Mathematical Biosciences and Engineering, 16(4): 1874–1892.
DOI: 10.3934/mbe.2019091
Received: 03 December 2018
Accepted: 31 January 2019
Published: 07 March 2019

http://www.aimspress.com/journal/MBE

Research article

Secure data storage based on blockchain and coding in edge computing

Yongjun Ren[1,2], Yan Leng[1,2], Yaping Cheng[1,2] and Jin Wang[3,4,*]

[1] School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, China
[2] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science & Technology, Nanjing, China
[3] Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, China
[4] School of Information Science and Engineering, Fujian University of Technology, Fujian, China

*** Correspondence:** Email: jinwang@csust.edu.cn; Tel: +8618014849250.

Abstract: Edge computing is an important tool for smart computing, which brings convenience to data processing as well as security problems. In particular, the security of data storage under edge computing has become an obstacle to its widespread use. To solve the problem, the mechanism combing blockchain with regeneration coding is proposed to improve the security and reliability of stored data under edge computing. Our contribution is as follows. 1) According to the three-tier edge computing architecture and data security storage requirements, we proposed hybrid storage architecture and model specifically adapted to edge computing. 2) Making full use of the data storage advantages of edge network devices and cloud storage servers, we build a global blockchain in the cloud service layer and local blockchain is built on the terminals of the Internet of things. Moreover, the regeneration coding is utilized to further improve the reliability of data storage in blockchains. 3) Our scheme provides a mechanism for periodically validating hash values of data to ensure the integrity of data stored in global blockchain.

Keywords: smart computing; edge computing; secure data storage; blockchain; coding

1. Introduction

Technological progress such as Internet of Things (IoT), Big Data analytics, cloud computing, machine learning and artificial intelligence in recent years has allowed the design of new Smart computing systems in smart environments aiming to facilitate users' lives. Smart computing is widely used in transportation, energy, environmental protection, smart city, healthcare, entertainment, and social media.

In smart computing, more and more applications are putting large amounts of data into cloud servers for computing or storage. And it can solve the problem of limited storage capacity and insufficient computing speed of intelligent terminals and other devices. In addition, users do not need to care about the specific structure, management mode and maintenance of computing and storage, nor need to worry about the technical issues such as expansion and fault tolerance under the cloud computing environment. They only need to buy from Cloud Storage Provider (CSP) as required, just like buying water, electricity and gas [1–3].

As the trend of Internet of things deepens, the number of terminals such as smartphones and smart glasses keeps increasing, making the growth rate of data far exceed the growth rate of network bandwidth. At the same time, many new applications, such as augmented reality and unmanned driving, put forward a higher demand for delay. According to the forecast of Cisco cloud index (GCI) in 2016, global data center traffic will reach 15.3 ZB by 2020. The Internet business solutions group (IBSG) also forecasts that the number of IoT devices will reach 50 billion by 2020 [4,5]. The interconnection of all things breaks the limit of the interconnection between the traditional things, which results in that the traditional cloud computing model cannot meet the application demand of the interconnection of all things [6,7].

There are three main reasons why cloud computing model cannot meet the application demand of Internet of everything [8–10].

(1) Multi-source heterogeneous data processing. The perception layer data of the Internet of things is at a massive level, and there are frequent conflicts and cooperation between the data, with strong redundancy, relevance, real-time and multi-source heterogeneous characteristics. The integration of heterogeneous multi-source data and real-time processing demands brings great challenges to cloud computing which cannot be solved.

(2) Bandwidth load and resource waste. Cloud service is a kind of centralized service computing with high degree of aggregation. Users send data to the cloud for storage and processing, which will consume a large amount of network bandwidth and computing resources. At the same time, a large number of user visits will increase network traffic, which will lead to service interruption, network delay and other problems.

(3) Limited resources. In the interconnection mode of everything, network edge devices are usually resource-limited (storage, computing capacity, battery capacity, etc.), and the energy consumption of long-distance transmission of data between edge devices and cloud computing centers is particularly prominent.

Due to the contradiction between the cloud computing model and the inherent characteristics of the Internet of everything, the centralized computing processing mode of cloud computing alone is insufficient to support the application operation and mass data processing in the context of the perception of the Internet of things. Moreover, cloud computing model has been unable to effectively solve problems such as cloud central load, transmission broadband and data privacy protection. Edge computing can effectively solve these problems. Edge computing is a new service model in which the data or task can be calculated and performed at the edge of the network near the data source.

In the edge network, any functional entity between the data source and the cloud computing center can run the edge computing platform of computing, storage and application of core capabilities, providing the end users with real-time, dynamic and intelligent service computing. Moreover, the combination of edge computing and existing cloud computing centralized processing model can effectively solve the problem of big data processing of cloud center and network edge [11–13].

Because edge computing can be combined with cloud computing, its network structure is usually fixed mode combining centralized with distributed mode. Therefore, its data storage mechanism is different from the traditionally centralized storage mechanism of cloud computing, as well as the common local storage and distributed storage mechanism. This article develops special research for this purpose. Our contribution is as follows.

(1) We propose a hybrid storage architecture that adapts to edge computing.
(2) We make full use of the advantages of edge network devices and cloud storage servers to make full use of their resources and avoid the waste of resources.
(3) We propose to use blockchain technology for data storage, which ensures the security of data storage in edge computing.

2. Related work

2.1. Smart computing and edge computing

Intelligent computing combined with Internet of Things (IoT), big data analytics, cloud computing, machine learning and artificial intelligence has many applications. For example, smart medical care collects body information, temperature, nutrition, humidity and motion information through IoT sensors, which are uploaded to the cloud server. When thousands of pieces of information form a big data database, cloud computing resources are used to perform powerful machine learning and artificial intelligence calculations to process this information. Ultimately get a report on human health, a healthy diet and a reasonable exercise recommendation, pre-symptom prediction of the disease, etc. [14,15].

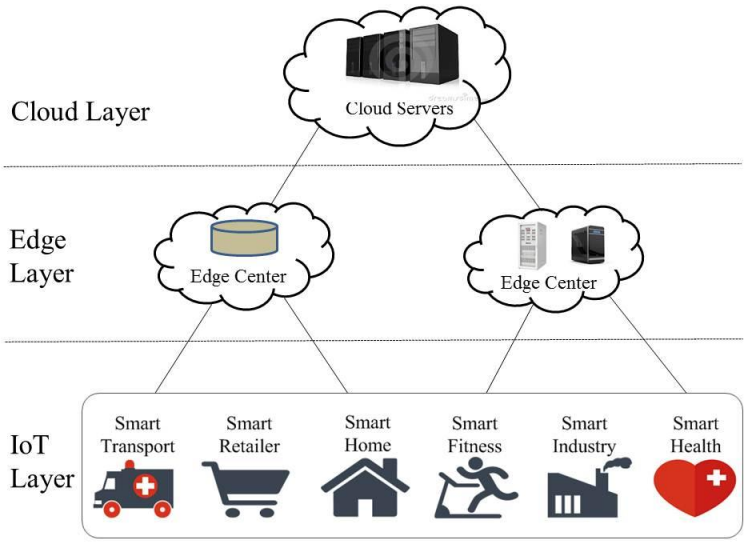

Figure 1. Architecture of edge computing.

Edge computing is also an important component of smart computing. The "edge" in edge computing is a relative concept that refers to any computing resource and network resource between the data source and the cloud service center. Edge computing allows terminal devices to migrate storage and computing tasks to network edge nodes, such as base station (BS), wireless access point (WAP), edge server, etc., which satisfies the computing power of the terminal equipment to expand demand and effectively saves computing tasks at the same time in the cloud resources transmission link between the server and terminal equipment. The architecture of Edge computing is shown in Figure 1, which mainly includes Cloud Layer, Edge Layer and IoT Layer [16].

(1) Cloud Layer mainly includes Cloud computing services and data centers. Cloud computing core services typically include three service patterns: infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS). Meanwhile, in the edge computing service mode, multiple cloud service providers are allowed to provide centralized storage and computing services for users at the same time. Therefore, large-scale computing migration between servers can be achieved by deploying multiple layers of heterogeneous servers, and real-time services and mobile agents can be provided to users in different geographic locations.

(2) Edge Layer: Edge Center is responsible for virtualization services and multiple management services. It is one of the core components in Edge computing, deployed by infrastructure providers, and equipped with multi-tenant virtualization infrastructure. Virtualization services provided by edge data centers can be used from third-party service providers to end users and infrastructure providers themselves. In addition, the edge side of the network tends to deploy multiple edge data centers which cooperate with each other while acting autonomously, but not disconnected from the traditional cloud. Therefore, it is possible to create a layered architecture interconnected by different network infrastructures to achieve a distributed collaborative computing service pattern. It is worth mentioning that the data security of the edge data center is always a concern of the end users [17].

(3) IoT Layer: edge network computing realizes the interconnection of IoT devices and sensors by integrating multiple communication networks from wireless network to mobile central network to Internet network. Mobile terminals in the IoT include all types of devices connected to the edge network (including mobile terminals and numerous IoT devices). They are not only the identity of the data consumer, but they can act as data providers participating in distributed infrastructure at all levels [18].

Due to the increase of data volume and the demand of real-time processing, the centralized data processing of cloud center will be transformed into the two-way computing mode of cloud and edge. Network edge devices not only act as service requesters, but also perform some computing tasks, including data storage, processing, search, management and transmission.

2.2. Edge computing and biological sciences

In recent years, smart computing technology has been widely used in many fields such as pattern recognition and artificial intelligence. With the implementation of the human genome project and the completion of more biological genome sequencing projects, the biological data are explosive and the traditional experimental determination methods are far from meeting the needs. The computational intelligence algorithm in smart computing has its unique advantages in the field of processing this kind of data with large volume, noise pattern and lack of unified theory. Edge computing is an important

tool in intelligent computing which has a wide range of applications in biological genetics and intelligent medicine.

Recent research shows that edge computing can provide nearby computing services, but generally only some simple data processing. For the sequencing industry, edge computing can assist in the preprocessing of some sequencing computations. Imagine the future, if you can realize the edge computing, the gene sequencing model will change, it will not send blood samples directly to a center again, instead of that the front end will do the pretreatment, the data of the pretreatment will transferred to the corresponding 'brain' when it need to be analyzed, and the analysis of the work is the core value. The new theories and technologies, such as intelligent edge computing, are bringing disruptive changes to tumor radIoTherapy, and breakthroughs will be made in the accuracy and efficacy of treatment. Intelligent omics radIoTherapy is expected to solve the dilemma faced by radIoTherapy for many years.

2.3. Data storage of internet of things

Data storage of internet of things can be divided into centralized storage, distributed storage and sensor network database. These strategies are discussed below, and their data storage and access costs are analyzed [18].

2.3.1. Centralized storage

Centralized storage [19–22] is the simplest data storage strategy. Each node transfers the collected perception data to the base station (sink node) for storage, while the data access directly obtains the data from the base station. Because the energy and storage space of the base station is not limited, the data can be stored for a long time and the data access does not consume the energy of the nodes in the network. At this point, the sensor network is only a means of data collection rather than data processing, because users can only get data from the base station database. In addition, when the network scale is large and the nodes are distributed densely, a large amount of data in the network needs to be transmitted. The nodes near the base station will consume energy too fast for forwarding data, which constitutes a bottleneck for the entire network. So it is not suitable for large-scale network.

All nodes in the centralized storage send data to the base station, and the cost of data access to the base station is negligible. So the total cost is $O(n\sqrt{n})$, where the base station receives data of all nodes at the same time. And the maximum cost of a single node is $O(n)$.

2.3.2. Distributed storage

Distributed storage [23–26] is a data-centric storage strategy. Its core idea is that the perceptual data generated by the node is not necessarily stored locally, but stored in other nodes using distributed technology. And an effective information intermediary mechanism is used to coordinate the relationship between the data store and data access to ensure that the data access request can be met. Under this policy, the data is stored according to the specific storage mechanism, and the query request also depends on the specific access mechanisms to obtain the data. These mechanisms include hash mapping, indexing, routing data and query requests by certain rules and so on. The advantage is that the distributed data storage well matches the distribution of the sensor network itself and the information mediation mechanism can guarantee that the data access request is met, while the disadvantage is that the information mediation requires additional costs.

In distributed storage, since the data is stored in s locations, the storage cost is $O(sn\sqrt{n})$. The query request must send any of the s locations at the cost of $O(n/s)$. If a node satisfies the query condition, the feedback cost is $O(n)$. When data store and query access intersect a single node, the maximum cost is $O(n)$ +RQuery.

2.3.3. Database in sensor network

Sensor network database [27–29] sets above three strategies as a whole which means closely combined traditional database technology, distributed technology and network technology, integrated the sensor nodes as perceptual data flow or data source and considered wireless sensor network as distributed perception database. From the logic concept, a data-centric sensor network database system with high performance was achieved [4]. Sensor network databases and data-centric routing complement each other. Routing is a bottom-up mechanism relative to data storage and access, while database relative data modeling and access is a top-down mechanism. Typical sensor network databases include OUGAR, Tiny-DB, PRESTO and StonesDB [30].

3. Problem statement

In edge computing environment, a large number of sensor nodes are usually deployed. The sampling data of most sensors (such as temperature sensor, GPS sensor, pressure sensor, etc.) is numerical, but there are also many sensors whose sampling values are multimedia data (such as traffic camera video data, audio sensor sampling data, remote sensing imaging data, etc.). Each sensor generates new sampling data frequently, and the system needs to store the latest version of the sampled data, and in most cases that all historical sampling values within a period of time such as one month in order to meet the requirements of traceability processing and complex data analysis. Since the data is massive, data storage will be an unprecedented challenge.

The same IoT systems can contain a variety of sensors, such as traffic sensor, hydrological sensors, geological sensors, meteorological sensors, biological medical sensors, etc. And each kind of sensors includes a number of specific sensors, which can be subdivided into GPS sensors, license plate recognition sensors, electronic photographic identification sensors, traffic flow sensors (infrared, coil, optics, video sensor), road sensors, sensors in perfect condition, etc. These sensors not only have different structures and functions, but also collect heterogeneous data. This heterogeneity greatly increases the difficulty of data storage. In addition， more complex traditional encryption algorithms, access control measures, identity authentication protocol and privacy protection methods cannot be applied in edge computing， on the one hand, because of the multi-source data fusion characteristics of edge computing and the superposition of mobile and Internet networks; on the other hand, there are resource limitations in storage, calculation and battery capacity of edge terminals.

Although edge computing has some locally or proximally advantage, it faces more serious security problems [31]. First of all, edge computing covers a lot of terminal devices, which have potential security problems. For example, 82% of Android devices have at least one of the 25 security vulnerabilities. Secondly, the network access of terminal devices in edge computing is diverse, and the security of these networks is difficult to guarantee, making them more vulnerable to attack. According to statistics, 80% of home routers use the default password and 89% of public wi-fi hotspots are unsafe. In addition, in edge computing, nodes are distributed across the local network and cloud, making it more difficult to implement security protection across domains. Moreover, some of the sensing

equipment has very limited resources, which makes many of the existing security protection technologies unavailable for direct use.

Blockchain is a kind of decentralized, tamper-resistant, traceability, jointly safeguard of distributed database. It can integrate traditional single-party maintenance of multiple isolated databases involving only its own business and distributed storing multiple nodes that are jointly maintained by multiple parties. Neither party can fully control these data which can only be updated according to strict rules and consensus, thus achieving trusted information sharing and supervision among multiple parties in an untrusted environment [32–34]. With the application of blockchain technology, trusted information can be shared among multiple parties without any third-party trusted institutions. Blockchain technology is well suited for edge computing where security demands are strong. However, the blockchain network is a completely distributed architecture, while the edge computing environment is a centralized distributed architecture. How to use blockchain technology for secure data storage in edge computing has become an outstanding problem that needs to be studied.

4. Data security storage mechanism based on blockchain and coding

In storage systems, redundancy is often introduced to improve system reliability. As an important redundant strategy, erasure code strategy gets more and more attention. Common remedy delete code for (n, k) maximum distance separable (maximum short separable, MDS) code, such as Reed - Solomon code. When data loss occurs to a storage node, the original data can be restored through redundant coding. Usually, when a node restores lost data, the amount of data transmitted is greater than the storage capacity of the node [35]. This results in huge bandwidth consumption.

To solve the problem of erasure code, Dimakis et al. proposed the regenerative code of optimized repair bandwidth [36]. As an improvement of erasure code, regenerative code not only maintains the property of erasure code MDS but also greatly reduced the network bandwidth required in the repair of failed data by introducing the concept of network coding [37–40]. Based on the regeneration code and blockchain, this paper constructs the data security storage mechanism in the edge computing environment.

4.1. Data security storage model in edge network

The data storage model in the edge network is shown in Figure 2. The storage model is divided into three layers and corresponding different components: (1) cloud server group, (2) global blockchain network, (3) edge network center, (4) local blockchain, (5) IoT terminal equipment.

(1) Cloud server cluster

In edge computing, the top layer is the cloud service layer. In the cloud service layer, there are a large number of cloud servers, which can provide the function of cloud storage service. Users can easily access data at anytime and anywhere, through any internet-connected device connected to the cloud. More importantly, the cloud server cluster has a large number of storage resources, which can provide infinite storage space for the edge networks with limited resources [41].

(2) Global blockchain network

To ensure the security of the data stored in the cloud server, we build a global storage mechanism based on blockchain technology. Each cloud storage server has a copy of the data block and can verify and store the data through the block structure, and use the consensus algorithm of distributed cloud storage server to generate and update the data.

In addition, in order to further improve the reliability and security of data, we use regenerative code to provide data redundancy. When a data loss error occurs in a cloud storage server, it can be recovered again by redundant regeneration code. Further, improve the reliability of cloud storage servers.

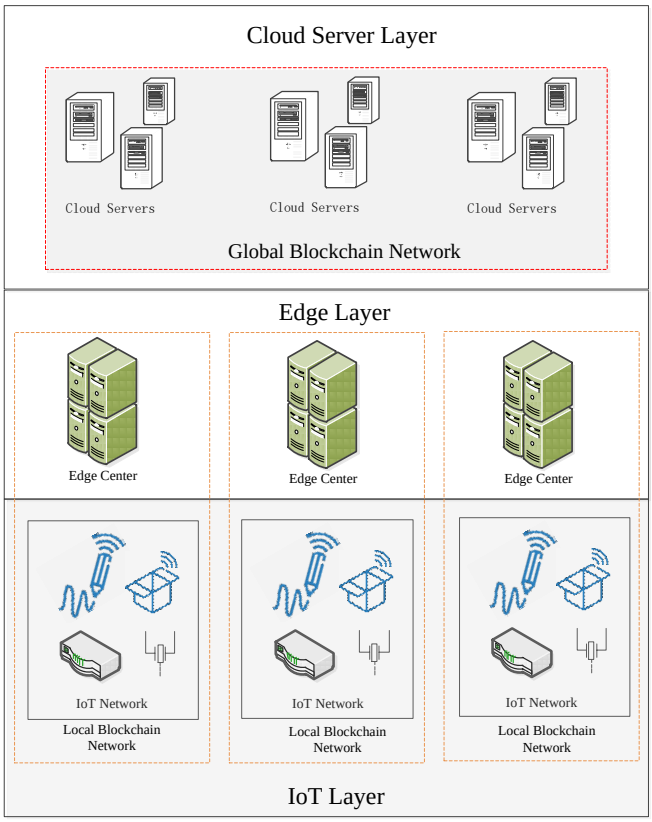

Figure 2. The model of data storage based on blockchain in edge computing.

(3) Edge Center

Edge Center is part of the Edge network that manages a set of IoT network devices. These IoT network devices form an edge network. IoT network devices with limited resources and capabilities are managed by the Edge Center and do not prevent them from participating in Edge networks due to their hardware limitations. At the same time, Edge Center is responsible for interacting with the upper layer cloud server, and transferring data and information in the local Edge network periodically to the cloud server for storage.

Any entity can be registered as an IoT network device. However, IoT network devices must be registered under the control of the Edge Center in order to avoid edge network devices being added to edge networks without the permission of edge center nodes. In addition, all registered IoT network devices in the system must belong to an edge center. Otherwise, no node can manage the device [42,43].

(4) Local blockchain

The Edge Center, along with the IoT network devices it manages, builds a local blockchain that stores the data collected by each IoT network device. Local blockchain is a lightweight blockchain due to the limited storage resources of IoT network devices. When the data stored in the local IoT device reaches the upper limit, the data is hashed. And the data is transmitted and stored to the local blockchain. At the same time, data in IoT devices will be deleted. Thus, IoT network devices can collect data again.

(5) IoT terminal equipment

In edge computing, various objects can be connected to the Internet through intelligent and non-intelligent sensors, so that intelligent applications such as information collection, environmental monitoring and health management can be realized [44,45]. The basic core is the information interaction between objects and objects as well as between people and objects. It extends the communication dimension of people in the world of information and communication technology from any time, any place, and any connection to anyone, to any connection to any object. Everything is connected via the Internet to form the Internet of things [46].

4.2. Data security storage combining blockchain and regenerative code

4.2.1. Cloud service layer data storage based on global blockchain

In the cloud service layer, there are many cloud servers which can be considered to have infinite storage space and can be used to store all data in the edge computing environment. However, although cloud storage brings convenience and economic benefits to users, but it also poses serious threats to users' data security. Particularly it brings many new challenges in guaranteeing the integrity and correctness of user data. Due to the influence of hardware, software, operating system or human operation, cloud server will inevitably change or even delete users' data by mistake. Therefore, data security has become the biggest obstacle to the further development of cloud storage services.

To this end, the system adopts the blockchain technology to directly build the global blockchain in the cloud storage server and store the data sent from each Edge Center. Blockchain technology has the characteristics of decentralization, non-tampering and trust removal of block data, which make the storage of the data more secure. Moreover, this system adopts the structure of cloud, and constructs the structure of distributed storage. Distributed data storage can further improve system reliability, availability and access efficiency, and is easy to expand.

At the same time, in order to further improve the reliability of the data, this paper continued to adopt the regenerative code technology in the cloud storage and carried out redundant coding and storage of the original data. It is shown in Figure 3.

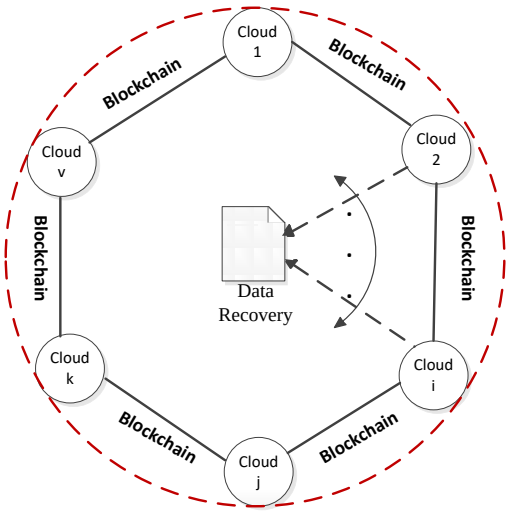

Figure 3. Data storage based on global blockchain in cloud layer.

In general, the storage system based on erasure code is to cut the original file into k original data blocks, then encode them and form n coding blocks. These coding blocks are stored in n different storage nodes. When the original data needs to be restored, k coding blocks are obtained from any k storage nodes and then decoded to recover the original data. Erasure code is much less expensive to store than multiple copies at the same fault tolerance. However, when repairing the failed data, erasure code needs to transfer k blocks from k nodes. The erasure code strategy increases network overhead compared to the multi-copy policy which only transmits 1 data block.

In regenerative code, each node stores a coding block composed of a coding segment. When repairing, the coding segment is firstly combined in a linear manner on the node providing the recovery data, so as to obtain the restoration block composed of b (b is far less than a) repair segments and then the restoration block is transmitted to the newborn node. The new node then decodes the collected repair blocks to get the invalid coding blocks. Although the number of connection nodes is greater than k, the regenerative code does not need to download the entire coding block. Therefore, the regenerative code fundamentally reduces the amount of data transmitted.

Regenerative codes can greatly reduce the amount of data transmitted when repairing, but the amount of data that needs to be read is much larger. Therefore, based on blockchain, this paper adopts the combination of data duplication and regenerative code technology to improve the reliability of data storage.

When data is stored in a cloud environment, the original data k is encoded and stored on n cloud servers, and each cloud server stores the encoded data with size of aps. When a cloud server fails, it can connect any of the remaining n-1 cloud servers to download d landscape for repair, which is called Repair Bandwidth. The parameter set of the regenerative code scheme can be expressed as {[n, k, d], (yes, yes, B)}, and its average repair bandwidth d is smaller than file size B. The Minimum Bandwidth restate-owned (MBR) Code has the lowest repair Bandwidth. The amount of data that needs to be stored during the data repair process of commonly used MBR code is too large, which increases the disk load of the system and limits the speed of the repair. Repair by Transfer (RBT) Code can only transmit data without any mathematical calculation when data is restored, so the amount of data to be read is the same as the amount of data to be transferred, which is more suitable for use in the storage system. Therefore, this system adopts RBT.

Suppose M and is a data matrix, and its form is as follows:

$$M = \begin{bmatrix} S & \mathrm{T} \\ -\mathrm{T}^{t} & 0 \end{bmatrix}$$

Where 0 is the zero vector of $(n-k)\times(n-k)$, T is the filling vector of $k\times(n-k)$, and S is an antisymmetric matrix. Antisymmetric matrix has the following properties: $S=-S^{t}$. As a result, $S[i,j]=-S[j,i]$. The coding matrix $\psi=[\phi\ \Delta]$ is a matrix of $n\times n$, where 0 in the $\Delta=\begin{bmatrix}0\\ I_{n-k}\end{bmatrix}$ is the zero vector, and I_{n-k} is the identity matrix. Thus, RBT is encoded as $C=\phi M\phi^{t}$. Its symmetric matrix is $C^{'}$. Let's say that for every behavior in matrix C, we have

$$c_{j}^{'}=\begin{cases} c_{j}[i]\ i\geq j \\ -c_{j}[i]\ i<j \end{cases}$$

The minimum repair bandwidth point in MBR codes is : $(\alpha,\beta)=(\frac{2Bd}{2kd-k^{2}+k},\frac{2Bd}{2kd-k^{2}+k})$. Because the system uses blockchain technology, it can use the copy of the blockchain stored in other cloud servers for data repair. Therefore, only the repair data of $(k-1)$ coding nodes is needed in data repair. In the case that data copies exist, according to literature [] : $C(\alpha)=B+\sum_{i=0}^{k-2}min\{b_{i},\alpha\}$, $\alpha\in[0,b_{k-2}]$, $b_{i}=(1-\frac{k-2-i}{d}\beta)$.

Therefore,

$$C(\alpha)=\begin{cases} B+(k-1)\alpha & \alpha\in[0,b_0] \\ B+b_0+(k-2)\alpha & \alpha\in(b_0,b_1] \\ \qquad\qquad \text{L} & \\ B+b_0+\text{L}\ +b_{k-3}\alpha & \alpha\in(b_{k-3},b_{k-2}] \end{cases}$$

The minimum value satisfying $C(\alpha)\geq B$ is

$$\alpha'=\begin{cases} \dfrac{B}{k-1}\ B\in[B,B+(k-1)b_0] \\ \dfrac{B-\sum_{j=0}^{i-1}b_j}{k-1-i}\ B\in[B+\sum_{j=0}^{i-1}b_j+(k-i-1)b_{i-1},B+\sum_{j=0}^{i-1}b_j+(k-i-2)b_i] \end{cases}$$

In addition, $\sum_{j=0}^{i-1}b_j=\sum_{j=0}^{i-1}(1-\frac{k-j-2}{d})\,\beta=\beta i[\frac{d-k+2}{d}+\frac{i-1}{2d}]=\beta g(i)$ there are:

$$\sum_{j=0}^{i-1}b_j+(k-i-2)b_i$$

$$=\sum_{j=0}^{i-1}(1-\frac{k-j-2}{d})\,\beta+(k-i-2)(1-\frac{k-2-i}{d})\beta$$

$$=\beta(i+1)[\frac{d-k+2}{d}+\frac{i-1}{2d}]+(k-i-2)(\frac{d+2-k}{d}+\frac{i}{d})\beta$$

$$=\beta(\frac{(k-1)(d-k+2)}{d}+\frac{(2k-i-3)}{2d})$$

From above: $\alpha'=\frac{B-g(i)\beta}{k-i-1}$. Therefore, the minimum repair bandwidth point with data copy is: $(\alpha',\beta')=(\frac{2Bd}{(2d-k+2)(k-1)},\frac{2Bd}{(2d-k+2)(k-1)})$

4.2.2. IoT layer data storage based on local blockchain

The IoT layer is a mass of sensors that sense information, work together to build local networks and connect disparate objects to the Internet. Compared with Edge Center and cloud server, a large number of IoT terminals have limited resources in storage, communication and computing. Thus, this paper proposes that each IoT network builds local blockchain.

Any IoT device in our system can store an encoded fragment of each data block. Even if each data block is not stored in its original form, each block is completely stored in a different IoT device, and each IoT device stores different fragments. In most cases, the initial block can be restored by combining any k code fragments provided by any node in the IoT network.

Most IoT terminals are sensors with very limited power. To some extent, the energy determines the performance of the local IoT network. Therefore, this paper proposes that energy - aware coding is applied to the data storage of local blockchain. In the process of information transmission in the local IoT network, when the intermediate node i sends the data, we hope to select the node with the most remaining energy in the neighbor node of i to send in order to reduce the probability of premature exhaustion of the energy of the next hop node. According to literature [], the current average residual energy(ARE) of each network device is as follows: $Are(j)=N(i)RE(j)/\sum_{k\in N(i)}RE(k)$. N(I) refers

to the number of neighbor nodes of node i. The ARE of network device is the relative value of the average residual energy in all the neighboring nodes of its previous hop node. By using the relative residual energy, a more enough node can be selected as the network device of information transmission in multiple nodes with sufficient energy which can be used for encoding and recovery of lost data. It is shown in Figure 4.

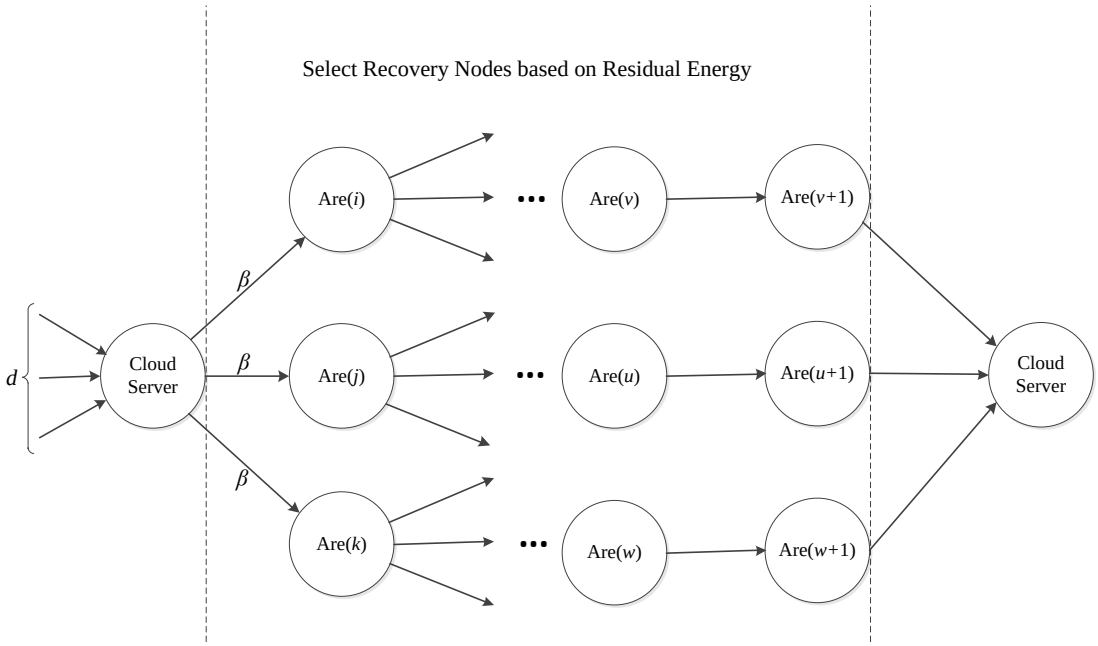

Figure 4. Data storage based on local blockchain in IoT layer.

The scheme of the IoT terminal equipment with high residual energy to collaborate on coding and data repair is as follows. The source file M is divided into many blocks and distributed on k local terminal devices, and the data of each local terminal device is divided into k parts. These nodes form a k by k matrix M_1, and M_i^T (I = 1,2... K), each column in the matrix M_1 represents the data information of each local terminal. The matrix M_1 is:

$$M_1 = \begin{bmatrix} a_1^1 & a_2^1 & \mathrm{L} & a_{k-1}^1 & a_k^1 \\ a_1^2 & a_2^2 & \mathrm{L} & a_{k-1}^2 & a_k^2 \\ & & \mathrm{L} & & \\ a_1^k & a_2^k & \mathrm{L} & a_{k-1}^k & a_k^k \end{bmatrix}$$

In order to increase the reliability of the system, the data of the source file is coded with the same code to generate $n-k$ redundant terminal equipment. The data structure of redundant terminal equipment is represented by the $k \times n$ order matrix M_2, and each column M_i^T $(i = k+1, \cdots, n)$ represents the data information of each redundant terminal equipment. The matrix M_2 is:

$$M_2 = \begin{bmatrix} a_{k+1}^1 & a_{k+2}^1 & \mathrm{L} & a_n^1 \\ a_{k+1}^2 & a_{k+2}^2 & \mathrm{L} & a_n^2 \\ & & \mathrm{L} & \\ a_{k+1}^k & a_{k+2}^k & \mathrm{L} & a_n^k \end{bmatrix}$$

All the storage codes in matrix M_1 and M_2 constitute a complete local network data storage system. When some nodes are damaged, it uses d local terminal devices to fix the data on these terminals. The data size that each node needs to download is the β, and the total number of data M_i^T that needs to be transmitted is $d-1$, to form the $k \times (d-1)$ order matrix. All the data used to repair the node is concentrated on the repair device, and the data used to repair the failed node is the $k \times d$ order matrix, and set as C. The linear relationship between the repaired data γ of each terminal device and the data size of the terminal is expressed as the matrix of initial α × γ infinite matrix $N(i)$ $(2 \le i \le n-k)$. Through linear calculation, the recycled data of each size α is finally obtained, which is represented by the matrix M':

$$M' = \begin{bmatrix} N^1 \\ N^2 \\ N^3 \\ \mathrm{M} \end{bmatrix} C$$

The data downloaded from recovery node is $\sum_{i=1}^{k}(d-\sum_{j=0}^{i-1} l_j)\,\beta > 0$, so there is the following formula: $k\alpha + \sum_{i=1}^{k}(d-\sum_{j=0}^{i-1} l_j)\,\beta > k\alpha$.

4.3. Data transfer

Our blockchain scheme includes the local blockchain and global blockchain. Local blockchain is created by edge centers. And they store the data collected by IoT devices and its the hash value. And global blockchain stores all data block. Data in local blockchains is periodically uploaded to the global blockchain. Our scheme provides a mechanism for periodically validating hash values of data to ensure data integrity. One hash value is calculated from data in global blockchain, the other is from local blockchains.

4.3.1. Data transfer from IoT device to local blockchain

At first, each IoT device of local edge network will be authenticated by its manager (edge center). The legal devices will be assigned a public-private key pair. A local blockchain is created by edge centers. Devices without public-private key pairs assigned by edge center cannot participate in the local blockchain.

All IoT devices of the edge network can collect data. When the collected data reaches the upper limit of an IoT device, the hash values of the data blocks are calculated. The hash values of the collected data is signed using the private key that distributed by the edge center.

And the hash values are broadcast to edge network for verification. After validation, the hash values are written to the local blockchain. The hash values can construct a Merkle tree. The root of the Merkle tree is included in the head of local blockchain. Moreover, the data is sent to edge center to storage. That is to say, the edge center is the global node of the local blockchain; while the IoT devices are the light nodes.

After the IoT device empties the stored data, it will be able to collect data again. The specific process is shown in Figure 5.

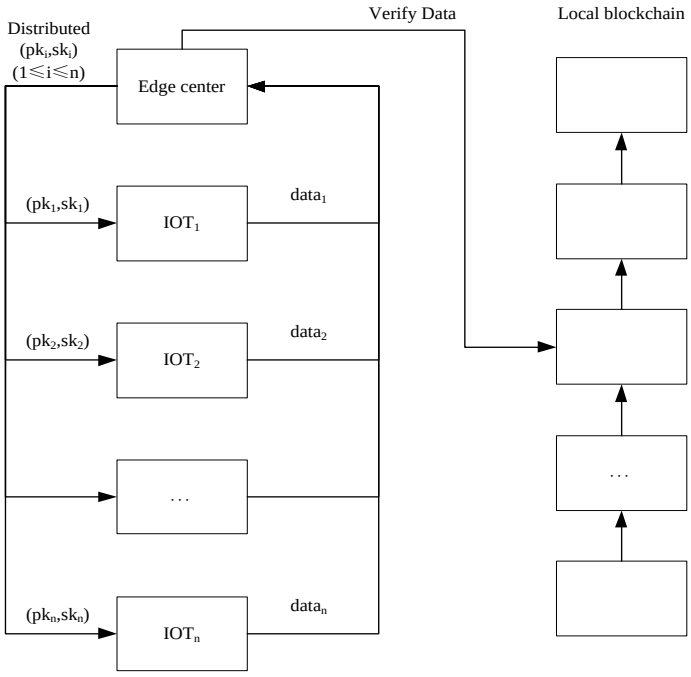

Figure 5. Data transfer of Local blockchain.

4.3.2. Data transfer from local blockchain to global blockchain

Data in local blockchains can be periodically uploaded to the global blockchain. The data uploaded by edge center is divided into blocks. Then, the hash values of data blocks are calculated by cloud servers. The hash values are compared with the hash values stored in the local blockchain to ensure that the data is not changed when uploaded to the cloud server. The specific process is shown in Figure 6.

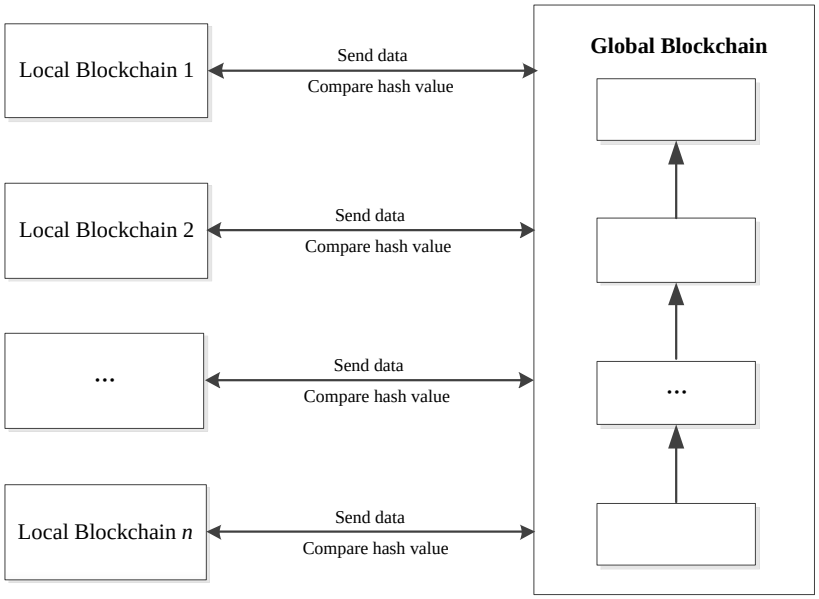

Figure 6. The verify of the cloud server.

The detailed procedures:

(1) Data in local blockchains is periodically uploaded to the global blockchain.
(2) The cloud servers compute the hash values uploaded to the global blockchain. Edge center compares the hash values with the hash values stored in the local blockchain. If they are equal, the data written to the global block chain is correct.
(3) After validation, the hash values are written to the global blockchain. The hash values can construct a Merkle tree. The root of the Merkle tree is included in the head of global blockchain.
(4) The verification process of the hash values mentioned above can be carried out periodically to ensure the integrity of data in the global blockchain.

4.4. Performance evaluation

4.4.1. Data storage based on global blockchain and regenerative code

We use Aliyun and Amazon S3. Compared with single cloud storage service, the response time of distributed cloud storage service composed of multiple cloud storage services is more stable and faster. This is because blockchain-based cloud storage provides multiple copies of data and is able to leverage the network bandwidth of multiple cloud storage services, overcoming the bandwidth shortfalls of a single cloud server.

In the case of blockchain-based replica repair, the repair bandwidth and disk I/O are reduced when using regenerative code to repair failed data. And as the redundancy m increases, it helps to repair the reduction in bandwidth and disk I/O. In addition, m can be effectively reduced for both average repair bandwidth and average disk I/O. When $m = i$, as a_1 decreases, n increases, and n_0 grows the fastest.

Because: $n = (\frac{\delta\sqrt{a_1/(k-m)}+\sqrt{(\delta^2 a_1+4a_1(k-m))/(k-m)}}{2a_1})^2 \times (k-m)+m$, $\delta$is the standard deviation of the normal distribution of the desired availability level. As a result, the bandwidth and disk I/O required to repair will be significantly reduced for highly available cloud storage devices. If the server is less available, you can introduce a high availability server as a replica repair node to avoid large β that consume large amounts of storage space. In addition, because average repair bandwidth is proportional to total storage, a decrease in average repair bandwidth also means a decrease in total storage. Therefore, in the regeneration code, because the α' doesn't change, the increase of m will lead to the increase of total storage, the maximum increase$(\beta-1)m\alpha'$.

4.4.2. Data storage based on local blockchain and regenerative code

Given that our system can significantly reduce the amount of storage required by nodes without significantly increasing CPU decryption, running blockchain on IoT devices is a very important step. We tested local blockchain and encoded data recovery based on remaining energy at Raspberry Pi. Raspberry pie is a small, single-board computer with the same hardware architecture of common IoT devices and smartphones, which perfectly reflects the processing power of average IoT devices or smartphones in experiments. Common coding algorithms don't often implement parallel computing, so you can easily increase the speed by using four threads of a quad-core ARM processor on Raspberry Pi.

The hash values calculation and the Merkle tree construction are efficient, which donot require too much computing and storage resources [29,33,35]. In order to analyze the influence of the

heterogeneity of nodes' computing power on the data repair process, we tested the influence of node residual energy coefficient and node's transmission and repair time on data. In the actual test process, the higher the value of ARE, the longer time the node can process the information and data; On the contrary, the shorter. Figure 7 shows how the repair time of the whole process changes with the remaining energy. The larger the residual energy value is the more stable the node is, and the longer the data processing time is.

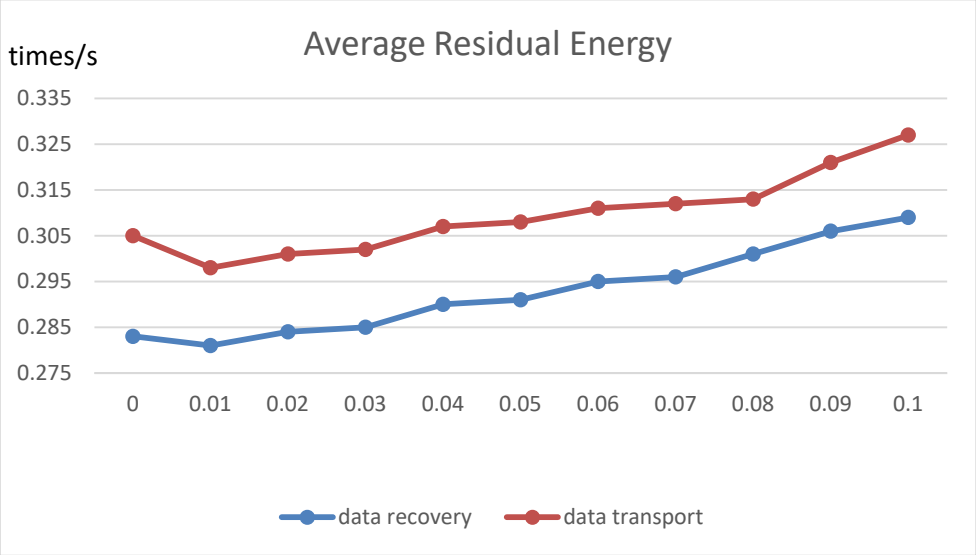

Figure 7. Influence of ARE on data recovery and data transport.

5. Conclusions

The security of data storage under edge computing has become an obstacle to its widespread use，which also affects the development of smart computing. In the paper, the mechanism combing blockchain with regeneration coding is proposed to improve the security and reliability of stored data under edge computing. At first, according to the three-tier edge computing architecture and data security storage requirements, hybrid storage architecture and model under edge computing are proposed. Secondly, making full use of the advantages of edge network devices and cloud storage servers, a global blockchain in the cloud service layer is built. In addition, the regeneration coding is utilized to further improve the reliability of data storage. Moreover, the local blockchain is built on the terminals of the Internet of things which realized the second verification. After the data is stored in the cloud, it can be compared and verified with the data in the local blockchain which further ensuring the data security. According to the residual energy of nodes, the terminals are selected, who are repaired with recovering data through regeneration coding. Thus, the resources of each device under edge computing can be fully developed and the waste of resources can be avoided.

Acknowledgments

This work is supported by the NSFC (61772280, 61772454, 61811530332, 61811540410), the PAPD fund from NUIST. Professor Jin Wang is the corresponding author.

Conflicts of interest

All authors declare no conflict of interest in this paper.

References

1. A. Khan, M. Othman, S. Madani, et al., A survey of mobile cloud computing application models, *IEEE Commun. Surv. Tut.*, **16** (2014), 393–413.
2. G. Dan, K. Jeremy, S. Evan, et al., Cloud-Trust—a security assessment model for infrastructure as a service (IaaS) clouds, *IEEE T. Cloud Comput.*, **5** (2017), 523–536.
3. J. Wang, Y. Cao, B. Li, et al., Particle swarm optimization based clustering algorithm with mobile sink for WSNs, *Future Gener. Comp. Sy.*, **76** (2017), 452–457.
4. S. Li, L. Xu, S. Zhao, et al., Internet of things: A survey, *J. Ind. Inf. Integr.*, **10** (2018), 1–9.
5. P. Ray, A survey on Internet of Things architectures, *J. King Saud University-Comput. Inf. Sci.*, **30** (2018), 291–319.
6. L. Bittencourt, R. Immich, R. Sakellariou, et al., The Internet of Things, fog and cloud continuum: integration and challenges, *Internet. Things*, **4** (2018), 134–155.
7. R. Olaniyan, O. Fadahunsi and M. Maheswaran, Opportunistic edge computing: concepts, opportunities and research challenges, *Future Gener. Comp. Sy.*, **89** (2018), 633–645.
8. P. Hu, D. Sahraoui and H. Ning, Survey on fog computing: architecture, key technologies, applications and open issues, *J.Netw. Comput. Appl.*, **98** (2017), 27–42.
9. S. Ola, E. Imad and C. Ali, IoT survey: an SDN and fog computing perspective, *Comput. Netw.*, **143** (2018), 221–246.
10. J. Zhang, Y. Zhao, B. Chen, et al., Survey on data security and privacy-preserving for the research of edge computing, *J. Comm.*, **39** (2018), 1–21.
11. T. Taleb, K. Samdanis, B. Mada, et al., On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration, *IEEE Commun. Surv. Tut.*, **19** (2017), 1657–1681.
12. M. Marjanović, A. Antonić, I. Podnar, et al., Edge computing architecture for mobile crowdsensing, *IEEE Access*, **6** (2018), 10662–10674.
13. D. Zeng, Y. Dai, F. Li, et al., Adversarial learning for distant supervised relation extraction, *Comput. Mater. Con.*, **55** (2018), 121–136.
14. Y. Tu, Y. Lin, J. Wang, et al., Semi-supervised learning with generative adversarial networks on digital signal modulation classification, *Comput. Mater. Con.*, **55** (2018), 243–254.
15. C. Yin, J. Xi, X, Sun, et al., Location privacy protection based on differential privacy strategy for big data in Industrial Internet of Things, *IEEE T. Ind. Inform.*, **14** (2018), 3628–3636.
16. S. N. Shirazi, A. Gouglidis, A. Farshad, et al., The extended cloud: review and analysis of mobile edge computing and fog from a security and resilience perspective, *IEEE J. Sel. Area. Comm.*, **35** (2017), 2586–2595.
17. E. B. Tirkolaee, A. A. R. Hosseinabadi, M. Soltani, et al., A hybrid genetic algorithm for multi-trip green capacitated arc routing problem in the scope of urban services, *Sustainability*, **10** (2018), 1–21.
18. W. D. Wang and J. Y. Lang, Reflection and prospect: precise radiation therapy based on bionomics/radionics and artificial intelligence technology, *Chinese J. Clin. Oncol.*, **45** (2018), 30648–30656.
19. R. Khanna, H. Liu and T. Rangarajan, Wireless data center management: sensor network applications and challenges, *IEEE Microw. Mag.*, **15** (2014), S45–S60.

20. C. Yang, D. Puthal, S. Mohanty, et al., Big-sensing-data curation for the cloud is coming: a promise of scalable cloud-data-center mitigation for next-generation IoT and wireless sensor networks, *IEEE Consum. Electr. M.*, **6** (2017), 48–56.
21. J. Wang, C. Ju, H. Kimet, et al., A mobile assisted coverage hole patching scheme based on particle swarm optimization for WSNs, *Cluster Comput.*, **3** (2017), 1–9.
22. R. Meng, S. Rice, J. Wang, et al., A fusion steganographic algorithm based on faster R-CNN, *Comput. Mater. Con.*, **55** (2018), 1–16.
23. B. Gong, P. Cheng, Z. Chen, et al., Spatiotemporal compressive network coding for energy-efficient distributed data storage in wireless sensor networks, *IEEE Commun. Lett.*, **19** (2015), 803–806.
24. Y. Yang, J. Miao, Y. Zhao, et al., Distributed information storage and retrieval in 3D sensor networks with general topologies, *IEEE ACM T. Network.*, **23** (2015), 1149–1162.
25. J. Wang, J. Cao, S. Ji, et al., Energy-efficient cluster based dynamic routes adjustment approach for wireless sensor networks with mobile sinks, *J. Supercomput.*, **73** (2017), 3277–3290.
26. L. Xiang, Y. Li, W. Hao, et al., Reversible natural language watermarking using synonym substitution and arithmetic coding, *Comput. Mater. Con.*, **55** (2018), 541–559.
27. O. Diallo, R. Joel, M. Sene, et al., Distributed database management techniques for wireless sensor networks, *IEEE T. Parall. Distr.*, **26** (2015), 604–620.
28. A. Al-Dhaqm, S. Razak, S. H. Othman, et al., CDBFIP: common database forensic investigation processes for Internet of Things, *IEEE Access*, **5** (2017), 24401–24416.
29. Y. Ren, Y. Liu, S. Ji, et al., Incentive mechanism of data storage based on blockchain for wireless sensor networks, *Mob. Inf. Syst.*, **2018** (2018), 1–10.
30. S. Fang, L. Xu and Y. Zhu, An integrated system for regional environmental monitoring and management based on Internet of Things, *IEEE T. Ind. Inform.*, **10** (2014), 1596–1605.
31. D. E. Kouicem, A. Bouabdallah and H. Lakhlef, Internet of things security: A top-down survey, *Comput. Netw.*, **141** (2018), 199–221.
32. C. Wei, Z. Wang, E. Jason, et al., Decentralized applications: the blockchain-empowered software system, *IEEE Access*, **6** (2018), 53019–53033.
33. F. Tschorsch and B. Scheuermann, Bitcoin and beyond: a technical survey on decentralized digital currencies, *IEEE Commun. Surv. Tut.*, **18** (2016), 2084–2123.
34. R. B. Uriarte and R. D. Nicola, Blockchain-based decentralized cloud/fog solutions: challenges, opportunities, and standards, *IEEE Commun. Standards Mag.*, **2** (2018), 22–28.
35. K. Govinda, P. Nin, V. Lalitha, et al., Codes with local regeneration and erasure correction, *IEEE T. Inform. Theory*, **60** (2014), 4637–4660.
36. J. Yao, K. Zhang, Y. Yang, et al., Emergency vehicle route oriented signal coordinated control model with two-level programming, *Soft Comput.*, **22** (2018), 4283–4294.
37. Y. Hu, L. Patrick, K. W. Shum, et al., Proxy-assisted regenerating codes with uncoded repair for distributed storage systems, *IEEE T. Inform. Theory*, **64** (2018), 2512–2528.
38. Y. Chen, L. Wang, X. Yan, et al., Mimic storage scheme based on regenerated code, *J. Commun.*, **39** (2018), 21–34.
39. J. Wang, C. Ju, Y. Gao, et al., A PSO based energy efficient coverage control algorithm for wireless sensor networks, *Comput. Mater. Con.*, **56** (2018), 433–446.
40. J. Wang, Z. Zhang and B. Li, An enhanced fall detection system for elderly person monitoring using consumer home networks, *IEEE T. Consum. Electr.*, **60** (2014), 23–29.
41. Y. Ren, J. Shen, D. Liu, et al., Evidential quality preserving of electronic record in cloud storage, *J. Internet Technol.*, **17** (2016), 1125–1132.
42. X. Li, A fast and exhaustive method for heterogeneity and epistasis analysis based on multi-objective optimization, *Bioinformatics*, **33** (2017), 2829–2836.
43. X. Li, J. Y. Peng, J. W. Niu, et al., A robust and energy efficient authentication protocol for Industrial Internet of Things, *IEEE Internet Things J.*, **5** (2018), 1606–1615.

44. J. Wang, J. Y. Cao, R. Simon, et al., An improved ant colony optimization-based approach with mobile sink for wireless sensor networks, *J. Supercomput.*, **74** (2018), 6633–6645.
45. J. Wang, Y. Gao, X. Yin, et al., An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks, *Wirel. Commun. Mob. Com.*, **18** (2018), 1–9.
46. X. Li, J. W. Niu, M. Bhuiyan, et al., A robust ECC based provable secure authentication protocol with privacy preserving for industrial internet of things, *IEEE T. Ind. Inform.*, **14** (2017), 3599–3609.

AIMS Press