International Journal on Semantic Web and Information Systems
Volume 20 • Issue 1

A Trusted Authentication Scheme Using Semantic LSTM and Blockchain in IoT Access Control System

Ge Zhao, The Third Research Institute of the Ministry of Public Security, China
Xiangrong Li, Beijing University of Technology, China*
Hao Li, Xi'an University of Arts and Science, China

ABSTRACT

In edge computing scenarios, due to the wide distribution of devices, complex application environments, and limited computing and storage capabilities, their authentication and access control efficiency is low. To address the above issues, a secure trusted authentication scheme based on semantic Long Short-Term Memory (LSTM) and blockchain is proposed for IoT applications. The attribute-based access control model is optimized, combining blockchain technology with access control models, effectively improving the robustness and credibility of access control systems. Semantic LSTM is used to predict environmental attributes that can further restrict user access and dynamically meet the minimum permission granting requirements. Experiments show that when the number of certificates is 60, the computational overhead of the proposed method is only 203s, which is lower than other state-of-the-art methods. Therefore, the performance of the proposed schema in information security protection in IoT environments shows promise as a scalable authentication solution for IoT applications.

KEYWORDS

Attribute-based access control, Blockchain, Edge computing, Internet of Things, Long Short-Term Memory

INTRODUCTION

Since the beginning of the 21st century, a new round of digital technology revolution and production transformation is sweeping the world (Gupta & Quamara, 2020; Liu et al., 2022). Digital science and technologies such as the internet of things (IoT), blockchain, and artificial intelligence are the core driving forces of this revolution. These disruptive technologies have become the prelude to the new era of the fourth industrial revolution (Al-Qerem et al., 2020; Gupta et al., 2023b; Mamta et al., 2021). The development of science and technology has never been isolated or closed, so the integration and innovation of different technologies will create enormous productivity and promote human civilization to a new and higher level (Chander et al., 2022; Gupta et al., 2023a; Tiwari & Garg, 2022).

DOI: 10.4018/IJSWIS.341233

*Corresponding Author

Blockchain technology is a deep technological transformation of the current, highly centralized internet technology, which has the characteristics of openness and transparency, decentralization, and robustness to tampering (Gaurav et al., 2022; Hu et al., 2022; Raj & Pani, 2022). The decentralized architecture of blockchain, its distributed computing modes, and smart contract collaboration can solve the problems of traditional IoT architectures (Ferrag & Shu, 2021; Khanam et al., 2022; Kiran et al., 2022). First, blockchain systems adopt a decentralized design based on ethernet, with each node directly connected to the others. To avoid a failure of the central node leading to downtime of the entire system, there is no unified central control node (Bamakan et al., 2021; Cao et al., 2019; Kshetri, 2017; L. Wu et al., 2018). Second, blockchain systems use globally-agreed smart contract collaboration to ensure the stability and availability of the entire system and its control processes. Finally, during the operation of IoT systems, a large amount of data will be generated, which requires stable, reliable, and tamper-proof multi-point backup storage, posing a new challenge to traditional storage architectures (Guo et al., 2021; Huang et al., 2022). The blockchain-based InterPlanetary File System adopts a distributed storage architecture, which distributes the entire file system based on the storage capabilities of different nodes, spreading the storage pressure of the system across multiple nodes, and ensuring storage efficiency and security through verification mechanisms, multipoint redundancy, and multipoint read/writes (Alotaibi, 2019; Chaganti et al., 2022; Xie, et al., 2019).

Intelligent IoT technology is commonly deployed in complex and diverse application environments, which pose higher requirements for the security and stability of the computational processes of IoT devices. There are two main security requirements for IoT systems and applications: single-point security and system security. Single-point security refers to the ability of a single IoT device to ensure its own security and resilience to external interference, while system security refers to the ability of the entire IoT system to operate normally after one or more IoT devices have been removed from the system due to failure or external interference (Alizadeh et al., 2020; Hui et al., 2019). Currently, the single-point security of IoT devices mainly relies on the integrity of firmware design, in the design process of which efficient software design processes and standardized methods are used to improve the reliability and stability of the firmware's operation, thus reducing the possibility of IoT devices being interfered with and maliciously attacked (Kavita & Dakshayani, 2022). System security is mainly achieved by the reliability of the IoT system's design. By detecting the abnormal behavior of a single device, marking and eliminating abnormal devices, and enhancing Byzantine fault-tolerance, the entire system becomes more resistant to interference. Even if some IoT devices are under malicious control, the IoT system can still function normally, which greatly improves its resistance to interference and robustness. These security requirements face challenges such as complex external environments and heterogeneous terminals for IoT devices, making it difficult to incorporate these terminal devices into different IoT systems using a common organizational framework (Tibrewal et al., 2022; Q. Zhang et al., 2021). The technical characteristics of blockchain can precisely meet these security requirements for IoT systems. First of all, blockchain adopts a unified computational mode for all terminal devices, thus realizing global collaboration through smart contracts and ensuring single-point security. Second, blockchain systems ensure the consistency of global data and actions through a global consensus mechanism, which can allow the normal operation of the entire system even if some nodes fail. Taking Bitcoin as an example, a proof of work (PoW)-based consensus mechanism is used to ensure that a single computational node can only rely on its own results to compete with other nodes, and it allows for the dynamic exit and joining of any node, while having a tolerance of maliciously node hijacking of 50% (Khattak et al., 2020; Singh et al., 2020).

From the current stage, combining blockchain technology with IoT applications has very important practical significance. On the one hand, the introduction of real-world IoT applications and data into blockchain networks provides the possibility for the current relatively closed blockchain technologies to be adopted. The characteristics of blockchain can solve issues such as multi-party trust, single point of failure elimination, data security and trust, and terminal authentication in IoT environments.

The existing methods do not fully consider the characteristics of incompatible platforms, widely dispersed devices, complex application environments, and limited computational and storage capabilities prevalent in IoT environments, which result in low authentication and access efficiency.

To overcome these shortcomings, a blockchain technology and attribute-based access control (ABAC)-based information security protection scheme is proposed in the IoT environment. The innovation points of the proposed method are summarized in the following:

1. Through the optimization of the ABAC model and combining blockchain technology with access control models, a framework that mainly includes policy enforcement points (PEPs), policy administration points (PAPs), attribute authority (AA), and policy decision points (PDPs) was proposed, which improved the robustness and credibility of IoT-based access control systems effectively.
2. A signed token was used as a transferable asset that allows accessing various devices and applications in complex IoT environments while ensuring the security of interactions away from the blockchain network. A semantic long short-term memory (LSTM) network was used to predict environmental attributes in ABAC access control policies, enabling users to obtain minimum access rights. By training on public data sets, the model's prediction accuracy is improved, allowing users to obtain the minimum access permissions.

RELATED WORK

The IoT seamlessly interconnects heterogeneous devices and objects. Liu and Zhang (2020) designed and implemented a blockchain-based data integrity detection method for IoT systems based on the requirements of industrial IoT and the characteristics of blockchain systems. X. Wu et al. (2019) focused on blockchain applications in smart cities and designed a privacy-based IoT technology solution based on spatiotemporal smart contract services. A blockchain-based distributed access control system was designed by Rahman et al. (2019) from the perspective of information security in IoT environments, providing a new approach to solving the security issues of distributed IoT systems. Novo (2018) utilized the distributed storage of blockchain systems to design a computing framework that can store and analyze big data in smart city systems using IoT technologies, expanding the usage boundaries of the IoT system and enhancing the usability of smart city solutions. Yu et al. (2018) summarized and evaluated the performance evaluation schemes for blockchain consensus algorithms with regard to four aspects: algorithm throughput, consensus incentive, degree of decentralization, and consistent security. Fuzzy set theory was used in Bamakan et al. (2020) to formulate behavioral pattern analysis data for blockchain systems at the microarchitecture level, and machine learning methods were used to quantitatively analyze these data, laying a good foundation for the design of dedicated hardware specific to blockchain systems. Zhu et al. (2020) conducted in-depth research on the throughput efficiency evaluation of CPU and GPU, opening up new ideas for device computational power evaluation. A scalable parallel fragmentation protocol for blockchain systems was proposed in Lee et al. (2010), which utilized cryptographic mechanisms to ensure security. M. Zhang et al. (2020) expanded the original backbone protocol of the Bitcoin system, and effective piecemeal capacity expansion was achieved, which improved the scalability of the Bitcoin system. A blockchain asynchronous consensus architecture was proposed in Wang et al. (2021), which to some extent solved the impossible triangle problem that has long plagued blockchain-based applications.

Blockchain systems have achieved certain scalability using fragmentation schemes without sacrificing system security and decentralization capability. Ren et al. (2021) proposed a fragmented blockchain architecture like OmniLedger, which adopted a new consensus scheme to enable blockchain systems to scale. The linearization and corresponding time of Byzantine fault-tolerant algorithms have been discussed in depth in Kokoris-Kogias et al. (2018), and new ideas were put forward for designing fault-tolerant IoT systems. A CP-ABE scheme was proposed in Zhang and Ren (2021) to protect the privacy of ciphertext policies, while in Tsaur et al. (2022) a blockchain-based protection mechanism was proposed, which can reduce the demand for storage space and improve system security. In Xiao (2021) an information security protection method was proposed based on the fusion of big data and public BT.

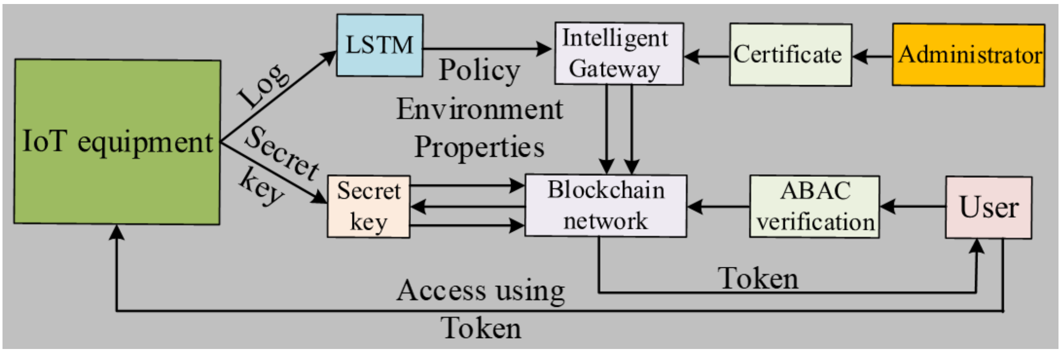
Based on the above analysis, the combination of blockchain technology and smart contract technology can solve the limitations of traditional access control solutions and achieve trusted distributed access control. However, in many existing solutions, frequent user access and massive data processing have brought new challenges to the access control of the IoT, namely low throughput and high time consumption. When enforcing access control in a large-scale environment with a large number of users and highly frequent access requests, these two issues may impose a huge burden on administrators and users.

To address these challenges, a private data access control model based on blockchain technology is proposed. This model can track the request records, response records, and access records of private data through distributed networks and consensus authentication mechanisms. LSTM is used to effectively predict environmental attributes in ABAC access control policies, allowing users to obtain the minimum access permissions. At the same time, the system realizes dynamic management of private data access control policies solves the access control problem of private data on the IoT and supports efficient user access.

METHOD

The process of IoT device users obtaining and applying access permissions is shown in Figure 1. (a) The IoT device is stored in the token configuration file of the smart contract through a key (key-exchange ECDH algorithm) and ID. (b) After the administrator verifies the certificate, the policy configuration contract is called to publish the ABAC policy in the blockchain network. (c) The user requests access rights to a device or application, and the latter verifies whether the user attributes meet existing ABAC policies through a policy verification contract. (d) After ABAC authentication is passed, a token is requested from the corresponding IoT device or application. The device contract generates a token based on the request information and signs it. The blockchain records the authorization and transfers the token as an asset to the corresponding user. (e) Users use tokens to access devices or applications. (f) LSTM is used to extract semantic features from access logs to learn environmental attributes and predict subsequent user visits. The IoT system updates ABAC policies by modifying ABAC access policy environment properties and calling policy configuration contracts.

Figure 1. Overall process of IoT device users obtaining and applying access permissions

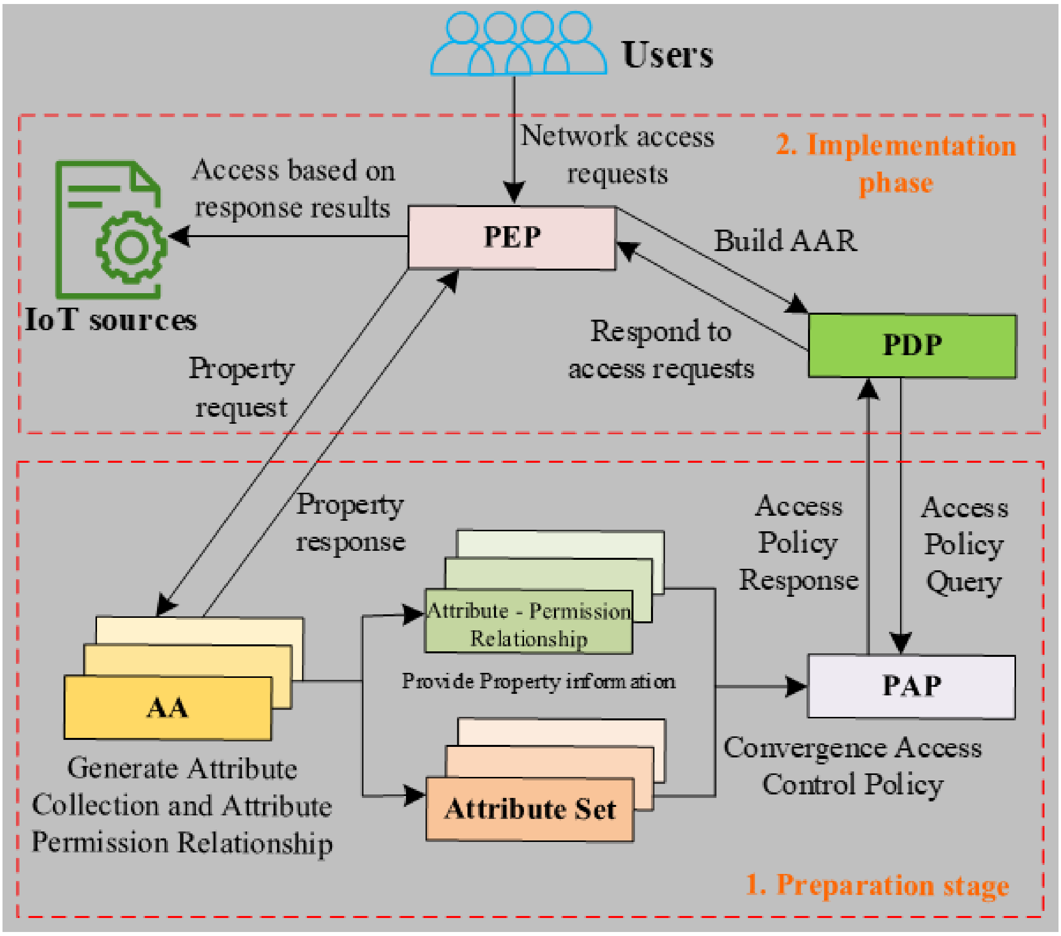


ABAC Method Based on Blockchain Technology

The proposed blockchain-based ABAC framework is shown in Figure 2. The framework mainly includes four core parts: a PEP, a PAP, an AA, and a PDP. The entire access control workflow can be divided into the preparation phase and the execution phase, as shown in Figure 2.

1. In the preparation stage, the AA is responsible for generating and storing attribute sets and permission relationships in blockchain transactions. The policy publisher will then publish access control policies in the blockchain. The PAP will describe, collect, and integrate access control policies in blockchain transactions in combination with attribute information, and the PDP will evaluate the access request.
2. During the execution phase, when the PEP receives a request from a user to perform an operation on a resource, the PEP first analyzes the request. Then, based on the attribute information obtained from the AA, the PEP generates an attribute access request (abbreviated as AAR in the figure), and sends it to the PDP. The PDP queries the PAP and requests the set of access control policies related to IoT resources, performs access control evaluations, and finally sends the judgment result response back to the PEP. The PEP then is allowed to perform authorized-access operations on IoT resources based on the response results.

Figure 2. ABAC framework based on blockchain technology


Users
Network access requests
2. Implementation phase
Access based on response results
PEP
Build AAR
IoT sources
Respond to access requests
PDP
Property request
Property response
Access Policy Response
Access Policy Query
Attribute - Permission Relationship
Provide Property information
PAP
AA
Attribute Set
Convergence Access Control Policy
Generate Attribute Collection and Attribute Permission Relationship
1. Preparation stage

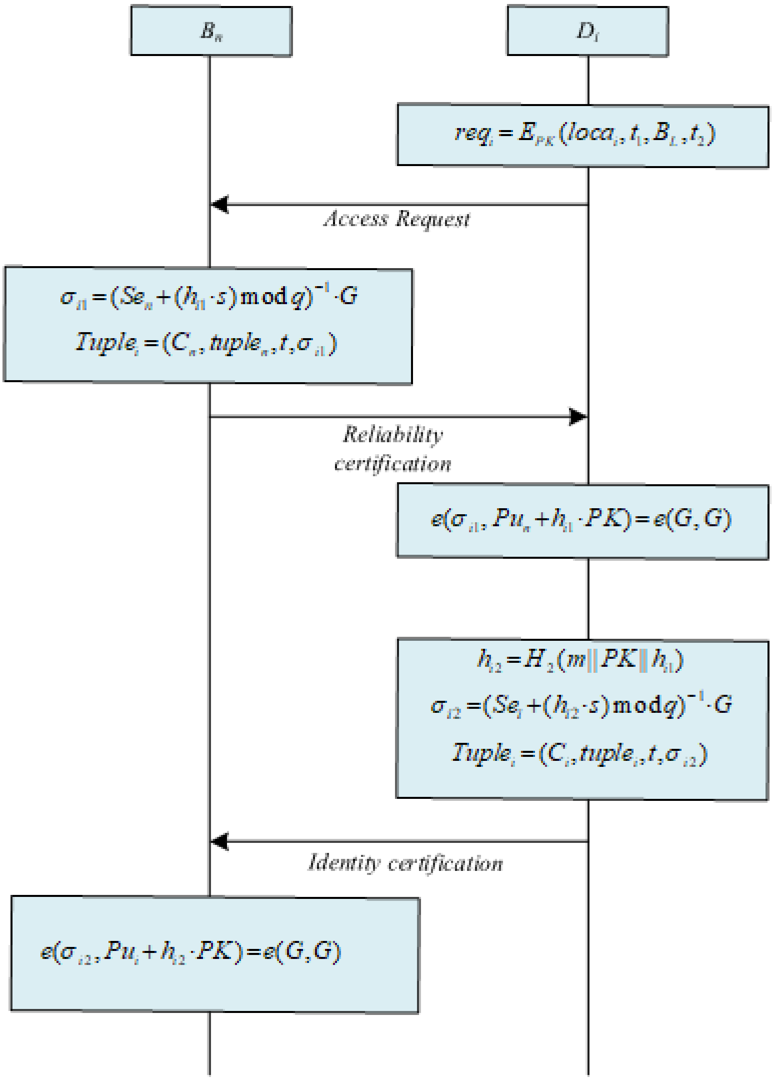
UAS Identity Authentication Scheme

During the authentication process, there may be malicious nodes in the blockchain. In this section, a pre-signature mechanism is designed to achieve reliable verification of blockchain edge nodes, and then the device is authenticated through the primary signature to achieve highly secure two-way identity authentication. The authentication process is shown in Figure 3, with the specific steps described below.

Step 1: When IoT device D_i is within the coverage range of the edge node B_n, the former first sends an access request message req_i to the latter:

$$req_i = E_{PK}(loca_i, t_1, B_L, t_2) \tag{1}$$

Figure 3. Identity authentication process


B_n
D_i
$req_i = E_{PK}(loca_i, t_1, B_L, t_2)$
Access Request
$\sigma_{i1} = (Se_n + (h_{i1} \cdot s) \bmod q)^{-1} \cdot G$
$Tuple_i = (C_n, tuple_n, t, \sigma_{i1})$
Reliability certification
$e(\sigma_{i1}, Pu_n + h_{i1} \cdot PK) = e(G, G)$
$h_{i2} = H_2(m \| PK \| h_{i1})$
$\sigma_{i2} = (Se_i + (h_{i2} \cdot s) \bmod q)^{-1} \cdot G$
$Tuple_i = (C_i, tuple_i, t, \sigma_{i2})$
Identity certification
$e(\sigma_{i2}, Pu_i + h_{i2} \cdot PK) = e(G, G)$

where $loca_i$ represents the geographical location of the device D_i, t_1 represents the sending time of the access request message req_i, and B_L represents the last accessed node on the device.

Step 2: After node B_n receives the request, it first checks whether device D_i has been previously connected here. If there is an access record at time t_2, this indicates that the reliability of the access request message req_i is high. At this time, B_n generates a reliability proof message $Tuple_n$ and sends it to D_i:

$$\sigma_{i1} = (Se_n + (h_{i1} \cdot s) \bmod q)^{-1} \cdot G \tag{2}$$

$$Tuple_i = (C_n, tuple_n, t, \sigma_{i1}) \tag{3}$$

where σ_{i1} represents the pre-signature generated by B_n, C_n represents the certificate issued by the Certificate Authority for node B_n, and $tuple_n$ is the query path of C_n in the Merkle Patricia Tree (MPT).

Step 3: The device performs a reliability verification on the edge node. D_i first retrieves the public key in the node certificate C_n and calculates whether the key value formed by the $tuple_n$ path corresponds to the node's public key hash value. Then, the MPT root value is calculated based on the path node's hash value. Finally, the node signature is verified using the characteristics of the bilinear map:

$$e(\sigma_{i1}, Pu_n + h_{i1} \cdot PK) = e(G, G) \tag{4}$$

If Equation (4) holds, the identity of Edge Node B_n is reliable.

Step 4: After device D_i has verified that the identity of node B_n is reliable, an authentication message $Tuple_i$ is generated and sent to B_n:

$$h_{i2} = H_2(m \,||\, PK \,||\, h_{i1}) \tag{5}$$

$$\sigma_{i2} = (Se_i + (h_{i2} \cdot s) \bmod q)^{-1} \cdot G \tag{6}$$

$$Tuple_i = (C_i, tuple_i, t, \sigma_{i2}) \tag{7}$$

where h_{i2} represents the device message summary. m is the device authentication message clear text, and σ_{i2} represents the primary signature generated by the device.

Step 5: The edge node authenticates the device. If the tuple is consistent, it will indicate that the device certificate is valid. Finally, the node verifies the device signature:

$$e(\sigma_{i2}, Pu_i + h_{i2} \cdot PK) = e(G, G) \tag{8}$$

If Equation (8) holds, the device passes the identity authentication and is allowed to access the blockchain edge network.

Smart Contract

Smart contracts are at the core of the ABAC access control implementation of Hyperledger Fabric. In the proposed schema, smart contracts are divided into three types: policy configuration, token configuration, and policy verification contracts. During the contract invocation process, the standard ABAC policy operation parameter structure is defined, which includes the enumeration type operate field and the actual ABAC policy structure field.

1. Policy configuration contract

The policy configuration contract mainly provides methods for publishing, modifying, querying, and removing ABAC policies based on blockchain networks. Some of the core methods are as follows: Because Couch DB is used as a state database, and Couch DB supports complex queries similar to MongoDB, Hyperledger Fabric encapsulates the ABAC policy query method, supporting batch ABAC policy queries based on {AS, AO} collections.

2. Token configuration contract

As a trusted credential for a device on the chain, a permission token encapsulates the permissions and permission holders for different devices. Recording device permissions on the chain ensures the robustness to tampering and verifiability of the permission records. Users can customize and control corresponding devices between nodes through the verification of permission tokens and the execution of corresponding smart contracts. Cloud storage services encrypt, transmit, and store offline data through permission tokens issued by the storage service of the on-chain verification node. The functions of this blockchain network include permission token generation, transfer, and destruction. When a new IoT device is added to the blockchain network, a permission token generation node requires a permission token to generate a transaction. After the transaction is published on the chain, all nodes in the blockchain can verify the permissions of the corresponding device and the permission holder (the home resident or public gateway node) through the transaction, and nodes with permissions for the device can control it.

Household node A is taken as an example to chain up IoT device M. The transaction input of this token is set to null, and the output is the public key address B of node A. It also contains field information such as device information, permissions, and a list of permission holders. Finally, a digital signature is issued using the private key of node A. After the transaction has been packaged into a block, the nodes in the network can verify the signature by outputting the address A to determine whether A is allowed to control the device. Permission token transfer: When it is necessary to transfer the permissions of a device to other nodes, such as a hotel temporarily handing over control of the internal equipment of a room to the tenant, the node performs a short-term or persistent transfer of permissions through the generation of a token transaction. In token transactions, the input points to the last token transaction that contains the address of the IoT device M, while the output is the public key address B of the new control node. The permission field can be used to transfer the assigned read/write permissions, while the expiration time field can be used to limit the duration of permission usage, generally using the block height as the time benchmark. If it is necessary to replace the gateway node of the IoT device, the permission holder can also be permanently reassigned. Finally, the transaction is signed through the private key of node A and published on the chain.

3. Policy validation contract

The policy verification contract is used to verify whether a user's access request meets existing ABAC policies, including three elements: attribute of subject (AS), attribute of object (AO) and attribute of environment (AE). AS represents the attributes of a user, including three types: user account, user role, and user organizational structure. AO represents the object attribute, which is identified as the unique ID assigned to the industrial IoT. AE represents the environmental attributes required for access control.

The check access method is the core of implementing ABAC access control. First, the relevant ABAC access control policies are queried based on the incoming AS and AO. If there is no corresponding access control policy (ACP), an exception is returned; if there is an access control policy, it is determined whether the user's environment meets the requirements of the ABAC policy AE. If not, an exception is again returned. If the ABAC policy requirements are met, the token configuration contract is called to generate the required token and return it to the user.

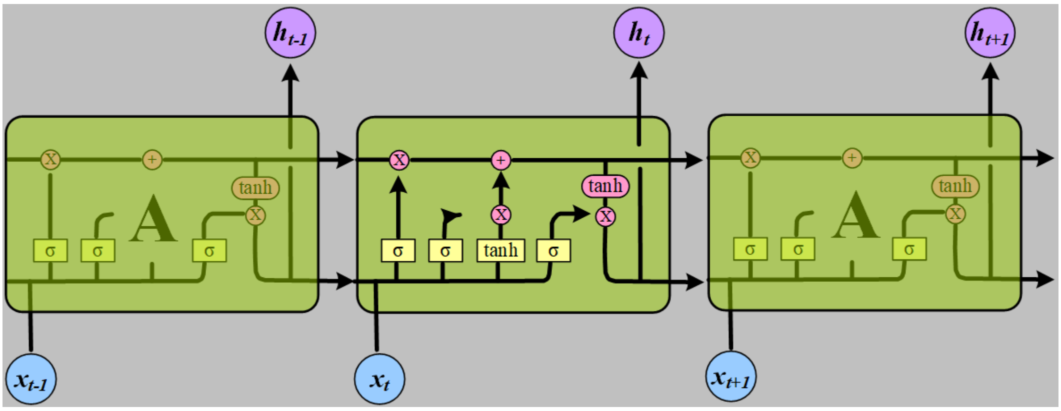
4. Environmental attribute learning

LSTM is used to learn environmental attributes from logs and to predict subsequent access environment attributes. In LSTM, C_t and C_{t-1} represent the memory units of the current time and the previous time, respectively, h_t and h_{t-1} represent the hidden units of the current time and the previous time, respectively, i_t represent the input gate of the current time, f_t is the forget gate, o_t is the output gate, X_{nmt} is the value of the n-th feature quantity in the m-th time segment of the t-th day. The LSTM network state update process can be represented as (Rossini, et al., 2023):

$$\begin{aligned}
&C_t = f_t * C_{t-1} + i_t * C_t', \\
&C_t' = \tanh(W_c[h_{t-1}, X_{nm(t-1)}] + b_c), \\
&f_t = \sigma(W_f[h_{t-1}, X_{nmt}] + b_f), \\
&i_t = \sigma(W_i[h_{t-1}, X_{nmt}] + b_i), \\
&o_t = \sigma(W_o[h_{t-1}, X_{nmt}] + b_o), \\
&h_t = o_t * \tanh(C_t), \\
&\sigma(\cdot) = \frac{1}{1+e^{-(\cdot)}}.
\end{aligned} \tag{9}$$

where, W_c, W_f, W_i and W_o represent the weights of memory units, forget gates, input gates, and output gates, b_c, b_f, b_i and b_o represent their respective bias coefficients. The structure of LSTM is shown in Figure 4.

Figure 4. The structure of LSTM



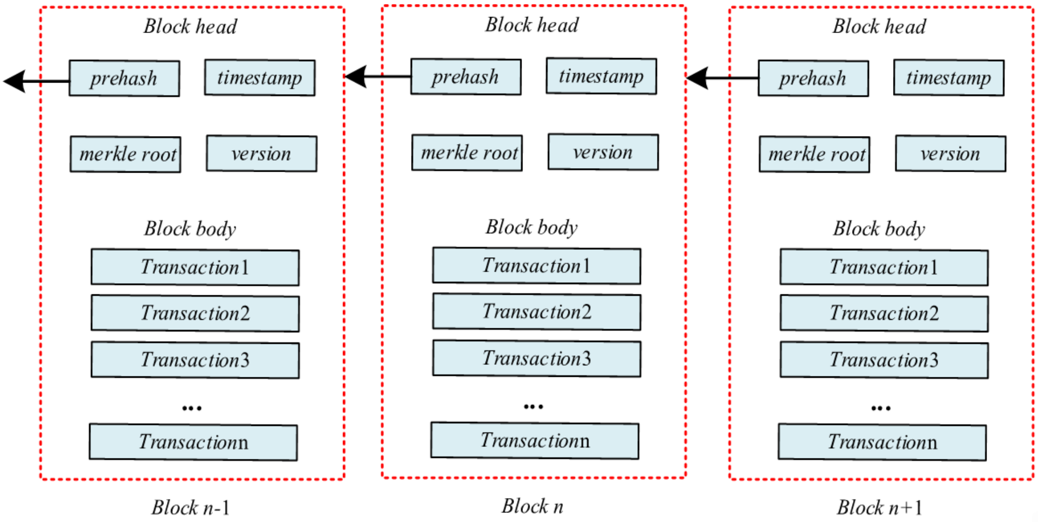
Blockchain Network

The structure of the blockchain is shown in Figure 5. Each block is connected through the hash value of the previous block. The block header mainly contains the previous block's hash value, a timestamp, the Merkle tree root, the version number, and other information. The previous block hash value connects the blocks together. The existence of timestamps makes the blockchain data traceable, and users cannot ignore information that has been linked. The version number information is used to represent the rule version referenced by the transaction in the block. The Merkle tree is a hash binary tree whose roots provide a guarantee that blockchains can quickly verify transactions. The block body stores specific information about a series of transactions (abbreviated as TX).

The chain structure of blockchain ensures extremely high data security. An attacker attempting to modify a transaction of a block on the chain for some reason, such as increasing their account balance or erasing their operational traces, will be forced to modify the Merkle tree root of that block. Since the hash of the entire block depends on the Merkle tree root, the attacker must recalculate the hash of the block. Due to the overall structure of blockchain, each block after the tampered block will contain a hash value pointing to the previous block. Therefore, attackers have to recalculate the hash value of each block after the tampered block. During this period, the blockchain will continue to extend forward, and the number of calculations required for attackers to complete attacks will also increase. In theory, attackers need to control more than 51% of the computing power of the entire network to achieve this. It is obvious that the cost of conducting such an attack is difficult for attackers to bear.

In the proposed strategy, the hash algorithm used in the blockchain network is the SHA-256 algorithm, which is a relatively complex hash algorithm with a 256-bit digest. Blockchain networks based on SHA-256 can be applied to some secure applications and protocols, including Bitcoin, SSH, and IPC. The algorithm includes steps such as message preprocessing, padding, and digest calculation, resulting in the generation of a 256-bit hash string for each text message. In the process of hash mapping, it can be ensured that each hash corresponds to a unique input, and the calculation ensures the impossibility of finding an input with a given hash value, thus ensuring high security in the process.

Figure 5. Blockchain structure



Improved Raft Algorithm

To ensure the legitimacy of data added by nodes to the blockchain, the use of consensus algorithms can ensure that all nodes in the blockchain network reach consensus on the new blocks to be added to the chain. Even if there is a certain number of malicious or faulty nodes on the chain, consensus algorithms can ensure that other nodes can reach consensus on the legitimacy of new transactions. For IoT environments based on edge computing, an improved Raft consensus algorithm is proposed in this paper. The overall flow chart is shown in Figure 6. First, all consensus nodes are clustered, and a sub-leader is elected in each sub-cluster using the Raft election method. The other nodes in the sub-cluster are the follower nodes of the sub-cluster. Then, a main cluster is formed by all the sub-leaders, and the Raft algorithm is applied again. During the election process of each cluster, due to the decrease in the number of nodes, in order to prevent multiple candidate nodes from competing and causing deadlock, a vote conversion algorithm is introduced. When multiple candidate nodes do not receive more than half of the votes, the candidate with the largest number of votes will become the leader node. Through the use of multiple clusters, the number of leader nodes in the cluster is increased, so there is no longer reliance on the consensus of a single leader node, and multiple leader nodes share all the consensus tasks, thereby improving the efficiency of the algorithm.

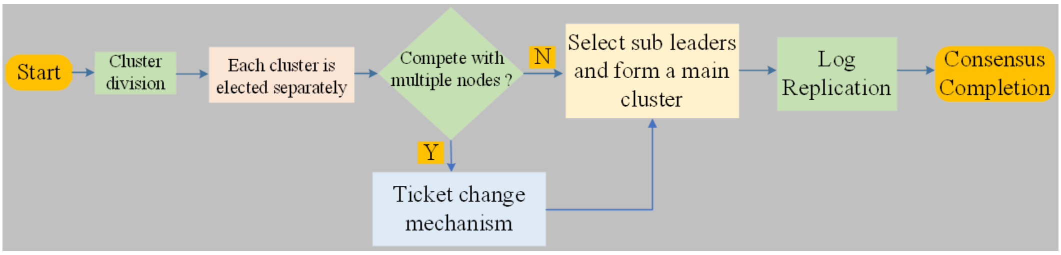
Safety Analysis

In this access control system, the proposed solution can effectively alleviate vulnerabilities in the IoT, such as man in the middle attacks, replay attacks, and unauthorized access, thereby effectively ensuring the privacy and security of private data on the IoT. Because the blocks storing data can be obtained through any node, this can lead to data leakage in transparent blockchain networks. The proposed solution in this article optimizes the ABAC model by combining blockchain technology with access control models, and based on the Fabric platform for operation control, effectively improving the robustness and credibility of the access control system. In order to ensure the legitimacy of data added by nodes to the blockchain, an improved Raft consensus algorithm is used to ensure that all nodes in the blockchain network reach consensus on the new blocks to be added to the chain. In the process of identity authentication, in order to deal with potential malicious nodes in the blockchain, a pre signature mechanism is designed to achieve reliable verification of blockchain edge nodes, and then the device is authenticated through the main signature, achieving highly secure bidirectional identity authentication. In addition, private data in smart contracts can only be accessed through authentication by access control policies, and the semantic LSTM-based access attribute prediction method can enable users to obtain the lowest access permissions. In summary, the proposed scheme can effectively ensure the security of private data.

Case Analysis

Let us assume the existence of an industrial IoT workshop A where the proposed access control strategy is applied to allow access to a device. Only employees with subject attributes containing <UserRole

Figure 6. Algorithm flow of improved raft


Start
Cluster division
Each cluster is elected separately
Compete with multiple nodes ?
N
Y
Select sub leaders and form a main cluster
Log Replication
Consensus Completion
Ticket change mechanism

= “Engineer”, UserOrg = “Devices Management Group”> should have secure access, and employees are required only to access the device to ensure the security of the system. The process of applying for access control system from IoT management system is shown in Figure 7.

Assuming that the subnet segment of workshop A is between 192.168.3.198 and 192.168.4.255, as long as the host and guest attributes meet the requirements, the environmental attribute of the IP address is also required to further ensure access security. Compared to RBAC, ABAC relies on more attribute restrictions to realize finer access control granularity and ensure minimum permission criteria and access security.

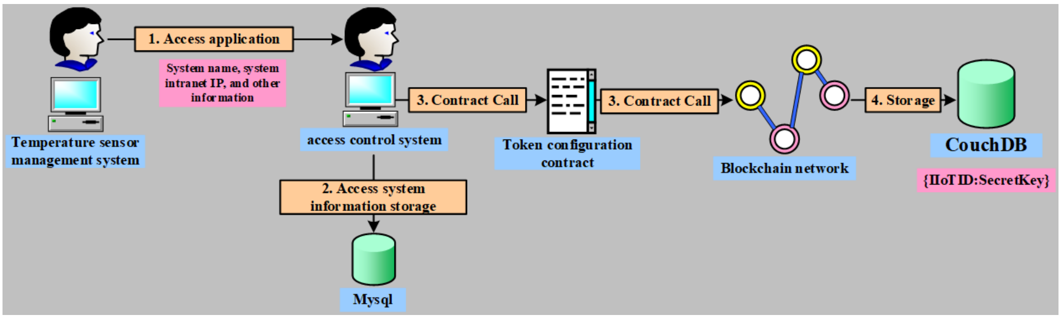
First, the administrator of the device submits an access request, which is approved by the access control system administrator. After approval, a unique ID of industrial internet-of-things (IIoTID) will be generated based on the intranet IP of the sensor management system, and a unique key will be generated. Here, it is assumed that the generated IIoTID and key are A0000001_ 170.30.30.10: 7a8183bc769768293327a58fc9910c8d8ec0129d52f ae331300123a1956f587f. Afterward, the access control system calls the token configuration contract method of the blockchain network and stores the properties of the device in the state database.

After introducing the proposed access control strategy into the device, it is necessary to add corresponding access control policies to meet the access control requirements. Based on these requirements, the access control system administrator can filter the subject and object attributes and develop initial access control strategies. If and only if the access subject property contains <UserRole = “Engineer”, UserOrg = “DevicesManagement Group”>, the IIoTID of the access object is A0000001_ 170.30.30.10, and the application time is before 13:58:30 on September 12, 2023, then access permissions will only be granted if the access IP is between 192.168.3.198 and 192.168.4.255. After the access control system administrator submits the policy addition, the blockchain network calls the policy configuration contract’s add policy method, and it stores the newly-added policy ID and policy ontology in the blockchain network’s state database. The new policy ID is calculated using SHA256 based on the subject and object attributes of the ABAC policy, ensuring its global uniqueness.

Compared to other access control methods, the introduction of the IP address as an environmental attribute restriction into the ABAC method allows the effective verification and filtering of the access control system without the need for additional IP address filtering modules, thus reducing the complexity of the access control system.

In the environment attribute prediction module, based on the past access logs, an LSTM is used to predict the possible environment attributes (such as access traffic and access frequency) in the next time period, extract the subject and object attributes from the logs, and then call the update strategy method of the blockchain network policy configuration contract to update the corresponding ABAC strategy. In this example, it is assumed that the predicted access traffic in the next time period is limited to 28311552 bytes, and the frequency is limited to six requests per unit time (minute).

Figure 7. The process of applying for access control system from IoT management system


1. Access application
System name, system intranet IP, and other information
Temperature sensor management system
access control system
3. Contract Call
Token configuration contract
3. Contract Call
Blockchain network
4. Storage
CouchDB
{IIoTID:SecretKey}
2. Access system information storage
Mysql

Compared to other access control methods, the introduction of environmental attribute restrictions such as traffic, frequency, or time restrictions in the ABAC method allows the effective monitoring of the access control system without the need for deployment of flow-limiting modules or additional rules in frequency modules, thus greatly simplifying the management of the system.

Let us assume now that an existing employee named Tom needs to access the device, and his subject attributes include <UserRole = "Engineer", UserOrg = "Devices Management Group">. When Tom first enters the system, he will trigger the access control system for ABAC verification. At this point, the blockchain network will call the validation policy method for policy validation contracts. First, a search for an ABAC policy based on Tom and the subject-object attributes of the device is conducted. If a policy cannot be found, this indicates that there is no permission to access it; if a policy is found, a verification check is conducted on whether Tom's environmental attributes meet the static environmental attribute requirements of the ABAC policy. At this point, assuming that Tom's device IP address is 192.168.4.52, which meets the ABAC access control rules, the blockchain network calls the token configuration contract to generate a token for Tom based on these environment attributes.

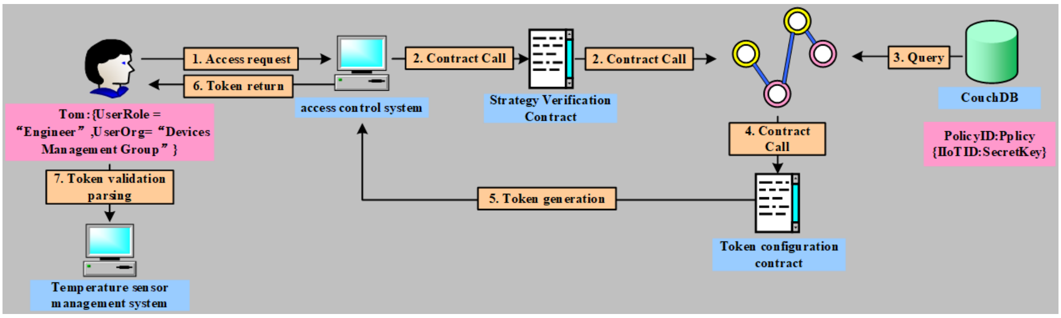
Meanwhile, suppose an existing employee named George with the main attributes <UserRole ="Worker", UserOrg = "Devices Management Group"> attempts to access the device. When George triggers an ABAC verification by the access control system, the blockchain network will call the policy verification method of the corresponding contract, but no ABAC policy with <UserRole = "Worker", UserOrg = "Devices Management Group"> and <IIoTID="A0000001_170.30.30.10"> will be found, so an exception will be returned by the access control system. Thus, the device will deny George's access due to the lack of a valid token. Similar to George, consider an employee named John, who attempts to access the device outside of workshop A with primary attributes <UserRole = "Engineer", UserOrg = "Devices Management Group">, but an IP address of 192.168.5.80. When John triggers ABAC verification by the access control system, the blockchain network calls the policy verification method of the policy verification contract and detects that John's IP address does not meet the policy requirements. Similar to John, the blockchain network will return an exception to the access control system.

After obtaining the token, Tom can use it to access the device. The authentication module of the device only needs to verify the token to ensure the security of the transactions. The process of obtaining access to the IoT management system is shown in Figure 8.

First, Tom's own key is used to apply HMACSHA256 on the header and payload of the token to verify the signature, ensuring that the token content has not been tampered with. After successful verification, BASE64URLDECODEER is applied on the payload content to access to the host, guest, and environmental attributes. The environmental attribute restrictions are submitted to the traffic and frequency restriction modules for subsequent access monitoring.

If a blockchain network is not used, as long as malicious users obtain permission to store authorization records in the database, they can delete their own authorization records conceal their

Figure 8. The process of obtaining access to the IoT management system


1. Access request
6. Token return
2. Contract Call
2. Contract Call
3. Query
access control system
Strategy Verification Contract
CouchDB
Tom :{UserRole = "Engineer" ,UserOrg= "Devices Management Group" }
4. Contract Call
PolicyID:Pplicy {IIoTID:SecretKey}
7. Token validation parsing
5. Token generation
Token configuration contract
Temperature sensor management system

traces. Compared to traditional distributed access control, combining blockchain networks with ABAC allows nodes to store complete data and make authorization records tamperproof.

RESULTS AND DISCUSSION

The configuration of the simulation test environment for this experiment is shown in Table 1.

The Hyperledger Fabric blockchain network nodes deployed in this system test, the nodes to which the smart contracts belonged, and the LSTM environment attribute prediction nodes are shown in Table 2.

Algorithm Training

Consensus Latency for Blocks of Different Sizes

Consensus latency refers to the time it takes for the entire process, i.e., from the moment the primary node generates a new block until the new block is finally recognized as a legitimate block via consensus and added to the blockchain. Therefore, consensus latency is mainly composed of the message generation, transmission, and processing times. The consensus latency is measured for blocks with

Table 1. Hardware and software environment

Experimental environment	Specific information
CPU	AMD Ryzen 5 2600 Six-Core
Operating system	Ubuntu 20.04.2 LTS
GPU	RTX 3060
Memory	16Gb
Hard Disk	500Gb SSD
Hyperledger Fabric	v2.4.1
Docker	v20.10.12
Docker Compose	v1.22.0
Blockchain Programming Language	Golang

Table 2. Docker node type and number

Node type	Quantity
Peer Node	4
Predict Node	1
Policy Verification Contract Node	4
Token Configuration Contract Node	4
Policy Configuration Contract Node	4
CA Node	2
Order Node	1
Couch DB Node	4
Tools Node	1
Zookeeper	1
Kafka Node	1

different numbers of packaged transactions, i.e., for blocks of different sizes. Blockchain nodes will verify all transactions packaged in a block. The results for each block type were measured 10 times and then averaged, which are shown in Figure 9 below.

When the block size is less than 1,800 transactions, the consensus delay is at the millisecond order of magnitude, which would be acceptable for most IoT application scenarios. Moreover, when the number of transactions contained in a block is below 1,000, the consensus latency remains relatively steady. When the number of transactions increases above 1,000, the consensus latency will suddenly increase and grow rapidly as the number of blocks increases. The reason for this is that when the number of transactions in a block exceeds 1,000, the processing capacity range of the blockchain nodes is exceeded, causing excessive load on the entire blockchain network and reducing the efficiency of consensus. Therefore, to ensure reasonable consensus delay, the size of the blocks should be kept below 1,000 to maximize the operational efficiency of the blockchain.

Algorithm Time Consumption

To test the concurrency performance of the system, stress testing was implemented by simulating different numbers of clients. In this experiment, 50, 100, 250, 500, and 750 clients were created for concurrency testing, and the processing time was only counted for three types of contracts. In this test, the ABAC additional environment attributes only included traffic and frequency restrictions. In Figure 10, it is clear that as the concurrency increases, the total time cost of ABAC policy validation increases, and the average time cost decreases and then stabilizes. Therefore, throughput has not shown a significant downward trend as concurrency increases.

Learning Rate Analysis

The TensorFlow deep learning library and Keras deep learning top-level library serve as tools for constructing an LSTM network structure, and the training process was optimized using the Adam optimizer. The hyperparameters included the training time step t, the batch size, and the training frequency epochs. Through repeated training, the range of hyperparameter values was determined to be as follows: The step size range of t was 1~60, the batch size was set 64, the number of epochs was set to 500, and the activation function used was the tanh function. The evaluation indicator used to assess for prediction accuracy was the mean absolute percentage error (MAPE), smaller

Figure 9. Consensus delay for different block sizes

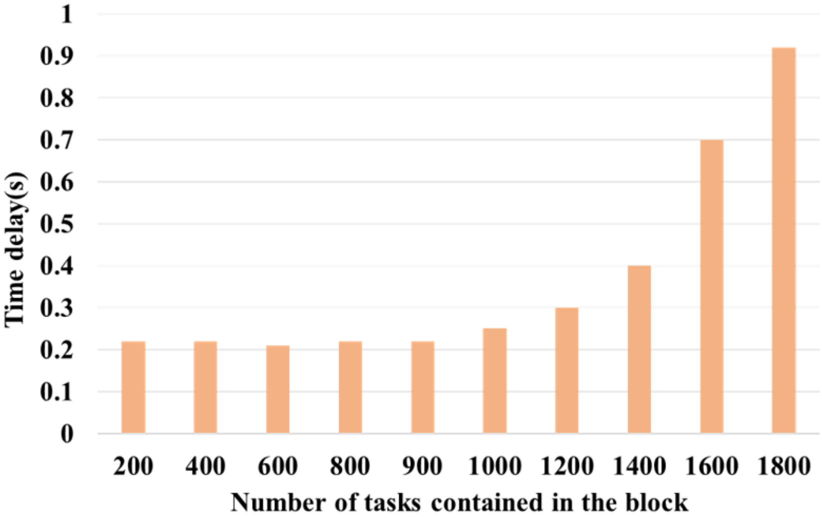
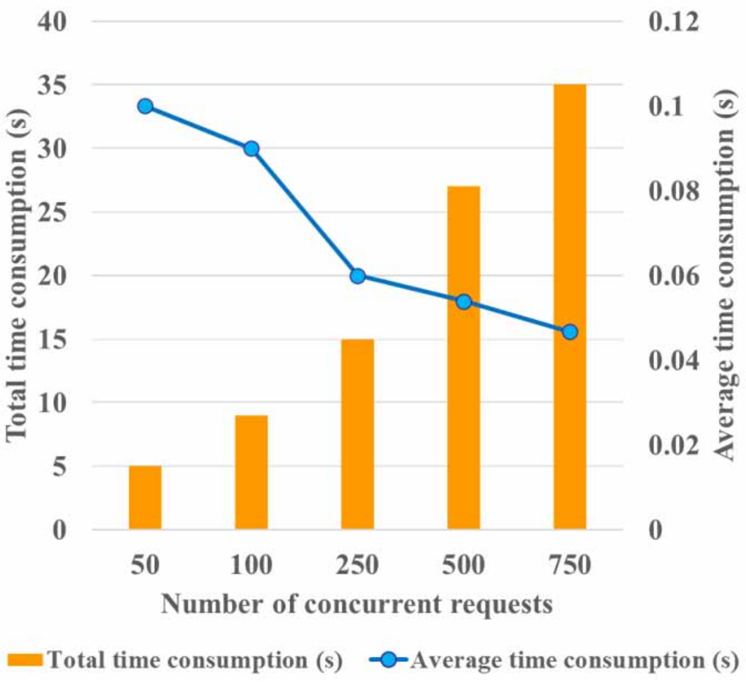


Figure 10. Policy verification contract time cost



values of which correspond to a lower deviation between the predicted value and the true value, and consequently a better prediction effect.

In order to evaluate the performance of the proposed LSTM based environmental attribute prediction method, experiments were conducted under the conditions of a publicly available data set (Narouei et al., 2017), which includes four categories: iTrust, IBM, Cyberchair, and Collected ACP, including 2,477 text data, all of which were manually annotated. Summarize the data from four types of data sets for experimentation, and divide the data set into training, validation, and testing sets in a ratio of 60%, 20%, and 20%. The statistical information of the data set is shown in Table 3.

Different learning rates and step sizes were tested to predict the properties of ABAC, and the MAPE results obtained are shown in Table 4. The results show that a high learning rate can easily lead to overfitting, while a low learning rate can lead to underfitting. The lowest MAPE value was obtained when the learning rate was 0.001, which corresponded to the best prediction effect.

Table 3. Data set statistics information for ACP

Dataset	Number of ACP	Number of Non-ACP	Total
iTrust	967	664	1 631
IBM	169	232	401
Cyberchair	140	163	303
Collected ACP	125	17	142
Total	1401	1 076	2477

Table 4. MAPE values for different learning rates and step sizes

Steps	Learning rate			
	0.1	0.01	0.001	0.0001
1	0.3057	0.2557	**0.1732**	0.2246
12	0.2839	0.2468	**0.1564**	0.1957
24	0.2598	0.2120	**0.1378**	0.1859
36	0.2246	0.1905	**0.1442**	0.1540
48	0.2425	0.2205	**0.1698**	0.1889
60	0.2654	0.2362	**0.1825**	0.2064

Block Size Analysis

The throughput of the system was tested under different block sizes. In the proposed architecture, the block size is related to the maximum amount of information count and the absolute maximum byte count. Two sets of simulation experiments were conducted under the same hardware conditions, with the maximum information count parameter set to 10, 50, 100, 150, 200, 350, 300, 350, 400, 450, and 500, and the absolute maximum byte count set to 1M, 5M, 10M, 15M, 20M, 30M, 35M, 40M, 45M, 45M, and 60M. The throughput performance of the system under different maximum information count values is shown in Figure 11, and corresponding performance under different absolute maximum byte count values is shown in Figure 12. The results indicate that larger block sizes result in a greater system throughput. The size of the blocks can be adjusted according to the demand of the actual IoT environments to increase system throughput. However, this value needs to be adjusted reasonably according to the application requirements in order to avoid resource wastage.

Figure 11. Throughputs for different maximum information count values

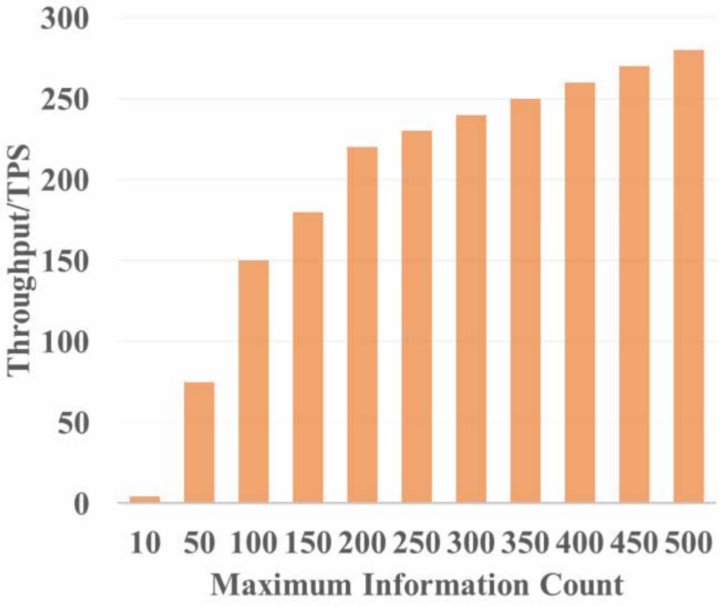
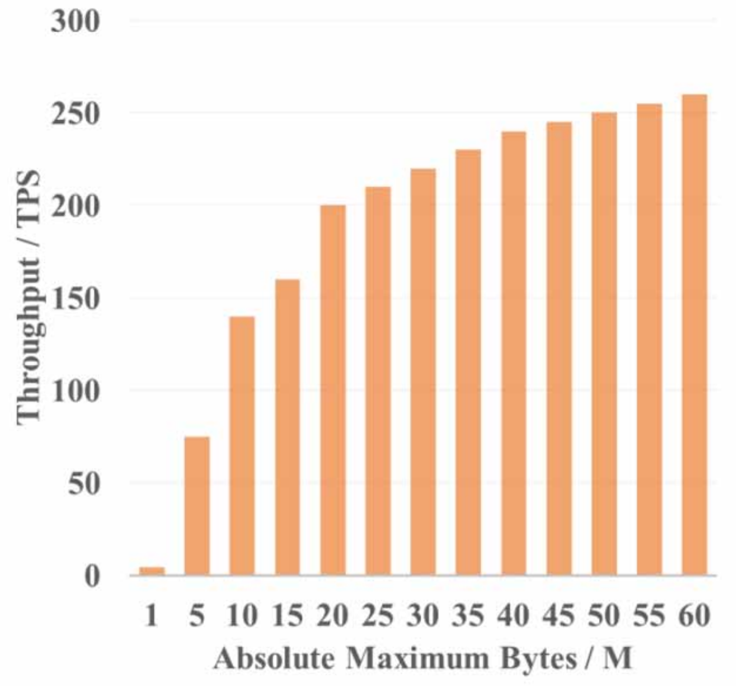


Figure 12. Throughputs for different absolute maximum byte values



Comparative Experiment Based on Consensus Algorithm

The throughput of the system was tested in different consensus algorithms. There are three consensus algorithms on the Hyperledger Fabric platform, namely Solo, Kafka, and Raft. Solo is a single-node mode that is not suitable for IoT environments; therefore, only the throughput of Kafka and Raft under different concurrent requests was tested experimentally, and the results are shown in Figure 13.

The results indicate that in the proposed strategy, the throughput of Raft is higher than that of Kafka. Meanwhile, Raft has the same fault tolerance characteristics as Kafka and can ensure the reliability of the system under high throughput conditions. Therefore, Raft's high throughput can meet the needs IoT applications.

Experimental Comparisons

Comparison of Communication Success Rates

In a real IoT environment, there may be a large number of malicious nodes in the network. When participating in the interaction between nodes, these malicious nodes may intentionally send error messages, resulting in errors in the information sent by normal nodes. At the same time, without the constraints of smart contracts, normal nodes may be hijacked by malicious nodes and become malicious themselves. As shown in Figure 14, due to the constraints of smart contracts on nodes, the success rate of establishing communication between nodes improves to a certain extent and is more stable as the network operation time increases. When smart contracts are employed, the bit error rate is controlled effectively, which indicates that the introduction of smart contracts can suppress the spread of malicious nodes effectively, thereby improving the security of the entire network.

Figure 13. Throughput obtained using different consensus algorithms for the proposed strategy

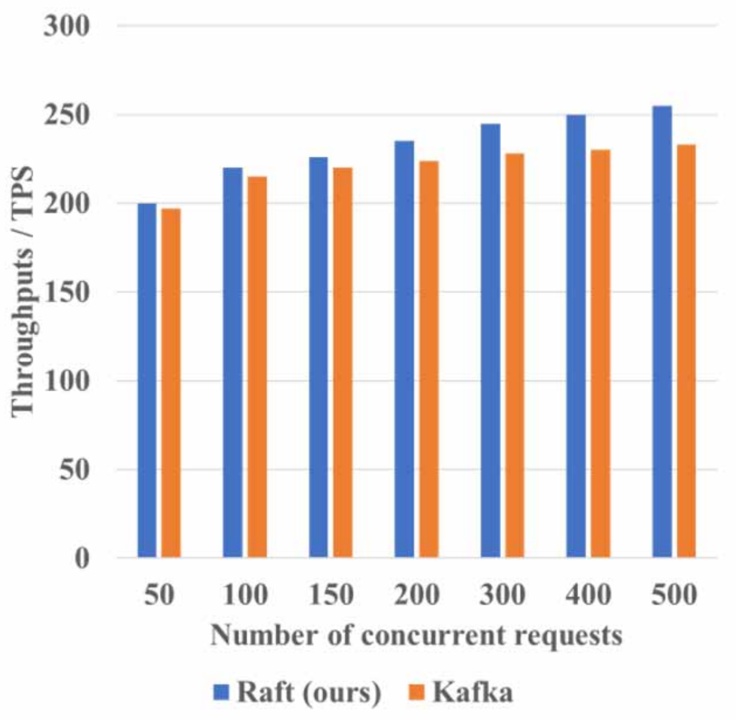
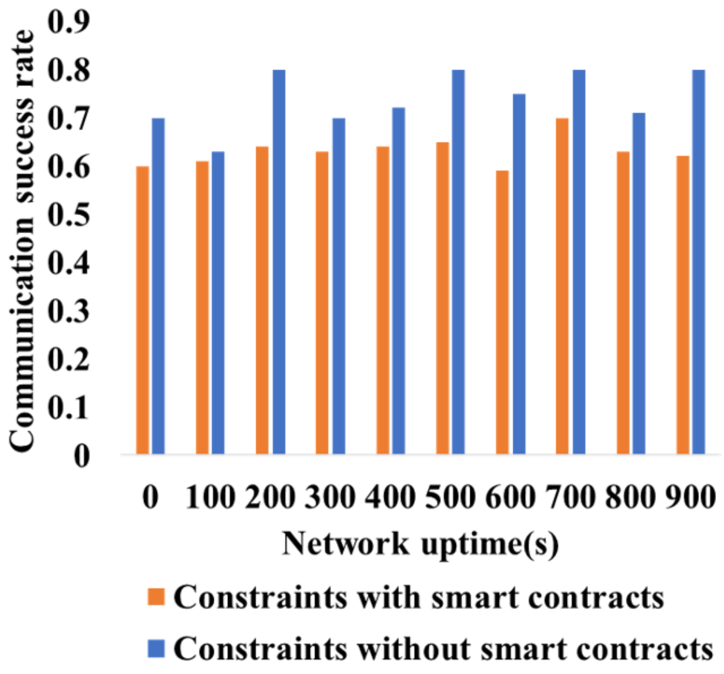


Figure 14. Impact of malicious nodes on communication success rate after using smart contract



Comparison of Information Exchange Rates

Node A was selected from domain F2 to randomly establish communication connections with other nodes in the trusted domain and simulate the process of information sharing between nodes in an IoT environment, and the results are listed in Figure 15. Here, the proportion of the number of other nodes that node A establishes connections to within the same trusted domain and within a certain period of time to the total number of nodes is used to evaluate the information interaction rate. Higher node information interactions rate per unit time indicates a higher degree of information sharing between devices in the trusted domain that contains the node. As shown from the results, after the introduction of smart contracts, the information exchange rate between the nodes is improved to a certain extent compared to the absence of smart contracts. This improves the efficiency of information sharing between nodes in a real IoT environment.

Comparison of Different Key Exchange Algorithms

In order to evaluate the performance of the proposed information security protection method, the proposed method was compared with CP-ABE (M. Zhang et al., 2020), CcBAC (Jiang, et al., 2023) and IFBT (Ren et al., 2021) under the same experimental conditions. The experimental results are shown in Figure 16.

A node A in domain F2 was selected and was allowed to establish connections with other nodes in the same trusted domain using the general ECDH key exchange algorithm and the CP-ABE, CcBAC, and IFBT algorithms, respectively. This experiment aimed to simulate a scenario where devices need to establish frequent connections with nodes with different computing power for information interaction in a real IoT environment. As the number of secure connections increased, the time required by the proposed algorithm was shorter compared to the other algorithms of the comparison. This is because the proposed optimization of the ABAC model combines blockchain with access control models, reducing the running time of the access control system effectively.

Figure 15. Impact of smart contracts on information interaction rate

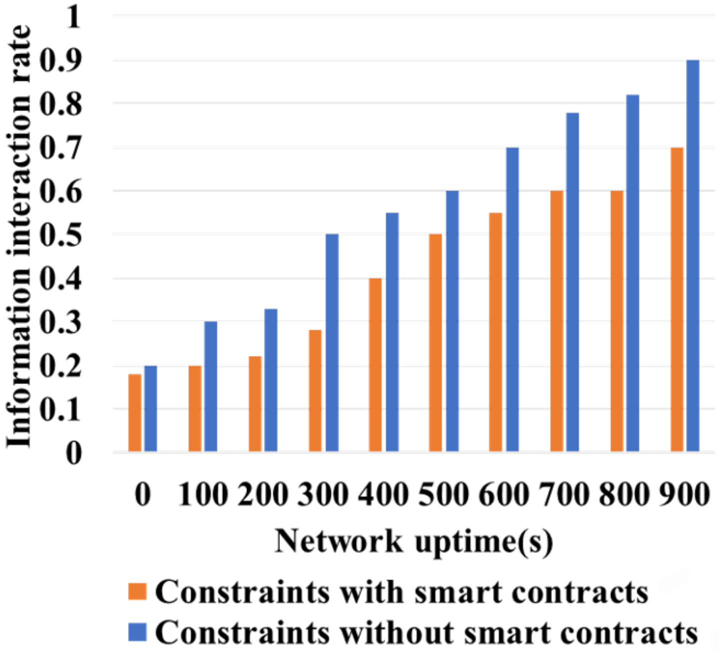
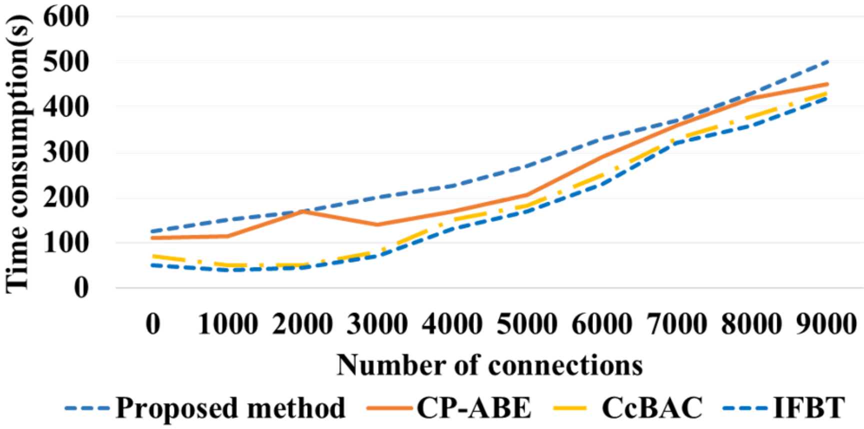


Figure 16. Comparison of different key exchange algorithms

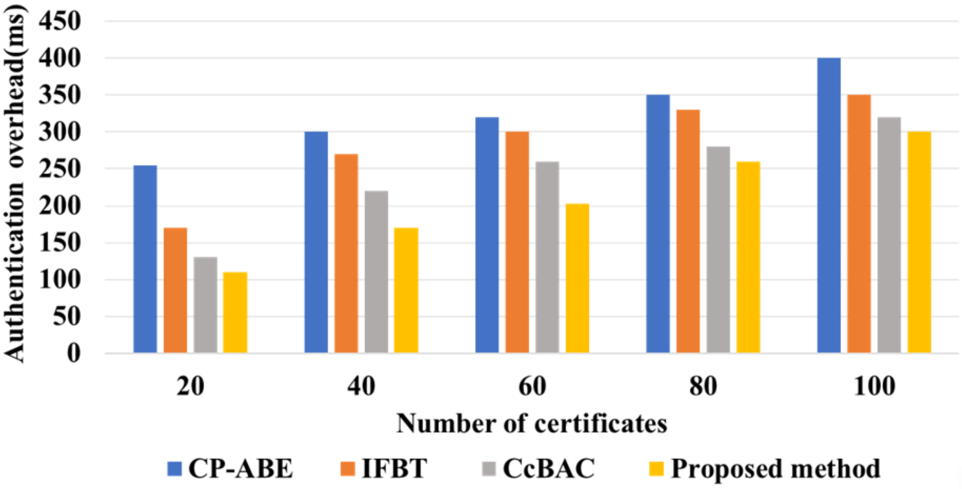


Comparison of Computational Costs Between Different Authentication Schemes

The computational overheads required for multiple authentications are compared for different algorithms, which is listed in Figure 17.

Among them, the CcBAC scheme quickly surpasses the CP-ABE scheme, while the overhead of the scheme proposed in this article was the lowest. When the number of certificates was 60, the computational overhead of the proposed method was only 203ms, which was lower than the times of the comparison methods. This is because the use of a signed token as a transferable asset allows access to various devices and applications in complex IoT environments while ensuring the security of interactions away from the blockchain network. The proposed method for predicting environmental attributes in ABAC access control policies using LSTM allows users to obtain the minimum access rights required.

Figure 17. Comparison of authentication calculation costs



Comparison of Election Times and Throughput of Different Algorithms

To prove the effectiveness of the proposed information security protection method, in this section the proposed algorithm is compared with the existing methods from the perspectives of latency and throughput. In the consensus algorithm, the delay represents the time required for the client to initiate a request and receive a response. In this experiment, only the election delay of the algorithm was recorded. First, the election times of the compared algorithms were recorded as the number of consensus node increased, as shown in Figure 18. As the number of nodes increases, the proposed method shows significantly improved election times compared to the other methods.

Throughput can directly affect the performance of consensus algorithms, as it affects the ability to process a number of requests per unit time. In this article, the number of consensus log entries completed per unit time was used as the throughput metric, which indicated the number of transactions that reached consensus within a fixed time. In the experiment, the client circularly initiated 1000 transaction proposals at a speed of 200 TPS, as shown in Figure 19. The experimental results show that the improved algorithm had higher throughput for the same number of nodes in the system. As the number of nodes increased, the system throughput tended to decrease.

CONCLUSION

The incompatibility of centralized platforms with IoT technologies inhibits the secure collaboration and sharing of information between IoT devices. Moreover, due to the wide distribution of devices, complex application environments, and limited computing and storage capabilities, their authentication and access control efficiency is low. In response to the above issues, in this article, the ABAC model is optimized, and blockchain technology is combined with access control models, while signed tokens are used as transferable assets. This allows effective access control between various devices and applications in complex IoT environments. The LSTM algorithm is used to predict environmental attributes and further restrict user access. A pre-signature mechanism is also designed to verify the reliability of the blockchain edge nodes. The experimental results show that the proposed data security protection scheme can solve the security and access control issues of IoT data storage, and

Figure 18. Comparison of election times

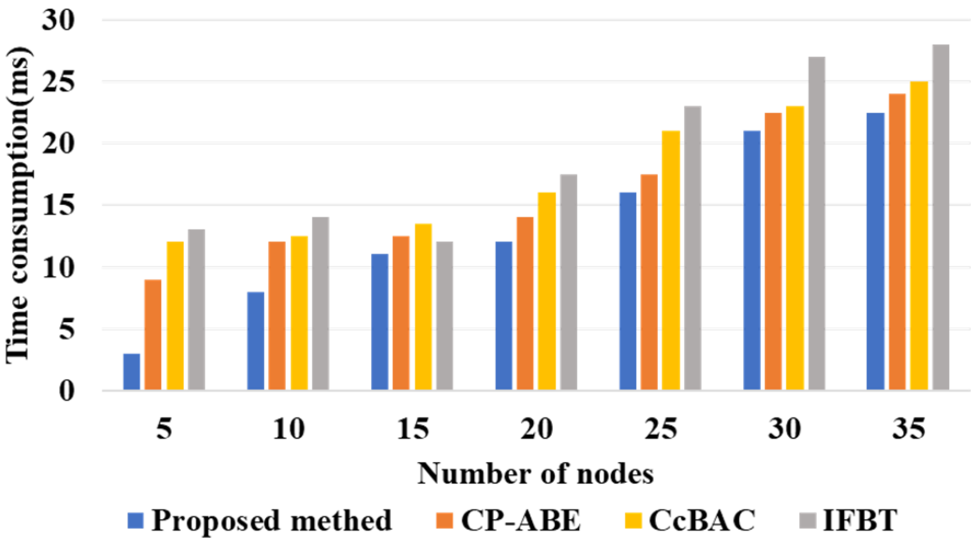
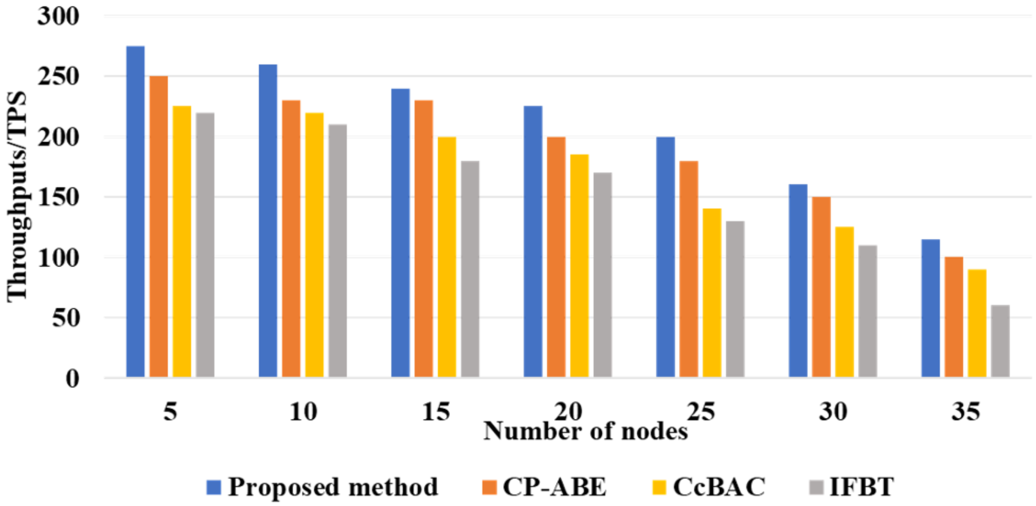


Figure 19. Throughput comparison of different algorithms



it meets the operational requirements of IoT systems. Therefore, the proposed solution is expected to be applied to real IoT environments.

The Raft consensus algorithm used in the proposed solution performed outstandingly in terms of throughput and fault tolerance, but its ability to resist malicious attacks needs to be improved. Therefore, a novel malicious attack detection model will be designed and introduced into the proposed scheme to further enhance the security of the system. The proposed solution has achieved good performance on the Fabric platform. In order to improve its scalability, it will be deployed on other platforms for experimentation, ensuring system security and stability. In addition, the proposed solution will be integrated with other IoT security mechanisms to enhance the predictive ability of semantic LSTM models in more complex environments.

ACKNOWLEDGEMENT

This research was supported by the Special Fund Project for Basic Scientific Research Business of the Ministry of Public Security (C22282)

REFERENCES

Al-qerem, A., Alauthman, M., Almomani, A., & Gupta, B. B. (2020). IoT transaction processing through cooperative concurrency control on fog–cloud computing environment. *Soft Computing*, *24*(8), 5695–5711. doi:10.1007/s00500-019-04220-y

Alizadeh, M., Andersson, K., & Schelen, O. (2020). A survey of secure IoTs in relation to blockchain. [JISIS]. *Journal of Internet Services and Information Security*, *10*, 47–75. doi:10.22667/JISIS.2020.08.31.047

Alotaibi, B. (2019). Utilizing blockchain to overcome cyber security concerns in the IoTs: A review. *IEEE Sensors Journal*, *19*(23), 10953–10971. doi:10.1109/JSEN.2019.2935035

Bamakan, S. M., Motavali, A., & Bondarti, A. B. (2020). A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Systems with Applications*, *154*, 1133–1152. doi:10.1016/j.eswa.2020.113385

Bamakan, S. M. H., Najmeh, F., & Ahad, Z. (2021). Di-ANFIS: An integrated blockchain-IoT-big data-enabled framework for evaluating service supply chain performance. *Journal of Computational Design and Engineering*, *8*(8), 676–690. doi:10.1093/jcde/qwab007

Cao, Y., Jia, F., & Manogaran, G. (2019). Efficient traceability systems of steel products using blockchain-based industrial IoTs. *IEEE Transactions on Industrial Informatics*, *16*(9), 6004–6012. doi:10.1109/TII.2019.2942211

Chaganti, R., Varadarajan, V., Gorantla, V. S., Gadekallu, T. R., & Ravi, V. (2022). Blockchain-based cloud-enabled security monitoring using IoTs in smart agriculture. *Future Internet*, *14*(9), 250–262. doi:10.3390/fi14090250

Chander, S., Vijaya, P., & Dhyani, P. (2022). A parallel fractional lion algorithm for data clustering based on mapreduce cluster framework. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–25. doi:10.4018/IJSWIS.297034

Ferrag, M. A., & Shu, L. (2021). The performance evaluation of blockchain-based security and privacy systems for the IoTs: A tutorial. *IEEE Internet of Things Journal*, *8*(24), 17236–17260. doi:10.1109/JIOT.2021.3078072

Gaurav, A., Psannis, K., & Peraković, D. (2022). Security of cloud-based medical internet of things (miots): A survey. [IJSSCI]. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–16. doi:10.4018/IJSSCI.285593

Guo, T., Yu, K., Srivastava, G., Wei, W., Guo, L., & Xiong, N. N. (2021). Latent discriminative low-rank projection for visual dimension reduction in green internet of things. *IEEE Transactions on Green Communications and Networking*, *5*(2), 737–749. doi:10.1109/TGCN.2021.3062972

Gupta, B. B., Chui, K. T., Gaurav, A., Arya, V., & Chaurasia, P. (2023a). A novel hybrid convolutional neural network-and gated recurrent unit-based paradigm for IoT network traffic attack detection in smart cities. *Sensors (Basel)*, *23*(21), 8686. doi:10.3390/s23218686 PMID:37960386

Gupta, B. B., Gaurav, A., Arya, V., & Kim, P. (2023b). A deep CNN-based framework for distributed denial of services (DDoS) attack detection in internet of things (IoT). In *Proceedings of the 2023 international conference on research in adaptive and convergent systems*, (pp. 1-6). IEEE. doi:10.1145/3599957.3606239

Gupta, B. B., & Quamara, M. (2020). An overview of internet of things (IoT): Architectural aspects, challenges, and protocols. *Concurrency and Computation*, *32*(21), e4946. doi:10.1002/cpe.4946

Hu, B., Gaurav, A., Choi, C., & Almomani, A. (2022). Evaluation and comparative analysis of semantic web-based strategies for enhancing educational system development. [IJSWIS]. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–14. doi:10.4018/IJSWIS.302895

Huang, L., Nan, R., Chi, K., Hua, Q., Yu, K., Kumar, N., & Guizani, M. (2022). Throughput guarantees for multi-cell wireless powered communication networks with non-orthogonal multiple access. *IEEE Transactions on Vehicular Technology*, *71*(11), 12104–12116. doi:10.1109/TVT.2022.3189699

Hui, H., An, X. S., Wang, H. Y., Ju, W. J., Yang, H. X., Gao, H. J., & Lin, F. H. (2019). Survey on blockchain for internet of things. *Computer Communications*, *136*, 10–29. doi:10.1016/j.comcom.2019.01.006

Jiang, W., Li, E., Zhou, W., Yang, Y., & Luo, T. (2023). IoT access control model based on blockchain and trusted execution environment. *Processes (Basel, Switzerland)*, *11*(3), 723. doi:10.3390/pr11030723

Kavita, S., & Dakshayani, G. (2022). A sliding window blockchain architecture for the internet of things. *2022 5th International Conference on Advances in Science and Technology (ICAST),* (pp. 45-48). IEEE. doi:10.1109/ICAST55766.2022.10039664

Khanam, S., Tanweer, S., & Khalid, S. S. (2022). Future of internet of things: Enhancing cloud-based IoT using artificial intelligence. [IJCAC]. *International Journal of Cloud Applications and Computing*, *12*(1), 1–23. doi:10.4018/IJCAC.297094

Khattak, H., Tehreem, K., Almogren, A. S., Ameer, Z., Din, I. U., & Adnan, M. (2020). Dynamic pricing in industrial IoTs: Blockchain application for energy management in smart cities. *Journal of Information Security and Applications*, *55*, 1026–1035. doi:10.1016/j.jisa.2020.102615

Kiran, M. A., Pasupuleti, S. K., & Eswari, R. (2022). Efficient pairing-free identity-based signcryption scheme for cloud-assisted IoT. [IJCAC]. *International Journal of Cloud Applications and Computing*, *12*(1), 1–15. doi:10.4018/IJCAC.305216

Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., & Ford, B. (2018). Omniledger: A secure, scale-out, decentralized ledger via sharding. *2018 IEEE Symposium on Security and Privacy (SP).* IEEE. doi:10.1109/SP.2018.000-5

Kshetri, N. (2017). Can blockchain strengthen the IoTs? *IT Professional*, *19*(4), 68–72. doi:10.1109/MITP.2017.3051335

Lee, V. W., Kim, C., Chhugani, J., Deisher, M. E., Kim, D., Nguyen, A. D., Satish, N., Smelyanskiy, M., Chennupaty, S., Hammarlund, P., Singhal, R., & Dubey, P. K. (2010). Debunking the 100x GPU vs. CPU myth: An evaluation of throughput computing on CPU and GPU. *Proceedings of the 37th Annual International Symposium on Computer Architecture*, (pp. 451-460). ACM. doi:10.1145/1815961.1816021

Liu, R. W., Guo, Y., Lu, Y., Chui, K. T., & Gupta, B. B. (2022). Deep network-enabled haze visibility enhancement for visual IoT-driven intelligent transportation systems. *IEEE Transactions on Industrial Informatics*, *19*(2), 1581–1591. doi:10.1109/TII.2022.3170594

Liu, Y. H., & Zhang, S. (2020). Information security and storage of IoTs based on block chains. *Future Generation Computer Systems*, *106*, 296–303. doi:10.1016/j.future.2020.01.023

Mamta, Gupta, B., Li, K., Leung, V.C., Psannis, K.E., & Yamaguchi, S. (2021). Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system. *IEEE/CAA Journal of Automatica Sinica*. IEEE.

Narouei, M., Khanpour, H., Takabi, H., Parde, N., & Nielsen, R. D. (2017). Towards a top-down policy engineering framework for attribute-based access control. *Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies*, (pp. 103–114). Association for Computing Machinery. doi:10.1145/3078861.3078874

Novo, O. (2018). Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet of Things Journal*, *5*(2), 1184–1195. doi:10.1109/JIOT.2018.2812239

Rahman, M. A., Rashid, M. M., Hossain, M. S., Hassanain, E., Alhamid, M. F., & Guizani, M. (2019). Blockchain and IoT-based cognitive edge framework for sharing economy services in a smart city. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 18611–18621. doi:10.1109/ACCESS.2019.2896065

Raj, M. G., & Pani, S. K. (2022). Chaotic whale crow optimization algorithm for secure routing in the IoT environment. [IJSWIS]. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–25. doi:10.4018/IJSWIS.300824

Ren, L. Y., Paul, A. S. W., & Bernard, W. (2021). Improving the performance of blockchain sharding protocols with collaborative transaction verification. *2021 IEEE International Conference on Blockchain (Blockchain)*, (pp. 462-469). IEEE. doi:10.1109/Blockchain53845.2021.00071

Rossini, M., Zichichi, M., & Ferretti, S. (2023). On the use of deep neural networks for security vulnerabilities detection in smart contracts. *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, (pp. 74-79). IEEE. doi:10.1109/PerComWorkshops56833.2023.10150302

Singh, R., Dwivedi, A. D., & Srivastava, G. (2020). IoTs based blockchain for temperature monitoring and counterfeit pharmaceutical prevention. *Sensors (Basel)*, *20*(14), 3951–3962. doi:10.3390/s20143951 PMID:32708588

Tibrewal, I., Srivastava, M., & Tyagi, A. K. (2022). Blockchain technology for securing cyber-infrastructure and IoTs networks. *Intelligent Interactive Multimedia Systems for e-Healthcare Applications, 8*, 337-350. 10.1007/978-981-16-6542-4_17

Tiwari, A., & Garg, R. (2022). Adaptive ontology-based IoT resource provisioning in computing systems. [IJSWIS]. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–18. doi:10.4018/IJSWIS.306260

Tsaur, W. J., Chang, J. C., Chen, C., & Chen, L. (2022). A highly secure IoT firmware update mechanism using blockchain. *Sensors (Basel)*, *22*(2), 530–541. doi:10.3390/s22020530 PMID:35062490

Wang, H. J., Wu, Z. F., Li, Y. L., Yan, Z. H., & Ma, J. W. (2021). Architecture design and application of distributed power trading system based on blockchain asynchronous consensus. *2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE),* (pp. 35-41). IEEE. doi:10.1109/AEMCSE51986.2021.00015

Wu, L., Du, X., Wang, W., & Lin, B. (2018). An out-of-band authentication scheme for IoTs using blockchain technology. *2018 International Conference on Computing, Networking and Communications (ICNC).* IEEE. doi:10.1109/ICCNC.2018.8390280

Wu, X., Kong, F., Shi, J., Bao, L., Gao, F., & Li, J. (2019). A blockchain IoTs data integrity detection model. *AISS*, *19*, 2025–2032. doi:10.1145/3373477.3373498

Xiao, J. (2021). Information security management of sharing economy based on blockchain technology. *Wireless Communications and Mobile Computing*, *12*(8), 1032–1042. doi:10.1155/2021/9931460

Xie, L., Ding, Y., Yang, H., & Wang, X. (2019). Blockchain-based secure and trustworthy IoTs in SDN-enabled 5G-VANETs. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 56656–56666. doi:10.1109/ACCESS.2019.2913682

Yu, H., Yang, Z., & Sinnott, R. O. (2018). Decentralized big data auditing for smart city environments leveraging blockchain technology. *IEEE Access : Practical Innovations, Open Solutions*, *7*(2), 6288–6296. doi:10.1109/ACCESS.2018.2888940

Zhang, M., Li, J., Chen, Z., Chen, H., & Deng, X. (2020). Cycledger: A scalable and secure parallel protocol for distributed ledger via sharding. *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS).* IEEE. doi:10.1109/IPDPS47924.2020.00045

Zhang, Q., Li, Y., Wang, R., Liu, L., Tan, Y., & Hu, J. (2021). Data security sharing model based on privacy protection for blockchain-enabled industrial IoTs. *International Journal of Intelligent Systems*, *36*(1), 94–111. doi:10.1002/int.22293

Zhang, Z., & Ren, X. (2021). Data security sharing method based on CP-ABE and blockchain. *Journal of Intelligent & Fuzzy Systems*, *40*(2), 2193–2203. doi:10.3233/JIFS-189318

Zhu, L., Chen, C., Su, Z., Chen, W., Li, T., & Yu, Z. (2020). Bbs: Micro-architecture benchmarking blockchain systems through machine learning and fuzzy set. *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA).* IEEE. doi:10.1109/HPCA47549.2020.00041

APPENDIX

List of abbreviations

Abbreviations	Full name
IoT	internet of things
LSTM	long short-term memory
ABAC	attribute-based access control
PoW	proof of work
MPT	Merkle Patricia tree
PEP	policy enforcement points
PAP	policy administration points
AA	attribute authority
AS	attribute of subject
AO	attribute of object
AE	attribute of environment
MAPE	mean absolute percentage error
PDP	policy decision points
ACP	access control policy

Ge Zhao, M.D. of Computer Science, Associate Researcher. Graduated from Xi'an Jiao Tong University in 2003. Worked in The Third Research Institute of the Ministry of Public Security. Her research interests include network security and iot security.

Xiangrong Li, Ph.D of Computer Science, Associate Researcher. His research interests include network security, trusted computing, mobile security and iot security.

Hao Li,Ph.D of Computer Science. Worked in Xian Key Laboratory of IOT Engineering. His research interests include Internet of Things Project and Intelligent Transportation.