission through the result of JC.

The specific flow of the blockchain-based access control management mechanism is shown in Fig 7.

3.3 SC-ABAC-based inter-device access control process design

This section will introduce in detail the blockchain and edge computation-based IoT access control process implemented in the above, as shown in Fig. 8, taking a device A accessing an edge server to send an access request to a resource B as an example; its access control process includes system initialization phase, registration phase, access phase, and authorization phase.

3.3.1 System initialization phase

The relevant nodes at the edge of the network by default are all edge nodes, including Edge Gateways and Edge Servers. The edge node (E_n) needs to calculate its own public and private keys based on the *ECC* algorithm first. The details are as follows:

E_n selects two non-negative integers *a, b* less than p among p elements (p is a prime number) to obtain the elliptic curve $E_p(a, b)$ and then selects a point p of order n on the elliptic curve such that the number multiplied by $n * p = 0$.

E_n randomly selects a number from positive integers smaller than n as his private key $\mathrm{PKE}_{n,}$ obtains the public key PKE_n according to $\mathrm{PKE}_n * p$, and discloses the following information after calculating the hash of the physical address (Media Access Control Address, MAC) from the following Eq. (1): $\{\mathrm{PKE}_n, E_p(a, b), P, p, Hash\}$.

$$\mathrm{Hash} = \mathrm{SHA256}(\mathrm{MAC}, P(x), P(y)) \tag{1}$$

Blockchain creates Genesis blocks and stores E_n information, attributes, and permissions.

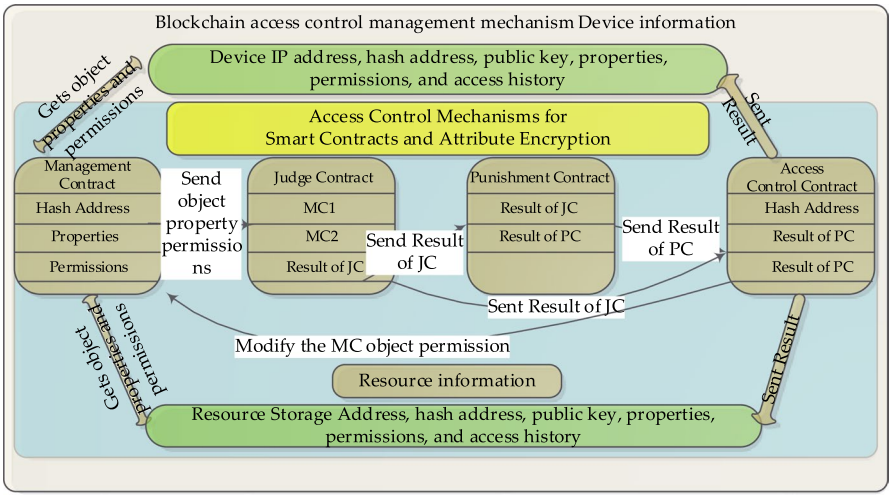

Fig. 7 Flow chart of SC-ABAC based smart contract implementation

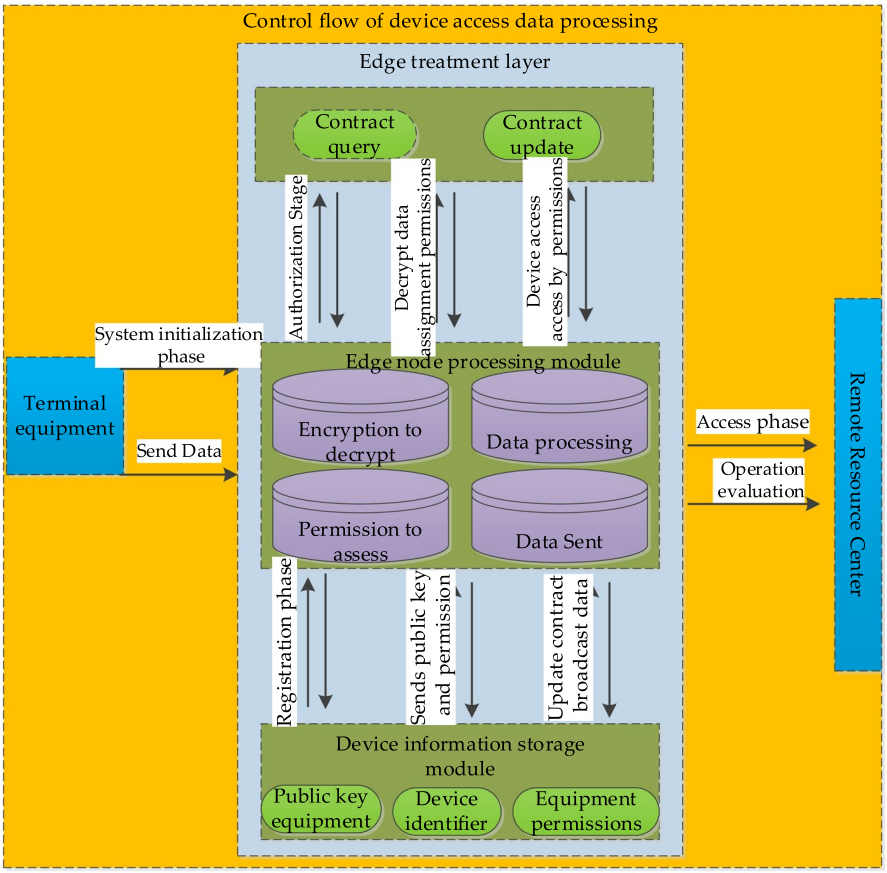

Fig. 8 Processing control flow chart of specific device access data

3.3.2 Registration phase

1) E_n calculates the *ECC* key pair and the hash value of each access device, assigns the specific attributes of the device such as sensing, sending and receiving data, analyzing, controlled, etc., assigns permissions based on the attributes, and sends the information of the device {registration *ID, device public key* pk_{IoT}, *device attributes, device permissions, device MAC address, Hash*} to the blockchain for storage.
2) E_n stores the private key pk_{IoT} of the device in the local data, which is only viewed by the edge gateway.
3) Blockchain invokes smart contract to store device information one by one.

3.3.3 Access phase

1) The device generates a random prime number $r(r < n)$, generates a random key pair$(r, R(x, y))$ by the *ECC* algorithm, and encrypts the sent data M with a digital signature based on *secp256k1* Elliptic Curve Digital Signature Algorithm *(CDSA)*,

and the specific signature process is shown in Algorithm 1. By the analysis of the algorithm, the time complexity of the algorithm is O(log n).

Algorithm 1 ECDSASIGN signature algorithm

Input:

Device data M, device private key pKIoT, secp256k1 Elliptic curve=(p,a,b,P,h,h)

Output:

$Sign = (S_r, S_s)$

Process:

01: *begin*

02: *while true*

03: *do*

04:*r = random_prime();*$1 < r < n$

05: $R(x, y) = r * P$

06: $S_r = (R(x)\ mod\ n)$

07: *if* $S_r = 0$

08: *Re-select the new random prime r to generate* $R(x, y)$ *again*

09:*continue*

10: $S_r = r^{-1} * (Hash(M) + pK_{IoT})\ mod\ n$

11: *if* $S_s = 0$

12: *Re-select new random prime r to generate* $R(x, y)$ *again*

13:*continue*

14: *done*

15: *return* $Sign = (S_r, S_s)$

16: *end*

2) En receives the data M from the device when it is accessed and applies the public key pK IoT to the blockchain for signature and decryption verification, as shown in Algorithm 2. By the analysis of the algorithm, the time complexity of the algorithm is O(n).

Algorithm 2 ECDSASIGN signature verification algorithm

Input:

Device data digest e = Hash(M), signature result $Sign =$

(S_r, S_s)*, device public key* pk_{IoT}*, secp256k1 ellipticcurve* $T = (p, a, b, P, h, h)$

Output:

Signature verification result true or false;

Process:

01: *begin*

02: $U1 = (e * Sign(S_s)^{-1}) \ mod \ n$

03: $U2 = \left(\left(Sign(S_r)\right) * \left(Sign(S_s)\right)^{-1}\right) \ mod \ n$

04: $\mathrm{R}(x_1, y_1) = U1 * P + U2 * pK_{IoT} n$

05: *If* $Sign(S_r) == R(x_1) \ mod \ n$

06: *Signature Verification succeeded*

07: *return true*

08: *else*

09: *return false*

10: *end*

2) E_n decrypts the data M, obtains the hash address value of the device and the data requested for access, sends a message to the data storage center to obtain the hash address value of the data, and sends it to the blockchain.

3.3.4 Authorization Stage

(1) The management contract (MC) obtains the corresponding permission from the decrypted hash value and sends it to the judgment contract (JC).
(2) JC compares the device permission and data permission size, determines whether it can be read/written or illegal operation according to the result, and sends it to the Access Control Contract (ACC) and the Punishment Contract (PC).
(3) PC decides to implement a penalty or reward mechanism for the device based on the result of JC and sends the result to ACC.
(4) ACC sends the query result of access control to En based on the result of JC, and according to the result of PC, MC raises or lowers the authority of this device, or removes this device in the device table.

3.3.5 Optimization of PoW consensus algorithm

Considering that the traditional PoW consensus algorithm combines the previous block hash value, the transaction data of this block, the timestamp, and the random number *nNonce* in the block together to calculate the corresponding hash value by *SHA256* algorithm, the self-addition of *nNonce* is used to get the final hash value is obtained by self-addition of *nNonce* to the first nBits of 0. Since the hash value varies greatly each time, it takes too much time to get a hash that matches the first nBits of 0. Therefore, in this paper, we consider a random serialization operation for the random number *nNonce,* change the initial value to a random number, and make a judgment when calculating the Hash value each time, if the first bit of the obtained Hash value is 0, the possibility of getting 0 for the first nBits of *nNonce* after self-addition is small, so *nNonce* is set as a random number; and if the first bit is not 0, then *nNonce* is set as a random number on the if the first bit is not 0, *nNonce* is self-added to the existing base. The specific algorithm is as follows. By the analysis of the algorithm, the time complexity of the algorithm is O (log n).

Blockchain difficulty nBits, block data DataHash, random number nNonce previous block hash prevHash, maximum random number range N, current timestamp Date.

Output:

Current block final hash Block Hash

Process:

Input:

01: *begin*

02: *while true*

03: *do*

04: $nNonce = random(), 1 \leq nNonce \leq N$

04: $\mathrm{R}(x_1, y_1) = U1 * P + U2 * pK_{IoT}n$

05: $Data = (DataHash + nNonce + prevHash)$

06: $BlockHash = sha256(Data + Date)$

07: *If* $nBits\ before\ BlockHash \neq 0$

08: *If the first bit of* $BlockHash \neq 0$

09: $nNonce = nNonce + 1$

10: *else*

11: *break*

12: *done*

13: *return BlockHash*

14: *end*

4 Methodology to identify applicability of blockchain to a use case

Bitcoin and other cryptocurrencies were the first places where blockchain technology was used in a way that worked. Later, smart contract platforms like Ethereum, which make it possible for the public to own assets and apps that use smart contracts, were found to be useful. The goal of all information stored on blockchains is to make sure it can't be changed and is shared forever.

Enterprise blockchains were made much later. They were inspired by the decentralized nature of public blockchains, but they have very different requirements. A lot of research has been done on how blockchain technology could be used to solve big problems in the business world. Business collaboration is already hard at work in the corporate blockchains that are now being built. Companies that work together came to agreements that led to these blockchains. Because of this, enterprise blockchains only need a small amount of decentralization to work well. Enterprise blockchains can do more than just settle money or tokens. They can be used for a wide range of other things as well. Some of these uses are verifying events and managing assets, contracts, identities, and other types of specialized data. When a company's blockchain is too centralized, it loses some of its usefulness. This is because there would only be a small number of trusted entities in the ecosystem, and all of the industrial partners depend on those entities. Even though Bitcoin's distributed ledger technology has a lot of possible uses, its usefulness is reduced when a company's blockchain is too centralized. Because of this, the easiest and most practical thing to do would be to use a central database that would be run by the same group. In the next section, we'll talk about when and why a blockchain might be useful and important for a business setting. Some of these conditions are: The solutions offered by blockchain technology may be able to cause big changes in many different industries because they are decentralized and open. Using blockchain technology instead of more traditional methods, on the other hand, can be a risky, time-consuming, and expensive endeavor. This means that companies in every industry will have to weigh the pros and cons of using blockchain technology versus more traditional methods of either centralization or decentralization. To come up with a methodical way to figure out if blockchain technology is a good solution or not, we started by doing a deep analysis of a wide range of real business problems and reading the relevant scientific literature. This gave us a place to start with our investigation, so we could move forward. In academic research, a model was made to figure out if blockchain technology can be used for collaborative product modification. did research on different ways that blockchain technology could be used.

5 Results and discussion

5.1 Experimental setup

5.1.1 Experimental environment setup

The experiment was conducted on a personal computer (PC) running Windows 11 Professional i5-10,210 2. 10 GHz 16G running memory and three CC2530s (8051 CPU, 8 KB running memory) with a temperature and humidity sensor DHT11 module, block chained by Node.Js version 10.16.0, are implemented. By using three CC2530s as IoT devices, the acquired temperature and humidity data are stored in the PC as IoT data, and the PC is used as an edge node to achieve secure access to the devices and data based on SC-ABAC. The host configuration is shown in Fig. 9.

5.1.2 Access control policy design

Two groups of access control policies are designed in this experiment, including device type and operation permission. For device type, where administrator M0 manages cloud C1, edge node E2, and terminal device D3, the sub-device of cloud is smart gateway G4, the sub-device of edge node and the terminal device are the underlying IoT device d5 ~ d7, and the IoT device manages data D8; the operation permission includes four operations of data creation, update, read, and delete, as shown in Fig. 10.

5.2 Security analysis

In the security analysis for the SC-ABAC model, this paper assumes that there is a risk of data leakage in the end device, while there are malicious users in the edge device En, bad service providers in the storage node, and malicious attackers in the network. Internet of Things devices are vulnerable to network attacks. The distributed architecture of edge computing naturally has the characteristics of resisting such attacks and has higher reliability and fault tolerance.

5.2.1 Risk of data leakage for terminal devices

SC-ABAC encrypts the data uploaded by terminal devices and the corresponding authority information on the transaction side and stores them in DataHash. The original data are transferred to the data center for storage. The original data are

Fig. 9 Configuration of the host

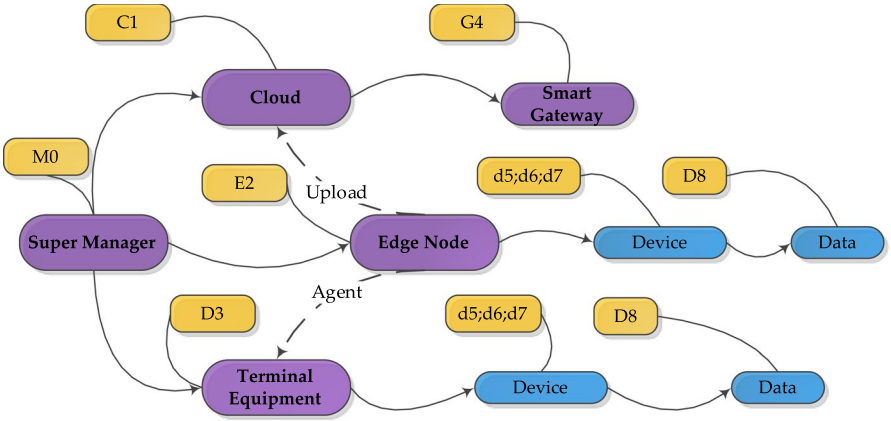

Fig. 10 Design of access control policies

transferred to the data center for storage, and access to the data requires En to have the corresponding DataHash. Only when the device gets the correct DataHash and has the corresponding permission will the original data be returned. The data transfer is in P2P (Point-to-Point) mode, and there is no third-party capture. The original data can be there is no malicious modification of the storage node.

5.2.2 Targeting malicious users in edge devices

SC-ABAC generates a corresponding HashID for all access En with MAC to store and ensure the corresponding ID is unique when adding devices each time, while access En has different read and write permissions for different devices based on function, and there is a corresponding penalty mechanism for each override, which can effectively reduce the survival time of malicious users in the network.

5.2.3 Targeting malicious users in edge devices

The data of SC-ABAC are stored in the storage node in the form of a smart contract, and the upload, access, and expiration of the original data are recorded in a time-stamped chain structure, and the information is recorded synchronously by the whole node, so the illegal access to the storage node can be traced and cannot be tampered with.

5.2.4 Against malicious nodes in the network

SC-ABAC manages terminal devices with relatively weak performance as En agents. Terminal devices do not touch the external network, while En, as an edge device, has a high-security protection capability and can resist external malicious totals, and qualitative analysis can improve the security protection capability of the system. The storage node is then assumed by En with the strongest protection capability.

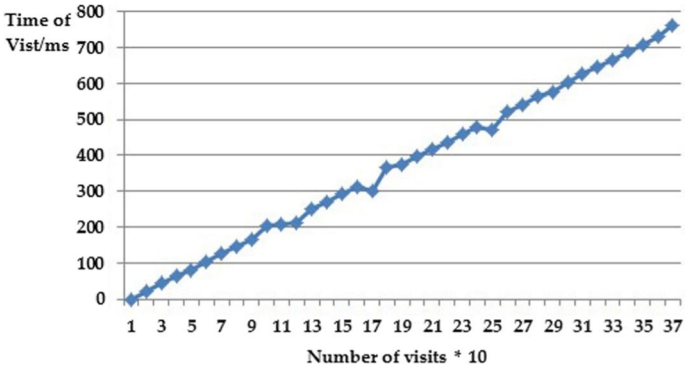

Fig. 11 Time consumption with different number of visits

Fig. 12 CPU Utilization for Continuous Access Processes

5.3 Blockchain system testing

5.3.1 Performance testing of the blockchain system

After the security analysis of the model, this paper detects the performance of this blockchain system by setting different numbers of accesses of the subject to the object. The performance is determined by setting a high concurrency of consecutive accesses, and the CPU utilization of the access process is viewed by the resource monitor and task manager.

Figure 11 shows the access time consumed for a total of 500 consecutive accesses to different data for different subjects, corresponding to the average consumption for every 10 accesses. It can be seen that as the number of accesses increases, the time spent on each access increases linearly, and the average time spent on each access at the 500th access is about 1.3 s. This is because the access records are stored in the blockchain at the end of each access, which increases the access query data each time, thus causing the access time to increase linearly with the increase in the number of accesses.

Figure 12 shows the CPU utilization in the continuous access process, and it can be seen that the CPU utilization is about 20% on average during the access process, indicating that the access process of this system is relatively stable in terms of CPU usage.

5.3.2 System security testing

In this paper, the system uses the PoW consensus mechanism for blockchain generation, so the security test of blockchain means the test of the consensus mechanism. In this paper, the PoW security analysis case proposed in the literature [3] (Yen 2019) is used, and the block creation between honest nodes and attackers is described by a binomial random process.

An attack is considered successful only if the attacker's block creation rate is greater than the block generation rate of all honest nodes in the chain. Thus, assume that the attacker creates the next block with probability q, the honest node creates the next block with probability p, and the blockchain is caught up by the attacker starting from block z. Then, the probability of the attacker catching up to the true block length is shown in Eq. (2).

$$q^z = \begin{cases} 1, p \le q \\ \left(q/p\right)^z, p > q \end{cases} \tag{2}$$

And the number of blocks that determine success for the attacker can be viewed as a Poisson distribution with the expectation value as in equation (3).

$$\lambda = z * \frac{q}{p} \tag{3}$$

Multiply it with equation (2), that is, the attacker can exceed the real block length of the honest node under the length of the block created during the attack, that is, the probability of success of the attack as in equation (4), where k is the number of blocks created by the attacker.

$$P = \sum_{k=0}^{\infty} \frac{\lambda^k * e^{-\lambda}}{k!} * \begin{cases} 1, k > z \\ q/p^{(z-k)}, k \le z \end{cases} \tag{4}$$

Translated into equation (5) which is the success probability of the attacker.

$$P = 1 - \sum_{k=0}^{z} \frac{\lambda^k * e^{-\lambda}}{k!} * \left[1 - q/p^{(z-k)}\right] \tag{5}$$

Through simulation, it can be seen that when the attacker's probability of creating the next block q increases (i.e., the arithmetic power increases), the higher the probability of a successful attack, and a perfect attack can be achieved when the attacker has 50% of the arithmetic power of the whole network, while the lower the probability of creating the next block by itself, the more difficult it is to attack, so the system in this paper controls the number of blocks in the blockchain to be at least 6 blocks, and the security of the system can be guaranteed when the attacker's own arithmetic power does not exceed 30% of the whole network. We have shown different types of contract time consumption at different difficulty levels in Fig 13.

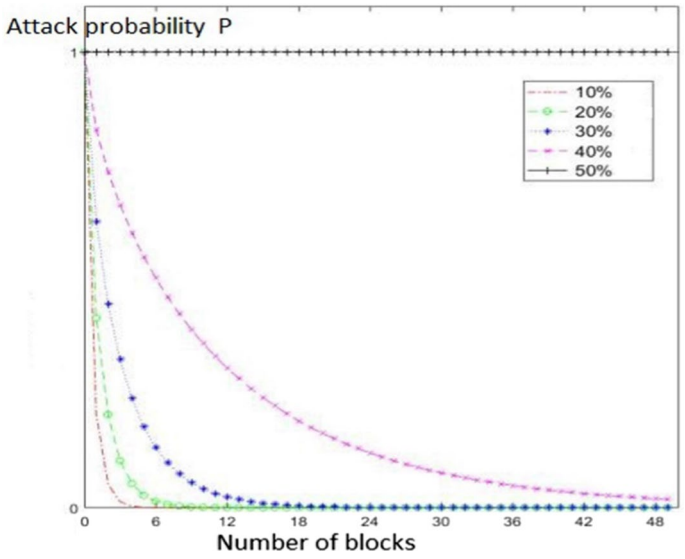

Fig. 13 Three types of contract time consumption at different difficulty levels

5.4 Performance analysis of SC-ABAC

5.4.1 Latency comparison under different smart contract

By simulating concurrent requests from multi-threaded clients, the processing times of ManageAdd (), QueryData (), and PublishToACC () are tested for different numbers of concurrent requests at PoW difficulty 0 and difficulty 2, as shown in Fig. 14. It can be seen that adding and judging contracts hardly take time under different concurrent request counts, which proves the superiority of edge nodes in IoT scenarios. Based on the need to complete consensus, access records and synchronize queries in the blockchain during the query process; the time consumption increases with the number of requests, resulting in an increase in time. The difficulty differentiation is not significant, making the time required for PoW negligible compared to the query time. However, in comparison, the scheme proposed in this paper still has high application value. The overhead of edge nodes is not analyzed in this section as a subsequent study. This formula simulates data flow between nodes. "Transmission time" is the time it takes one node to send data to another. The period between the second node's data receipt and the first node's acknowledgment is the confirmation time. blockchain's decentralized nature and need for consensus. The complete latency may include the time it takes all nodes to validate a new block or transaction and reach a consensus after broadcasting it.

Latency = (Time taken to send data + Time taken to confirm receipt) / 2.

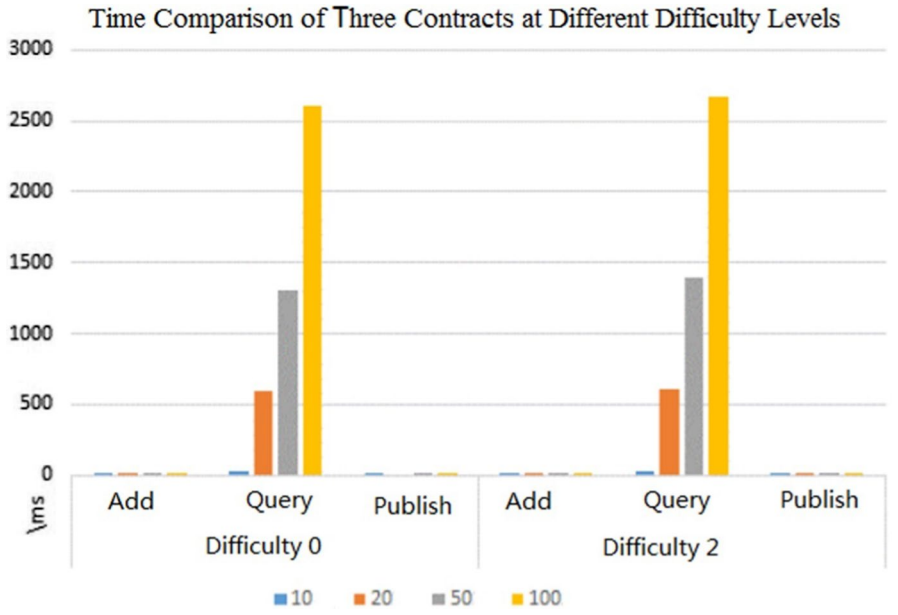

Fig. 14 Three types of contract time consumption at different difficulty levels

5.4.2 Optimization of consensus mechanism

Optimization is performed for the PoW consensus mechanism, and the data consistency efficiency of the distributed system is tested by comparing the cost time of building different blocks with the same number of nodes and difficulty of 3 for the access control scheme and PoW consensus mechanism in this paper. The number of blocks in the experiment is set from 1 to 100, and the results are shown in Fig. 15. The blue part shows the optimized PoW consensus mechanism in this paper, and the gray part shows the original PoW consensus mechanism, where the highlighted peak is the time-consuming waiting time when reaching consensus. In terms of average time, the construction time per 100 blocks is reduced by 17.11% compared to PoW. It can be seen that the optimization in this paper outperforms PoW in the case of a growing number of blocks.

Table 2 represents a comparison of our attribute-based access control model, which integrates blockchain and edge computing to enhance the security and privacy of IoT systems, with other relevant models in the field.

Table 1 compares our attribute-based access control method to earlier studies. Blockchain and edge computing safeguard IoT privacy and security.

- "Security Model" is the current access control strategy.
    - "Computing Paradigm" refers to cloud, edge, or fog computing environments where the paradigm is implemented.

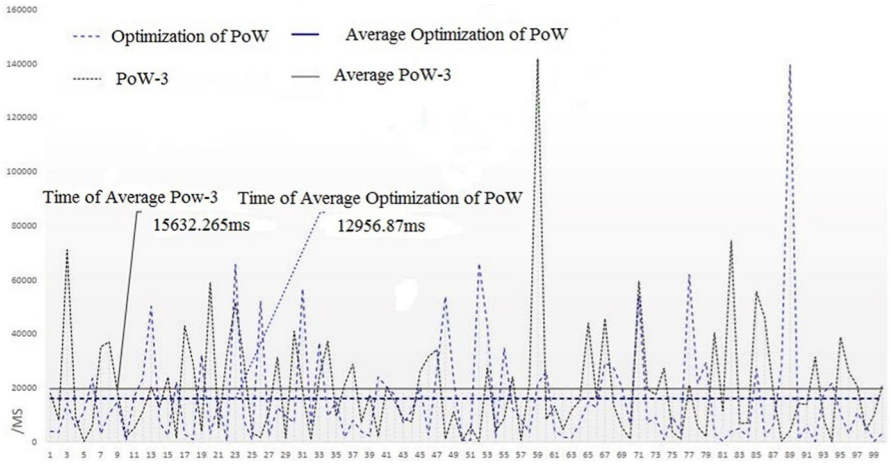

Fig. 15 Comparison of optimization algorithms and PoW based on difficulty 5

Table 2 Comparative results of our attribute-based access control model that leverages blockchain and edge computing for IoT system security and privacy

	Our work	[18]	[6]	[4]
Security model	Attribute based	Role based	Policy based	Attribute based
Computing paradigm	Edge computing	Cloud computing	Fog computing	Edge computing
Blockchain integration	Yes	Yes	No	Yes
Access control latency (ms)	1.3 (500th access)	1.7 (500th access)	0.9 (500th access)	1.5 (500th access)
Max. concurrency (users)	500	400	600	450
CPU utilization (%)	75	80	70	80
Data privacy feature	Yes	No	Yes	Yes
IoT integration	Yes	No	Yes	No

- The "Blockchain Integration" column indicates that the study's model included blockchain technology.
- "Access Control Latency" refers to the 500th person's access delay.
- The "Max. Concurrency" setting determines how many users can log in at once.
    - "CPU Utilization" is the proportion of the central processing unit used.
- The "Data Privacy Feature" shows if the model protects user data.
    - "IoT Integration" specifies whether the research uses IoT devices.

6 Conclusion and limitations

The main contribution of this paper is to design an access control model for terminal devices and resources through the edge nodes of the Internet of Things in combination with blockchain technology, aiming to provide a new solution for the deficiencies of access control research on edge computing and blockchain in the current Internet of Things security system. The specific work is to design the access control architecture of integrated edge computing on the basis of the original integrated edge computing Internet of Things system. The SC-ABAC model is proposed by combining smart contract in blockchain with attribute-based access control. Considering the security performance, an access control management mechanism based on blockchain is implemented based on ECC and ECDSA, and the security reliability of the model and architecture is verified by experiments. In terms of average time, the construction time per 100 blocks is reduced by 17.11% compared to PoW. The inadequacies and future work direction of this paper are, considering that this research only focuses on the access control of data under the edge Internet of Things and does not deal with the existing environment of data and personal privacy information, to try to combine attribute-based encryption to build an efficient and privacy-protecting edge Internet of Things environment.

Acknowledgements Researchers appreciate the funding support provided by "Scientific research projects of Jilin Education Department in 2023. Project No.: JJKH20230022KJ

Author contributions CZ, SBG Conceptualization, Formal analysis, Methodology, Writing—original draft, Data curation, Investigation, Software, Validation; Visualization; SBG Supervision, Writing—review & editing; ASR and PS Resources, Project administration, Writing—review & editing- SBG;

Funding Scientific research projects of Jilin Education Department in 2023. Project No.: JJKH20230022KJ.

Data availability Not Applicable.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethical approval The are no human and animal subjects in this article, and informed consent is not applicable.

Consent to participate All authors consent to participate in this research work. This research work may not be of direct benefit to us. Our participation is completely voluntary.

Consent to publish All authors give consent for the publication of the research paper in your esteemed journal.

References

1. Puliafito C, Mingozzi E, Anastasi G (2017) Fog computing for the internet of mobile things: issues and challenges. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP) (pp 1–6). IEEE. https://doi.org/10.1109/SMARTCOMP.2017.7947010

2. Shi W, Cao J, Zhang Q, Li Y, Xu L (2016) Edge computing: vision and challenges. IEEE Internet Things J 3(5):637–646. https://doi.org/10.1109/JIOT.2016.2579198
3. Yang Q, Lu R, Rong C, Challal Y, Laurent M, Wang S (2019) Guest editorial the convergence of blockchain and iot: opportunities, challenges and solutions. IEEE Internet Things J 6(3):4556–4560. https://doi.org/10.1109/JIOT.2019.2921235
4. Hu VC, Kuhn DR, Ferraiolo DF et al (2015) Attribute-based access control [J]. Computer 48(2):85–88
5. Dwivedi AD, Srivastava G, Dhar S, Singh R (2019) A decentralized privacy-preserving healthcare blockchain for IoT. Sensors 19(2):326. https://doi.org/10.3390/s19020326
6. Lin C, He D, Kumar N, Huang X, Vijayakumar P, Choo KKR (2019) HomeChain: a blockchain-based secure mutual authentication system for smart homes. IEEE Internet Things J 7(2):818–829. https://doi.org/10.1109/JIOT.2019.2944400
7. Ren Y, Leng Y, Cheng Y (2019) Secure data storage based on blockchain and coding in edge computing [J]. Math Biosci Eng 16(4):1874–1892. https://doi.org/10.3934/mbe.2019091
8. Tuli S, Mahmud R, Tuli S et al (2019) FogBus(2019): A blockchain-based lightweight framework for edge and fog computing [J]. J Syst Softw 154:22–36. https://doi.org/10.1016/j.jss.2019.04.050
9. Ren Y, Zhu F, Sangaiah AK (2019) Identity management and access control based on blockchain under edge computing for the industrial internet of things. Appl Sci 9(10):2058. https://doi.org/10.3390/app9102058
10. Omar YA, Goyal SB, Varadarajan V (2021) Apply blockchain technology for security of IoT devices. In: 2021 Emerging Trends in Industry 4.0 (ETI 4.0), 2021, pp. 1–6, https://doi.org/10.1109/ETI4.051663.2021.9619295
11. Cheng GJ, Huang ZJ, Deng SG (2020) IoT data management based on blockchain and edge computing [J]. J Internet Things 4(02):1–9
12. Andaloussi Y, El Ouadghiri MD (2018) Access control in IoT environments: feasible scenarios. Procedia Comput Sci 130(2018):1031–1036. https://doi.org/10.1016/j.procs.2018.04.144
13. Kumar R, Kumar P, Aljuhani A, Islam AKMN, Jolfaei A, Garg S (2022) Deep learning and smart contract-assisted secure data sharing for IoT-based intelligent agriculture. IEEE Intell Syst. https://doi.org/10.1109/MIS.2022.3201553
14. Kumar P, Kumar R, Kumar A, Franklin AA, Garg S, Singh S (2022) Blockchain and deep learning for secure communication in digital twin empowered industrial IoT network. IEEE Trans Netw Sci Eng. https://doi.org/10.1109/TNSE.2022.3191601
15. Nwosu AU, Goyal SB, Rajawat AS, Islam SMN, He J, Aslam M (2022) An innovative blockchain-based secured logistics management architecture: utilizing an RSA asymmetric encryption method. Mathematics 10(24):4670. https://doi.org/10.3390/math10244670
16. Hieb J, Schreiver J, Graham J (2012) Using bloom filters to ensure access control and authentication requirements for SCADA field devices. In: Critical Infrastructure Protection VI: 6th IFIP WG 11.10 International Conference, ICCIP 2012, Washington, DC, USA, March 19-21, 2012, Revised Selected Papers 6 (pp 85-97). Springer Berlin Heidelberg
17. Wang SY, Zou SH (2021) Blockchain and capability based access control mechanism in multi-domain IoT [J]. J Appl Sci Electron Inf Eng 39(01):55–69
18. Bedi P, Goyal SB, Kumar J, Kumar S (2021) Blockchain integrated framework for resolving privacy issues in smart city. In: Chakraborty C, Lin JCW, Alazab M (eds) Data-Driven Mining, Learning and Analytics for Secured Smart Cities. Advanced Sciences and Technologies for Security Applications. Springer, Cham. https://doi.org/10.1007/978-3-030-72139-8_6
19. Zhang JG, Hu XH (2021) Research on improved access control mechanism of internet of things devices based on ethereum[J]. Comput Eng 47(4):32–39
20. Omar HOM, Goyal SB, Varadarajan V (2021) Application of sliding window deep learning for intrusion detection in fog computing. In: 2021 Emerging Trends in Industry 4.0 (ETI 4.0) (pp 1-6). IEEE. https://doi.org/10.1109/ETI4.051663.2021.9619421
21. Rajawat AS, Bedi P, Goyal SB, Alharbi AR, Aljaedi A, Jamal SS, Shukla PK (2021) Fog big data analysis for IoT sensor application using fusion deep learning. Math Problems Eng 2021:1–16. https://doi.org/10.1155/2021/6876688
22. Singh Rajawat A, Bedi P, Goyal SB, Shukla PK, Zaguia A, Jain A, Monirujjaman Khan M (2021) Reformist framework for improving human security for mobile robots in industry 4.0. Mobile Inf Syst 2021:1–10. https://doi.org/10.1155/2021/4744220

23. Goyal SB, Bedi P, Kumar J et al (2021) Deep learning application for sensing available spectrum for cognitive radio: an ECRNN approach. Peer-to-Peer Netw Appl 14:3235–3249. https://doi.org/10.1007/s12083-021-01169-4
24. Nwosu AU, Goyal SB, Bedi P (2021) Blockchain transforming cyber-attacks: healthcare industry. In: Innovations in Bio-Inspired Computing and Applications: Proceedings of the 11th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2020) held during December 16–18, 2020 11 (pp 258-266). Springer International Publishing. https://doi.org/10.1007/978-3-030-73603-3_24
25. Goyal SB, Bedi P, Kumar J, Ankita (2022) Realtime accident detection and alarm generation system over IoT. In: Kumar R, Sharma R, Pattnaik PK (eds) Multimedia Technologies in the Internet of Things Environment, Volume 2. Studies in Big Data, vol 93. Springer, Singapore. https://doi.org/10.1007/978-981-16-3828-2_6
26. Nyamtiga BW, Sicato JCS, Rathore S, Sung Y, Park JH (2019) HomeChain: a blockchain-based secure mutual authentication system for smart homes. Electronics 8(8):828. https://doi.org/10.3390/electronics8080828
27. Ali G et al (2020) xDBAuth: blockchain based cross domain authentication and authorization framework for internet of things. IEEE Access 8:58800–58816. https://doi.org/10.1109/ACCESS.2020.2982542
28. Nyame G, Qin Z, Obour Agyekum KOB, Sifah EB (2020) An ECDSA approach to access control in knowledge management systems using blockchain. Information 11(2):111. https://doi.org/10.3390/info11020111