

Architecture Design of Distributed Encrypted Storage and Computing Based on Fog Computing

Qiang Gu ^{1, a}

¹CHINA MOBILE GROUP JIANGSU CO.,LTD

^a guqiang@js.chinamobile.com

Abstract. With the progress of informatization, the problems of data transmission delay, untimely response and theft and tampering of user privacy data are becoming increasingly prominent. This paper proposes a distributed encryption storage and computing architecture based on fog computing, which uses the scalability and low latency of fog computing to solve the problems of data silos and response delays. Blockchain technology is introduced to build decentralized storage to prevent data tampering and ensure secure transmission and sharing. Hierarchical privacy protection measures are designed to store data in a hierarchical and classified manner to accurately control the scope of information acquisition. This method has good application characteristics and can effectively solve the problems of government data collection and calculation.

Keywords: Fog Computing; Distributed Storage;;Big Data; Privacy Protection,;Access Control.

1. Introduction

In recent years, with the advancement of informatization, the demand for government affairs of urban residents has shifted to the information platform, and the amount of data has grown rapidly. The spatiotemporal data stream is transmitted to the cloud through the Internet of Things for cleaning and storage[1]. However, due to the large number of terminal sensors and data collection apps, the data transmission and storage process is time-consuming, resulting in delayed or unavailable cloud response.

Existing research mainly focuses on the combination of blockchain and the Internet of Things, or the single combination of cloud computing and fog computing. For example, research[2] proposed a hierarchical capacity supply solution that uses random sorting and optimization algorithms to solve network latency problems. Reference[3] designed a network edge component scheduling algorithm to achieve detailed control of data scheduling. In terms of the combination of blockchain and the Internet of Things, research focuses on trusted design and application. Research[4] used the Shamir secret sharing algorithm to encrypt private keys and publish them on the blockchain to protect transmitted data. Literature[5] combined blockchain, environmental signatures and RSSI to form a distributed heterogeneous Internet of Things authentication system. At present, the research results are widely used in the fields of electricity[6-8], agriculture[9], transportation [10], and medical care[11].

In terms of fog computing resource scheduling, Hoseiny[12] proposed a volunteer computing method that efficiently allocates tasks through the Min-CCV and Min-V algorithms. Choudhari[13] proposed a priority-based task scheduling algorithm to improve the efficiency of fog computing tasks and reduce time costs. Li[14] proposed a scheduling strategy based on deep reinforcement learning to solve the scheduling problem through a hierarchical framework and algorithm. Reference[15] proposed the application of fog computing in a distributed greenhouse environment and designed a fog computing model for the intelligent greenhouse Internet of Things to improve the temperature control accuracy and stability. The above studies mostly solve the fog computing problem from a single or two directions, but lack innovation in the application of fog computing combined with Internet of Things cloud computing, have not formed an overall technical framework, lack practical applications, and are difficult to cope with the task requests and data processing needs of a large number of terminal devices.

Based on existing research, this paper proposes a new idea of combining fog computing[16] with blockchain, and proposes a distributed encrypted storage and computing architecture based on fog computing for government data collection and computing. The main innovations of this paper are as follows:

1. A distributed encryption storage method based on fog computing is proposed. This method focuses on the collection and calculation of government data. On the basis of considering the actual application scenarios and data characteristics, a scheme for integrating IoT terminal data and performing hierarchical encryption transmission is proposed.

2. It is proposed to improve the existing terminal data islands and untimely data response problems by utilizing the scalability and low latency characteristics of fog computing and combining with the Internet of Things.

3. The connection between the terminal and the management center is designed to reduce the pressure of data transmission. And a computing and storage application framework that conforms to the characteristics of government data is built to enable the overall architecture to have better practical application capabilities.

2. Overall model design

This architecture design combines the Internet of Things with fog computing, calculates and manages data through fog nodes, and uses the AMQR protocol to achieve high-reliability communication and message routing. Data is transmitted between fog nodes through network protocols, and the processing results are returned to the cloud and IoT devices. Then, priority indicators are used to monitor the access of IoT devices integrated with smart contracts, and the resource allocation of edge servers is controlled according to the rules of smart contracts. Then, a hierarchical encrypted storage architecture is designed to process the massive data generated by the interaction between blockchain and IoT to protect sensitive information. Finally, data is encrypted for transmission to solve the problem of data transmission encryption between blockchain nodes and the cloud.

2.1 Underlying Design of Distributed Storage Based on Fog Computing

The bottom layer of IoT devices includes various client devices, which communicate with each other, request edge server resources or send data to external servers, mainly processing personal data. We combine IoT devices with fog computing resources, use smart contracts for interface interaction, and assist data interaction through blockchain. The IoT gateway performs behavior detection, monitors access to edge node devices, calls computing storage resources and controls malicious attacks, improving information processing speed and blockchain efficiency.

To solve the problem of high consumption of blockchain computing resources, we combine distributed storage architecture and fog computing to divide IoT edge devices into central node computing resources, mobile computing resources and small fixed computing resources (fog nodes). The central computing resource platform deploys powerful computing resources, assists terminal mobile resource computing and provides storage services. Small fixed computing resources form fixed fog computing nodes. Mobile computing resources (such as mobile phones and cars) are used for data collection and feedback. Through this integration, the computing and storage efficiency of the overall architecture is improved.

2.2 Distributed Big Data Privacy Information Storage Architecture Combined With Blockchain

In the actual application process, the user information transmitted by the mobile terminal device often contains a large amount of private information, which needs to be encrypted for storage and transmission of sensitive information. Finally, the data transmission process is shown in Figure 1.

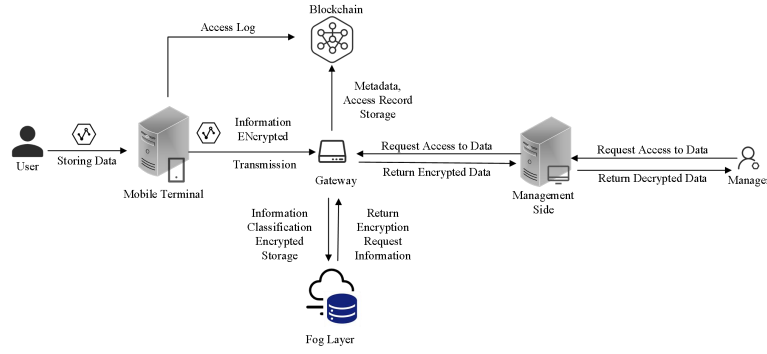


Fig. 1 Distributed big data privacy information storage architecture

The user encrypts and transmits the data to the small fog computing resource node for storage and calculation through the mobile terminal, and the metadata design of all information is stored on the block node of the blockchain to prevent tampering of important data. Among the data, sensitive data related to users needs to be encrypted and stored. If the data needs to be called, an application must be submitted during the use process, and the identity of the applicant must be verified to confirm whether it has access rights. Nodes responsible for processing requests need to submit information to small fixed computing resource nodes, namely fog nodes.

Multiple fog nodes are combined to form an independent transaction processing layer, which uniformly processes all information processing requests. The connection between fog nodes is shown in the Figure 2:

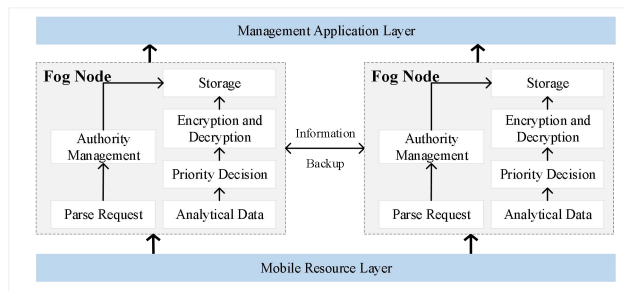


Fig. 2 Fog Node Internal Design

Set up a docking interface for the information of terminal IoT devices, and the interface is responsible for exchanging information about mobile devices and related service requests. Use the Retroshare protocol as a communication protocol between fog nodes to transfer important private data stored by users inside the nodes. Among the nodes, the module responsible for storage mainly stores a large amount of transactional data generated by users in their daily work, marks the source of the data, and performs asymmetric encrypted storage of sensitive information. According to the blockchain distributed storage idea, the data is backed up to the adjacent fog nodes at the same time to ensure that the data has the ability to restore the original data when it is subjected to malicious attacks. At the same time, when the user needs to call the stored data, he needs to use the private key in the digital signature to encrypt the information, and then enter the request reading module to verify the digital signature. If the verification result matches the hash value, the download can be performed. step. In order to solve a large number of concurrent requests in actual application scenarios, a separate module is designed for transaction priority processing. The final processed data will be transmitted to the cloud data application center.

This design can effectively reduce the amount of data stored in the cloud, and at the same time reduce the time it takes for users to obtain data. Cloud computing tasks are effectively offloaded to fog nodes. Compared with edge computing, fog computing has better scalability and can cope with the problem of overall equipment expansion brought about by the subsequent expansion of user groups at any time. At the same time, fog nodes can also access cloud computing resources later, and maintain traceability by updating the transaction list in the blockchain.

2.3 Sensitive data hierarchical encryption authorized access

This section designs a dynamic hierarchical processing model for user privacy-sensitive data access roles, which aims to effectively protect user privacy data and improve the interactivity of access control. Through interactive management, the model realizes the processing of user privacy data access requests, the identification of access roles, the implementation of access control policies, and the application of authorization decision-making for access information.

For the fog computing resource end, the main tasks are to store data and control access to data calls. Data types are divided according to different privacy levels, different access permissions are set, and access to the general information database and the privacy information database is classified according to the authorization level, and the access level is changed according to the dynamic changes in user identity permissions. For the mobile end, as a collector of data resources, after collecting data indiscriminately, irrelevant data is eliminated, and then the data is classified. Data with important privacy levels is encrypted and transmitted and stored in the privacy information database, and the rest of the data is stored in the general information database.

2.4 Computing resource scheduling design

In order to effectively solve the problem of effective resource scheduling when transactional requests are issued at the same time, we propose a method for judging request priority based on transaction size and urgency. First, a priority comparison function is defined to compare the priorities of two tasks. In the actual parameter collection, the name, transaction size and urgency of the task are received as parameters and stored in the corresponding class attributes. In this way, unique identification and task attributes are generated for each task object. Then compare the transaction size of the task. If a task has a large amount of transactions, it is considered to have a higher priority. If two tasks have the same transaction volume, we further compare their urgency. Higher urgency will result in higher priority. Finally, if the transaction volume and urgency of two tasks are the same, it is determined that the two tasks have the same priority. By determining the priority of tasks, computing resources can be allocated reasonably, the execution order of tasks can be optimized, and the overall performance and efficiency of the system can be improved. The specific ideas are as follows:

Step 1: Define the task class. The task class ('Task') contains two attributes: task name ('task name') and priority ('priority').

Step 2: The constructor of the task class receives the task name and priority as input parameters and initializes the task object.

Step 3: Create a task list ('tasks'), which contains three tasks with priorities of "low" ('low'), "medium" ('medium'), and "high" ('high').

Step 4: Allocate fog computing resources and execute tasks. For each task in the task list, call the task's 'assign_fog_resource' method to allocate fog computing resources and perform corresponding operations. If the priority of the task is "high" ('high'), allocate fog computing resources with high priority, and then execute fog computing operations of high priority tasks. If the priority of the task is "medium" ('medium'), allocate fog computing resources with medium priority, and then execute fog computing operations of medium priority tasks. The priority is "low" ('low'), and so on.

Tasks are then prioritized based on urgency. The tasks in the system allocate fog resources according to their priority requirements, and perform corresponding fog computing operations. There are three levels of priority for tasks: high, medium, and low. In the main program, a task list is created, and the task list is traversed in order, and fog resources are allocated according to the priority requirements of the tasks and corresponding fog computing operations are performed. By allocating fog resources according to the priority requirements of tasks, resources can be optimally utilized according to the importance and urgency of tasks, and the execution efficiency of tasks and the performance of the overall system can be improved.

Step 1: Define the task class `Task`, which contains three attributes: task name (`task name`), transaction size (`transaction size`), and urgency (`urgency`).

Step 2: Constructor: Initialize the task object and receive the task name, transaction size, and urgency as input parameters.

Step 3: Define the task comparison function `comparePriority` method: used to compare the priorities of two tasks. If the transaction size (`transaction size`) of the current task is larger than that of another task, return `True`; if the transaction sizes are equal, compare the urgency (`urgency`), and the task with higher urgency has higher priority; if the transaction size of the current task is smaller than that of another task, return `False`.

Step 4: Create a task list `tasks` and note the urgency.

Step 5: Sort tasks by priority `sorted_tasks`: Use the `comparePriority` method to sort the task list and determine the priority of the task based on the transaction size and urgency.

Step 6: Output the sorted task list, traverse the sorted task list, and output the name, transaction size, and urgency of each task in turn.

Through the above steps, priority-based task sorting is achieved, and tasks are effectively sorted by transaction size and urgency.

3. Practical application of communication information

In actual applications, the personal account information, business needs and communication logs of communication users constitute the core of business big data. These data are stored in the internal database of the enterprise. In order to improve storage and utilization efficiency, the data is classified and processed. High-level permissions are required to access highly sensitive data, such as name, ID number, account password, and communication number; middle-level management permissions can access regular business data, such as communication time, location, and billing; business personnel permissions can access low-confidentiality data, such as business packages, service logs, business activation and expiration time. The data originates from the Internet of Things and mobile device transmission, and the process includes resource sorting, request coordination, resource allocation, logging and upload management.

During the data capture process, the Internet of Things gateway uses the smart contract event listener to receive and analyze requests and automatically execute operations. Sort the request list according to predefined rules, determine the urgency, and generate a filtered request list for execution. The request is sent to the resource management module for resource allocation. After the execution starts, the identity of the request initiator is verified, and the database information is accessed after confirming the permissions, and sensitive data is returned in a graded manner. During the whole process, all data will be recorded and saved on the fog end. The specific process is shown in Figure 3.

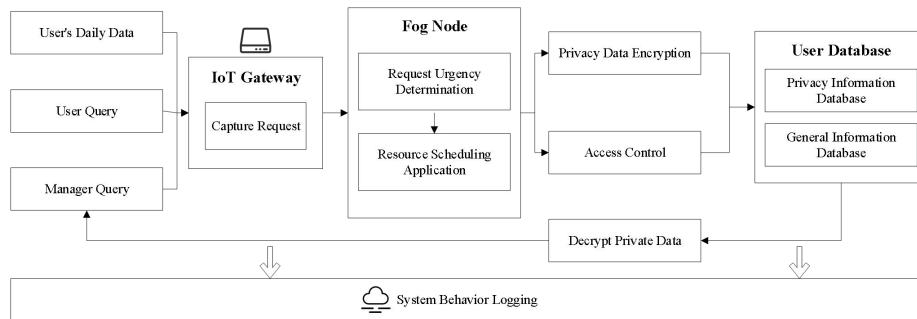


Fig. 3 Schematic diagram of the actual application process

4. Summary

This paper proposes a distributed encrypted storage and calculation method based on fog computing. According to the actual application scenarios and data characteristics, this method proposes a scheme to integrate the terminal data of the Internet of Things and carry out hierarchical encrypted transmission. Among them, by utilizing the scalability and low-latency characteristics of

fog computing, combined with the Internet of Things, the problems of terminal data islands and untimely data response are solved. In order to reduce the pressure of data transmission, a computing and storage application framework that conforms to the characteristics of government data is built on the cloud, which improves the actual application ability of the overall architecture. In future work, the framework proposed in this paper can be further improved and perfected, combined with specific use cases for detailed description and empirical analysis. In addition, more innovative technologies and methods can be explored to further improve the efficiency and security of big data storage and computing.

References

- [1] Ding W, Zou J, Zhao Z. A multidimensional service template for data analysis in highway domain[J]. *International Journal of Internet Manufacturing and Services*, 2020, 7(4): 290-306.
- [2] Kiani A, Ansari N, Khreishah A. Hierarchical capacity provisioning for fog computing[J]. *IEEE/ACM Transactions on Networking*, 2019, 27(3): 962-971.
- [3] Charántola D, Mestre A C, Zane R, et al. Component-based scheduling for fog computing[C]//*Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion*. 2019: 3-8.
- [4] Tan L, Yu K, Yang C, et al. A blockchain-based Shamir's threshold cryptography for data protection in industrial internet of things of smart city[C]//*Proceedings of the 1st Workshop on Artificial Intelligence and Blockchain Technologies for Smart Cities with 6G*. 2021: 13-18.
- [5] Mostafa A. Blockchain-based distributed authentication Mechanism for internet-of-things devices[C]//*Proceedings of the 2020 9th International Conference on Software and Information Engineering (ICSIE)*. 2020: 159-164.
- [6] Baozhi Z, Junyan Y, Rongsheng L, et al. Research on the Application of Blockchain technology in Ubiquitous Power System Internet of Things[C]//*Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications*. 2019: 118-123.
- [7] Mohammed S, Fiaidhi J, Ramos C, et al. Blockchain in eCommerce: A special issue of the ACM transactions on internet of thingsBlockchain in eCommerce: A special issue of the ACM transactions on internet of things[J]. *ACM Transactions on Internet Technology (TOIT)*, 2021, 21(1): 11-55.
- [8] Baozhi Z, Junyan Y, Rongsheng L, et al. Research on the Application of Blockchain technology in Ubiquitous Power System Internet of Things[C]//*Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications*. 2019: 118-123.
- [9] Jie C, Guixiang Z, Junhui W, et al. Research and Progress on The Application of Blockchain Technology in Agricultural Product Traceability Systems[C]//*Proceedings of the 6th International Conference on Big Data and Computing*. 2021: 206-211.
- [10] He M, Bai F, Zhang C, et al. A Blockchain-Enabled Location Privacy-preserving under Local Differential Privacy for Internet of Vehicles[C]//*Proceedings of the 2022 4th Blockchain and Internet of Things Conference*. 2022: 84-91.
- [11] Khaloufi H, Abouelmehdi K, Beni-Hssane A. Fog computing for smart healthcare data analytics: An urgent necessity[C]//*Proceedings of the 3rd international conference on networking, information systems & security*. 2020: 1-5.
- [12] Hoseiny F, Azizi S, Shojafar M, et al. Joint QoS-aware and cost-efficient task scheduling for fog-cloud resources in a volunteer computing system[J]. *ACM Transactions on Internet Technology (TOIT)*, 2021, 21(4): 1-21.
- [13] Choudhari T, Moh M, Moh T S. Prioritized task scheduling in fog computing[C]//*Proceedings of the ACMSE 2018 Conference*. 2018: 1-8.
- [14] Li H, Ota K, Dong M. Deep reinforcement scheduling for mobile crowdsensing in fog computing[J]. *ACM Transactions on Internet Technology (TOIT)*, 2019, 19(2): 1-18.
- [15] Yuan G M, Wang M N, Ding C J, et al. Design of greenhouse intelligent sensing and control system based on IoT and fog computing[J]. *Sensors and Microsystems*, 2020.
- [16] Stojmenovic I, Wen S, Huang X, et al. An overview of fog computing and its security issues[J]. *Concurrency and Computation: Practice and Experience*, 2016, 28(10): 2991-3005.