Hindawi
International Journal of Distributed Sensor Networks
Volume 2023, Article ID 5963039, 13 pages
https://doi.org/10.1155/2023/5963039

Research Article

Improved Private Data Protection Scheme for Blockchain Smart Contracts

Sheng Hu

Chongqing Industry Polytechnic College, Chongqing 401120, China

Correspondence should be addressed to Sheng Hu; husheng@cqipc.edu.cn

Received 21 March 2023; Revised 19 June 2023; Accepted 2 August 2023; Published 24 August 2023

Academic Editor: Yuedong Xie

Data security and privacy protection are critical challenges that constrain the advancement of edge computing. Similarly, blockchain technology faces constraints in addressing security issues linked with edge computing due to its scalability limitations. To tackle these challenges and promote the development of blockchain technology, this paper presents a scheme that enhances privacy data protection in blockchain smart contracts using edge computing and a master-slave multichain architecture. Firstly, we propose a master-slave multichain architecture based on the traditional single chain and integrate it with a three-layer edge computing structure to address security issues on the edge side. We also design a signature authentication scheme utilizing ECC integrated with blockchain encryption technology. Secondly, we incorporate the role-based access control (RBAC) model with smart contracts to finely divide user privileges, construct an interdomain role-based access control (ID-RBAC) model, and provide detailed access authentication process designs for both within and between domains. Finally, experimental results demonstrate that our proposed scheme can effectively resist various attacks, significantly improve algorithm efficiency, and maintain a system overhead of less than 160 p, with a maximum transaction throughput of nearly 310 tx/s.

1. Introduction

As the use of blockchain technology continues to grow, its security faces constant challenges, requiring ongoing evolution of its underlying technology. Literature [1] introduces a method for privacy protection in blockchain. In literature [2], an analysis and comparison of three types of blockchain privacy protection technologies are presented. With the increasing amount of data stored in blockchain systems, it becomes crucial to address issues such as data query efficiency, as well as data security and reliability. This paper focuses on analyzing data security from the perspective of data management security. In literature [3], the issue of data storage is raised as blockchain systems evolve to handle specific business data. The paper categorizes blockchain data into identity data and data privacy, aiming to analyze associated security concerns. From the standpoint of blockchain credibility, literature [4] examines the credibility of the blockchain network. Smart contract technology, which acts as the interface between the current blockchain system and business operations, is susceptible to vulnerabilities. The data management model in blockchain systems exhibits flaws, resulting in challenges for application development and a high degree of coupling between the system and applications. Blockchain data security sharing refers to the process by which different nodes on the blockchain read and utilize data shared by other nodes to protect node identity privacy and prevent data leakage. In literature [5], blockchain query technology is analyzed from two perspectives: general query processing and trusted query processing. Due to the use of data redundancy in blockchain systems to ensure data integrity, all nodes in the blockchain network must store backups of all the data. However, as time passes, the high redundancy of data on the blockchain system places a significant memory burden on each node. When the data to be stored in the blockchain network exceeds the storage capacity of most nodes, it not only reduces the difficulty for malicious nodes to carry out malicious activities but also compromises reliability, potentially leading to security issues within the blockchain system.

Due to the significantly higher cost of on-chain storage in blockchain systems compared to general databases, efforts are being made to reduce data redundancy by adopting a combination of on-chain and off-chain storage approaches. As the blockchain system evolves, data storage can be categorized into on-chain storage and on-chain collaborative storage. On-chain storage refers to the storage of data directly on the blockchain, where all nodes within the network are required to store the complete data. On the other hand, on-chain collaborative storage involves the blockchain system storing only the metadata of data uploaded by users, while the complete data is stored in selected nodes. This paper examines the challenges associated with on-chain storage and on-chain collaborative storage and presents several solutions to address these issues. The aim is to optimize data storage efficiency while ensuring the integrity and security of the blockchain system.

In the study of smart contracts, it has been observed in literature [6] that there is a high rate of code repetition. To address this issue, a method of contract updating with differentiated code was proposed. Additionally, a loosely coupled smart contract model, as suggested in literature [7], tackles the challenges associated with costly upgrades and development of certain contracts, as well as redundant code storage. Literature [8] introduces an adaptive smart contract algorithm. Regarding the security of blockchain data during network transmission, literature [9] focuses on ensuring security by imposing restrictions on network node access through a certificate authority (CA). Another area of research involves encrypting and obfuscating data to prevent nonowners from accessing the true value of the data, even if it is made public. Several methods such as ring signature, homomorphic encryption, and sequentially preserving encryption are employed for data encryption on the blockchain. For example, Blindcoin utilizes a coin-based mechanism for transaction privacy protection [10], Monroe Coin employs ring signatures for encrypted data protection [10], and Zcash employs zero-knowledge proof for encrypted data protection [11]. BlockOPE is a data protection method that employs sequence-protected encryption [12].

The Internet of Things refers to a large network that combines various sensing devices with networks. At present, the growth rate of the network data has far exceeded the load limit of the network bandwidth of the centralized processing mode and has been unable to meet the multidimensional, real-time service requests of IoT. Edge computing has emerged as a novel computing paradigm [13], offering an advantageous environment for real-time communication, collaboration, and storage among diverse IoT devices. In this context, the edge nodes that offer services are geographically distributed, diverse, and operated in an uncertain environment. Consequently, these edge nodes are exposed to potential security risks when connected to malicious nodes. Such malicious nodes can bypass the security authentication mechanism and launch attacks on other edge devices, thereby compromising user privacy. In order to promote the development of edge computing in IoT applications, it is urgent to solve the problem of secure access to its data.

Blockchain, as an emerging information processing technology, possesses inherent strengths in security and access control [14]. The white paper on blockchain edge computing technology highlights the mutually beneficial and synergistic relationship between edge computing and blockchain technology. However, it is worth noting that blockchain 1.0 and blockchain 2.0 have compromised scalability in their pursuit of decentralization [15]. Furthermore, as the trading volume continues to increase, the overall performance of the system becomes increasingly limited by the maximum capacity of individual nodes, resulting in a developmental bottleneck [16]. In addition, the security of blockchain network is guaranteed based on the fact that each node stores all transactions on the chain for verification, so it needs high scalability to support its high security. These problems prevent it from playing a maximum role in the security of edge computing, which does not conform to the service purpose of large bandwidth and low delay of edge computing.

In light of the blockchain performance bottleneck, some scholars have proposed on-chain and off-chain capacity expansion schemes. However, there is a mutual restriction relationship between security and performance when applying capacity expansion schemes [17], especially in cross-chain interaction. Therefore, in deploying an edge computing network architecture based on capacity expansion, security interaction between edge trust domains constituted by blockchain should be considered.

Furthermore, there are difficulties in identity authentication, access control, and privacy protection in the process of secure interzone device interaction [18]. Currently, the existing approach to building interdomain trust involves setting up a central authority within the domain to manage and authenticate identities. However, due to the centralized authentication mode, mutual identity verification cannot be carried out between node devices, leading to a lack of cross-domain identity trust. As a result, IoT devices in different trust domains are unable to easily access each other, and data cannot be shared securely [19].

To address these challenges of data security and cross-domain authentication in the edge computing and blockchain network architecture, this paper proposes a highly scalable distributed trusted authentication model that supports cross-domain access control. We design a master-slave multichain structure and integrate edge computing to achieve this goal. The main contributions of this paper include the following:

(1) To address the scalability bottleneck of blockchain, we designed a scalable master-slave multichain structure using off-chain expansion methods. A large number of cross-domain operations were transferred to the main chain, reducing the transaction burden on the slave chains. The slave chains can provide data availability proof without submitting all transaction data to the main chain

(2) Based on the implementation ideas of on-chain expansion schemes, we integrated the master-slave multichain structure into edge computing and

deployed a three-tier architecture. We designed a secure access process for edge nodes based on elliptic curve cryptography (ECC). By endowing trust to the edge through the master-slave chain, we improved the security and computational effectiveness of the network architecture

(3) We proposed an interdomain role-based access control (ID-RBAC) model based on role-based access control (RBAC) and designed fine-grained access control policies combining smart contracts and role permissions

(4) Based on the ID-RBAC model, we designed intradomain and interdomain access control mechanisms, including specific identity authentication processes and cross-domain data management methods

This paper is divided into five main sections: the Introduction, the State of the Art, the Methodology, the Result Analysis and Discussion, and the Conclusion.

2. State of the Art

2.1. Master-Slave Multichain Design. In order to break the bottleneck of traditional single-chain performance, a master-slave multichain structure is designed, as shown in Figure 1 (Node0 is the communication node, and the rest are ordinary nodes). The master-slave multichain structure includes one main chain (MC) and several slave chains (SC). As a trusted platform within the domain, SC manages the access operations within the domain and defines common nodes and communication nodes. Common nodes are responsible for data storage, and communication nodes are the hubs of network interaction connected to the MC. MC is a bridge between chain interactions, used to resolve cross-chain requests, to achieve trusted identity authentication.

MC defines communication node and cache node. Communication node interacts with SC network to realize interconnection and interworking between chains. Cache node caches cross-domain data through CouchDB status database. Communication nodes constitute the index of MC and SC and connect multiple SC to form an infinitely extended master-slave multichain, which has good flexibility and expansibility. As a certificate management server, membership service provider (MSP) participates in the maintenance of the local blockchain ledger and conducts identity verification and certificate issuance for nodes that join the blockchain. The advantage of the master-slave chain structure is its strong scalability. The slave chain can be dynamically extended, so that the performance of the whole system is not limited to a certain chain, and its scalability bottleneck is broken. The main chain, as a trusted authentication platform, keeps the hash time lock of its transactions and maintains the atomicity of interchain transactions.

2.2. Distributed Security Architecture Based on Master-Slave Multichain. In the master-slave multichain integrated edge computing, a three-layer distributed security architecture is designed, as shown in Figure 2, including the device layer, slave network and master network, and three layers of bottom-up service. The device layer provides trusted computing services for the upper layer. After successfully completing the identity registration process, the edge devices in this layer gain access to the slave network and become the "miners" of the slave network, referred to as edge nodes (En). En preprocesses data and stores it in the SC node. The SC provides a secure data storage environment and intradomain access control for devices in the current domain. Communication nodes in the SC and MC work together to maintain reliable communication and provide services for cross-domain access control. The backbone network supports access across different SC domains. The three-tier architecture covers the core functions of blockchain and edge computing, providing distributed security services from different layers of storage, network, and computing. Slave chains and edge nodes in the architecture can be developed on demand, which is an infinitely expanding alliance.

(1) The lower layer is the device layer, which has two functional modules: awareness module and device management module. In the device management module, a cryptographic-based secure access process is designed. Devices can become legitimate En only after being verified by the process

(2) The middle layer is the slave link network, which contains two functional modules: data processing and intradomain access control. In the data processing module, En preprocesses the data collected at the equipment layer, packages the data into blocks after unified format, and stores them in SC books. The intradomain access control module is the core functional module of the middle layer, which realizes the intradomain access management of the Internet of Things

(3) The upper layer is the main chain network. As a trusted sharing platform between domains, this layer is the controller for data cross-domain management. The cache module and interzone access control module are configured on this layer. The interzone access control module jointly manages the cross-domain access behavior with the SC, and the cache module caches the cross-domain data and related information

2.3. Interzone Access Control Model Based on RBAC

2.3.1. ID-RBAC Model. In the access control scheme of the Internet of Things, common access control models include attribute set and permission set. The set protection measures of these methods are not secure enough and the design cost is high [20]. Under edge computing and block multichain architecture, the access of multiple devices and multiple nodes intensifies the complex relationship between access and permission, making authorization management tedious. At the same time, frequent application, authorization, and access operations increase the cost of authority management. Role-based access control model can decouple the relationship between users and permissions, support hierarchical permission hierarchy

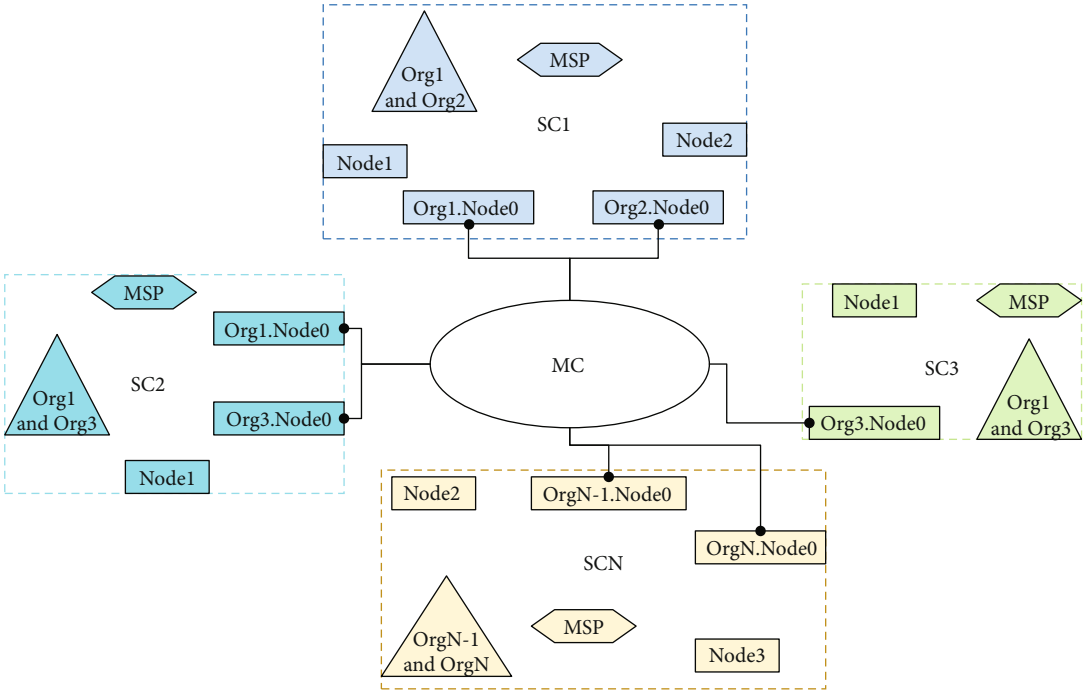

Figure 1: Master-slave multichain structure.

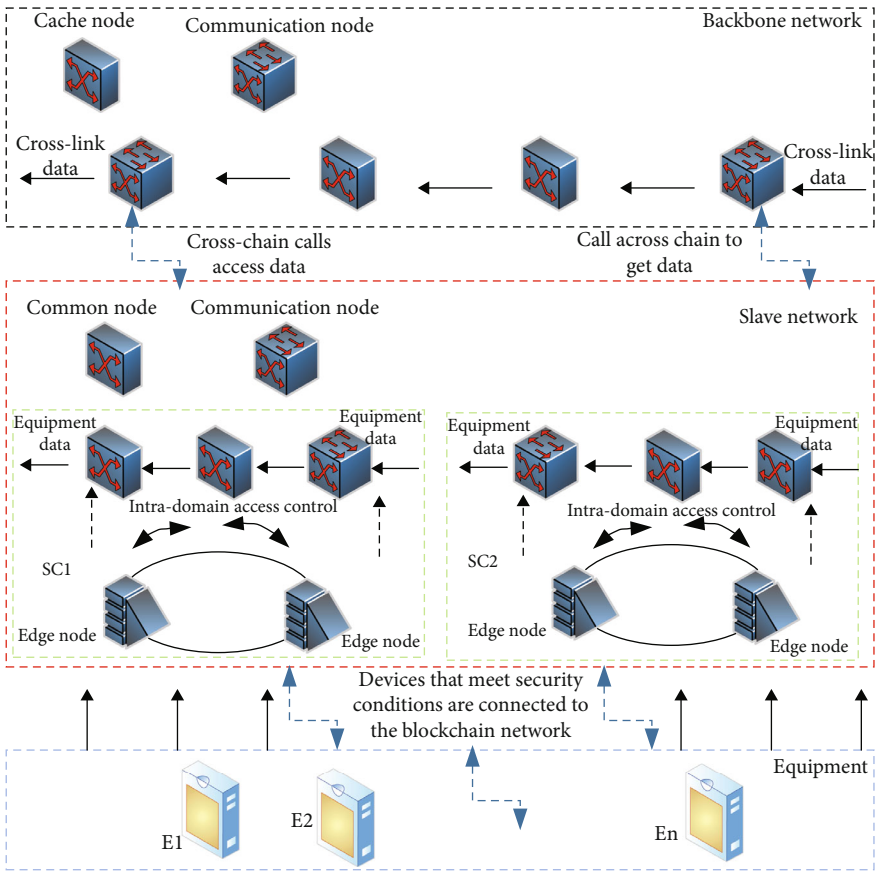

Figure 2: Distributed security architecture.

and permission inheritance, and greatly simplify the management of permissions. However, because user permissions are divided according to roles, it is difficult to achieve fine-grained access control, which usually requires creating more specific role versions or designing other mechanisms [21]. Smart contracts have high accuracy and intelligence and can be automatically executed without human involvement. Smart contracts are used to assign access user roles consistent with their identity information, so that the change of the relationship between the recipient role and the user is more frequent than the change of the authority corresponding to the role [22], which can reduce the complexity of authorization and reduce the management cost of the organization. Based on RBAC model, this paper uses smart contracts to logically divide users, roles, and permissions and proposes an interdomain access control model, ID-RBAC.

Under the ID-RBAC model, a dynamic authorization mechanism is designed to enable the adjustment of user permissions based on their different states. Authorization rules are defined to determine the permissions assigned to users. For instance, if role A possesses permission I and satisfies the authorization policy within the intradomain access process, permission I is granted. However, in the interdomain access process, if the permission fails to meet the cross-domain authorization policy, permission I is restricted. This means that the same role should have the same permissions for the same access content. However, in ID-RBAC, permissions may vary during intradomain and interdomain access due to the fine-grained contract mechanism. The ID-RBAC model is flexible for complex authorization states in different domains. When a role needs to be added to the system, the role can be written to the contract mechanism for dynamic permission authorization. ID-RBAC has better scalability and flexibility than traditional RBAC.

2.3.2. Contract Design of ID-RBAC. The process of role allocation and access control is realized by intelligent contract. Five contracts are designed based on ID-RBAC, which, respectively, realize data storage management, private data access control, user role management, data cross-domain access control and cross-domain data cache, and forwarding.

(1) Data management contract (DMC) {address, resource, attribute}: the DMC is responsible for managing data on the blockchain and performs domain and classification management of the data through the DMC. An address signifies the IoT domain where a data resource is located, and an attribute represents the data type of the resource

(2) Private data contract (PDC) {DMC attribute, Policy, time}: the PDC utilizes the data attribute from the DMC to construct a privacy data set (PDS). The PDS is stored in a private database (private-DB), which is exclusively accessible to a specific member of the system

(3) Role management contract (RMC) {user attributes, user address, PDC-Policy result, user}, {Role 1, Role 2,..., role N}, {Authority 1, authority 2,..., authority N}: the RMC handles the management of roles in the system. When a new user joins the network, the RMC matches the user's identity based on their attributes and the domain they belong to. A user can hold multiple roles concurrently

Table 1: Division of domain rights.

Role	Permissions
IoTl member	The SC1 ledger corresponding to IoT1 can be entered
IoT2 member	The SC2 ledger corresponding to IoT2 can be entered
IoT3 member	SC3 ledger corresponding to IoT3 can be entered

According to the different Internet of Things domains, users can be divided into codomain and exotic users, and each domain maintains a domain from the chain, so each domain has a corresponding ledger. Therefore, the division of permissions between domains is mainly for the ledger maintained by the domain. Table 1 lists the permissions of domains. It is intended to clearly indicate the permissions of different users.

To enable fine-grained access control, the data is classified, and different access thresholds are established. Based on the role management contract (RMC), users within the domain are categorized into four levels: level I, level II, level III, and level IV. The corresponding rights for each level are outlined in Table 2. It is designed to show the relationship between these permissions for corresponding users.

(4) Cross-domain contract (CrossD) {access attribute, RMC result}: according to the request operation parsing access attribute and combined with RMC to obtain the corresponding permissions after cross-domain invocation

(5) Cache contract (cacheC) {CrossD result, attribute}: executes CacheC through CrossD result and classifies cache according to data attribute

2.4. Security Analysis. According to the characteristics of blockchain, its security is analyzed from the following aspects.

(1) Witch attack

Each new user must register with the SC and pass the SC identity authentication before joining the affiliate system. Therefore, this scheme can effectively prevent witch attacks.

(2) Long-range attack

The scheme requires each node to check the latest block regularly, which can make sure that at least one check is included in the period of return of credits, so the node will not choose the longest chain created by the attacker. Therefore, this scheme can effectively contain long-range attacks.

Table 2: Relationship between role levels and permissions.

User	Role	Permissions
User A	IV level	Access to the ledger's public-DB data
User B	III level	Public-DB data and private-DB1 data in the ledger can be accessed
User C	II level	Access to public-DB data and private-DB2 data in the ledger
User D	I level	Public-DB, private-DB1, and private-DB2 data can be added, deleted, or modified in the audit book

(3) Internal attacks

In the initialization phase, the user uses the digital signature sign sent by the SC and his/her own public key to create an account that is unique. In addition, distributed storage can also prevent malicious users in the system from impersonating other legitimate users to launch internal attacks.

(4) Modify the attack

The header of each block contains the hash value of the previous block and the timestamp of the current block, which ensures that the data has been modified or removed. Therefore, modification attacks are not effective against the scheme.

3. Methodology

3.1. Edge Initialization. In order to ensure the security of edge nodes providing computing services, the secure access process as shown in Figure 3 is designed, including initial value setting, registration, and identity authentication process. The initialization of the initial value is performed by En, and the public and private key pairs are generated through ECC as the public parameters required by the system. The identity information Enc is obtained by combining the MAC (media access control) address value of the device, and it is packaged and stored on the blockchain to register itself with the blockchain network. When device A sends an access request to device B, as shown in Figure 3, device B authenticates A's identity through the blockchain network.

The specific process of edge node authentication is as follows:

(1) Initial value setting

An En joining a blockchain network uses ECC to calculate public and private keys. When Q satisfies a prime greater than 3 on a finite field Fp, the integer modulo p has an equation

$$y^2 = x^3 + ax + b(\text{mod } p), \tag{1}$$

where $a, b \in p, \text{Ep}(a, b)$, taking any number K to get the private key SK. TakeQas the base point on the elliptic curve, generate the public key, and broadcast the public key to the whole network. The expression is shown in

$$\text{PK} = Q^*\text{SK}. \tag{2}$$

(2) Registration

Enter the MAC address and SK value of En into Formula (3) to calculate the hash. Enc(SK, Hash) is obtained by encrypting it through SK, and Enc is stored locally and on blockchain to complete registration.

$$\text{Hash} = \text{SHA256}(\text{MAC} + \text{SK}). \tag{3}$$

(3) Identity authentication

Before becoming a miner, En must undergo a consensus recognition process, which involves the verification of its identity by all network nodes. When node A initiates an action, such as accessing node B, node B checks whether A's identity information exists on the blockchain. The following scenarios may occur:

If A's identity information is found on the blockchain, the encrypted message Enc(SK, Hash) is decrypted using A's public key (PK). By obtaining Dec(SK, Hash), node B can compare it with Enc(SK, Hash). If the comparison results match, it verifies that the node is legitimate and authenticated. Any discrepancy suggests that the node has been compromised or forged by malicious nodes.

If A's identity information is not found on the blockchain, the node is considered invalid, and the connection is terminated. This authentication process prevents nodes from being forged or impersonated, ensuring that data is not delivered to malicious nodes.

Through this process, En establishes an initial trust relationship with the data ledger, which enhances the overall security and integrity of the system.

3.2. Domain Initialization. The MSP that issues and verifies certificates generates public and private keys for the IoT domains and entities that join the master-slave network and publishes the domains and users to the master-slave network.

(1) Publishing domains $(\text{Hash}_i(\text{pk}_{\text{MSP}_i}), \text{Sign}_i, \text{Hash}_i)$, where $\text{Hash}_i(\text{pk}_{\text{MSP}_i})$ is the hash value of pk_{MSP_i} and Sign_i and Hash_i, respectively, represent the digital signature and hash operations carried out by domain X

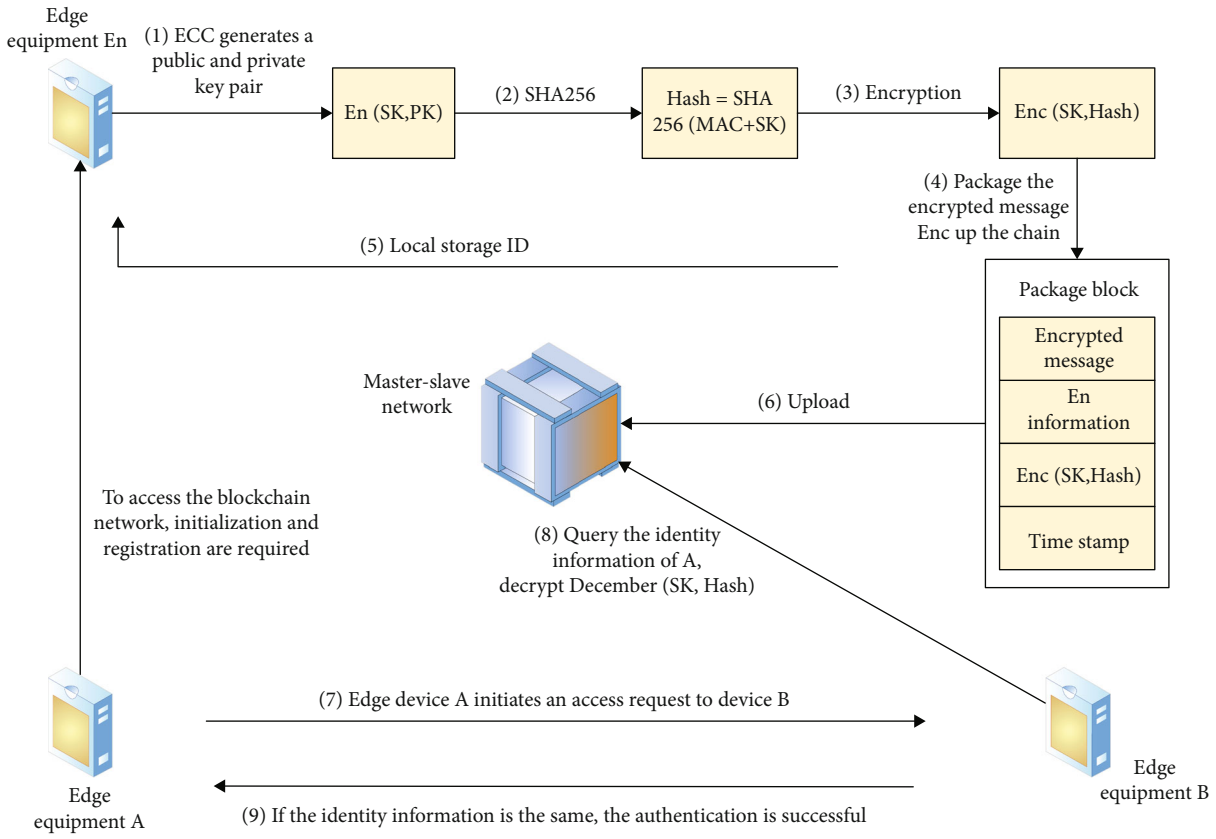

Figure 3: The edge node authentication process of the proposed model.

(2) Publish users $(\text{Hash}_i(\text{pk}_{\text{PGx}}), \text{Hash}_i(\text{pk}_{\text{MSP}_i}), \text{State})$, where pk_{PGx}, represents the public key of user Pix in domain I and State indicates whether user identity information is available

(3) Verify $(\text{pk}_{\text{PGx}}, \text{Sign}_i(\text{sk}_{\text{PGx}}, T), T, \text{Hash}_i(\text{pk}_{\text{MSP}_i}))$, where T is a random number for hash computation

(4) Store the verification results in the master-slave chain ledger

(5) After the blockchain network is authenticated, it starts to respond to the requests of the publishing domain $(\text{Hash}_i(\text{pk}_{\text{M}_{\text{SP}_i}}, \text{Sign}_i, \text{Hash}_i, \text{Sta}_{\text{BC}})$ and the publishing user $(\text{Hash}_i(\text{pk}_{\text{PGx}}), \text{Hash}_i(\text{pk}_{\text{MSP}_i}), \text{State})$. Sta_{BC} means to maintain the blockchain network information of the newly published domain

3.3. Access Authorization

3.3.1. Private Data Access Management. PDC and RMC provide hierarchical protection for private data. PDC defined privacy data sets (privacy dataset) for the privacy data of each domain, and RMC accessed and managed the access policy PolicyT defined in PDS.

Definition 1. The following is the privacy data access policy: PolicyT (privacy data attribute Type, Domain where the data resides, and access permission P). It is specifically expressed as

$$\text{Policy}_T = \left\{ \begin{array}{c} (\text{Type 1, IoT 1, Level} - \text{IV}) \\ \left(\text{Type 2, IoT 2,} \left\{ \begin{array}{l} \text{Level} - \text{IV,} \\ \text{Level} - \text{III} \end{array} \right\} \right) \cdots \\ \left(\text{Typex, IoTx, ,} \left\{ \begin{array}{l} \text{Level} - \text{IV,} \\ \text{Level} - \text{I, } \cdots \end{array} \right\} \right) \end{array} \right\}. \tag{4}$$

The private data access management process is shown in Figure 4. Let Org1 have data for three attributes {public-DB1 (attribute 1, attribute 2), private-DB1 (attribute 3)}. Attribute 3 is divided into private data sets and stored in private databases. Organization members on the same chain can share data in public-DB1, while data in private-DB1 is Org1 private. Private data policy Policy_1 is expressed as

$$\text{Policy}_1 = \{\text{Type 3, IoT 1, Level}_{\text{IV}} - \text{Org1}\}. \tag{5}$$

3.3.2. Intrazone Access. In-domain access refers to the access behavior of different IoT systems within the same marginal IoT domain. As shown in Figure 5, user A and user B are

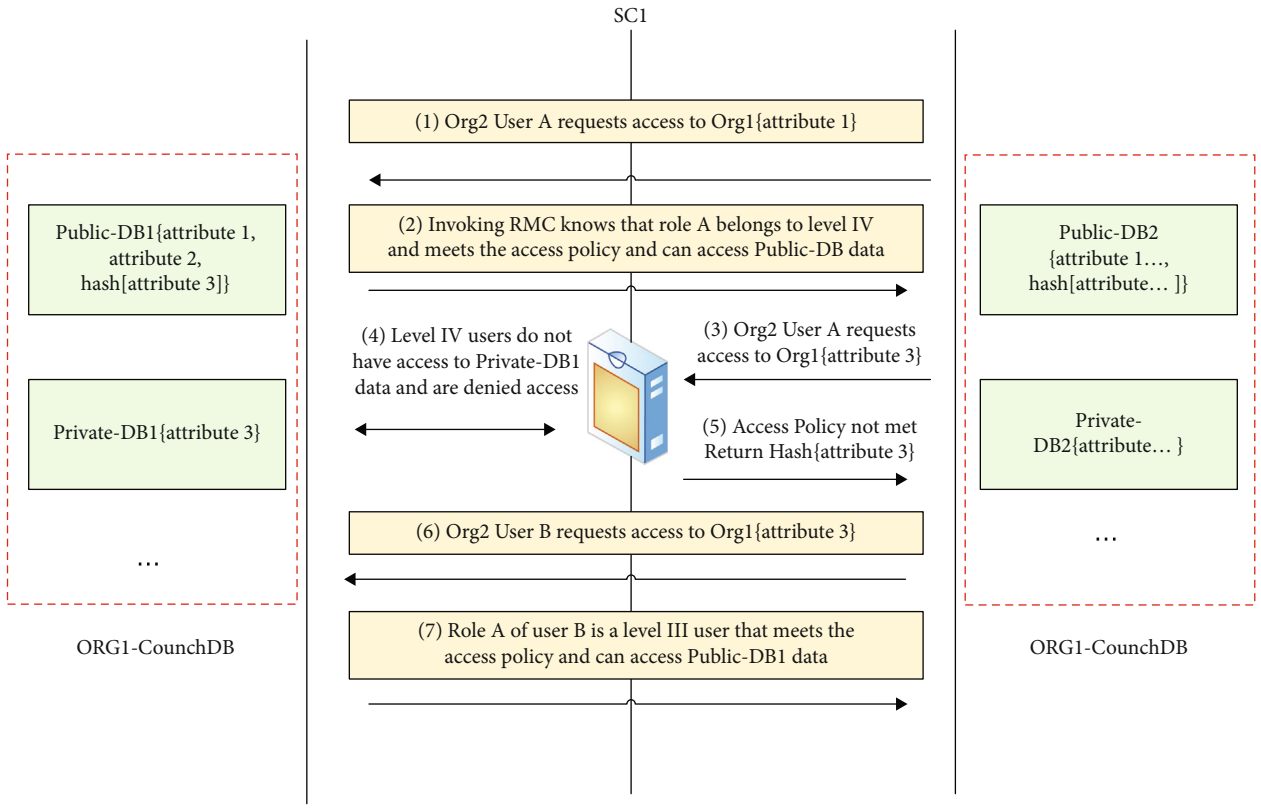

Figure 4: The proposed model of private data access management flow.

assumed to be in the domain managed by SC1, and user A publishes access behaviors to user B in SC1.

(1) IoT1-A sends a request with the status of identity registration to the MSP. After the authentication is successful, the MSP returns the identity certificate Cert_G to the MSP. Time indicates the validity time of Cert_G.

$$A \longrightarrow \text{MSP} : \left\{ \text{pk}_{\text{PG}}, \text{Status}, \text{Sig}_{\text{sk}_{\text{IoT1}}} \left(\begin{array}{c} \text{sk}_{\text{PG}}, \\ \text{Hash}_G(\text{pk}_{\text{PG}}) \end{array} \right) \right\},$$
$$\text{MSP} \longrightarrow A : \text{Cert}_G \left\{ \text{sig}_{\text{MSP}_{\text{pk}}} (\text{States} : \text{True}, \text{Time}) \right\} \tag{6}$$

(2) A package its Cert_G, its own information and access content Ac into access request Ar. After the Ar packaging is successful, it will be written into the account book, and the block height will be increased by 1. PackagedAr indicates the current Ar packaging status. The status can be success or failure

The Ar is sent to SC1 for parsing verification, where Dn represents the data set accessed by A. SC1 is A slave network that maintains the domain where A resides. As a trusted authentication platform for access within the domain, records in the access process are stored on SC1 for historical tracing. In addition, fine-grained access control policies in the domain are implemented by smart contracts on SC1.

$$A \longrightarrow \text{SC1} : \left\{ \text{pk}_{\text{PG}}, \text{Sig}_{\text{sk}_p} \left(\text{Ar} \left(\begin{array}{c} \text{Cert}_G, \text{Owner}, \\ \text{Ac}(\text{where}, \text{Dn}) \end{array} \right) \right), N_1 \right\}. \tag{7}$$

(3) SC1 parses the Ar received, confirms that the current behavior is intradomain access, and then performs identity authentication through MSP

If the Cert_G authentication is valid, record the authentication result and modify the status information of A in the SC1 ledger. If the authentication fails, A rejection message is sent to user A, indicating that user A is an invalid user or does not belong to the local domain, and the access is terminated.

(4) After authentication, SC1 invokes RMC and automatically matches roles for A according to Dn, then generates access token according to role permissions, and sends it to IoT1-B accessed by A. Deadline is the validity period of access token. If the access cannot be completed within the token deadline, the user

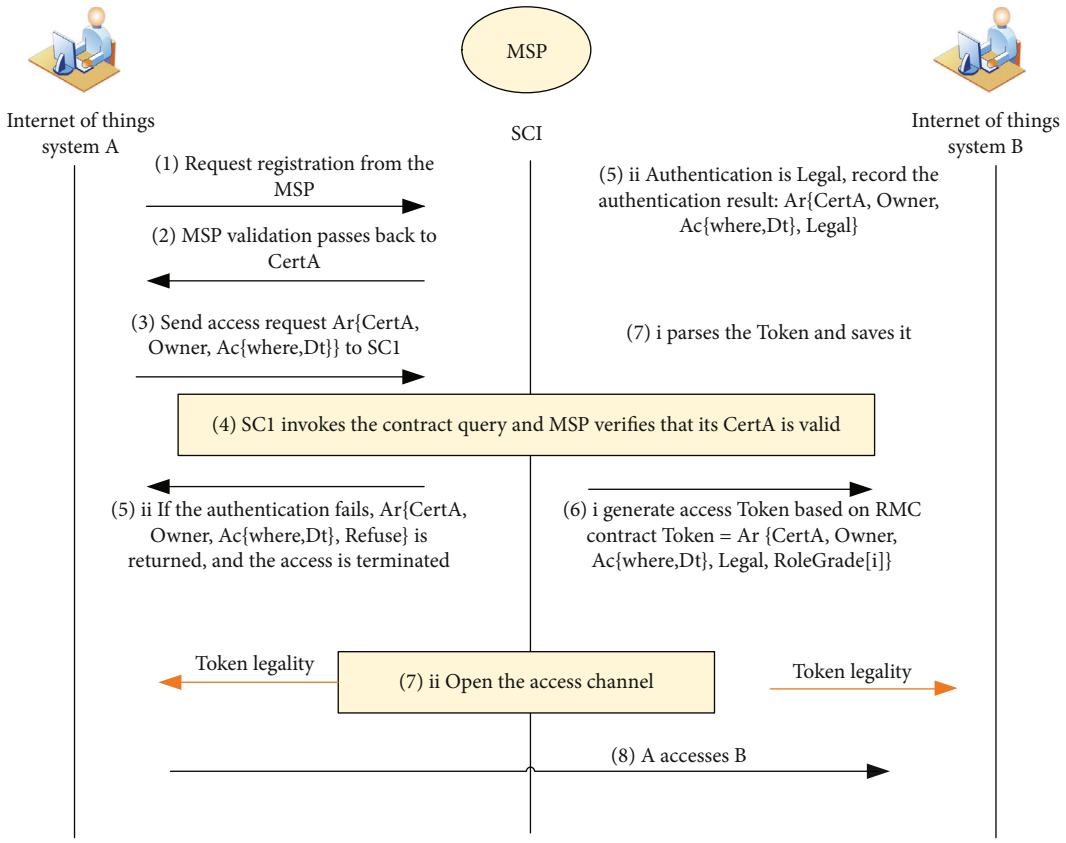

Figure 5: Different IoT system access processes under the same edge IoT domain.

needs to reauthenticate to generate a new token for security reasons

(5) IoT1-B saves the token after resolution and opens the access channel

3.3.3. Interzone Access. Interdomain access refers to the access behavior occurring in different edge IoT domains. Figure 6 shows the access operation between IoT1-C and IoT2-D, in which SC2 responds to the access and MC plays a scheduling role.

(1) IoT1-C sends a request with the status of identity registration to MSP. After passing the authentication, MSP returns the identity certificate Cert_C to MSP. Time indicates the validity time of Cert_C.

$$A \longrightarrow \text{MSP} : \left\{ \text{pk}_{\text{PC}}, \text{Status}, \text{Sig}_{\text{sk}_{\text{IoT1}}} \left(\begin{matrix} \text{sk}_{\text{PC}}, \\ \text{Hash}_c(\text{pk}_{\text{PC}}) \end{matrix} \right) \right\},$$
$$\text{MSP} \longrightarrow A : \text{Cert}_C \left\{ \text{sig}_{\text{MSPpk}}(\text{States} : \text{Ture}, \text{Time}) \right\} \tag{8}$$

(2) IoT-C sends cross-domain authentication request to MC after getting Cert_C and submits the content of access request.

$$C \longrightarrow \text{MC} : \left\{ \text{pk}_{\text{PCx}}, \text{Sig}_{\text{sk}_{\text{PC}}} \left(\text{Ar} \left(\begin{matrix} \text{Cert}_C, \text{Owner}^2, \\ \text{Ac}(\text{where}, \text{Dn}) \end{matrix} \right) \right), N_1 \right\} \tag{9}$$

(3) The main chain MC will parse the Ar received and mark this access state as cross-domain. The access user is then authenticated.

$$\text{MC} \longleftarrow \left\{ \text{Sig}_{\text{sk}_{\text{WC}}} \left(\text{Ar} \left(\text{Ac} \left(\begin{matrix} \text{Cert}, \text{Owner}, \\ \text{from}, \\ \text{where}, \text{Dn} \end{matrix} \right), \text{CrossAr} \right) \right), N_2 \right\} \tag{10}$$

After the Cert_C is authenticated, CrossT is sent to SC2. Cert_C is invalid and authentication fails. Therefore, the user

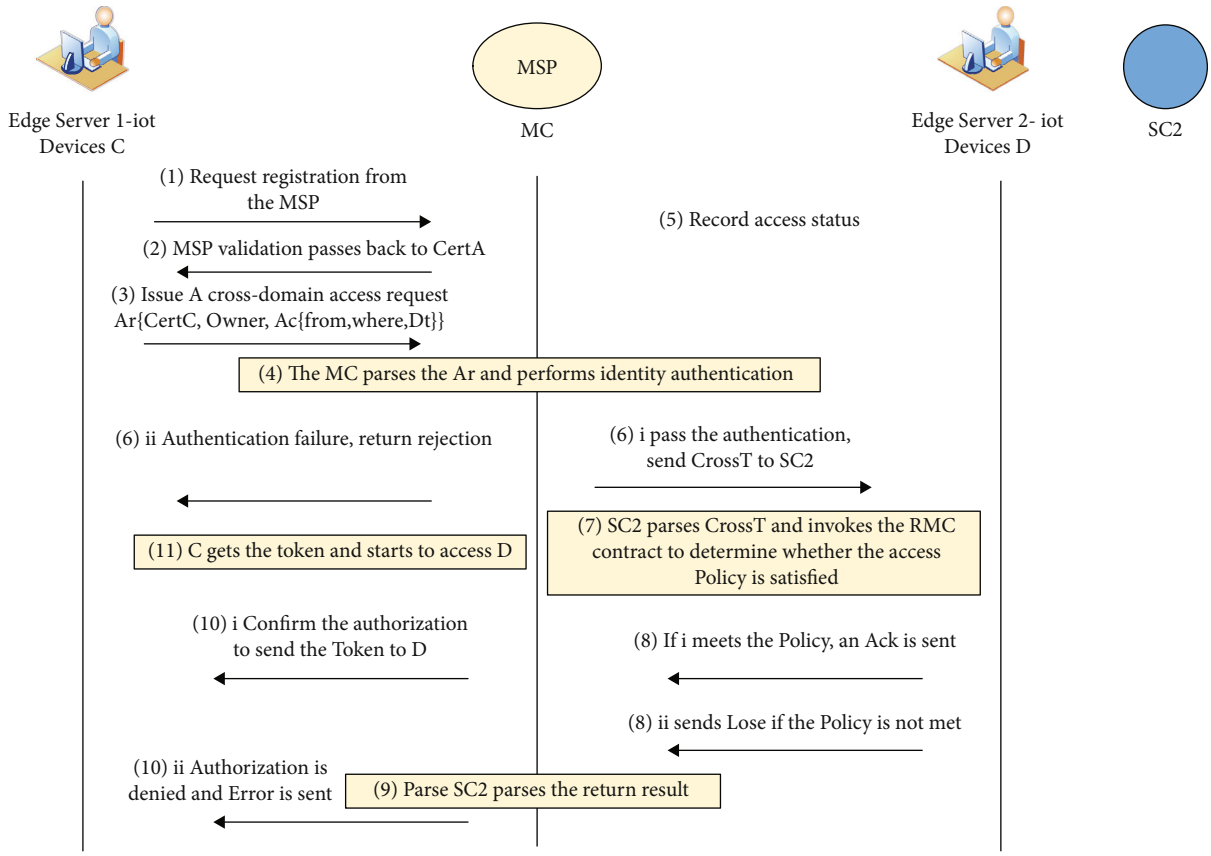

Figure 6: Interzone access control flow.

has no access right. The user returns a denial message, and the cross-domain access process ends.

(4) SC2 analyzes CrossT and invokes RMC to query the role of C to determine whether the current role of C meets the Policy

If the Policy is met, a confirmation message is sent to the MC. If the Policy is not met, an error message is displayed.

(5) MC parses the authorization information returned by SC2

To confirm authorization, CrossD and CacheC contracts are called to obtain and cache data. Open C's access channel to D, and send access tokens to IoT-C. If authorization is rejected in the parsing result, the MC sends an error message to C, indicating that the data set is not open to the user.

If the data belongs to the PDS set, the user who has passed the identity authentication but does not meet the access policy of the PDS set in the domain to be accessed is not authorized to be "activated," and such users have no right to access the user. Therefore, C cannot access the content to be accessed. If you continue to access the domain, you can reselect the collection to be accessed and return. Modify the access content information in Ac, select other access sets, and repackage Ar to initiate a new round of access process.

$$\mathrm{Ac} \longrightarrow \mathrm{Ac}' : \left\{ \mathrm{Ac}\left(\mathrm{where}, \mathrm{Dn}'\right), N_5' \right\}. \tag{11}$$

(6) C starts to visit D after receiving the token

4. Result Analysis and Discussion

In this paper, three aspects of the algorithm's operating efficiency, system overhead, and transaction throughput TPS (transaction per second) of blockchain are tested. By comparison, you can specifically observe the performance. In the experimental process, the third-generation B+ Raspberry Pi (RPi) was used as the hardware carrier to simulate nodes. The Remix-IDE was employed as the development tool for Ethereum smart contracts, and programs were written using Solidity language.

4.1. Algorithm Efficiency. The experiment uses the number of iterations and the ratio of the number of iterations of the improved algorithm to the number of iterations of the traditional algorithm to measure the efficiency of the algorithm. Since the operation efficiency is affected by the binary sequence length of the exponential x of the algorithm, the operation efficiency of the algorithm can be understood by changing its sequence length and observing the changes of

the number of iterations and its ratio. The comparison experiment of algorithm efficiency is shown in Figure 7.

When the binary sequence length of exponential x is more than 100 bits, the curve of iteration times of the traditional algorithm grows rapidly, while the curve of iteration times of the improved algorithm grows slowly. The maximum ratio of iteration times of the improved algorithm and the traditional algorithm is less than 0.50, and the efficiency of the improved algorithm is 3 times higher than that of the traditional algorithm. In addition, with the increase of the sequence length, the ratio curve of iteration times presents a downward trend, and the gap between the two iteration times curves becomes larger and larger, indicating that the larger the sequence length, the better the effect of the improved algorithm and the higher the operation efficiency than the traditional algorithm.

4.2. System Overhead. The system uses credits instead of virtual tokens for circulation and will consume gas during the operation of smart contracts. According to the real-time data of Huobi, 1 gas is assumed to be 0.0078 p. The experiment measured the cost of the system by running the smart contract for one period and observing its consumption points. By comparison, the medical record safe storage and access scheme in literature [23], smart grid data safe storage and sharing scheme in literature [24], and battery health data sharing scheme in literature [25] were selected. The comparison experiment of system overhead is shown in Figure 8.

In literature [23], the integral consumption of the scheme is above 420 p, with the highest integral consumption. This is because the scheme adopts the form of intelligent contract throughout the implementation process, leading to excessive system overhead. Compared with the scheme in literature [24, 25], the integral consumption of the scheme in literature [23] is more than 1/2 less. This is because both schemes only use smart contracts in the data sharing stage, so the integral consumption is greatly reduced. However, the integral consumption of the scheme in this paper is less than 165 p, which is significantly reduced compared with the scheme in literature [23]. Compared with the scheme in literature [24, 25] (over 170 p), there are also some improvements. This is because the scheme in this paper stores encrypted data packets in DD down the chain, while only data storage index is stored on the chain. The system overhead is lower than that of traditional solutions that store data directly on the chain. Therefore, the cost of this scheme is the least compared with other schemes.

4.3. Transaction Throughput. The transaction throughput of blockchain refers to the number of transactions completed per unit time. This experiment sets the block size to a fixed value of 1 MB, tests the number of transactions at different times by setting different block generation times, and then takes its average value and draws a curve. The security storage and access scheme of medical records in literature [23], the security storage and sharing scheme of smart grid data in literature [24], and the battery health data sharing scheme in literature [25] are also selected for comparison. Figure 9 shows the comparison of TPS experimental results.

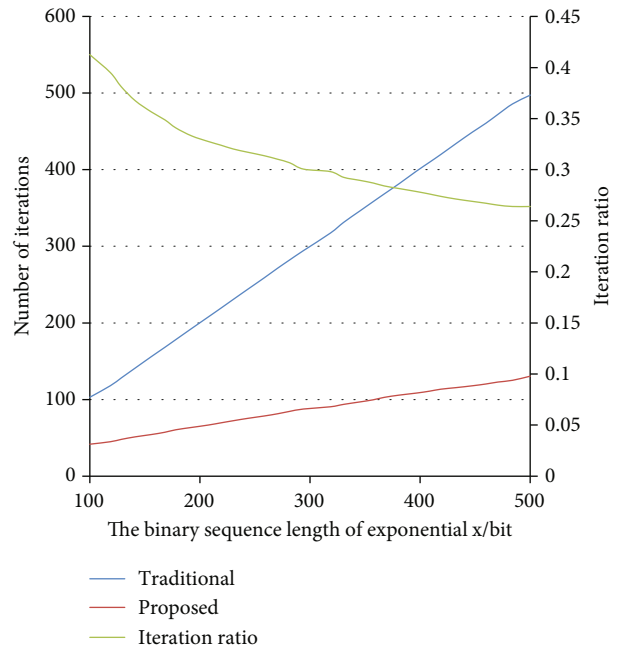

Figure 7: Experimental results of algorithm efficiency comparison.

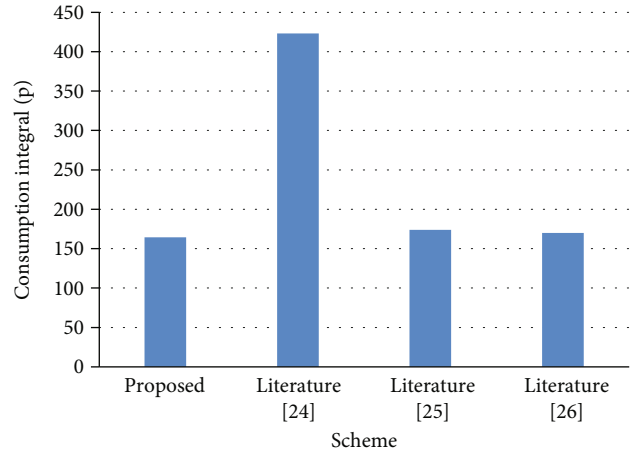

Figure 8: Comparative experimental results of system overhead of different schemes.

The TPS of all schemes decreased with the increase of the block out time, because the increase of the block out time means that the number of transactions processed by the blockchain decreased during the same time. In literature [23], the scheme has the worst performance, with a maximum of just over 50 tx/s. This is due to the excessive use of smart contracts, which leads to excessive system overhead and reduced TPS. The maximum of the scheme in literature [24] is close to 200 tx/s, the maximum of the scheme in literature [25] is just over 250 tx/s, and the maximum of this scheme is close to 300 tx/s. This is because all the schemes in literature [24, 25] adopt the traditional PoW consensus mechanism, while the scheme in this paper has improved

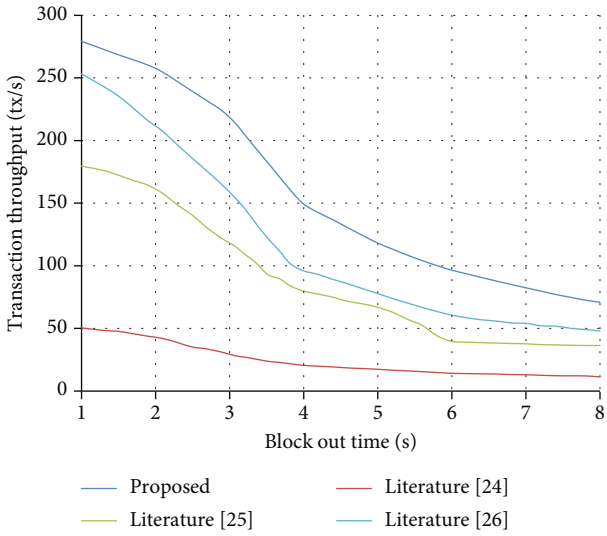

Figure 9: Comparative experimental results of TPS of different schemes.

to some extent on the basis of PoS consensus mechanism. Therefore, the TPS of this scheme is higher than that of other schemes.

5. Conclusion

Blockchain technology has gained popularity for its decentralized nature, data transparency, and traceability, leading to its widespread application in various industries. However, most blockchain applications have predominantly focused on the financial sector. This bias stems from blockchain's origins in virtual currency and its well-suited consensus mechanism for financial use cases. While blockchain holds promise as a security technology, it faces challenges in addressing the security concerns of edge computing. Issues such as low computing efficiency, high energy consumption for consensus, and scalability bottlenecks hinder its ability to fully meet the security demands of edge computing environments. To address these limitations, this paper proposes a master-slave multichain structure that integrates edge computing. This structure forms the basis of a distributed secure trusted authentication model called the interdomain role-based access control (ID-RBAC) model. The ID-RBAC model aims to overcome data isolation challenges by establishing secure connections between domains. It also employs a fine-grained access control method to prevent unauthorized data access and excessive authorization. Experimental results indicate that the proposed scheme effectively mitigates various attacks, significantly improves algorithm efficiency, maintains a system overhead of less than 160 p, and achieves a maximum transaction throughput of nearly 310 tx/s. By combining blockchain technology, edge computing, and the ID-RBAC model, this research offers a novel approach to address security concerns and enhance data accessibility in distributed systems.

Data Availability

The labeled data set used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

There is no conflict of interest between the submitter and the unit and others.

Acknowledgments

This work is supported by the 2022 Chongqing Industry Polytechnic College School-Level Teaching Reform Project "Research on the ideological and political path of higher vocational courses based on information security risk and risk assessment" (No. 2022GZYKCSZ010).

References

[1] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, pp. 45–58, 2019.

[2] Z. Sun, Y. Wang, Z. Cai, T. Liu, X. Tong, and N. Jiang, "A two-stage privacy protection mechanism based on blockchain in mobile crowdsourcing," *International Journal of Intelligent Systems*, vol. 36, no. 5, pp. 2058–2080, 2021.

[3] B. K. Daniel, "Big data and data science: a critical review of issues for educational research," *British Journal of Educational Technology*, vol. 50, no. 1, pp. 101–113, 2019.

[4] S. J. Hsiao and W. T. Sung, "Employing blockchain technology to strengthen security of wireless sensor networks," *IEEE Access*, vol. 9, pp. 72326–72341, 2021.

[5] H. Y. Paik, X. Xu, H. D. Bandara, S. U. Lee, and S. K. Lo, "Analysis of data management in blockchain-based systems: from architecture to governance," *IEEE Access*, vol. 7, pp. 186091–186107, 2019.

[6] V. Ravi, "The prevalence of repeating fast radio bursts," *Nature Astronomy*, vol. 3, no. 10, pp. 928–931, 2019.

[7] Y. Yin, Y. Li, B. Ye, T. Liang, and Y. Li, "A blockchain-based incremental update supported data storage system for intelligent vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4880–4893, 2021.

[8] S. Tanwar, R. Kakkar, R. Gupta et al., "Blockchain-based electric vehicle charging reservation scheme for optimum pricing," *International Journal of Energy Research*, vol. 46, no. 11, 2022.

[9] G. Dhiman, J. Rashid, J. Kim, S. Juneja, W. Viriyasitavat, and K. Gulati, "Privacy for healthcare data using the byzantine consensus method," *IETE Journal of Research*, pp. 1–12, 2022.

[10] L. H. Zhu, B. K. Zheng, M. Shen, F. Gao, H. Y. Li, and K. X. Shi, "Data security and privacy in bitcoin system: a survey," *Journal of Computer Science and Technology*, vol. 35, no. 4, pp. 843–862, 2020.

[11] S. Capraz and A. Ozsoy, "Personal data protection in blockchain with zero-knowledge proof," *Blockchain Technology and Innovations in Business Processes*, vol. 219, pp. 109–124, 2021.

[12] J. Nie, Z. Wang, X. Huang, G. Lu, and C. Feng, "Uniform continuous and segmented nanofibers containing a π-conjugated oligo (p-phenylene ethynylene) core via "living" crystallization-driven

self-assembly: importance of oligo (p-phenylene ethynylene) chain length," *Macromolecules*, vol. 53, no. 15, pp. 6299–6313, 2020.

[13] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020.

[14] Y. Zhang, R. H. Deng, S. Xu, J. Sun, Q. Li, and D. Zheng, "Attribute-based encryption for cloud computing access control: a survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–41, 2020.

[15] C. De Souza, R. Newbury, A. Cosgun, P. Castillo, B. Vidolov, and D. Kulić, "Decentralized multi-agent pursuit using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4552–4559, 2021.

[16] J. G. Reiter, W. T. Hung, I. H. Lee et al., "Lymph node metastases develop through a wider evolutionary bottleneck than distant metastases," *Nature Genetics*, vol. 52, no. 7, pp. 692–700, 2020.

[17] Y. J. Chung, "Multi-alternative retrofit modelling and its application to Korean generation capacity expansion planning," *The Journal of Information Systems*, vol. 29, no. 1, pp. 75–91, 2020.

[18] L. Duan, J. Fan, Y. Wang et al., "Interaction mechanism between nitrogen conversion and the microbial community in the hydrodynamic heterogeneous interaction zone," *Environmental Science and Pollution Research*, vol. 30, no. 3, pp. 5799–5814, 2023.

[19] J. Abou Jaoude and R. G. Saade, "Blockchain applications–usage in different domains," *IEEE Access*, vol. 7, pp. 45360–45381, 2019.

[20] J. Zhao, J. Qi, Z. Huang et al., "Power system dynamic state estimation: motivations, definitions, methodologies, and future work," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 3188–3198, 2019.

[21] Q. Zhang, J. Luan, Y. Tang, X. Ji, and H. Wang, "Interfacial design of dendrite-free zinc anodes for aqueous zinc-ion batteries," *Angewandte Chemie International Edition*, vol. 59, no. 32, pp. 13180–13191, 2020.

[22] C. Abbate, "On the role of knowers and corresponding epistemic role oughts," *Synthese*, vol. 199, no. 3-4, pp. 9497–9522, 2021.

[23] Y. Chen, S. Ding, Z. Xu, H. Zheng, and S. Yang, "Blockchain-based medical records secure storage and medical service framework," *Journal of Medical Systems*, vol. 43, no. 1, pp. 1–9, 2019.

[24] M. Fan and X. Zhang, "Consortium blockchain based data aggregation and regulation mechanism for smart grid," *IEEE Access*, vol. 7, pp. 35929–35940, 2019.

[25] X. Hu, C. Zou, X. Tang, T. Liu, and L. Hu, "Cost-optimal energy management of hybrid electric vehicles using fuel cell/battery health-aware predictive control," *IEEE Transactions on Power Electronics*, vol. 35, no. 1, pp. 382–392, 2020.