LBlockchainE: A Lightweight Blockchain for Edge IoT-Enabled Maritime Transportation Systems

Yu Jiang, Xiaolong Xu, Honghao Gao, *Senior Member, IEEE*, Adel D. Rajab, Fu Xiao, *Member, IEEE*, and Xinheng Wang, *Senior Member, IEEE*

***Abstract*— Blockchain can help edge IoT-enabled Maritime Transportation Systems (MTS) in solving its privacy and security problems. In this paper, a lightweight blockchain called LBlockchainE is designed for edge IoT-enabled MTS to guarantee the security of sensor data stored in an edge computing environment. To save the resources of edge servers on ship, a data placement strategy is proposed. To encourage edge servers to positively contribute to storing data generated by sensor devices, storage resource consumption is employed as an influencing parameter, and servers with abundant resources are selected for priority storage. The data placement strategy also takes care of the access delay between servers and selects the nodes with the least access and storage costs as the priority storage choice. LBlockchainE applies the low-energy-consumption characteristics of Proof of Stake to determine the ownership of bookkeeping rights through a small number of competitive calculations and the resources of the node. Experimental results indicate that compared with Ethereum, the consensus mechanism of LBlockchainE consumes less energy and occupies less storage space. On average, the new system uses 1.6% less time and consumes 78% less battery power compared with traditional blockchain systems. In comparison to the random storage, the best storage, and the optimal data storage strategies, the proposed strategy maintains the same message costs.**

***Index Terms*— Edge IoT-enabled MTS, blockchain, edge computing, LBlockchainE, security.**

I. Introduction

THE rapid development of the Internet of Things (IoT) has made the connection between things closer, including agriculture, medical industry, and manufacturing. The number of IoT devices worldwide is forecast to almost triple from 8.74 billion in 2020 to more than 25.4 billion IoT devices in 2030 [1]. The IoT technology also provides an informatization method for maritime transportation. There are various research applications in existing transport systems, including navigation, road condition detection, and autonomous driving [40], [41], [42], [54]. The inclination sensor and acceleration sensor on the ship need to upload the collected data of the inclination angle and acceleration during the running of the ship to the server to monitor the status of the ship and ensure the safety of the ship with the help of cloud computing [2], [3], [4], [43], [44]. However, as the number of IoT devices grows, so does the amount of data produced in the network, making it difficult for cloud computing to handle large volumes of business. In addition, data transmission delays can cause users to have a bad experience. Therefore, edge computing [4], [5] is proposed to provide services to the Maritime Transportation Systems (MTS) that are closed to edge servers and have a short delay. An edge cloud [3], [6], [13] is made up of edge servers in ships that are close together to use server collaboration in order to complete tasks from sensor devices more efficiently and in real time. Sensor devices that rely on distributed edge servers owned by various enterprises and suppliers are dispersed and have limited processing capabilities. Due to the abundance of maritime information, MTS was hit by cyber-attacks [42]. In the year 2020, the Mediterranean Shipping Company was hit by a cyber-attack.

A blockchain [7], [21], [23] composed of blocks in a chain structure has strong immutability and resistance to tampering by malicious nodes, which is similar to a distributed database. Blockchain technology can ensure that every node in the blockchain has the right to verify the block via decentralization. However, blockchain issues have prompted many scholars in academia to discover new research directions and propose their improved blockchain architecture [8], [9], [10], [11].

The edge-IoT architecture [12] meets the deployment of blockchain so that the blockchain can play an important role in guaranteeing data security. However, the sensor devices cannot bear the operation of the blockchain, and the blockchain can be deployed on the edge server. In order to save the resources of the edge server and use more resources for the services of edge devices, it is necessary for MTS to develop a lightweight blockchain, which can not only retain the tamper proof ability of the blockchain, but also save more computing and storage resources.

Manuscript received 30 October 2021; revised 3 January 2022 and 18 February 2022; accepted 3 March 2022. Date of publication 16 March 2022; date of current version 8 February 2023. This work was supported by the National Natural Science Foundation of China under Grant 62072255. The Associate Editor for this article was A. K. Bashir. *(Corresponding authors: Xiaolong Xu; Honghao Gao.)*

Yu Jiang is with the Jiangsu Key Laboratory of Big Data Security Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: 1219043720@njupt.edu.cn).

Xiaolong Xu and Fu Xiao are with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: xuxl@njupt.edu.cn; xiaof@njupt.edu.cn).

Honghao Gao is with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China (e-mail: gaohonghao@shu.edu.cn).

Adel D. Rajab is with the College of Computer Science and Information Systems, Najran University, Najran 61441, Saudi Arabia (e-mail: adrajab@nu.edu.sa).

Xinheng Wang is with the School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China (e-mail: xinheng.wang@xjtlu.edu.cn).

Digital Object Identifier 10.1109/TITS.2022.3157447

1558-0016 © 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: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

In this paper, a lightweight blockchain for edge IoT-enabled MTS (LBlockchainE) is proposed to address security problems in the edge IoT-enabled MTS. The data generated by the sensor devices on the ship is stored in the edge server supported by LBlockchainE so that the data are not tampered with and the MTS have higher privacy and operability. The main contributions of this paper are summarized as follows:

1) A lightweight blockchain called LBlockchainE is proposed for edge IoT-enabled MTS. To select suitable nodes for storing data, a data location strategy takes into account the number of storage resources in the server and the access cost between edge servers. To save more computing and storage resources, a new Proof of Stake (PoS) suitable for edge IoT-enabled environments, which is combined with competition mechanism, is proposed to save resource consumption.

2) LBlockchainE is used in the edge IoT-enabled service composition to guarantee the reliability of the Quality of Service (QoS) with the assistance of step factor modification and the use of the cotangent perturbation to improve the firefly algorithm (FA) to balance local and global searches, eventually obtaining the optimal solution.

3) In edge IoT-enabled MTS that is being built, a lightweight blockchain platform is realized. The computing clusters that make up the edge IoT-enabled experimental platform are divided into three layers: data access, data placement, and consensus. The data access layer establishes a peer-to-peer network and realizes the client access to the blockchain network, as well as block synchronization and other services. The data placement layer implements the data location strategy designed to store data in the edge IoT-enabled MTS. The consensus mechanism in the consensus layer is based on the PoS improvement, which involves determining the right to pack block based on the storage contribution of the edge server.

The remainder of this paper is organized as follows: Section II discusses related works. Section III proposes a lightweight blockchain called LBlockchainE. Section IV describes the LBlockchainE-based prototype system. Section V discusses the experiments and security analysis. Finally, Section VI concludes the paper by summarizing the main contributions of this paper and commenting on future work.

II. Related Work

A. IoT and MTS Security

In recent years, many security and privacy issues have emerged as a result of large-scale devices in IoT. In [14], [15], authors introduced the security challenges and privacy issues in IoT without discussing the solutions to these challenges. Alaba *et al.* [16] discussed the privacy issues in the smart grid and the main security vulnerabilities in the IoT. Al-Garadi *et al.* [17] discussed IoT security threats that are related to inherent or newly introduced threats are presented, and various potential IoT system attack surfaces and the possible threats related to each surface. Some researchers have proposed some solutions to security problems. Sicari *et al.* [18] studied the solutions of access control and trust management in IoT. Al-Garadi *et al.* [17] provided a comprehensive survey of ML methods and recent advances in DL methods that can be used to develop enhanced security methods for IoT systems. Alzahrani *et al.* [19] proposed an access control solution for Constrained Application Protocol (CoAP)-based IoT services. Kouicem *et al.* [20] discussed the benefits of new methods, such as blockchain and software-defined networking, to the security and privacy of the IoT. Gao *et al.* [45] found that malicious nodes may intercept or discard data packets, which might interfere with the transmission process and cause privacy leakage.

TABLE I
METHODS TO SOLVE THE SECURITY PROBLEMS

Reference	Methods
[51]	Access control, trust management
[23]	Machine learning, deep learning
[55]	Security information sharing, identity verification
[41-45]	Blockchain
[56]	Relay collaboration
[54]	Transforming business process execution language into a timed automaton

The analysis of MTS security has undoubtedly received considerable attention as presented in Table I. Huang *et al.* [46] analyzed that the existing IoT-enabled MTS are prone to single points of failure and malicious attacks, and cannot provide stable services. Teixeira *et al.* [26] simulated a combination of different attack scenarios, such as false data injections and zero-dynamics attacks. Mohanta *et al.* [47] analyzed that due to low processing power and storage capacity, smart things are vulnerable to attacks because existing security or encryption technologies are not applicable. Gao *et al.* [48] proposed an automatic method of transforming Business Process Execution Language into timed automata for formal verification, with the aim of formalizing timed privacy requirements for the IoT service composition. Gupta *et al.* [49] has developed an identity-based secure information sharing scheme for MTS to maintain proper identity verification measures. Liu *et al.* [50] explored the safety performance of MTS based on digital twins and IoT. This paper aimed to use blockchain technology to address MTS security concerns.

B. Blockchain

Blockchain technology has been widely used in cryptocurrency since the emergence of Bitcoin [7]. Ethereum [21] expands the application fields of the blockchain that deploys smart contracts to construct various decentralized applications (DApp). A blockchain is a distributed ledger made up of blocks in a chain structure with the connection between them maintained by storing the hash value of the previous block. Each block in the blockchain packs transactions within a certain period. The blockchain has so far not only acted as a platform for cryptocurrency transactions but has also been widely used in data security and DApp, deriving many scenarios and applications [22]. Fabric, a Hyperledger subproject for enterprises, is a blockchain development platform provided by Hyperledger [23], which is affiliated with the Linux

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

Foundation. In recent years, the project has become a widely used enterprise-level programmable blockchain, on which many leading Internet companies developing blockchain applications were based. IOTA [24] was proposed as the directed acyclic graph structure to improve the speed and scalability of confirming transactions in the blockchain. IOTA is expected to solve the inefficiencies of traditional blockchains and provide a new direction for blockchain development as a typical representative of the third-generation blockchain.

Each node in the blockchain is required to store complete ledger data so that the node can verify the data in each block anytime. A newly generated block must be broadcast on the network, according to the blockchain's operation mechanism. Each node will store more blocks as the blockchain's running time increases, taking up more storage space and incurring higher transmission costs. Therefore, block storage overhead should not be ignored. The reduction of the transmission and storage costs of blocks has become a hot topic in the current blockchain research. To allow nodes to form a community, Xu *et al.* [8] proposed a consensus unit method in which each autonomous fragmentation stores block data. However, for public chains, the method has a significant query overhead. By regularly creating snapshots of the blockchain, Marsalek *et al.* [9] proposed a compressible blockchain architecture that can reduce the amount of data required to be downloaded and stored. Dorri *et al.* [10] proposed a scalable blockchain for the IoT, which applies a distributed trust method for reducing the processing overhead for verifying new blocks. Xu [11] proposed a new blockchain protocol that can not only reduce storage overhead but also withstand large-scale node drops.

C. Integration of Blockchain and MTS

The deployment of blockchain systems is complicated by the limited resources available in edge IoT-enabled MTS. These resources need not only to support related application services but also cache data frequently accessed by sensor devices, making it impossible to require each node to store complete blockchain data. Multiple copies of data are usually placed in multiple data centers in cloud computing to respond to a large number of users' requests. Therefore, access delays, load balancing between data centers, data center energy consumption, and network transmission between data centers need to be taken into account. To reduce the overhead of block transmission and storage, the blockchain system in edge computing should optimize and improve the blockchain.

Poor interoperability, device resource limitations, and privacy leakage are all risks associated with the IoT system. Some studies have looked at how blockchain and IoT can work together to improve interoperability and security in the IoT system [27]. Dai *et al.* [28] applied blockchain to the medical IoT and proposed solutions to combat COVID-19 and solve the medical IoT system's privacy issues. Yang *et al.* [29] proposed a decentralized trust management system for vehicle networks based on blockchain technology, which applies blockchain to save the trust value of vehicles and combines Proof of Work (PoW) and PoS to maintain a reliable and consistent blockchain. Biswas and Muthukkumarasamy [30] proposed a secure IoT framework for smart cities; it integrates blockchain and smart devices to provide a secure communication platform. Dorri *et al.* [31] proposed a lightweight blockchain suitable for the IoT and tested the method in smart homes. Shen *et al.* [51] presented an efficient block-chain-assisted secure device authentication mechanism BASA for cross-domain Industrial Internet of Things to preserve the privacy of devices.

Liao *et al.* [34] proposed a blockchain-enabled intelligent task offloading strategy that mitigates various types of attacks and increases the credibility of network entities by quantifying the task offloading successful events. Munusamy *et al.* [35] designed an edge-centric framework for analyzing the real-time data at the edge of the networks with minimum latency and power consumption while meeting the security and privacy issue of MTS. Li *et al.* [37] proposed a blockchain-based collaborative edge knowledge inference (BCEI) framework for edge-assisted multi-robot systems, providing trusted edge collaborative inference in the presence of malicious nodes. Feng *et al.* [38] proposed a blockchain-enabled efficient and secure data sharing model, applying blockchain and attribute-based encryption to ensure the security of instruction issues and data sharing, and coping with the drone's open and untrusted environment. Dibaei *et al.* [39] explored cybersecurity defense mechanisms based on machine learning and blockchain against cyber-attacks in the vehicle network. Shen *et al.* [52], [53] solved the problem of protecting data privacy in the Vehicular Social Networks and smart cities by using the combination of blockchain and support vector machines.

The above-mentioned researches only focus on the application of blockchain in the IoT to solve the cyber-attacks and security issues, but do not take into account the problems of the blockchain. The deployment of the blockchain requires high storage and computing capabilities. Therefore, it is necessary to design a blockchain that is suitable for deployment in the edge IoT-enabled environment, especially MTS. This paper aims to explore the design of a lightweight blockchain suitable for edge IoT-enabled MTS to solve the security problems in the system.

III. System Model and Problem Formulation

A. System Model

Edge computing is derived from the content distribution network, in which edge servers are deployed near the edge devices to provide services to reduce data transmission delay and can meet the requirements of real-time equipment. To form the edge IoT-enabled MTS architecture, MTS relies on edge computing to process collected data and make instant decisions [12].

The architecture of edge IoT-enabled MTS is presented in Fig. 1. Edge IoT-enabled MTS contains a three-tier architecture, namely, core cloud, edge cloud, and sensor devices. An edge cloud is composed of an edge orchestrator that manages the collaborative work in the edge cloud and interacts with other edge clouds and many edge nodes in ship that

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

handle client tasks. When an edge cloud has difficulty in processing tasks, it will access the core cloud via core networks and rent computing and storage resources to solve the overload problem [13]. The server can request data from other edge servers if the data requested by the devices does not exist on the adjacent server. To summarize, all edge servers can work together to complete tasks that are dependent on distance to meet real-time application requirements.

B. A Lightweight Blockchain Called LBlockChainE

In the edge IoT-enabled MTS, sensor or smart devices collect data and then upload them to the edge server on ship nearby. The edge server accepts access from devices without guaranteeing the reliability of the data storage environment. To enhance data security and integrity, blockchain technology is employed as a reliable tool for data. The lightweight blockchain-designed workflow for the edge IoT-enabled MTS is presented in Fig. 2. Data is uploaded to the edge cloud, and metadata is generated. The data location strategy helps the edge cloud select appropriate edge nodes, and the metadata item stores the data storage information. Each edge node participating in the blockchain network competes for the rights to pack metadata items into blocks.

In LBlockChainE, each node participating in packing blocks is assigned a pair of keys in the system as a unique identification. The public key is distributed to the public and simultaneously generates the node's account, on which the transactions of other nodes to the node are based. The private key used to encrypt data can act as a digital signature, ensuring the data's integrity or verifying the sender's identity. When the node packs the block successfully, it will leave its signature in the block. The packing node will receive the block reward after the other nodes in the network verify that the block is correct. The miner's reward is set to an "edge cloud coin" for simplicity's sake, and the edge cloud coin will be used in the following section.

The structure of the block in LBlockchainE is presented in Fig. 2. The hash, timestamp, index, Merkle tree root of the previous block, and the mining difficulty (which is the mathematical difficulty) are all contained in the block header. The block body holds all of the block's transaction data. The metadata is stored in the block body in the edge IoT-enabled MTS, which will be covered in greater detail in the next section. The current block's hash is added to the block to ensure connectivity between them. Furthermore, each block saves the storage node's location and the data associated with it.

C. Problem Formulation

The entire data must be saved by all nodes participating in the blockchain. The edge server in the edge cloud must store not only the data generated by the devices but also various network data caches. Contrary to the core cloud's powerful data center, the edge server's limited resources cannot support each node saving all of the data. Therefore, the data should be stored in nodes selected by the storage strategy. To reduce the cost of data storage, metadata composed of multiple attributes containing the basic information of the data item [32] is proposed. Metadata is lighter and simpler compared with data items. Following the data, a corresponding metadata is generated, which is used to replace data items and packed into blocks to reduce the block storage overhead. Instead of saving copies in all nodes, it is necessary to select appropriate nodes to store the data uploaded by edge devices. As a result, to reduce data placement overhead and make network transmission easier, an efficient data placement strategy is required.

The server's long-term operation will consume a significant amount of power, which can be attributed to the server's storage cost. When the edge cloud responds to a client or cloud request that they need corresponding data, it will request other servers that store data. The transmission of data between servers will occupy network bandwidth, which results in transmission costs. Some nodes affiliated with different manufacturers are unwilling to provide their resources to store data copies. Relevant parameters are set to incentivize resource-rich nodes to ensure the load balance of each node in the edge cloud. A data placement strategy that takes power consumption and network transmission costs into account is proposed and sets a balanced cost to encourage nodes with more remaining resources to store.

In this scenario, the blockchain is installed on each edge server in the edge cloud. Some nodes should be selected for data storage. The power consumption cost (BC) is then calculated. To meet clients' access at any time, the server must be running at all times, which consumes a lot of power. The BC of the node is defined as follows:

$$bc_i = a_i \mu_i + b_i \tag{1}$$

where a_i denotes the positive factor representing the power consumption of the edge server es_i in the peak; μ_i, the operating rate of the edge server es_i; and b_i, the power consumption of the server es_i in the idle state.

The access overhead that other servers incur when they access data that they do not have is known as the network transmission cost (AC). The network transmission cost is as follows:

$$a_{ij} = \begin{cases} 0, & i = j \\ link(i, j) + hop(i, j), & i \neq j \end{cases} \tag{2}$$

where $link(i, j)$ denotes the data transmission delay between the edge servers es_i and es_j, and $hop(i, j)$ denotes the route propagation delay between the edge servers es_i and es_j.

The cost of network transmission is determined by the length of the connecting link and the number of hops forwarded by the router because edge servers are connected by optical cables. When the two nodes are identical, the network transmission cost is 0. When the two nodes are different, the network transmission cost depends on the link length of the connection corresponding to the data propagation delay and the number of hops, which corresponds to the router's data transmission delay. Both of them that occupy a major position in the total delay of the entire data cost is considered in this blockchain.

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

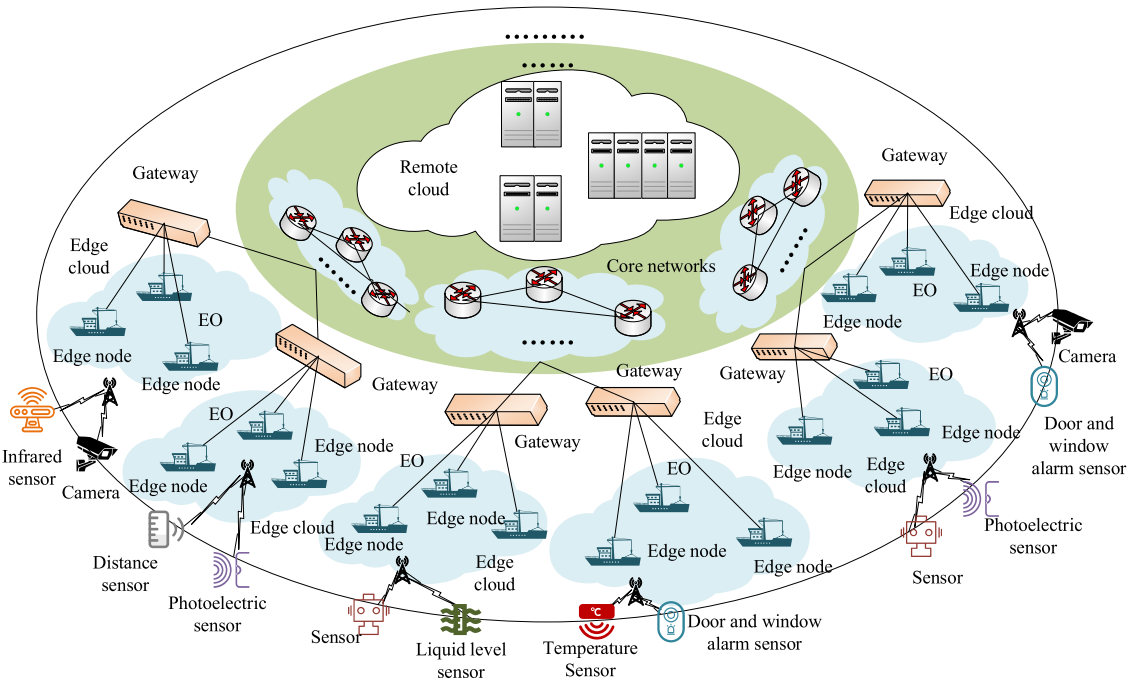

Fig. 1. Edge IoT-enabled MTS architecture.

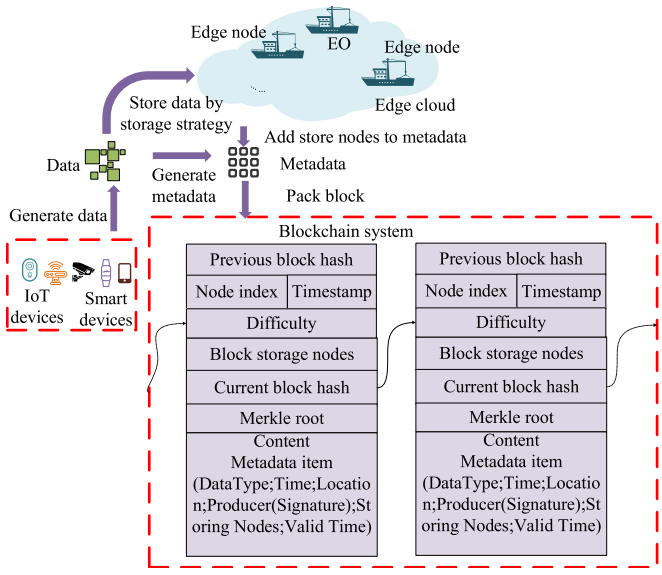

Fig. 2. The lightweight blockchain designed for edge IoT-enabled MTS.

A set of incentive parameters for node storage data copies is defined as an incentive cost (SC). The SC of the edge server es_i can be defined as follows:

$$s_i = \begin{cases} \dfrac{D_k}{ST_{to}(i) - ST(i)}, & ST(i) \neq 0 \\ 0, & ST(i) = 0 \end{cases} \tag{3}$$

where $ST_{to}(i)$ denotes the total storage capacity of the edge server es_i; $ST(i)$, the used storage capacity; and D_k, the size of the data k. If there are no remaining resources on the server, the incentive cost is ∞, and no more data will be stored on the server later.

The problem of data placement can be modeled as a multi-constraint optimization problem. The power consumption, network transmission, and incentive costs are integrated as follows:

$$\begin{aligned} min \ & \sum_{i\in V}\sum_{j\in V} a_{ij}x_{ijk} + \mathrm{A}\sum_{i\in V}(s_i + bc_i)y_{ik} \\ s.t. \ & \sum_{k} y_{ik}ST(k) \leq ST_{to}(i), \quad (\forall i \in V) \\ & \sum_{i\in V} x_{ijk} \geq 1, \quad (\forall j \in V) \\ & y_{ik} - x_{ijk} \geq 0, \quad (\forall i, j \in V) \\ & x_{ijk}, \quad y_{ik} \in \{0, 1\} \end{aligned} \tag{4}$$

where A denotes the factor for amplification used to increase the weight that is set to 100000, and x_{ijk} is an access variable. This variable A is added to increase the weight of the first part of the objective function, because the value calculated in the first part is much larger than the second part in (4). If $x_{ijk} = 1$, the request of the edge server es_j for access to the data item k will be assigned to the edge server es_i. y_{ik} is a storage variable. When $y_{ik} = 1$, the data item k will be placed on the edge server es_i. Constraint (1) ensures that the total size of data stored on the edge server es_i does not exceed the server's storage capacity. Constraint (2) ensures that at least one edge server stores a copy of the data item k, and constraint (3) ensures that the data item k assigned to the edge server es_i places a copy of the data item k on the server accessed.

D. Consensus Mechanism

PoW, which is used in Bitcoin, allows nodes to use their computing power to compute hash in order to compete with each other, and the ratio of computing power determines the probability of a node to be automatically selected. PoW has

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

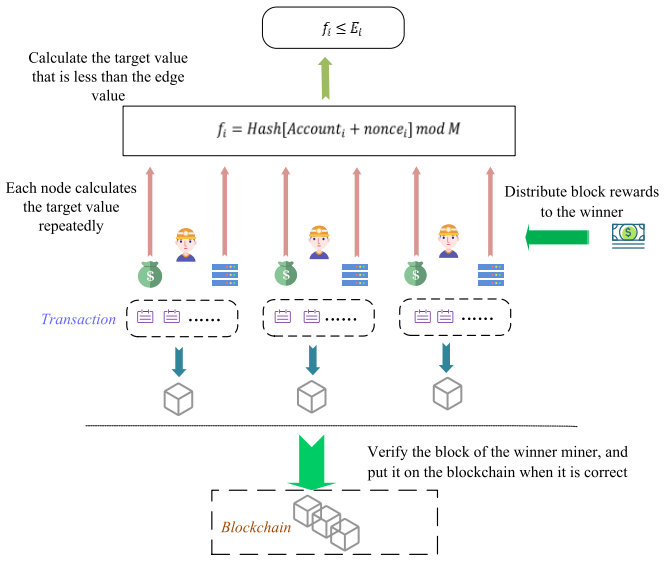

Fig. 3. Consensus mechanism designed in LBlockchainE.

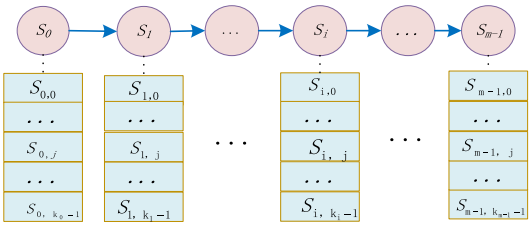

Fig. 4. Service chain and candidate services.

the disadvantage of consuming much energy, conflicting with the edge server with finite resources. To represent the rights of nodes, PoS employs a set of resources. To reduce the edge cloud's internal storage resource consumption, a new PoS containing the competition mechanism is proposed, which fully takes into account the contribution of each edge server to store data. The edge cloud coin owned by the node and the amount of data stored give the corresponding node the right to pack the block.

In this paper, LBlockchainE can encourage all nodes in the edge cloud to store data positively. As presented in Fig. 3, the consensus mechanism should be set in such a way that nodes with more data storage are favored.

In LBlockchainE, an edge value (E) is defined. The greater the edge value, the more obvious it is that the node has a block-mining advantage. The edge value is as follows:

$$E_i = C_i Q_i t \tag{5}$$

In (5), C_i denotes the number of the edge cloud coin owned by the edge server es_i, which can be obtained by mining new blocks. An edge cloud coin is awarded to a node in LBlockchainE for mining a block, which can also be traded between nodes. t denotes the elapsed time between the previous block and this block. Q_i is the amount of data stored in the edge server es_i. The amount of data is as follows:

$$Q_i = ST(i)/D_{unit} \tag{6}$$

where D_{unit} represents the size of the data block, which is 25 MB, to measure the amount of data stored by the edge server es_i.

In (5), t is set in that the node may have insufficient edge cloud coins and stored resources. Due to the small range, a small edge value will cause the target value to fall below the standard. The range can be increased by using the time from the previous block to the current block, causing the target value to grow. The target value f_i is as follows:

$$f_i = Hash[Account_i + nonce_i] \bmod M \tag{7}$$

where $Account_i$ denotes the account address of the edge server es_i; $nonce_i$, the random number calculated by the edge server es_i; and M, the maximum possible value. When f_i satisfies (8), the right to mine the block is obtained by the edge server es_i.

$$f_i \le E_i \tag{8}$$

where E_i denotes the edge value of the edge server es_i, which is determined by the edge cloud coin owned by the edge server es_i, the amount of the stored data, and the elapsed time from the previous block. The range of E_i will be greater when the edge server es_i has more edge cloud coins and stores more data. To prevent the elapsed time between two blocks from being large, causing the range of E_i to be larger also, time expectation t_0 is used to control the elapsed time. The easier it is for the edge server es_i to obtain the smaller f_i, the larger the range of E_i. When f_i does not meet the above formulation, $nonce_i$ needs to be changed until one node first obtains a f_i, meeting the above formulation.

E. Service Composition Based on LBlockchainE

In this paper, LBlockchainE is applied to the service composition to highlight its applicability. Edge servers affiliated with different companies are geographically dispersed in an edge cloud, which is deployed in base stations, routers, switches, and so on. Users have public access to these infrastructures, in which services are extremely vulnerable to malicious attacks. LBlockchainE can be applied to enhance the credibility of the QoS of the web services deployed in the edge servers, preventing malicious nodes from tampering with the QoS of web services.

The improved FA is used to apply LBlockchainE to service composition [33]. Cotangent initialization is employed to generate the random initial value, and cotangent disturbance is used to overcome the shortcoming of the algorithm of being stuck easily at a locally optimal value. To balance local and global search, the step factor and attractiveness formulation in FA is reset. With the above methods, FA is improved to suit high-dimensional service composition.

The brighter individuals in the population can attract each firefly, according to FA. FA is applied to the service composition in this paper, and each composition can be regarded as a firefly individual that will be attracted by better compositions, gradually approaching them. The position update formulation is expressed as follows:

$$X_i(t+1) = X_i(t) + \beta_{ij}\left(X_j(t) - X_i(t)\right) + \alpha\varepsilon_i \tag{9}$$

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

where $X_i(t)$ denotes the position of the firefly individual i in the iteration t; $X_j(t)$, the position of the firefly individual j in the iteration t; $\alpha\varepsilon_i$, a random term; and generally, ε_i, a random number, $\alpha \in [0, 1]$. r_{ij} is the Euclidean distance between the individual firefly i and the individual firefly j. The attraction formulation β_{ij} is improved as follows:

$$\beta_{ij} = (\beta_0 - \beta_{min})\, e^{-\gamma r_{ij}^2} + \beta_{min} \tag{10}$$

where β_0 denotes the maximum attraction representing the attraction of the firefly in the original position, and the minimum attractiveness β_{min} is set to ensure that the attractiveness between individuals is not zero. γ is the light absorption coefficient which can be taken as a constant. If β_{min} is zero, the degree of attraction β approaches 0 as the distance between individuals approaches infinity, which will lead to the lack of traction on other individuals and the ability to find the best.

If β_{min} is 0.25, when the distance is larger, even if $(\beta_0 - \beta_{min})e^{-\gamma r_{ij}^2}$ approaches 0, the attraction is not 0 due to β_{min}, which can guarantee that there is minimum attraction between individuals.

During the convergence process of FA, the firefly individuals should ideally gather and converge to a point in (9). The step factor is getting close to zero.

In general, the step factor α is set to 0.5, and ε_i is set to a random number. The third term in (9) is a completely random term that cannot be used during the algorithm when α is a fixed value. In this paper, to give full play to the function of α in the algorithm, α is set to dynamically change. To exert effective detection ability in the early stages of the algorithm, the random item must have a large random compensation. In the later stage of the iteration, it needs to perform effective searchability locally, and the step needs to be gradually reduced. Based on the above analysis, the step formulation is set as follows:

$$\alpha(t+1) = \alpha(t) \cdot e^{-0.5t/T_{max}} \tag{11}$$

As the algorithm progresses and the iteration increases, α gradually approaches 0 from 0.5.

In the early stages of the service composition algorithm, cotangent initialization is used, and when the algorithm reaches the local optimum, cotangent perturbation is required. The formulation is as follows:

$$a_{n+1} = cot\,(a_n) \tag{12}$$

where a_n denotes the one-time value of the cotangent sequence; the initial value of the cotangent sequence should be satisfied with $a_0 \in (0, \pi)$ and iterated through the cotangent sequence formulation to generate a pseudo-random sequence.

In the service composition scenario, the individual brightness is specifically set to the fitness value, that is, the value of the fitness function, which is used to evaluate the pros and cons of the service composition.

Reliability, response time, execution cost, and availability are four QoS parameters included in the service composition. The fitness function is as follows:

$$f = \alpha R + \beta A + \gamma \frac{1}{T} + \eta \frac{1}{C} \tag{13}$$

where f denotes the fitness value; R, the product of the reliability of each service in the service chain; T, the sum of the response time of each service in the service chain; C, the sum of the execution costs of each service in the service chain; and A, the product of each service availability in the service chain. α is the weight of R, β is the weight of A, γ represents the weight of T, η represents the weight of C, and $\alpha + \beta + \gamma + \eta = 1$.

Assuming a service chain $L_c = s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_j \rightarrow \cdots \rightarrow s_{m-1}$, s_j shown in Fig. 4 belongs to the service category S_j:

$$\begin{aligned} &Max\left(\alpha R + \beta A + \gamma \frac{1}{T} + \eta \frac{1}{C}\right) \\ &s.t.\ getOutput\left(s_j\right) == getInput\left(s_{j+1}\right), \quad s_j \in L_c \\ &\quad \alpha + \beta + \gamma + \eta = 1 \\ &\quad R = \prod_{i=1}^{n} Rel_i \quad T = \sum_{i=1}^{n} T_i \\ &\quad C = \sum_{i=1}^{n} C_i \quad A = \prod_{i=1}^{n} Avail_i \end{aligned} \tag{14}$$

where s_j denotes the service j in the service chain; $getInput()$ and $getOutput()$ can obtain the input set and output set of the corresponding service. α, β, γ, and η represent the weight of each QoS attribute. R, T, C, and A respectively represent the product or sum of each QoS attribute of the service chain, where R and A denote the reliability that is the product of the reliability of each service and availability that is the product of the availability of each service, respectively. T and C represent the execution time that is the sum of the time of multiple services and the execution cost that is the sum of the total execution cost of multiple services, respectively.

The algorithm includes three steps:

Step 1: Globally search for a feasible service chain and find candidate services for each service in the service chain.

Step 2: Initialize the firefly individual chaotically through the cotangent sequence.

Step 3: Calculate each firefly's fitness function value, weigh the benefits and drawbacks of each firefly, choose the best individual firefly as the object to be learned, and keep other fireflies up to date. When a single firefly falls into a local optimum, that individual should try to get out of it as much as possible. Individual fireflies' positions are re-initialized according to the cotangent sequence when cotangent disturbance is used. After re-initialization, it is necessary to search locally to find the local optimal value. The global optimal value will be replaced if the locally searched optimal value is better than the historical global optimal combination plan, and the chaotic disturbance will be over. Otherwise, it is necessary to repeat the chaotic disturbance until a new global optimal solution is found or the cotangent disturbance iteration meets the limit.

Algorithm 1 shows the steps of service composition algorithm. Calculate the fitness function value of each individual firefly, evaluate the pros and cons of each individual firefly, select the best individual firefly as the object to be learned, and update other firefly individuals. When one firefly individual

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Service Composition Algorithm

Input: N; I; D; β_0; β_{min}; α_0; γ ; $scandiadateList$
Output: G
$BestBorder$ = setBestBorder()
for $i = 1$ to N **do**
//firefly initialization
X_i = Initialize ()
//caculate the fitness value
F_i = ComputeFitness(X_i)
end for
// compare and update the global optimal firefly
index = GetGlobalBest(Fi)
$fBest = F_{index}$
$G = X_{index}$
while $fBest! = BestBorder$ **and** $i! = I$ **do**
for $i = 1$ to N **do**
// update each firefly
X_i = UpdatePosition(X_i)
F_i = ComputeFitness(X_i)
end for
index=GetGlobalBest(F_i)
$fBest = F_{index}$
$G = X_{index}$
end while
return G

falls into a local optimum, the individual should jump out of the local optimum as much as possible. By adopting cotangent disturbance, the position of individual firefly is reinitialized according to the cotangent sequence. After re-initialization, it is necessary to search locally to find the local optimal value. If the locally searched optimal value is better than the historical global optimal combination plan, the global optimal value will be replaced, and this chaotic disturbance ends. Otherwise, it is necessary to repeat the chaotic disturbance until a new global optimal solution is found or the cotangent disturbance iteration meets the limit.

The service registry, which is managed by the edge orchestration manager, saves the QoS attribute values of the services, which are also saved in LBlockchainE. The QoS value must be recalculated and saved in LBlockchainE after each service ends. The candidate service set, which is the firefly's brightness, determines the fitness value of the corresponding service composition. The composition optimization method applies LBlockchainE to realize the credibility of QoS attributes and provide users with reliable services. As shown in Fig. 5, each QoS value change will be stored as a transaction in the block, and each node participating in packing the block can verify whether the QoS has been tampered with, so the QoS value obtained by the service combination needs to be obtained from the edge Obtained from the orchestrator, the user can query in the chain to check whether it has been tampered with.

IV. Prototype System

A lightweight blockchain platform for edge IoT-enabled MTS is designed and built based on LBlockchainE. The blockchain platform structure is presented in Fig. 6.

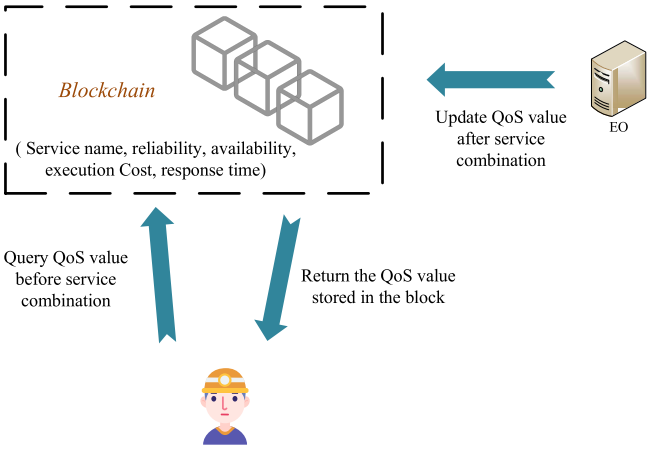

Fig. 5. Service composition with blockchain.

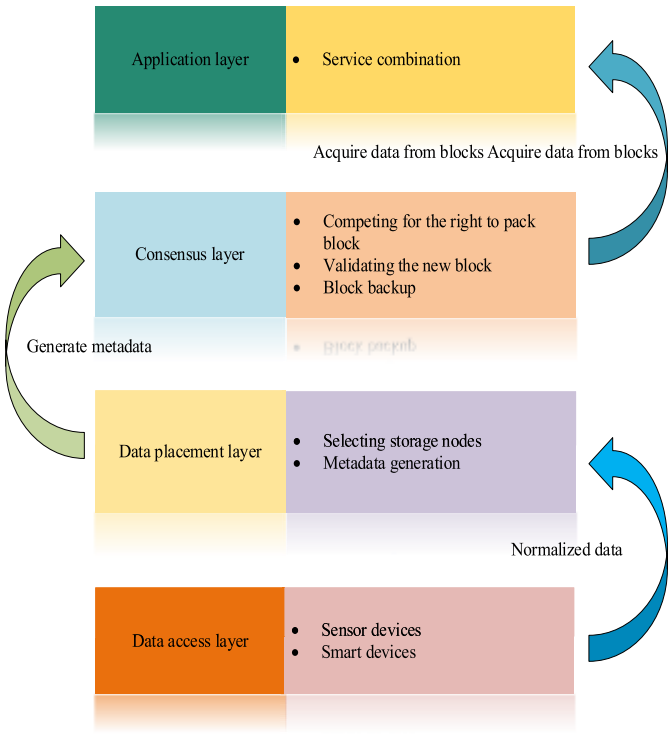

Fig. 6. LBlockchainE-based platform structure.

A data access layer, a data placement layer, and a consensus layer are all included in the structure. In addition, there is an application layer that provides services for specific applications by obtaining the support of the above three layers.

A. Data Access Layer

Sensor devices upload data that includes the type of data, the location where the data is generated, and the specific data to the edge cloud. A peer-to-peer network is what the blockchain environment is all about. As shown in Fig. 7, PeerNetwork class and RpcServer class need to be established. The PeerNetwork class that contains PeerThread class used to help each node access other nodes allows nodes joining the blockchain system to communicate with each other, synchronize blocks, and so on. Each PeerThread class will call

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

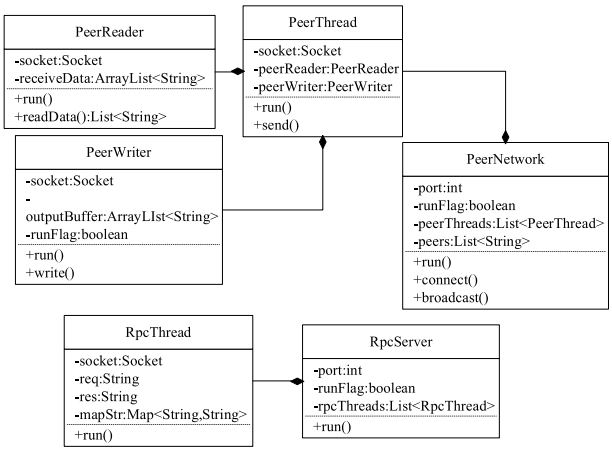

Fig. 7. Data access class diagram.

the PeerReader class and the PeerWriter class for message transmission between blockchain nodes.

When the IoT devices upload data to the blockchain system, the entire blockchain needs to process client access by the RpcServer class. The RpcServer class contains the RpcThread class, which the client accessing the blockchain generates. The system will return the response to the client after parsing the corresponding request.

B. Data Placement Layer

As shown in Fig. 8, the data placement strategy involves the realization of three classes that include Node class, Particle class, and ParticleState class. Each blockchain node's account information, storage resources, edge cloud coins, and account information are all stored in the Node class. The node's right to pack blocks is directly influenced by the related resources and edge cloud coins. According to the related information, the edge cloud orchestrator selects the best node to store data. The objective function is presented in (4). The particle swarm optimization algorithm is used to calculate the nodes suitable for storing data, which is the Particle class containing the ParticleState class used to initialize the position of the particle and perform calculation operations. The data and corresponding metadata will be broadcast to the blockchain network. The selected nodes will store the data.

C. Consensus Layer

The blockchain disseminates both the metadata and data. The metadata is broadcast in the network once the data is stored on the selected node, and each node packs the received metadata. The consensus mechanism, which is at the heart of the blockchain, determines which node will get the right to pack blocks and thus the block rewards in the end. The consensus mechanism class we designed in the blockchain defines the method of obtaining the packing rights. In each round, the node calculates the account address and nonce to obtain the corresponding target value, according to (7). The right to pack a block that can be broadcast and verified by a blockchain node is obtained when the target value is calculated

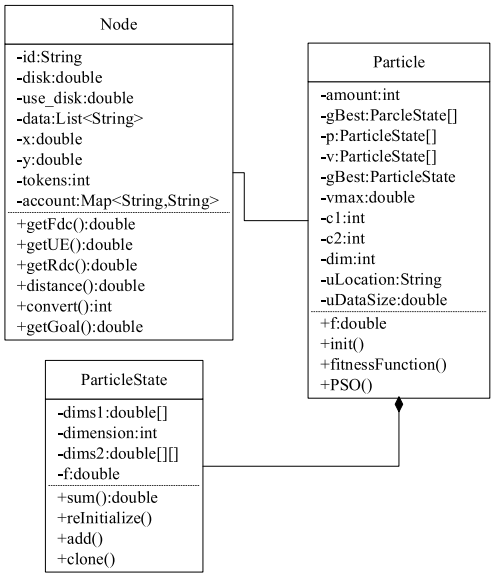

Fig. 8. Data placement class diagram.

first within the edge value. The node that packs the block will get the reward after the verification is passed.

D. Application Layer

LBlockchainE, which is designed in this paper, aims to support the security of the data in the edge-IoT and provide a secure storage environment for the services. In this paper, LBlockchainE is used to provide security for the service composition to prevent malicious nodes from tampering with related web services' QoS attributes.

V. Experiments and Security Analysis

A. Experimental Environment

Docker is used for simulation to test the performance of LBlockchainE, in which containers are built through It is also used to simulate each edge server in the edge IoT-enabled MTS. Sockets are used to transmit data between nodes that are run by containers. Fig. 9 describes the experimental environment. The experiments are conducted on four edge cloud computing systems. Several ship nodes make up each edge cloud system. The simulation was performed on a computer equipped with an Intel Core i7-8700K processor and 16 GB of RAM. In the experiment, it is assumed that the position of each node is located in the area of 200 m$*$200 m, and the expected interval time is set to 20 s.

The number of nodes on this blockchain platform will be limited by equipment, as thousands of nodes are added to the real blockchain. In this experiment, the size of each data item is about 700 B, and the generated block size is 670 B.

B. Performance Statement

Power consumption, block transmission cost, message transmission overhead, and data access delay are all factors in the experiment. When a block is packed, the power consumption represents how much power is used. PoW, which consumes a lot of power resources in Ethereum, is compared with the

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

consensus mechanism in LBlockchainE. The power consumption status of consensus mechanisms in different blockchain systems can be determined by measuring power consumption over the same number of blocks.

After a node successfully packs a block, the block will be distributed to all nodes in the blockchain network for verification. The transmission of a large number of block transmission messages will occupy the network bandwidth because a block contains multiple transaction information. The amount of block information transmitted in the network under the same number of blocks is utilized to calculate the block transmission cost. In LBlockchainE, the complete data information stored in the block is replaced with metadata, which can reduce network bandwidth resources.

Message transmission overhead refers to the transmission overhead of messages in the blockchain network. Data transmission overhead will be generated as the data is only stored on some nodes. The network transmission overhead must be greater than the data placement strategy in LBlockchainE if all nodes store data.

Data access between nodes will inevitably cause some delay because some nodes store data. To ensure that LBlockchainE can effectively control the time delay, the data placement strategy is compared with the random storage strategy, optimal storage strategy, and other strategies.

C. Performance Analysis

1) Power Consumption: To demonstrate the performance of the consensus mechanism designed in this paper, the consensus mechanism is compared with the consensus mechanism of Ethereum and that of Huang *et al.* [25] in the experiment. The power consumption is measured by a power meter. In the experiment, the electricity change of the same number of blocks packed is measured for each blockchain. In this experiment, the number of nodes in the blockchain is set to 5, as is the difficulty of the block. As presented in Fig. 10, with the increase in the number of blocks, although the power consumption of the three consensus mechanisms is increasing, the gap between the three gradually widens. Ethereum consumes significantly more power than LBlockchainE, whereas LBlockchainE consumes slightly less energy than that of Huang *et al.* The results show that LBlockchainE reduces the average power consumption by 92.21 % and 77.67%, respectively, compared with the other two.

The above is a comparison of power consumption. In addition, the mining ability is also reflected in the CPU occupation. The competition in the blockchain is reflected in the calculation of the hash, and the calculation work in this experiment is undertaken by the CPU, so CPU usage percentage can be measured to show the power consumption. In this experiment, three comparison subjects run in the same experimental environment, and the average CPU occupation was measured under packing a block. It can be seen from Fig. 11 that LBlockchainE designed in this paper occupies less CPU resources than the other two, and reduces the resource occupancy by 75.41% and 16.04%, respectively, compared with the other two. The results are generally consistent with the measurement of power consumption, which saves resource consumption.

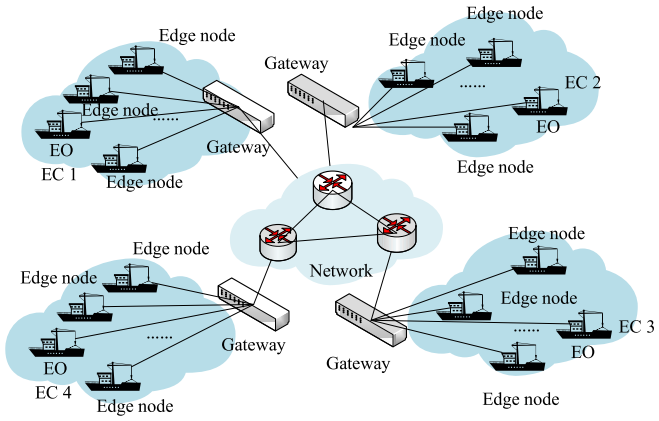

Fig. 9. Experimental environment.

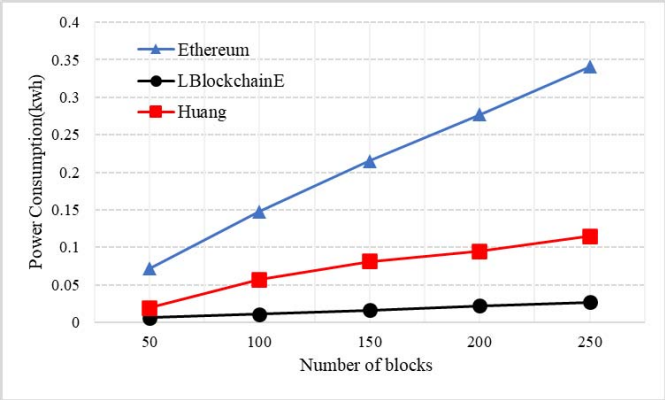

Fig. 10. Power consumption.

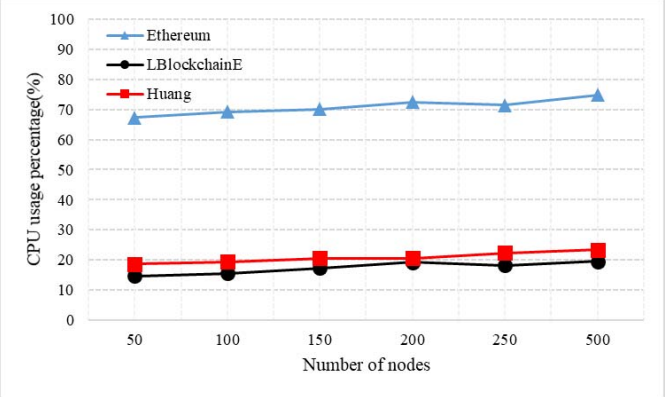

Fig. 11. CPU usage percentage.

The packing block condition in LBlockchainE is set to the target value within the range of the edge value, and calculating to the specified difficulty requires a large number of hash calculations if only PoW is utilized. Similar to the consensus mechanism of Huang *et al.* [25], the consensus mechanism of LBlockchainE also involves hash calculation and the competition mechanism, but the energy consumption of LBlockchainE is slightly lower than that of Huang *et al.*, but both involve hash calculation.

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

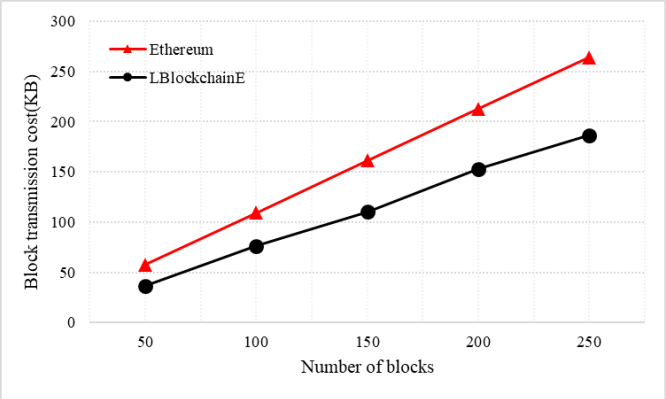

Fig. 12. Block transmission cost.

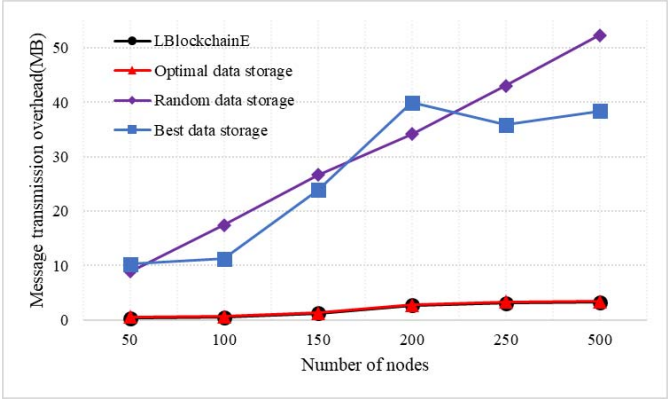

Fig. 14. Message transmission overhead.

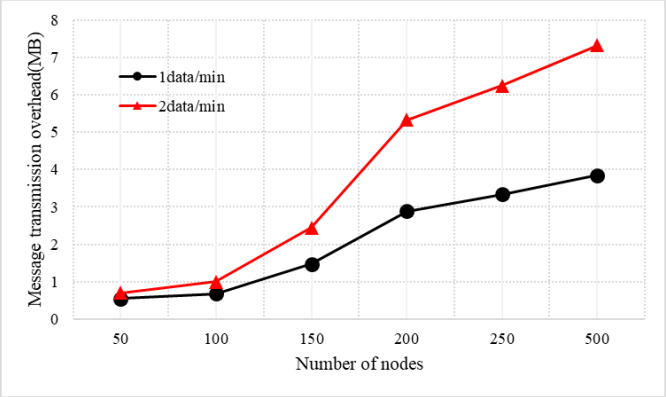

Fig. 13. Message transmission overhead.

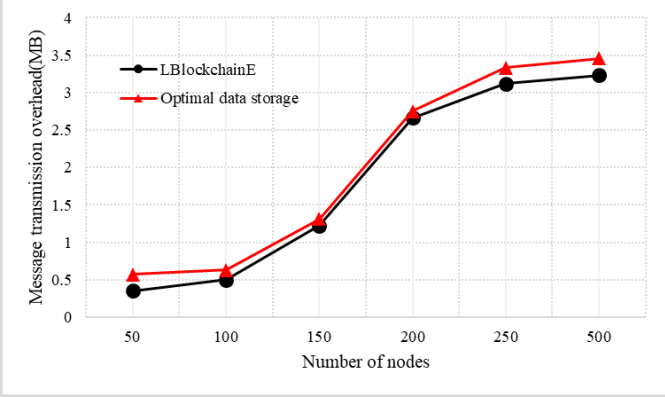

Fig. 15. Message transmission overhead.

2) Block Transmission Cost: Metadata item is used to replace data item and is packed into blocks to save the storage cost of the block. When compared with full data items, a metadata item extracts a portion of the information from them, saving a lot of space. Ethereum and LBlockchainE are compared. As presented in Fig. 12, as the number of packed blocks increases, the storage cost of the two-block formats also gradually increases. The results show that LBlockchainE reduces the average block transmission cost by 30.14% compared with Ethereum. Since blocks need to be verified by blockchain nodes before they can be put into the chain, broadcasting the blocks in the network will cost numerous transmission bandwidths. LBlockchainE reduces the cost of block transmission in the network when compared with Ethereum, according to the results of the experiments.

3) Message Transmission Overhead: A comparison between generating one data item per minute and generating two data items per minute is performed to evaluate the data placement strategy of LBlockchainE. The experiment lasted 7 h. The message transmission overhead increases as the number of nodes increases, as presented in Fig. 13, but it is obvious that the message transmission overhead is not large.

Under the same standard of generating 1 data item per minute, as presented in Fig. 14, the experiment was conducted under 50, 100, 150, 200, 250 and 500 nodes, respectively, and the data placement strategy in LBlockchainE is compared with the random storage, the best storage, and the optimal data storage strategies [25]. The experimental results indicate that as the number of nodes increases, the average message transmission cost per node using the random storage strategy also gradually increases. The optimal data storage strategy and the message transmission overhead of LBlockchainE are nearly identical. The results show that compared with the other three strategies, LBlockchainE reduces the average message transmission consumption by 93.96%, 93.52% and 8.5%, respectively.

As presented in Fig. 15, the message transmission overhead of LBlockchainE is only slightly better than the optimal data storage strategy in that the latter only takes the link distance between nodes into account, not considering the transmission delay; thus, more nodes will be selected for storing data, and data transmission between nodes will produce higher transmission overhead than LBlockchainE. The random storage strategy will randomly select nodes for storage without considering factors such as node resources and transmission cost, whereas the best storage strategy is for nodes with a small number of resources without considering the transmission cost and incentive factors. Thus, two strategies will produce extremely high message transmission overhead.

4) Data Access Delay: To evaluate the impact of the data placement strategy in LBlockchainE, it is compared with the random storage, best storage, and optimal data storage strategies. The four strategies are performed under the same network environment. The length of the link and the number of route hops are used to calculate the access cost. LBlockchainE has a lower data access cost than the other strategies, as presented in

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

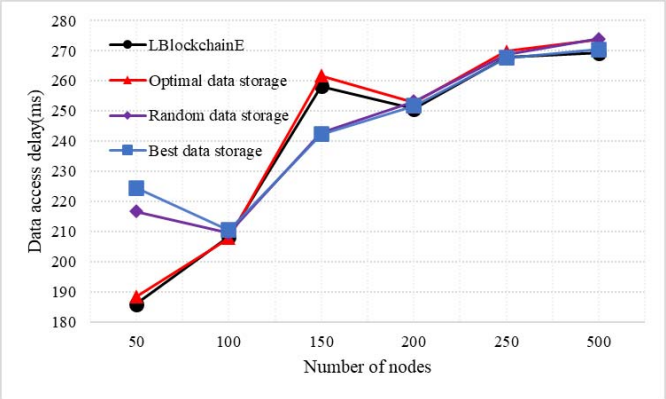

Fig. 16. Data access delay.

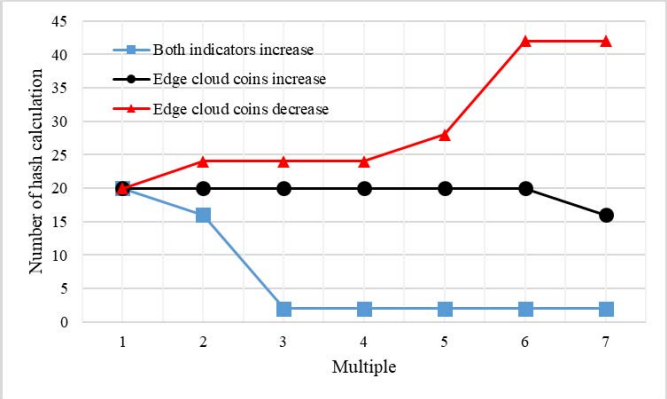

Fig. 17. The number of hash calculations required by the consensus mechanism in this paper.

Fig. 16. The results show that compared with the other three strategies, LBlockchainE reduces the average data access delay by 1.67%, 2.15% and 0.83%, respectively.

As the number of nodes increases, both LBlockchainE and the optimal data storage strategy achieve higher data access delay. The delay of LBlockchainE is slightly smaller than that of the optimal data storage strategy in that the latter only considers the link length without taking routing and forwarding delay between nodes into account, which will generate additional access delay. The random storage and best storage strategies have an unstable increase in latency, but both are higher than LBlockchainE. The random storage and optimal storage strategies take a longer time to access data between nodes and have an unstable delay.

In addition, the hash calculation in this paper depends on the size of the target value, which affects this difficulty. Therefore, although a large number of hash calculations are reduced overall, the security of the entire environment can be maintained, which relies on the verification of blocks and transactions. Miners will still maintain the balance of the entire blockchain ecology for long-term benefits. As shown in Fig. 18, this experiment tested the number of hash calculations required by PoW in the difficulty of 1 to 5. As the difficulty increases, the number of hash calculations increases almost exponentially. As presented in Fig. 17, when the contribution of data storage and the edge cloud coin owned by the node increase, the target value increases, the easier it is for the node to obtain the value within the range, and the number of hash calculations becomes less. The consensus mechanism in this paper can save more calculations and it can reach the target range with fewer rounds for the same block. When an indicator drops, such as the amount of stored data decreases, or the number of edge cloud coins owned decreases, it will increase the difficulty of obtaining blocks.

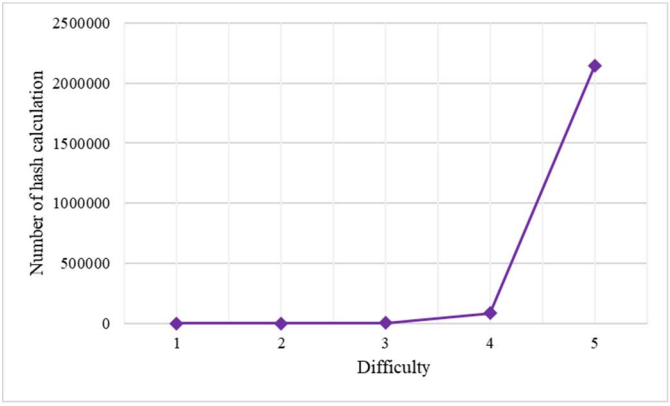

Fig. 18. The number of hash calculations required by PoW.

D. Security Analysis

The SM2 encryption algorithm, which is an elliptic curve public key cryptographic algorithm with higher security than RSA-2048 and above and belongs to a completely exponential encryption algorithm, is employed to generate the node account in LBlockchainE. Since the computational complexity is completely exponential, the difficulty of cracking is much higher than that of RSA. The key in this paper is divided into a public and a private key, and the account of each blockchain node is directly generated from the public key and released. The miner uses the private key to sign the block after it has been generated and signed. Other verifiers can use the public key issued by the miner to verify the signature in order to guarantee the legitimacy of the signer.

It is necessary to avoid a majority attack due to the adoption of PoW. When the malicious nodes in the network are combined and master more than 50% of the hash power, it is possible that a malicious node will launch a fork attack in the original blockchain that a fork is created to replace the previous chain relying on powerful computing power and the transaction on the previous chain will be canceled. In another case, the malicious node wants to modify the originally packed data and repack the tampered data to put it in the chain for malicious purposes. Once a transaction has been confirmed by k blocks, it is usually set that it cannot be withdrawn. The success probability of a fork attack exponentially decreases as a function of k, and generally, k takes 6. The node can obtain the right to pack the block only if the target value obtained through PoW is within the range determined by the edge cloud coin and the stored data. Even if a node has more computing resources, it is difficult to obtain block packing rights when it contributes little to the overall storage of the edge cloud and does not have sufficient edge cloud coins. In (5), C_i represents the edge cloud coin owned by the node; Q_i, the amount of

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

data stored in the node; and t, the time elapsed between this block and the previous block. Due to the limitation, the value is relatively stable. E_i depends on the change of C_i and Q_i. When C_i and Q_i are very small, it is still difficult for the node having a high computing power to get f_i within the target value E_i.

However, there is another special case in which edge cloud coins can be obtained through continuous mining, thereby accumulating a large amount of edge cloud coins, increasing the range of peripheral value to enhance the possibility of obtaining block packaging rights. This problem is left to be solved later.

VI. Conclusion

This paper proposes LBlockchainE for edge IoT-enabled MTS, a lightweight blockchain that fully considers the characteristics of the edge IoT-enabled MTS and removes some parts of the traditional blockchain that are not suitable for the edge computing environment. In light of the edge server's and IoT devices' limited resources on ship, a data placement strategy is devised that not only reduces the storage cost of edge servers where sensor data is stored but also enables rapid resource access between edge servers. To reduce the storage occupation of the block, metadata items are introduced in this design, which only contains the basic information and storage locations of the data items, which greatly saves storage resources. LBlockchainE combines the competition mechanism and Proof of Stake to determine the ownership of bookkeeping rights through a small number of competitive calculations and the node's resources, taking into account the limited resources in edge IoT-enabled MTS. This paper also uses the designed blockchain to ensure the reliability of the service composition, thereby verifying the effectiveness of the blockchain in edge IoT-enabled MTS. In the future, the incentive mechanism of each node in the system will be studied to encourage each node to actively contribute its resources in order to ensure smooth operation of the blockchain platform for edge IoT-enabled MTS.

Acknowledgment

The authors would like to thank editors and reviewers for their comments in helping them improve the quality of this paper.

References

[1] Statista. (2021). *Number of IoT Connected Devices Worldwide 2019–2030*. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[2] M. Armbrust *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[3] M. Du, Y. Wang, K. Ye, and C. Xu, "Algorithmics of cost-driven computation offloading in the edge-cloud environment," *IEEE Trans. Comput.*, vol. 69, no. 10, pp. 1519–1532, Oct. 2020.

[4] K. Gai, J. Guo, L. Zhu, and S. Yu, "Blockchain meets cloud computing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2009–2030, 3rd Quart., 2020.

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[6] E. El Haber, T. M. Nguyen, and C. Assi, "Joint optimization of computational cost and devices energy for task offloading in multi-tier edge-clouds," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3407–3421, May 2019.

[7] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[8] Z. Xu, S. Han, and L. Chen, "CUB, a consensus unit-based storage scheme for blockchain system," in *Proc. IEEE 34th Int. Conf. Data Eng. (ICDE)*, Paris, France, Apr. 2018, pp. 173–184.

[9] A. Marsalek, T. Zefferer, E. Fasllija, and D. Ziegler, "Tackling data inefficiency: Compressing the bitcoin blockchain," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Communications/13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Rotorua, New Zealand, Aug. 2019, pp. 626–633.

[10] A. Dorri, "LSB: A lightweight scalable blockchain for IoT security and anonymity," *J. Parallel Distrib. Comput.*, vol. 134, pp. 180–197, Dec. 2019.

[11] Y. Xu, "Section-blockchain: A storage reduced blockchain protocol, the foundation of an autotrophic decentralized storage architecture," in *Proc. 23rd Int. Conf. Eng. Complex Comput. Syst. (ICECCS)*, Melbourne, VIC, Australia, Dec. 2018, pp. 115–125.

[12] J. Pan, J. Wang, A. Hester, I. Alqerm, Y. Liu, and Y. Zhao, "EdgeChain: An edge-IoT framework and prototype based on blockchain and smart contracts," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4719–4732, Jun. 2019.

[13] C. Li, J. Bai, Y. Chen, and Y. Luo, "Resource and replica management strategy for optimizing financial cost and user experience in edge cloud computing system," *Inf. Sci.*, vol. 516, pp. 33–55, Apr. 2020.

[14] M. B. M. Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: A survey," *Comput. Netw.*, vol. 148, pp. 283–294, Jan. 2019.

[15] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2702–2733, 3rd Quart., 2019.

[16] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey," *J. Netw. Comput. Appl.*, vol. 88, pp. 10–28, Jun. 2017.

[17] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, 3rd Quart., 2020.

[18] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Comput. Netw.*, vol. 76, pp. 146–164, Jan. 2015.

[19] B. Alzahrani and N. Fotiou, "Enhancing Internet of Things security using software-defined networking," *J. Syst. Archit.*, vol. 110, Nov. 2020, Art. no. 101779.

[20] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of Things security: A top-down survey," *Comput. Netw.*, vol. 141, pp. 199–221, Aug. 2018.

[21] Ethereum. (2020). *Ethereum Whitepaper*. [Online]. Available: https://ethereum.org/en/whitepaper/

[22] S. Q. Zeng, R. Huo, T. Huang, J. Liu, S. Wang, and W. Feng, "Survey of blockchain: Principle, progress and application," *J. Commun.*, vol. 41, no. 1, pp. 134–151, 2020.

[23] Hyperledger. (2018). *An Introduction to Hyperledger.* [Online]. Available: https://www.hyperledger.org/wpcontent/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf

[24] S. Popov. (2016). *The Tangle*. [EB/OL]. [Online]. Available: https://iota.org/IOTA_Whitepaper.pdf

[25] Y. Huang, J. Zhang, J. Duan, B. Xiao, F. Ye, and Y. Yang, "Resource allocation and consensus on edge blockchain in pervasive edge computing environments," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Dallas, TX, USA, Jul. 2019, pp. 1476–1486.

[26] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, Jan. 2015.

[27] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 88, pp. 173–190, Nov. 2018.

[28] H.-N. Dai, M. Imran, and N. Haider, "Blockchain-enabled Internet of Medical Things to combat COVID-19," *IEEE Internet Things Mag.*, vol. 3, no. 3, pp. 52–57, Sep. 2020.

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

[29] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019.

[30] K. Biswas and V. Muthukkumarasamy, "Securing smart cities using blockchain technology," in *Proc. IEEE 18th Int. Conf. High Perform. Comput. Communications, IEEE 14th Int. Conf. Smart City, IEEE 2nd Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Sydney, NSW, Australia, Dec. 2016, pp. 1392–1393.

[31] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Kona, HI, USA, Mar. 2017, pp. 618–623.

[32] X. Song, Y. Huang, Q. Zhou, F. Ye, Y. Yang, and X. Li, "Content centric peer data sharing in pervasive edge computing environments," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Atlanta, GA, USA, Jun. 2017, pp. 287–297.

[33] I. Fister, I. Fister, Jr., X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," *Swarm Evol. Comput.*, vol. 13, pp. 34–46, Dec. 2013.

[34] H. Liao, Y. Mu, Z. Zhou, M. Sun, Z. Wang, and C. Pan, "Blockchain and learning-based secure and intelligent task offloading for vehicular fog computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4051–4063, Jul. 2021.

[35] A. Munusamy *et al.*, "Edge-centric secure service provisioning in IoT-enabled maritime transportation systems," *IEEE Trans. Intell. Transp. Syst.*, early access, Aug. 12, 2021, doi: 10.1109/TITS.2021.3102957.

[36] J. O. Eichenhofer, E. Heymann, B. P. Miller, and A. Kang, "An in-depth security assessment of maritime container terminal software systems," *IEEE Access*, vol. 8, pp. 128050–128067, 2020.

[37] J. Li, J. Wu, J. Li, A. K. Bashir, M. J. Piran, and A. Anjum, "Blockchain-based trust edge knowledge inference of multi-robot systems for collaborative tasks," *IEEE Commun. Mag.*, vol. 59, no. 7, pp. 94–100, Jul. 2021.

[38] C. Feng *et al.*, "Efficient and secure data sharing for 5G flying drones: A blockchain-enabled approach," *IEEE Netw.*, vol. 35, no. 1, pp. 130–137, Jan. 2021.

[39] M. Dibaei *et al.*, "Investigating the prospect of leveraging blockchain and machine learning to secure vehicular networks: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 683–700, Feb. 2022.

[40] X. Xu, W. Liang, J. Zhao, and H. Gao, "Tiny FCOS: A lightweight anchor-free object detection algorithm for mobile scenarios," *Mobile Netw. Appl.*, vol. 26, pp. 2219–2229, Oct. 2021.

[41] X. Xu, J. Zhao, Y. Li, H. Gao, and X. Wang, "BANet: A balanced atrous net improved from SSD for autonomous driving in smart transportation," *IEEE Sensors J.*, vol. 21, no. 22, pp. 25018–25026, Nov. 2021.

[42] P. Li, X. Wang, H. Gao, X. Xu, M. Iqbal, and K. Dahal, "A dynamic and scalable user-centric route planning algorithm based on polychromatic sets theory," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 7, 2021, doi: 10.1109/TITS.2021.3085026.

[43] Y. Huang, H. Xu, H. Gao, X. Ma, and W. Hussain, "SSUR: An approach to optimizing virtual machine allocation strategy based on user requirements for cloud data center," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 670–681, Jun. 2021.

[44] X. Ma, H. Xu, H. Gao, and M. Bian, "Real-time multiple-workflow scheduling in cloud environments," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 4, pp. 4002–4018, Dec. 2021.

[45] H. Gao, C. Liu, Y. Yin, Y. Xu, and Y. Li, "A hybrid approach to trust node assessment and management for VANETs cooperative data communication: Historical interaction perspective," *IEEE Trans. Intell. Transp. Syst.*, early access, Nov. 30, 2021, doi: 10.1109/TITS.2021.3129458.

[46] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3680–3689, Jun. 2019.

[47] B. K. Mohanta, D. Jena, S. Ramasubbareddy, M. Daneshmand, and A. H. Gandomi, "Addressing security and privacy issues of IoT using blockchain technology," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 881–888, Jan. 2021.

[48] H. Gao, Y. Zhang, H. Miao, R. J. D. Barroso, and X. Yang, "SDTIOA: Modeling the timed privacy requirements of IoT service composition: A user interaction perspective for automatic transformation from BPEL to timed automata," *Mobile Netw. Appl.*, vol. 26, no. 6, pp. 2272–2297, Dec. 2021.

[49] B. B. Gupta, A. Gaurav, C.-H. Hsu, and B. Jiao, "Identity-based authentication mechanism for secure information sharing in the maritime transport system," *IEEE Trans. Intell. Transp. Syst.*, early access, Nov. 15, 2021, doi: 10.1109/TITS.2021.3125402.

[50] J. Liu, C. Li, J. Bai, Y. Luo, H. Lv, and Z. Lv, "Security in IoT-enabled digital twins of maritime transportation systems," *IEEE Trans. Intell. Transp. Syst.*, early access, Nov. 24, 2021, doi: 10.1109/TITS.2021.3122566.

[51] 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, May 2020.

[52] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7702–7712, Oct. 2019.

[53] M. Shen, J. Zhang, L. Zhu, K. Xu, and X. Tang, "Secure SVM training over vertically-partitioned datasets using consortium blockchain for vehicular social networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5773–5783, Jun. 2020.

[54] H. Gao, C. Liu, Y. Li, and X. Yang, "V2 VR: Reliable hybrid-network-oriented V2 V data transmission and routing considering RSUs and connectivity probability," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3533–3546, Jun. 2021.

Yu Jiang is currently working as a Researcher with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing, China. He used to work at the Institute of Big Data Research, Yancheng. His research interests include blockchain-based IoT and edge computing. He has two patents in the above field authorized by the State Intellectual Property Office of China as one of the key inventors. He has won a National Competition Award and several scholarships during university.

Xiaolong Xu received the B.E. degree in computer and its applications in 1999, the M.E. degree in computer software and theories in 2002, and the Ph.D. degree in communications and information systems in 2008.

He is currently a Professor with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China. He is also working with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing. He is a Senior Member of the China Computer Federation. He teaches graduate courses and conducts research in areas of cloud computing, big data, information security, and novel network computing technologies. As the leader of project teams, he has successfully completed a number of high-level research projects, including the projects sponsored by the National Science Fund of China. He has published more than 100 journals and conference papers as the first or corresponding author and five books. He authorized 52 patents by the State Intellectual Property Office of China as the First Inventor. He was rated as an Excellent Young Professor of Jiangsu Province in 2014, selected as the High-Level Creative Talents of Jiangsu Province in 2015, and won the Title of Outstanding Expert in the area of computer science and technology.

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.

Honghao Gao (Senior Member, IEEE) is currently with the School of Computer Engineering and Science, Shanghai University, China. He is also a Professor with Gachon University, South Korea. Prior to that, he was a Research Fellow with the Software Engineering Information Technology Institute, Central Michigan University, USA, and an Adjunct Professor with Hangzhou Dianzi University, China. Moreover, he has broad working experience in cooperative industry-university-research. He has publications in IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, IEEE INTERNET OF THINGS JOURNAL, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, IEEE NETWORK, *ACM TOIT*, *ACM TOMM*, *ACM TOSN*, *ACM TMIS*, and *Information Sciences*. His research interests include software formal verification, the Industrial IoT networks, vehicle communication, and intelligent medical image processing.

Prof. Gao is a fellow of the Institution of Engineering and Technology (IET) and the British Computer Society (BCS). He is a member of the EPSRC Peer Review Associate College of U.K. Research and Innovation, U.K., and a Founding Member of the IEEE Computer Society Smart Manufacturing Standards Committee. He was a recipient of the Best Paper Award at IEEE TII 2020 and EAI CollaborateCom 2020. He is the Editor-in-Chief for *International Journal of Intelligent Internet of Things Computing* (IJIITC); an Editor of *Wireless Network* and *IET Wireless Sensor Systems*; and an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *IET Intelligent Transport Systems*, *IET Software*, *International Journal of Communication Systems*, *Journal of Internet Technology*, and *Engineering Reports*. He is an external expert for reviewing and monitoring EU Project at European Union Institutions.

Adel D. Rajab received the bachelor's degree in computer science and information system and the master's and Ph.D. degrees in computer science and engineering from the University of South Carolina, USA. He is currently working as an Assistant Professor with the College of Computer Science and Information System (CSIS) and the Vice Dean of graduate studies for academic affairs at Najran University, Najran, Saudi Arabia. His research interests are robotics, drones, machine learning, and bioinformatic OBS networks.

Fu Xiao (Member, IEEE) is currently a Professor with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China. His research papers have been published in many prestigious conferences and journals, such as IEEE INFOCOM, IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, *ACM TECS*, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. His research interests are mainly in the areas of the Internet of Things and mobile computing. He is a member of the IEEE Computer Society and the Association for Computing Machinery.

Xinheng Wang (Senior Member, IEEE) received the B.E. and M.Sc. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 1991 and 1994, respectively, and the Ph.D. degree in electronics and computer engineering from Brunel University, Uxbridge, U.K., in 2001.

He is currently a Professor with the School of Advanced Technology and the Head of the Department of Mechatronics and Robotics, Xi'an Jiaotong-Liverpool University (XJTLU), Suzhou, China. Prior to joining XJTLU, he was a Professor with different universities in the U.K. He has been an Investigator or Co-Investigator of nearly 30 research projects sponsored from EU, U.K. EPSRC, Innovate U.K., China NSFC, and industry. He has authored or coauthored over 170 referred papers. He holds 15 granted patents, including one U.S., one Japan, four South Korea, and nine China patents. His current research interests include tactile internet, indoor positioning, the Internet of Things (IoT), acoustic localization, communications and sensing, and big data analytics for intelligent services, where he has developed the world's first smart trolley with an industry partner. His research has led to a few commercial products in condition monitoring, wireless mesh networks, and user-centric routing and navigation for group users.

Authorized licensed use limited to: ROBERT BOSCH. Downloaded on October 08,2023 at 06:18:42 UTC from IEEE Xplore. Restrictions apply.