A Blockchain-Based Cross-Domain and Autonomous Access Control Scheme for Internet of Things

Xiaohan Hao, Wei Ren, *Member, IEEE*, Yangyang Fei, Tianqing Zhu, and Kim-Kwang Raymond Choo, *Senior Member, IEEE*

Abstract—The volume, variety and value of data generated by Internet of Things (IoT) devices are expected to increase significantly in foreseeable future, hence, reinforcing the importance of secure and efficient access control solutions for these devices and their networks. However, existing access control solutions are not generally lightweight or scalable, particularly for geographically disperse, inexpensive resource constrained IoT devices. To tackle above challenges, we propose a lightweight consortium blockchain based architecture to enable intelligent autonomous access control for IoT devices. In our architecture, intelligent blockchain facilitates the storing of access policies, provision of authentication services for data access control, and trust evaluation for access request nodes through token accumulation mechanism. Specifically, the user's access request is approved only after it is confirmed by the blockchain network. To ensure the reliability of authenticity, a compromised resistant consensus algorithm is adapted and implemented to defend against at most $1/3$ compromised authenticators. In addition, a cross-domain and flexible access control model is not only used to support data sharing among various users but can also be used for access control for exceptional blockchain situations. We explain how our system meets our design goals of reliability, availability, confidentiality, integrity, lightweight, security and scalability. In addition, we also analyze the proposed system's performance from computational, storage and network overheads (e.g., running cryptographic algorithms on a Raspberry Pi 4B), and the findings suggest that the time to run typical cryptographic algorithms is in the microsecond range.

Index Terms—Blockchain, access control, cross-domain accessing, Internet of Things

1 Introduction

Internet of Things (IoT) has a growing potential in our life. For example, IoT devices have been and will continue to be deployed in smart cities, facilities, homes, military, and aggressive environments [1]. In addition, IoT devices can be installed around us, worn or carried by humans, and embedded in the human body (such as embedded medical IoT). Such devices can facilitate the tracking of users and their environment and collect sensitive private data. International Data Corporation (IDC) estimates that the quantity of IoT devices generated data may reach 79.4 zettabytes (ZB) [2] in 2025.

1.1 Motivations

Most IoT terminals are devices with limited computing, storage capacity, and network capabilities, so they cannot run complex security solutions. Therefore, a continuing challenge is designing secure and lightweight security solutions for IoT devices to ensure the security and privacy of static and transmitted data [3]. Access control is a common approach to ensure the confidentiality of data in storage and in transit, as well as allowing access to services and data for legitimate and authorized users and devices in blockchain situations. Given the increasing interdependence between systems, there is also a need to design access control solutions that support cross-domain authentication. Example scenarios include the need by government agencies (e.g., some federal

- *Xiaohan Hao is with the State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China, and also with the School of Computer Science, China University of Geosciences, Wuhan 430078, China. E-mail: xhhaozhk@163.com.*
- *Wei Ren is with the School of Computer Science, China University of Geosciences, Wuhan 430078, China, with the Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China, and also with the Yunnan Key Laboratory of Blockchain Application Technology, Kunming 650500, China. E-mail: weirencs@cug.edu.cn.*
- *Yangyang Fei is with the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, Henan, China, and also with the Henan Key Laboratory of Network Cryptography Technology, Zhengzhou, Henan 450001, China. E-mail: feiyangyang@pku.edu.cn.*
- *Tianqing Zhu is with the School of Computer Science, China University of Geosciences, Wuhan 430079, China. E-mail: tianqing.zhu@ieee.org.*
- *Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249 USA. E-mail: raymond.choo@fulbrightmail.org.*

Manuscript received 15 Feb. 2022; revised 30 Apr. 2022; accepted 31 May 2022. Date of publication 2 June 2022; date of current version 10 Apr. 2023.
This work was supported in part by the Foundation of Yunnan Key Laboratory of Blockchain Application Technology under Grants 202105AG070005 and YNB202103, in part by the National Natural Science Foundation of China under Grant 61972366, in part by the Provincial Key Research and Development Program of Hubei under Grant 2020BAB105, in part by the Foundation of Henan Key Laboratory of Network Cryptography Technology under Grant LNCT2020-A01, and in part by the Foundation of State Key Laboratory of Public Big Data under Grant PBD2021-02. The work of Kim-Kwang Raymond Choo was supported only by the Cloud Technology Endowed Professorship. (Corresponding author: Wei Ren.)
Digital Object Identifier no. 10.1109/TSC.2022.3179727

1939-1374 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

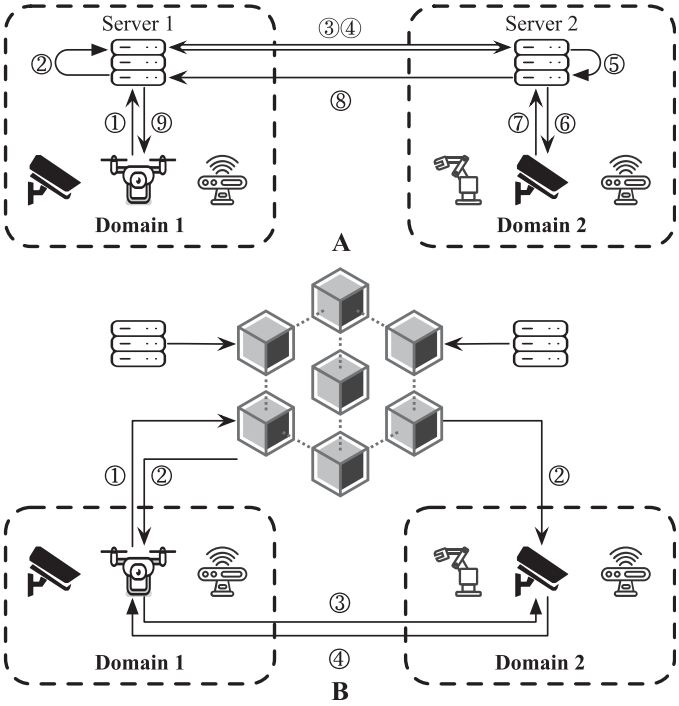


Fig. 1. Examples of cross-domain access control schemes, where A denotes a conventional approach and B describes the proposed blockchain-based approach.

or state law enforcement agency) gaining access to data or devices owned by other government agencies, private sector organizations or stakeholders. Similarly, organizations may also need to access IoT-related services or data owned by their vendors or organizations in the supply chain. To ensure the efficiency of access control scheme and enlarge the scale of blockchain network, we implement a cross-domain Role-Based Access Control (RBAC) model to control access devices in all access control domains autonomously.

However, many intermediate processes may prevent legitimate access requests from being responded to due to network delays, server bottlenecks and attacks. Typical process of traditional access control system is shown in Fig. 1 A. When device 1 in access control domain 1 tries to access resources of a device 2 in access control domain 2, it will go through the following steps:

1) Device 1 in domain 1 initiates an access control request to the target resource to the server in domain 1.
2) Server 1 verifies the identity of device 1 and confirms that it is a legitimate user of domain 1.
3) Server 1 and Server 2 authenticate each other's identity.
4) Server 1 sends the access request from device 1.
5) Server 2 determines whether the access request is legal.
6) For legitimate requests, server 2 requests resources from device 2.
7) Device 2 sends the requested resource to server 2.
8) Server 2 forwards the resource to server 1.
9) Device 1 obtains the requested resource from server 1.

1.2 Example

In conventional solutions, IoT devices in different access control domains need to communicate with the management server of the domain before accessing each other's resources, and then the server sends a request to the server in the target domain. Therefore, completing access requires a lot of complicated processes of multiple authentication and network communication between devices and servers, which consumes a lot of computing and network resources in large-scale IoT scenarios. In addition, these centralized servers may become the bottleneck of the system, causing excessive network delays and even paralysis.

While several methods have been proposed to address the above-discussed problems, it is also known that existing solutions also have a range of limitations. The schemes described in [4], [5] deploy access control at a device level, and rely on some centralized server(s). However, the cost of implementing access control policies and verification at the device level may exceed computing and storage capabilities of inexpensive IoT devices [4]. Besides, we have to also consider the risk of single point of failure and performance bottleneck in centralized solutions. Thus, this motivates us to design our solution that deploys access control on blockchain's peer-to-peer network. A simplified example is shown in Fig. 1 B. For example, when device 1 in domain 1 attempts to access the resources of device 2 in a different domain (i.e., domain 2 in our example), it will go through the following steps:

1) Device 1 sends an access request to blockchain network.
2) The blockchain network processes above request. For legitimate access requests, access tokens will be generated and stored on blockchain and synchronized with all devices.
3) Device 1 uses the token to directly access resources of Device 2.
4) Device 2 verifies the token and returns the requested resource.

1.3 Contributions

Using the blockchain network to replace original centralized server can simplify the access process between IoT devices in various contexts and reduce unnecessary computing and network resource overheads. In addition, distributed blockchain network can also reduce the risk of single points of failure and provide services for IoT devices more stably. Specifically, we design a based on cross-domain roles and conditions to facilitate access control within and between domains. To ensure lightweight requirements, access control policies are stored on blockchain in the form of transactions, and authentication processes are completed by full nodes of blockchain network. In addition, we use a practical Byzantine Fault Tolerance (PBFT) consensus algorithm to ensure the authenticity of our scheme. We also propose an intelligent method, which can be used for access control in exceptional situations through learning normal mode or realizing trust evaluation of access request nodes through intelligent learning. A summary of the contributions in this paper is as follows:

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

1) We design a lightweight consortium chain based architecture to facilitate intelligent autonomous access control for IoT devices, which can support temporary joining and leaving of IoT devices without incurring significant overheads during access control and authentication.
2) To ensure reliability, security and scalability, we propose a cross-domain access control model, which can support secure data sharing between IoTs in different domains.
3) We design a transaction script framework to implement condition-based access control on the top of the cross-domain role-based access control model.

The remainder of the paper is organized as follows. In Section 2, we review current blockchain-based access control schemes. In Sections 3, we introduce the relevant background and system model of proposed system. Our proposed scheme is presented in Section 4, and the evaluation is presented in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

There are solutions that not only use blockchain as a database, but also use blockchain to complete access control authentication. Ouaddah *et al.* [6], for example, proposed a full decentralized framework named FairAccess, to use pseudonyms for privacy protection purposes. Unlike bitcoin transactions, FairAccess introduces new transaction types for accessing permissions based on the Bitcoin blockchain and Bitcoin scripts, such as authorization, revocation, etc. However, based on the Bitcoin blockchain, FairAccess has flaws in performance and cannot adapt well to the IoT scenario. To address the challenges of VANET data security and privacy, Liu *et al.* [7] proposed a blockchain-based VANET data fine-grained access control scheme by combining blockchain and ciphertext-based attribute encryption technology. Instead of some untrusted third-parties, they used blockchain for identity management and established different VANET data access rights based on user attributes. To avoid the possibility of bias in trusted delegation services, Ali *et al.* [8] utilized blockchain to implement a framework for authorization and access control in IoT that meet the requirements of secure authorization services, such as decentralized and traceable. Feng *et al.* [9] utilized blockchain to implement a cross-domain authentication method for drones. It uses multiple signatures shared based on threshold to build the collaborative domain's identity federation and utilizes smart contracts for identity verification to secure cross-domain authentication and communication between domain devices. To improve the efficiency of edge computing, Ren *et al.* [10] designed an identity management mechanism combined with access control based on blockchain, which uses self-authentication cryptography to implement network entities' registration and authentication and provide authentication, audibility, and confidentiality for industrial IoT data. Sun used blockchain and role mapping technology to establish a credible and efficient cross-domain access control method, providing a reliable and verifiable cross-domain access control mechanism, thereby solving the limitations of a single server structure. Singh *et al.* [11] established a cloud-based centralized information sharing system with multiple security gateways to address cross-domain communication's privacy and security issues. They also provided data authentication and transaction algorithms to enable data to be distributed globally secure transactions and sharing between different domains. To address cross-domain communication's security and privacy issues, Shen *et al.* [12] proposed an efficient blockchain-assisted secure device authentication mechanism for cross-domain IIoT and an identity management mechanism for ensuring identity anonymity. They utilized consortium blockchain to build trust between devices to ensure the communication security of devices when they cooperate cross-domain tasks. To predict failures or monitor network security problems in real time, Lopez [13] studies how digital twins develop their context awareness and simulation technology, so as to update access control policies. To find ways to prove and track connections to govern or audit operations, Alcaraz [14] used blockchain to propose a three-tier interconnection architecture and various interconnection strategies to realize the safe connection between management entities, processes, and critical resources. According to the coupling degree of blockchain technology and the proposed scheme, they put forward the best strategy suitable for the smart grid.

TABLE 1
The Comparisons With Other Schemes

	Feng [9]	Jiang [15]	Tan [16]	Our scheme
Cross-Domain	✓	✓	×	✓
Intelligent and Autonomous	×	✓	✓	✓
Lightweight	Not mentioned	×	✓	✓
Scalability	✓	×	✓	✓
Resist Attacks	✓	Not mentioned	Not mentioned	✓
Reliability	Not mentioned	✓	×	✓

However, some of their works are not optimized for cross-domain access control, or not involving intelligent learning. We propose a lightweight consortium chain-based architecture to realize intelligent autonomous access control of IoT devices, support the temporary joining and leaving of IoT devices. Next, we propose a cross-domain access control model, which can support the access control between different domains to support secure data sharing between IoT. We also propose an intelligent method that can be used for access control in abnormal situations by learning normal patterns, as well as to achieve trust evaluation of access requesting nodes. The comparisons with other schemes are described in Table 1.

3 Preliminary

3.1 Blockchain

The blockchain system is actually a system that maintains the public data ledger, and all technical units are designed to better maintain this ledger. In the blockchain distributed network, consistent of data ledger of each node is kept

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

through a consensus algorithm such as PoW (Proof of work), PoS (Proof of Stake), PBFT (Practical Byzantine Fault Tolerance), Paxos, etc [17]. While these consensus algorithms stipulate the generation method of new blocks, they also create some thresholds for generating blocks. These thresholds are superimposed under the action of "hash pointer", so that attackers must pay a great price to successfully tamper with the block data without being discovered. Moreover, blockchain system can also expand the expression of ledger data through a powerful script system. It can even be considered that a blockchain system is actually a specially designed distributed database system. In this database, digital currencies can be stored, as well as smart contracts with more complex logic and a wider range of business data. Beginning from Bitcoin, early blockchains were geared towards digital currencies, such as Bitcoin and Litecoin [18]. At this stage we can think of blockchain as a system that supports digital currency contracts. Later, a more flexible system capable of supporting custom smart contracts appeared. Its representative is Ethereum [19]. It can be considered that Ethereum is an expansion of digital currency systems such as Bitcoin, but Ethereum still has built-in support for digital currencies, continuing the financial attributes of Bitcoin, and making the application of Ethereum more oriented to the financial sector. The next representative is the Hyperledger project, especially the Fabric sub-project. In this system, it goes beyond the application of the financial field and supports data definition in various fields.

3.2 Consensus Algorithm

Blockchain networks are interconnected by various of nodes through P2P communication protocol. Full nodes, also known as "miners", are the most important part of the network. They maintain the blockchain network and the blockchain ledger together. Every new block will be recognized only when those full nodes reach a consensus through consensus algorithm. Thus, consensus algorithms are so essential that it determines the overall scale of the blockchain network and the ability to resist attacks. Previous works have introduced, compared and analyzed mainstream consensus algorithms[20]. In this paper, we mainly introduce the PBFT consensus algorithm that the proposed system implemented.

The core problem to be solved by PBFT and consensus algorithms is how to maintain consistency of the cluster state in a distributed environment. In other words, given a set of operations for a set of data or services, we can end up with the same result on all devices in the cluster. In the PBFT, nodes have only two roles, "primary" and "replica". The two roles can be converted to each other in the "View". The maximum number of fault-tolerant nodes of PBFT is $(n-1)/3$, where n is the total number of nodes. This means that a cluster of 4 nodes can only tolerate one node for evil or failure at most.

At the beginning of PBFT, primary node is selected from all nodes by calculating $p = v \bmod n$, where v is current view number. First, the client sends a message m to the primary node p, and p starts a three-phase PBFT protocol: *pre-prepare*, *prepare* and *commit*.

3.2.1 Pre-Prepare Phase

After receiving message m from client, p constructs a *pre-prepare* message structure $<< PRE-PREPARE, v, n, d >, m >$ (d is message digest of m) and broadcasts it to all other nodes in the cluster.

3.2.2 Prepare Phase

Replica nodes will check it after receiving the request from p, and the message will be stored in the node if the check passes. When a node receives more than *2f+1,f=(n-1)/3* identical messages and the check passes, it will enter the *prepare* state and broadcast a message $<< PREPARE, v, n, d, i >>$, where i is the id of the node.

3.2.3 Commit Phase

Replica node will check the validity of *prepare* messages after receiving them. It will enter *commit* phase and broadcast *commit* message $< COMMIT, v, n, D(m), i >$ to other nodes in the cluster after it receives $2f+1$ (including itself) consistent *prepare* messages.

Finally, after receiving $2f+1$ consistent *commit* messages, replica node executes operations contained in message m.

PBFT algorithm is suitable for deployment in consortium or private blockchain networks. It can resist up to $1/3$ of node failures while maintaining high efficiency. That is very important in an unstable Internet of Things environment [21].

3.3 Access Control

Due to the needs of multi-domain information sharing, smart city IoT platforms need to process data from various platforms to support superior services. For example, based on the data of disaster prevention and traffic management systems, the best route for the ambulance to reach the destination can be calculated [22].

Unlike compulsory or discretionary access control, which directly grants users permissions, RBAC assigns permissions to roles. There are four main elements in RBAC, U (user), R (role), P (permission) and S (session). Administrators can assign P to R to specify permissions that the role has. Then they can specify the role of the user by assign R to U. In this way, users can access resources through permissions of the roles they hold. In addition, the setting of sessions ensures that users will not activate multiple conflicting roles that have assigned to them.

To ensure the efficiency of access control system and expand the scale of blockchain network, we have expanded traditional access control model and implement a cross-domain RBAC model to autonomously manage access to devices in all access control domain. The cross-domain RBAC model has the following components:

- S : the access subjects of access control model;
- O : the access objects of access control model;
- R : roles in traditional RBAC models;
- D : access control domains;
- D_iS : subject set in access control domain D_i;
- D_iO : object set in access control domain D_i;
- D_iR : role set in access control domain D_i;

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

- $SA_i \subseteq D_iS \times D_iR$: a many-to-many subject-to-role relation in access control domain D_i;
- $OA_i \subseteq D_iO \times D_iR$: a many-to-many object-to-role relation in access control domain D_i;
- $CA_i \subseteq D_jR \times D_iR$: a many-to-many role-to-role relation which mappings roles of other access control domains D_j to roles of access control domain D_i.

The design of cross-domain RBAC is similarly to the traditional RBAC, but two improvements have been implemented to better achieve cross-domain access control: (1) a set D of access control domains was defined in addition to the sets of subject, objects and roles. D_iS, D_iO and D_iR are subsets of S, O and D, respectively, and they represent the subjects, objects, and roles managed under access control domain D_i. So they satisfy the following relationship: $S = D_1S \cup D_2S \cup \cdots \cup D_nS$, $O = D_1O \cup D_2O \cup \cdots \cup D_nO$ and $R = D_1R \cup D_2R \cup \cdots \cup D_nR$. (2) A cross-domain role-role mapping assignment CA is partitioned into disjoint subsets: $CA = CA_1 \cup CA_2 \cup \cdots \cup CA_n$. If $(d_jr, d_ir) \in CA_i$, that means administrators of domain D_i has mapped role d_jr ($d_jr \in D_jR$) from other domain D_j to local role d_ir ($d_ir \in D_iR$), so that users d_js ($d_js \in D_jS$) in domain D_j who satisfies $(d_js, d_jr) \in SA_j$ can access object d_io ($d_io \in D_iO$) who satisfies $(d_io, d_ir) \in OA_i$ in domain D_i.

Furthermore, We have redesigned the blockchain transaction script so that the proposed blockchain system can support additional conditional judgments on top of the cross-domain RBAC model, and implement condition-based access control.

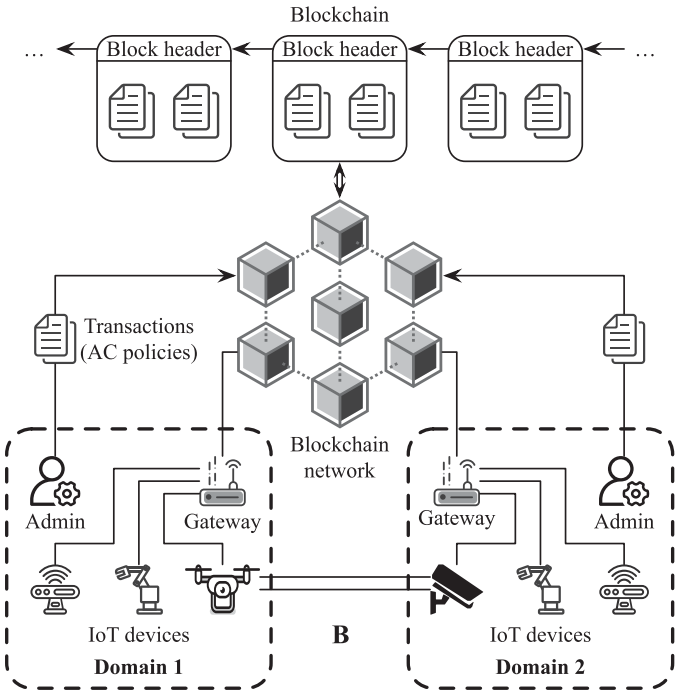


Fig. 2. System model.

4 Our Proposed Scheme

4.1 System Model and Framework

Next, we propose our system model and illustrate definition and function of each components. The system model of proposed system is depicted in Fig. 2. The system consists of three parts: blockchain, blockchain network, IoT terminals and service nodes and access control domain.

Blockchain. blockchain in our proposed system model stores transactions and executable smart contracts. Protected by cryptographic algorithms and consensus algorithms, transactions and smart contracts can hardly be tampered with without being detected. In addition, all operational records and the status of smart contracts can be traced and audited.

Blockchain Network. access control policies are deploying and executing on blockchain network to stabilize access control service.

Access Control Domain. in the proposed system, we divide all IoT terminals and other network devices into different access control domains according to application scenarios. The design of the access control domain does not prevent devices belonging to different domains from accessing each other. On the contrary, through the settings of the administrator, these devices can access each other legally and under protection. We use the proposed cross-domain RBAC model to achieve access control between devices from the same or different access control domain.

Multitude of IoT terminals may exist in an access control domain. These terminals are the main objects served by proposed system. Limited by resources on the scope of computing, storage and network, it is difficult for IoT terminals to act as full nodes to maintain blockchain network, back up complete blockchain data or join in consensus process. Therefore, there are many service nodes in the proposed scheme to maintain some operations of the entire system and provide services for IoT terminals. Service nodes are nodes such as NAS (Network Attached Storage) and API (Application Programming Interface) servers, which can obtain access resources or access IoT devices. They have enough computing resources and storage resources to access the blockchain network directly.

Full Nodes. the full node is mainly responsible for maintaining the blockchain network and blockchain ledger, receiving and processing transactions from administrators and IoT terminals, and packaging legal transactions into blocks and synchronizing these blocks through a consensus protocol.

Administrator. administrator manages the access control permissions of all device in the access control domain, sets roles for devices and configures which resources that the roles can access. In order to achieve cross-domain access control, the administrator also needs to manage the mapping between roles of other domains and roles of this domain.

Gateway. gateways and IoT terminals have a division of labor and work in conjunction with each other. Due to the limited computing resources, it is challenging for IoT devices to complete all blockchain functions. Gateways are full nodes of blockchain, which can be used to help IoT terminals to verify access authorization.

Other Terminals. there are other terminals that can be accessed in the access control domain, such as storage servers. Unlike IoT devices, these terminals can act as nodes of a blockchain network, store data from blockchain networks, and verify the status of smart contracts.

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

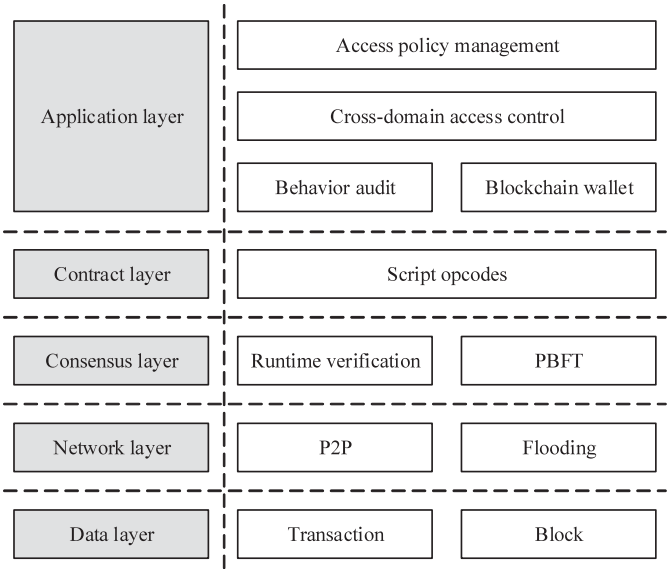


Fig. 3. Architecture of proposed system.

Our proposed system is based on a five-layer blockchain architecture design, including data, network, consensus and application layers. The composition structure of the proposed system is depicted in Fig. 3. In *data layer*, new transaction types, such as administrator transaction, user transaction and access base transaction, were designed to achieve an autonomous access control system. In addition, the system uses the RTXO (Reusable Transaction Output) and UTXO (Unspent Transaction Output) to easily trace operation records of any entity. In *network layer*, the existing IoT network protocols and transmission mechanisms, such as MQTT and Coap, are very practical and compatible with the proposed system, and we use these schemes directly. In *consensus layer*, we have appropriately expanded on the basis of the PBFT consensus protocol to meet the design goals. In *contract layer*, opcodes are designed to implement various operations required by entities. In *application layer*, we proposed a distributed autonomous cross-domain role-based access control system including access control management, access authorization, audit and wallet. In this paper, we will describe each layer in detail from the bottom to up except the network layer.

4.2 Adversary Model and Design Goals

Every part of the system model has the potential to be attacked by adversaries. An adversary can attack the blockchain network and paralyze the access control system. Gateways could be hacked to fail to provide blockchain services for IoT devices. An adversary can make a feint of being a legitimate device to destroy system. For example, it can pretend to be an administrator to destroy access control policies, pretend to be a subject to illegally obtain resources, or pretend to be an object to provide fake data for the subject. Concretely, adversaries may make some malicious behaviors as follows:

1) Attack on the blockchain network, directly modifies the data stored on blockchain to achieve tampering with access control policies or affect the execution results of access control.
2) Disguised as an administrator to tamper with access control policies.
3) Disguised as a gateway to cheat IoT devices that illegal subjects have legitimate access rights.
4) Disguised as a legal subject to illegally obtain resources.
5) Disguised as a legitimate object to provide false data to the subject.
6) Man-in-the-middle type attacks where the adversary appears on gateways in the form of IoT objects or sensors.

In this paper, we are committed to achieving cross-domain autonomous access control in the IoT environment. Users and IoT devices can easily and securely access resources in other domains through the proposed cross-domain RBAC model. The deployment and execution of access control policies do not rely on local server but the blockchain network to stabilize access control service provided by the system. Our design goals are as follows:

Reliability and Availability. reliability requires that the access control system can provide services stably over a long period of time, while availability requires that the access control system can provide services correctly under any circumstances.

Confidentiality and Integrity. confidentiality of access control system ensures that resources are not accessed without authorization. Integrity ensures that resources are not modified without authorization.

Lightweight IoT devices are mostly embedded devices with limited computing, storage, and network resources. Therefore, access control systems should consume as few resources as possible to run on these devices without affecting their normal operation. Lightweight means that the proposed system is resource-friendly in design, and IoT devices can withstand resource consumption during system operation.

Security. in the case of decentralization or multi-center threshold node resistance to compromise, it can prevent unauthorized access or malicious tampering, indicating that the system is safe.

Scalability. considerate IoT devices are usually deployed in IoT. In order to effectively manage access between these devices, the access control system must achieve great scalability.

Intelligent. considering that there are many and very complex IoT devices, ensuring the stability of the access mechanism in the event of an abnormal situation is an important goal. In addition, it is also one of our goals to realize the evaluation of access request nodes through intelligent learning.

4.3 Data Layer

The block structure of the proposed system is similar to that of Bitcoin blockchain. However, the proposed system does not use PoW as its consensus algorithm, so there is no difficult value and nonce on the structure. All fields in a block are described in Table 2.

There are three types of transactions: administrator transaction, user transaction and accessbase transaction. All transactions share the same data structure and perform

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

TABLE 2
The Description of Block

Field	Description
	block_header
block_size	Size of this block.
version	Indicates the version of protocol used by block for future expansion.
prev_block_hash	Hash value of **block_header** of the previous block.
merkle_root_hash	The root hash of the merkle hash tree formed by using hash value of each transaction as the leaf node.
timestamp	Timestamp when the block was generated.
signature	Full node signs this block with its private key.
public_key	Public key of the full node that packaged this block.
	block_body
tx_num	Number of transactions contained on the block.
txs	The list of transactions.

different operations with different opcodes. All fields in a transaction are shown in Table 3. *tx_in* means a data structure pointing to the UTXO or RTXO, and it includes *prev_hash*, *n* and *unlock_script*. Besides, *tx_out* means a data structure containing operations. If a *tx_out* is not referenced by *tx_in* of another transaction, it will become UTXO or RTXO, depending on whether it own **use_script* field. UTXO will be consumed once it is referenced by *tx_in* , while RTXO can only be consumed by specific *tx_in* . Before that, RTXO can be referenced by multiple *tx_in* . All fields in *tx_out* are including *operation*, *lock_script* and **use_script*.

4.3.1 Administrator Transaction (Tx_Admin)

The administrator of access control domain uses *Tx_Admin* to manage access control policies under that domain. All *Tx_Admin* must have an *tx_out* for the *tx_in* of the next *Tx_Admin*, so all *Tx_Admin* must also have an *tx_in* to point to that *tx_out*. By specifying different opcodes in *operation*, administrators can implement different operations on access control policies. According to operation types, *Tx_Admin* can be divided into *domain registration transactions* (*Tx_DR*), *user-role management transactions* (*Tx_SA*) and *role-right management transactions* (*Tx_OA*), *role-role management transactions* (*Tx_CA*) and *operation remove transactions* (*Tx_RM*). Transaction history of a single administrator may be as shown in Fig. 4 A.

Tx_DR must be the first transaction created by the domain administrator. This type of transaction has no *tx_in* and only one *tx_out* which will be a UTXO for the next *Tx_Admin*.

Tx_SA is used to add user-role relationships. *Tx_SA* role mappings has one *tx_in* and at least two *tx_out*. The *tx_in* points to the UTXO of the last *Tx_Admin*. One *tx_out* will be a UTXO for the next *Tx_Admin*, and the remaining *tx_out* will be RTXO for *Tx_User* and can only be consumed by *Tx_RM*.

Tx_OA is used to add role-right relationships. *Tx_OA* has only one *tx_in* refers to the UTXO of the last *Tx_Admin* and at least two *tx_out*. One of the *tx_out* will be a UTXO for the next *Tx_Admin*, and the remaining *tx_out* will also be UTXO and will be referenced by *tx_in* of transactions that confirm the right by resources.

Tx_CA is used to add role-role relationships mentioned in Section 4.1. *Tx_CA* has one *tx_in* and at least two *tx_out*. The *tx_in* points to the UTXO of the last *Tx_Admin*. One *tx_out* will be a UTXO for the next *Tx_Admin*, and the remaining *tx_out* contain necessary operations to establish role-role mappings.

Tx_RM is used to remove user-role mappings, role-right mappings and role-role mappings. *tx_in* of this transaction will point to UTXO or RTXO which add mappings that need to be removed.

TABLE 3
The Description of Transaction

Field	Description
tx_hash	Hash value of this transaction.
tx_type	Type of this transaction, `0x00` for accessbase, `0x01` for administrator and `0x02` for user.
tx_in_num	Number of inputs for this transaction contained in field **tx_in**.
tx_out_num	Number of output for this transaction contained in field **tx_out**.
timestamp	Timestamp when this transaction was generated.
	tx_in
pre_hash	**tx_hash** from previous transaction.
n	Sequence number of UTXO or RTXO of transaction.
unlock_script	Public key of the full node that packaged this block.
	tx_out
operation	A fixed-format statement containing opcodes through which entities perform operations they need.
lock_script	Similar to the script in Bitcoin, ensuring that only legitimate entity can unlock the **tx_out**.
***use_script**	For users using RTXO. Users can use this **tx_out** through the script, but cannot use their own **unlock_script** to unlock it.

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

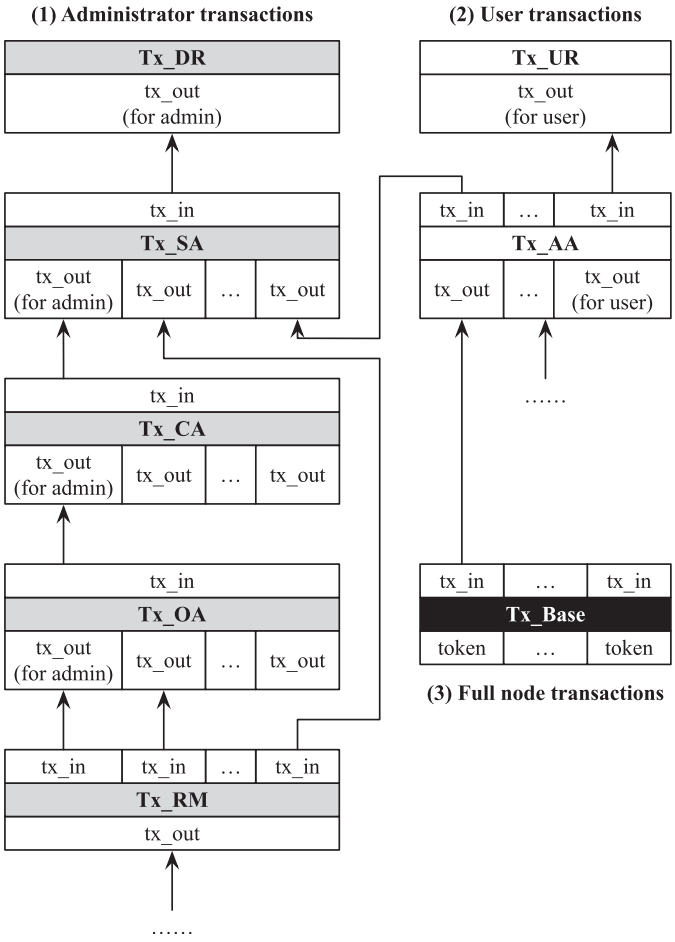


Fig. 4. Index relationship between inputs and outputs of transactions.

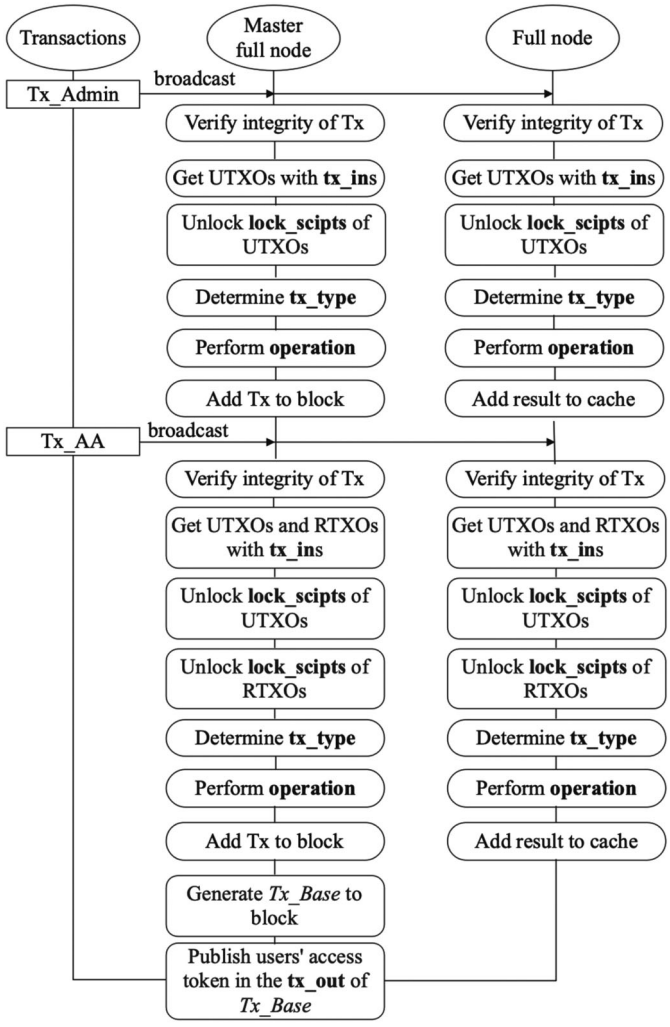


Fig. 5. Runtime verification.

4.3.2 *User Transaction (*Tx_User*)*

Users use *Tx_User* to register their information and apply for access to resources. Therefore, there are two types of transactions under *Tx_User*: *user registration transactions (Tx_UR)* and *access application transactions (Tx_AA)*. Transaction history of a single user may be as shown in Fig. 4 B.

Tx_UR must be the first transaction submitted by each user on the blockchain network. It will initialize basic information of the user, including the default credit value. Only after *Tx_UR* is confirmed by the blockchain network, can the user's identity be recognized. *Tx_UR* has no *tx_in* and only one *tx_out* for the user's next *Tx_User*.

Tx_AA is used to apply to blockchain network for access control rights to resources. It contains at least two *tx_in* and two *tx_out*. One of *tx_in* refers to the UTXO of the previous *Tx_User*, and the remaining *tx_in* refer to the RTXO of a *Tx_SA* which has added mapping of this user to a role in the domain. Number of *tx_out* of *Tx_AA* is the same as *tx_in* . Except for the *tx_out* for the next *Tx_AA*, the other outputs are used for confirmation.

4.3.3 *Accessbase Transaction (*Tx_Base*)*

Tx_Base is used in the consensus phase. After full nodes verify *Tx_AA*, *Tx_Base* is used to publish verification results and users' token for accessing resources. It has only one *tx_in* pointing to the *tx_out* of *Tx_AA* and only one *tx_out*.

4.4 Consensus Layer

Full nodes maintain the blockchain, validate every transaction they received and determine the next block to join the blockchain through the extended PBFT protocol. The protocol mainly divided into *runtime verification phase*, *consensus phase*.

4.4.1 *Runtime Verification Phase*

Every transaction generated by users or domain administrators will be broadcast to all full nodes. Each full node verifies received transactions and caches results to record whether these transactions are legal. The process is shown in Fig. 5, and the specific operation is as follows:

1) Verify integrity of the transaction with *tx_hash*;
2) Get UTXO or RTXO by *prev_hash* and *n* in *tx_in* of the transaction;
3) Try to use *unlock_script* to unlock *lock_script* of UTXO or RTXO. If unlock fails, the transaction is invalid and its status is be cached by full node. Otherwise, proceed to the next step;
4) Determine *tx_type* and perform *operation* of *tx_out* from the transaction. If the opcodes executed by full node does not match permission of *tx_type*,

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

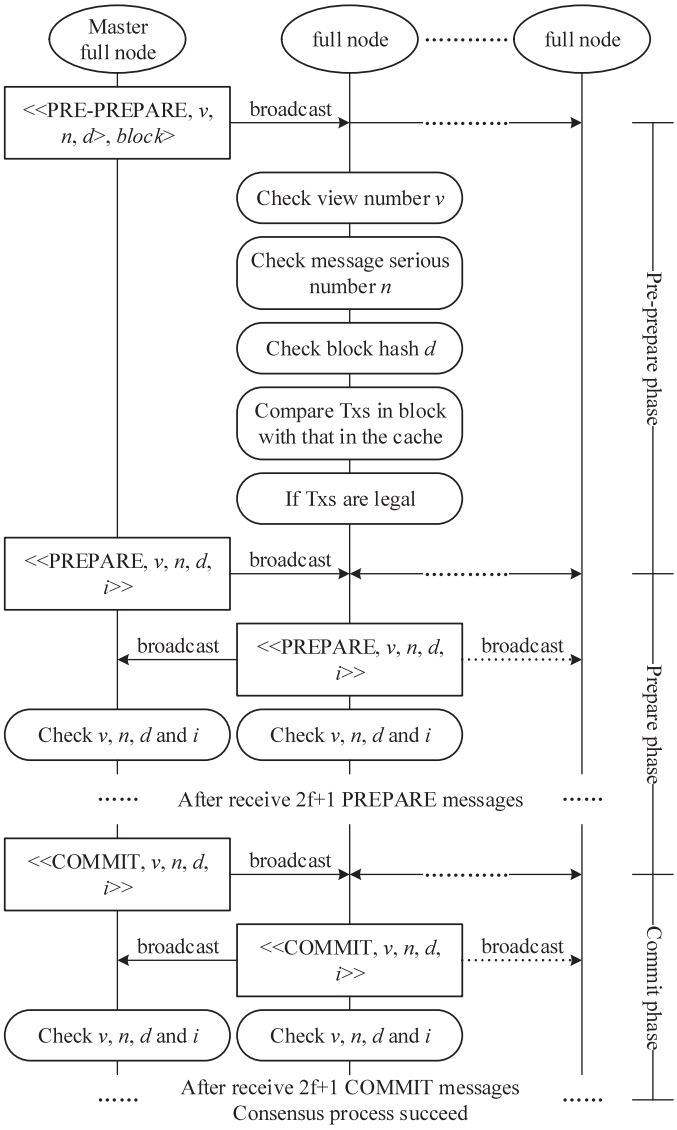


Fig. 6. Consensus phase.

operation will fail and the transaction will be considered a failure.

For the master full node, in addition to performing the above process, it also needs to generate *Tx_Base*. If access request of *Tx_AA* is valid and verified, then master full node will generate a *Tx_Base* and publish users' access token in the *tx_out* of *Tx_Base*.

4.4.2 Consensus Phase

PBFT algorithm introduced in 3.2 is used to synchronize new blocks between all full nodes. In *prepare* phase of used PBFT algorithm, all full nodes except the master full node will receive the block generated by the master full node. These full nodes will verify whether the transactions on the block successfully pass the verification in the *Runtime Verification Phase*. According to PBFT algorithm, only when more than 2/3 of all full nodes accept the block, can it be recognized by all nodes and join the blockchain. The consensus process is shown in Fig. 6.

4.5 Contract Layer

There are two types of scripts in the proposed system: (1) scripts used in *lock_script*, **use_script* and *unlock_script* to specify who can use the UTXO or RTXO, and (2) scripts used in *operation* to apply required operations. For the former type of scripts, the proposed system directly uses the script pattern of Bitcoin script system, such as P2PKH (Pay-to-Public-Key-Hash), P2PK (Pay-to-Public-Key), etc [18]. For the latter type of scripts, they will be parsed by full nodes and stored on blockchain as operation records. There are different opcodes that can be used for different *tx_type* are as follows:

- `op_user_role`: used only in *Tx_Admin* to add a role to a user;
- `op_role_right`: used only in *Tx_Admin* to grant the corresponding right to a role;
- `op_role_role`: used only in *Tx_Admin* to map a role from other access control domain to a role from local;
- `op_remove`: used only in *Tx_Admin* to remove a past operation;
- `op_register_domain`: used only in *Tx_Admin* to register a new access control domain;
- `op_register_user`: used only in *Tx_User* to register a new user;
- `op_authorize`: used only in *Tx_Base* to issue tokens to users who submit *Tx_AA*;
- `op_effective_time`: specify the effective time of the operation;
- `op_expiration_time`: specific the expiration time of the operation.
- `op_access_request`: used only in *Tx_User* to request access to resources.
- `op_condition`: add a condition, the instruction can only be executed when the condition is met.

As shown in Fig. 7, we describes the use of these opcodes in scripts for different *tx_type*.

4.5.1 Scripts for Administrator Transactions

Administrators manage access control rules through Scripts 1 to 5. They can specify other conditions that users need to meet when requesting access in $\langle condition \rangle$ of Script 2. Validity period of each token obtained by user is also set in $\langle condition \rangle$.

4.5.2 Scripts for User Transactions

Users need to register on blockchain via Script 6. After that, they can submit *Tx_AA* via Script 7. In order to generate $\langle token \rangle$, IP address and information needed to establish communication between user and accessed device are encrypted using public key of the device. After *Tx_AA* is verified by full nodes, $\langle token \rangle$ in Script 7 is directly copied to Script 8.

4.5.3 Scripts for Accessbase Transactions

There is only one kind of script for *Tx_Base*, and its form is as Script 8. `op_effective_time` and `op_expiration_time` specify the validity period of the operation, that is the time frame within which the token can be used.

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

op_register_domain (1)
op_user_role <*user_addr*><*role*> op_condition <*condition*> op_effective_time <*time*> (2)
op_role_right <*role*><*object_addr*><*resource*><*right*> op_effective_time <*time*> (3)
op_role_role <*role*><*outer_domain_addr*><*outer_role*> op_effective_time <*time*> (4)

op_remove op_effective_time <*time*> (5)
op_register_user (6)
op_access_request <*object_addr*><*resource*><*operation*> (7)

op_authorize <*token*> op_effective_time <*time*> op_expiration_time <*time*> (8)

Fig. 7. Virtual machine instructions.

4.6 Application Layer

In the application layer, access policy management, cross-domain access control, behavior audit and blockchain wallet are implemented based on the design of previous layers.

4.6.1 Access Policy Management

According to the cross-domain RBAC model we designed, three types of relationships (SA_i, OA_i and CA_i) need to be expressed as policies. Each operation of administrator on access control policy is translated into *operation* of *tx_out* of *Tx_Admin* and administrators can use *Tx_SA*, *Tx_OA* and *Tx_CA* to add these relationships to blockchain. After *Tx_Admin* is confirmed, the *tx_out* containing these *operation* will become UTXO or RTXO. Finally, each rule of access control policy will correspond to a UTXO or RTXO, and when a rule is to be removed, the administrator can submit a *Tx_RM* to use and consume the corresponding UTXO or RTXO.

4.6.2 Cross-Domain Access Control

Tx_AA submitted by user needs to use RTXO of *Tx_SA*. After receiving the *Tx_AA*, full nodes first determine whether the transaction is valid. They verify whether *unlock_script* can unlock *lock_script* or **use_script*. Through RTXO used by *Tx_AA*, full nodes can determine the access control domain D_1 and role d_1r_1 to which the user belongs. Then full nodes check *operation* and use $\langle object\_addr \rangle$ and $\langle resource \rangle$ in it to find the UTXO with the role d_1r_1 and right added. If the user requests access to device from other access control domain D_2, then full nodes need to find UTXO that added the mapping between role d_1r_1 and role d_2r_2 from D_2, and look for UTXO with the role d_2r_2 and right added. The request is invalid If no valid UTXO is found, or if the user requested permission do not match the permission added in UTXO. Otherwise, master full node will generate *Tx_Base* to publish $\langle token \rangle$.

The user accesses a device through the $\langle token \rangle$ published on blockchain. The accessed device only needs to query *Tx_Base* on blockchain, and confirm whether the $\langle token \rangle$ has been issued. If so, the device can trust the user and establish a connection with it through information in $\langle token \rangle$ to provide access to resources.

4.6.3 Behavior Audit

All transactions of users and administrators must index the specific *tx_out* of the previous transaction in *tx_in* of this transaction. Therefore, All transactions submitted by the user or administrator can be traced through the latest transaction. Consequently, administrators can audit and analyze past behavior of users to adjust the access control policy. Besides, it is possible to generate access authorization and request logs for a subject or an object within a period by tracing the blockchain data forward.

4.6.4 Blockchain Wallet

Blockchain wallet serves as the entrance to the blockchain. Users can use the wallet to obtain all UTXO and RTXO related to them so that they can aware of what resources they have access to. For the accessed IoT device, it only needs to extract the token related to itself from *Tx_Base* and store it in the wallet after receiving the block broadcasted by full nodes.

5 Analysis

In this section, we first analyze the performance of the proposed system on IoT devices. Next, we will illustrate how the proposed system achieves our design goals.

5.1 Performance Analysis

We mainly analyze the performance of proposed system from computing, storage and networking. We use a Windows PC to simulate 4 full nodes to form a blockchain network, use a MacBook Pro to simulate two gateways as the entrance of the IoT terminals to access the blockchain network, and use 2 Raspberry Pis to simulate 8 IoT terminals. That is, we open four processes on the Raspberry Pi, which occupies four different ports, and runs four node programs simultaneously. Experimental platform is shown in Table 4, and the experimental architecture is shown in Fig. 8.

5.1.1 Computing Performance

In the proposed system, IoT devices need to consume a lot of computing resources only when running cryptographic algorithms. Therefore, we use Raspberry Pi 4B to test various cryptographic schemes to evaluate the performance of the proposed system on IoT devices. Performance results of encrypting and decrypting 512 bytes of data using the mainstream symmetric encryption algorithm are shown in Table 5. When using ECC algorithm for key exchange, performance of encryption and decryption is shown in Table 6. For the performance of ECC algorithm on IoT devices, Max Mössinger *et al.* [23] also quantified consumption of ECC

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

TABLE 4
Experiment Platform

Platform	Feature
Windows PC	Processor: Intel(R) Core(TM) i7-8700 Base Frequency: 3.20 GHz Turbo Frequency: 4.60 GHz Memory: 16 GB
MacBook Pro	Processor: Intel(R) Core(TM) i7-6700HQ Base Frequency: 2.60 GHz Turbo Frequency: 3.50 GHz Memory: 16 GB
Raspberry Pi	Processor: Broadcom BCM2711 Frequency: 1.5 GHz Memory: 4 GB

signatures on an ARM-based device and the results show that IoT devices can afford this work.

In addition, we generated 4000 transactions on the experimental platform, and each IoT terminal processed 500 transactions. We finally calculated the average processing time of IoT terminal to generate a transaction and verify an access request. The time data obtained from 4000 experiments is shown in the histogram Fig. 9.

According to evaluation results, it can be concluded that mainstream secure asymmetric cryptography algorithms have excellent performance on the tested platform. In the experimental environment, the average time to generate a transaction is 5.12ms, the maximum is no more than 11ms, and the average time to verify a transaction is 3.67ms, and the maximum is no more than 8ms. The proposed system has good performance on IoT terminals in this experimental environment.

Such performance results benefit from our design to minimize the complex calculations of IoT terminals. When IoT device is the accessed one, it only needs to confirm whether the token presented by the visitor is included in the *Tx_Base* on blockchain to determine whether the visitor is legitimate. The token contains visitor's IP and the data needed to establish communication, and is encrypted using hybrid encryption scheme that can only be decrypted by the accessed device. The hybrid encryption algorithm uses a randomly generated key to encrypt the message, and then processes the key using recipient's public key. The processed information can only be decrypted by recipient's private key. Therefore, it can be considered that an attacker cannot use other user's token to access resources. In this case, only the process of decrypting the token requires a lot of computing resources. When IoT devices act as visitor, the processes that need to consume computing resources are to use hybrid encryption scheme to encrypt the token, and to sign the transaction in *unlock_script*.

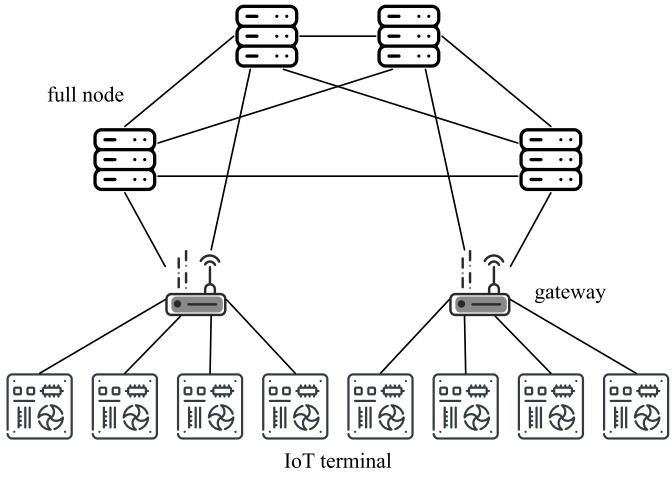


Fig. 8. Experiment architecture.

TABLE 5
Symmetric Encryption Performance

Algorithm	Encryption	Decryption
AES with 128bit key	62.50 μs	63.43 μs
AES with 192bit key	61.71 μs	63.06 μs
AES with 256bit key	62.35 μs	63.66 μs
3-DES with 64bit key	58.56 μs	58.64 μs
3-DES with 128bit key	58.56 μs	59.28 μs
3-DES with 192bit key	59.13 μs	59.94 μs
Camellia with 128bit key	63.97 μs	63.83 μs
Camellia with 192bit key	62.68 μs	62.85 μs
Camellia with 256bit key	62.29 μs	63.38 μs
chacha20 with 256bit key	56.74 μs	56.20 μs

5.1.2 Storage Performance

Although blockchain data will become larger and larger over time, IoT devices that are not blockchain maintainers do not need to store this data. With blockchain wallets, they only need to store a small amount of data related to themselves. As the party requesting access, they just need to store the UTXO and RTXO that they can use, and as the party being accessed, they only need to store the tokens related to themselves issued by *Tx_Base*.

Therefore, storage resources consumed by the proposed system on IoT devices are related to the number of roles that a device owned and the number of authorized tokens. According to our tests, 200kb of data can store more than 400 UTXOs or RTXOs and 100 tokens. Hence, IoT devices can easily withstand the storage resource consumption of the proposed system.

5.1.3 Network Performance

The main factors affecting the network performance of proposed system are the performance of gateways and the delay caused by consensus process of blockchain network.

TABLE 6
Asymmetric Encryption Performance

Elliptic Curve	Encryption	Decryption
SECP256K1	185 μs	180 μs
SECP192R1	180 μs	177 μs
SECP224R1	181 μs	178 μs
SECP256R1	185 μs	175 μs
SECP384R1	188 μs	181 μs
SECP521R1	208 μs	197 μs

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

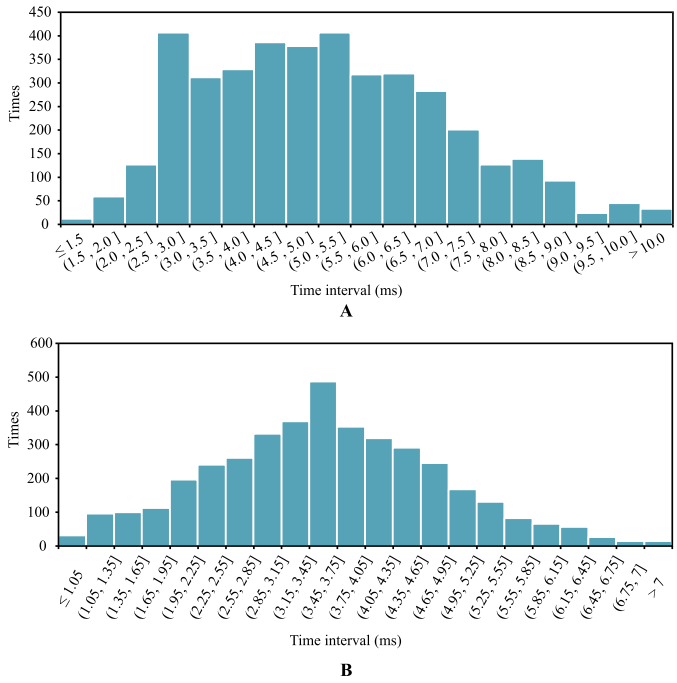


Fig. 9. Time histogram. A is the result of generating transactions, B is the result of verifying transactions.

Our previous works have verified in a specific experimental environment that the blockchain gateway has a network throughput of 27MB/s and can stably provide services for more than 10,000 IoT terminals [24]. The design of blockchain gateway achieves good network performance.

Under the experimental Architecture shown in the Fig. 8, the average network delay generated by each consensus process is 4.58ms. To evaluate the performance of PBFT consensus algorithm used, we continuously add nodes on the Windows PC, set the maximum network delay of the simulation to 5ms, generate 100 blocks under each condition, and calculate the average time, as shown in Fig. 10.

The experimental results point out that the mean time to reach a consensus increases exponentially when the nodes continue to grow. When the number of peers reaches 100, the mean time to reach a consensus in the experimental environment is close to 4s.

In the consortium chain scenario that does not require large-scale miner nodes, the PBFT consensus algorithm used by the proposed system is sufficient and can be applied in large-scale IoT scenarios under the deployment of blockchain gateways. But if there is a requirement for the number of miner nodes, then the consensus algorithm can be replaced on the basis of this scheme. There are also many researches aimed at the scalability of the PBFT algorithm [25], [26], [27], [28].

5.2 Functional Analysis

Next, we evaluated the proposed system for each design goal mentioned in Section 4.2.

5.2.1 Reliability and Availability

Blockchain network's distributed service architecture and compromise resistance consensus algorithm ensures that

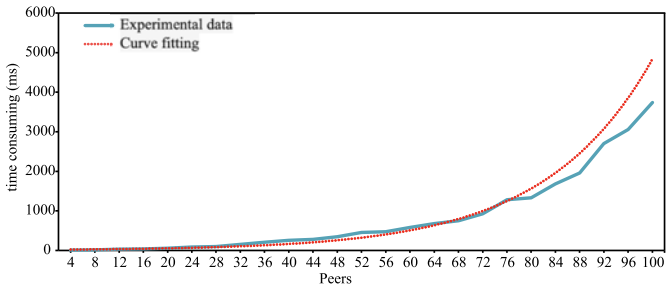


Fig. 10. PBFT algorithm scalability test.

the proposed system can provide services stably for all devices when less than $1/3$ of the full node rebellion. Hence, the more full nodes that maintain the blockchain network, the higher the reliability and availability of the proposed system. However, it will also make the consensus process consume more time.

5.2.2 Confidentiality and Integrity

Cryptographic technologies such as hash functions and digital signatures used by blockchain protect the confidentiality and integrity of data stored on blockchain. *unlock_script* of the submitted transaction contains a digital signature of this transaction calculated using the private key of the submitter, which ensures the integrity of the transaction.

5.2.3 Lightweight

As analyzed in Section 5.1, the proposed system is resource-friendly in design, and IoT devices can withstand the resource consumption during system operation.

5.2.4 Security

According to the compromise resistance PBFT consensus algorithm, the data of blockchain is secure in the case of less than $1/3$ of nodes rebelling. As the number of full nodes that maintain the blockchain increases, the number of malicious nodes that can be resisted also increases, and the blockchain network will be more secure. However the efficiency of the consensus process will also decrease.

Attackers cannot bypass the proposed system because the accessed devices only determine whether the visitor has permissions based on whether the token is on blockchain. Therefore, our proposed system is secure and reliable under the premise that blockchain network is not damaged.

unlock_script contains a digital signature of the transaction, so an attacker cannot forge the identity of another entity unless it obtains their private key. Therefore, the identity of the entity will not be forged it the private key is node leaked.

To resist man-in-the-middle type attacks, we uses tokens to access objects and then transmits data using the keys exchanged in tokens for encryption to protect data confidentiality.

5.2.5 Scalability

PBFT consensus algorithm limits the number of full nodes that can maintain the blockchain, and the communication

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

complexity of the original PBFT consensus mechanism is $O(n^2)$. It supports about 100 full nodes, and further increasing the number will significantly affect the efficiency of consensus process. However, there are many studies devoted to improving the original PBFT. For example, M. Yin *et al*. proposed Hotstuff consensus mechanism, which reduces the communication complexity to $O(n)$ based on achieving Byzantine fault tolerance [26]. The proposed scheme uses blockchain gateways to provide services for IoT devices, which can reduce the pressure on full nodes and significantly enlarge the number of devices that the proposed system can accommodate. The scalability of this architecture was verified in our previous work [24]. We also consider that if data is requested from a sensor, it needs to be processed and sent while keeping data collection, also known as so-called task-parallel processing. The above proof of scalability can prove that our scheme can solve multitasking problems in IoT devices.

6 Conclusion and Future Work

After describing our proposed blockchain-based cross-domain access control system and the associated performance evaluation in terms of computational, storage and network overheads, one can observe that our proposed system is sufficiently lightweight for deployment in IoT environments. Specifically, in our approach to achieve cross-domain access control, a cross-domain role and condition-based access control model is deployed, and each rule of the access control policy is parsed into operations stored in UTXOs or RTXOs. Administrators manage these UTXOs and RTXOs using blockchain transactions to achieve policy management. Both administrators' operations and users' access requests are confirmed by full nodes, and synchronized in all full nodes using a compromise resistance consensus algorithm (i.e., PBFT). Findings from the system evaluation also demonstrated that the proposed system can provide services at a stable level for all devices when less than $1/3$ of full nodes are compromised (i.e., *Reliability and Availability*). Besides, we also proposed an intelligent mode, which can be applied to access control under exceptional situations and evaluation request access nodes. We also demonstrated the security of our proposed approach.

There are, however, a number of potential extensions to this work. For example, we plan to introduce an incentive management module to the system to encourage nodes other than the full nodes to oversee the consensus process, as well as implementing a working prototype of the extended system in a real-world setting. Besides, we also expect that our method can learn normal modes for access control under exceptional situations or obtain trust evaluation of access requesting nodes through intelligent learning and token accumulation mechanism. First, through the token accumulation mechanism, intelligent learning, the credibility of the tokens is graded according to the total number and divided into three levels: high level, medium level, and low level. Second, intelligent learning can extract the common characteristics of high trust nodes. A feedback mechanism can be combined to generate an evaluation model and a feedback mechanism.

References

[1] N. Y. Philip, J. J. P. C. Rodrigues, H. Wang, S. J. Fong, and J. Chen, "Internet of Things for in-home health monitoring systems: Current advances, challenges and future directions," *IEEE J. Areas Commun.*, vol. 39, no. 2, pp. 300–310, Feb. 2021.

[2] M. Shirer and C. MacGillivray, The growth in connected IoT devices is expected to generate 79.4ZB of data in 2025, according to a new IDC forecast, Jun. 2019. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS45213219

[3] Y. Liu, X. Guan, Y. Peng, H. Chen, T. Ohtsuki, and Z. Han, "Blockchain-based task offloading for edge computing on low-quality data via distributed learning in the internet of energy," *IEEE J. Areas Commun.*, vol. 40, no. 2, pp. 657–676, Feb. 2022.

[4] D. Minoli and B. Occhiogrosso, "Blockchain mechanisms for iot security," *Internet Things*, vol. 1, pp. 1–13, 2018.

[5] P. Zhang, J. K. Liu, F. R. Yu, M. Sookhak, M. H. Au, and X. Luo, "A survey on access control in fog computing," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 144–149, Feb. 2018.

[6] A. Ouaddah, A. Abou Elkalam, and A. A. Ouahman, "Towards a novel privacy-preserving access control model based on blockchain technology in IoT," in *Proc. Europe MENA Cooperation Adv. Inf. Commun. Technol.*, 2017, pp. 523–533.

[7] H. Li, L. Pei, D. Liao, S. Chen, M. Zhang, and D. Xu, "FADB: A fine-grained access control scheme for VANET data based on blockchain," *IEEE Access*, vol. 8, pp. 85 190–85 203, 2020.

[8] G. Ali, N. Ahmad, Y. Cao, M. Asif, H. Cruickshank, and Q. E. Ali, "Blockchain based permission delegation and access control in Internet of Things (BACI)," *Comput. Secur.*, vol. 86, pp. 318–334, 2019.

[9] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K.-K. R. Choo, "Blockchain-based cross-domain authentication for intelligent 5g-enabled internet of drones," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 6224–6238, Apr. 2022.

[10] Y. Ren, F. Zhu, J. Qi, J. Wang, and A. K. Sangaiah, "Identity management and access control based on blockchain under edge computing for the industrial Internet of Things," *Appl. Sci.*, vol. 9, no. 10, 2019, Art. no. 2058.

[11] P. Singh, M. Masud, M. S. Hossain, and A. Kaur, "Cross-domain secure data sharing using blockchain for industrial IoT," *J. Parallel Comput.*, vol. 156, pp. 176–184, 2021.

[12] M. Shen *et al.*, "Blockchain-assisted secure device authentication for cross-domain industrial IoT," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 942–954, 2020.

[13] J. Lopez, J. E. Rubio, and C. Alcaraz, "Digital twins for intelligent authorization in the B5G-enabled smart grid," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 48–55, Apr. 2021.

[14] C. Alcaraz, J. E. Rubio, and J. Lopez, "Blockchain-assisted access for federated smart grid domains: Coupling and features," *J. Parallel Distrib. Comput.*, vol. 144, pp. 124–135, 2020.

[15] X. Jiang, F. R. Yu, T. Song, Z. Ma, Y. Song, and D. Zhu, "Blockchain-enabled cross-domain object detection for autonomous driving: A model sharing approach," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3681–3692, May 2020.

[16] L. Tan, N. Shi, K. Yu, M. Aloqaily, and Y. Jararweh, "A blockchain-empowered access control framework for smart devices in green Internet of Things," *ACM Trans. Internet Technol.*, vol. 21, no. 3, pp. 1–20, 2021.

[17] J. Spasovski and P. Eklund, "Proof of stake blockchain: Performance and scalability for groupware communications," in *Proc. 9th Int. Conf. Manage. Digit. EcoSystems*, ACM, 2017, pp. 251–258.

[18] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*, Newton, MA, USA: O'Reilly Media, Inc., 2014.

[19] A. H. Lone and R. Naaz, "Applicability of blockchain smart contracts in securing internet and IoT: A systematic literature review," *Comput. Sci. Rev.*, vol. 39, 2021, Art. no. 100360.

[20] S. J. Alsunaidi and F. A. Alhaidari, "A survey of consensus algorithms for blockchain technology," in *Proc. Int. Conf. Inf. Sci.*, 2019, pp. 1–6.

[21] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, and A. Rindos, "Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric)," in *Proc. IEEE 36th Symp. Reliable Distrib. Syst.*, 2017, pp. 253–255.

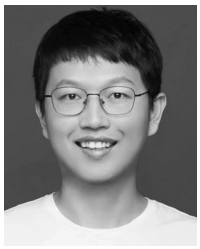
[22] T. Sasaki, Y. Morita, and A. Jada, "Access control architecture for smart city IoT platform," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng.*, 2019, pp. 717–722.

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.

[23] M. Mössinger, B. Petschkuhn, J. Bauer, R. C. Staudemeyer, M. Wójcik, and H. C. Pöhls, "Towards quantifying the cost of a secure IoT: Overhead and energy consumption of ECC signatures on an ARM-based device," in *Proc. IEEE 17th Int. Symp. World Wireless, Mobile Multimedia Netw.*, 2016, pp. 1–6.
[24] S. He, W. Ren, T. Zhu, and K.-K. R. Choo, "BoSMoS: A blockchain-based status monitoring system for defending against unauthorized software updating in industrial Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 948–959, Feb. 2020.
[25] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annu. Tech. Conf.*, 2014, pp. 305–319. [Online]. Available: https://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro
[26] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "HotStuff: BFT consensus with linearity and responsiveness," in *Proc. ACM Symp. Princ. Distrib. Comput.*, 2019, pp. 347–356.
[27] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer PBFT consensus for blockchain," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 5, pp. 1146–1160, May 2021.
[28] Y. Li *et al.*, "An extensible consensus algorithm based on PBFT," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov.*, 2019, pp. 17–23.

Xiaohan Hao received the graduate degree from the School of Computer Science, China University of Geosciences (Wuhan), China, in 2020. She is with Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China. Her research interests include access control, cryptography and blockchain.

Wei Ren (Member, IEEE) received the PhD degree in computer science from the Huazhong University of Science and Technology, China. Currently he is a full professor with the School of Computer Science, China University of Geosciences (Wuhan), China, since 2013. He was with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, USA, in 2007 and 2008, the School of Computer Science, University of Nevada Las Vegas, USA, in 2006 and 2007, and the Department of Computer Science, The Hong Kong University of Science and Technology, in 2004 and 2005. He has published more than 100 refereed papers, 1 monograph, and 4 textbooks. He has obtained 10 patents and 5 innovation awards. He is a distinguished member of the China Computer Federation.



Yangyang Fei received the PhD degree in mathematics from the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China, in 2018. His research interests include quantum information and quantum computation.

Tianqing Zhu received the BEng and the MEng degrees from Wuhan University, China, in 2000 and 2004, respectively, and the PhD degree in computer science from Deakin University, Australia, in 2014. She is currently a professor with China University of Geosciences, Wuhan, China. Prior to that, she was a lecturer with the School of Information Technology, Deakin University. Her research interests include privacy preserving, AI security and privacy, and network security.

Kim-Kwang Raymond Choo (Senior Member, IEEE) received the PhD degree in information security from the Queensland University of Technology, Australia, in 2006. He currently holds the Cloud Technology Endowed professorship with The University of Texas at San Antonio (UTSA). He is the founding co-editor-in-chief of ACM Distributed Ledger Technologies: Research & Practice, founding chair of IEEE TEMS Technical Committee on Blockchain and Distributed Ledger Technologies, an ACM Distinguished Speaker and IEEE Computer Society Distinguished visitor (2021 - 2023), and a Web of Science's Highly Cited researcher (Computer Science – 2021, Cross-Field – 2020). He is the recipient of the 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career researcher), the 2021 UTSA Carlos Alvarez College of Business Endowed 1969 Commemorative Award for Overall Faculty Excellence and the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, the British Computer Society's 2019 Wilkes Award Runner-up, He has also received best paper awards from *IEEE Systems Journal,* in 2021, *IEEE Computer Society's Bio-Inspired Computing Special Technical Committee (STC)* Outstanding Paper Award for 2021, EEE Conference on Dependable and Secure Computing (DSC 2021), *IEEE Consumer Electronics Magazine* for 2020, *Journal of Network and Computer Applications* for 2020. He has received the Outstanding wditor Award for 2021 from Future Generation Computer Systems.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: China University of Geosciences Wuhan Campus. Downloaded on September 06,2023 at 06:17:42 UTC from IEEE Xplore. Restrictions apply.