2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

A cross-blockchain based distributed crowdsouring system for Internet of Things

1st Min Peng
School of Mathematics and Information
Fujian Normal University
FuZhou, China
Email:pmachenll@163.com

2nd QinYang Miao
School of Mathematics and Information
Fujian Normal University
FuZhou, China
Email:18291972715@163.com

3*rd Hui Lin
School of Mathematics and Information
Fujian Normal University
FuZhou, China
Email:linhui@fjnu.edu.cn

4*th XiaoDing Wang
School of Mathematics and Information
Fujian Normal University
FuZhou, China
Email:wangdin1982@163.com

***Abstract*—With the development of Internet of Things (IoT) technologies, the mobile crowdsourcing has become the most effective and reliable method for data collection and processing. However, as huge amounts of data are generated in mobile crowdsourcing services, data privacy and security has become a key challenge for IoT. To address this challenge, a Cross-Blockchain based Distributed mobile Crowdsourcing System, named CBDCS, is proposed in this paper. In CBDCS, to overcome the privacy leakage problem caused by a single-point failure, the blockchain is used as a crowdsourcing task server, based on which an improved multi-subchain structure is designed. Then, based on multi-subchain, a task hierarchical storage mechanism is proposed, which stores different levels of tasks in corresponding subchains to improve privacy protection. In addition, CBDCS meet the demands of workers for tasks through a cross-subchain mechanism based on root&sub CA and smart contracts, while guaranteeing data privacy. Extensive simulation results demonstrate that the CBDCS can obtain high throughput, low overhead, and data privacy under various IoT scenarios.**

***Index Terms*—Blockchain, Cross-subchain, Mobile Crowdsouring, Internet of Things.**

I. Introduction

With the rapid development of advanced low-cost sensors, wireless communications, and network technologies, the Internet of Things (IoT) has emerged as a new communication paradigm, which promotes the diversification of smart applications, i.e., smart traffic control systems [1], smart cities [2], and Intelligence wireless communication [3]. The architecture of IoT is usually divided into three layers: perception layer, network layer, and application layer, which are responsible for data collection, data transmission, and data application respectively. In IoT, the mobile crowdsourcing is employed for effective data collection and processing, which are the prerequisites for IoT applications, i.e., city planning [4], smart navigation [5], traffic monitoring [6], etc.

Despite the mobile crowdsourcing is widely used and has the characteristics of high efficiency and low cost, the traditional mobile crowdsourcing technology has following deficiencies. For example, traditional mobile crowdsourcing technologies usually require a centralized task server, which rises a potential risk of privacy leakage caused by single point failure [7]. In addition, without effective data protection mechanisms, traditional mobile crowdsourcing technologies cannot prevent malicious workers from stealing private task data. To overcome above problems, a Cross-Blockchain based Distributed mobile Crowdsourcing System, named CBDCS, is proposed in this paper. The principal contributions of this paper are as follows.

1) To solve the problem of single point failure, we design a multi-subchain based blockchain structure, using multi-subchain as the mobile crowdsourcing task server.
2) To reduce the possibility for malicious workers to obtain private task information, we propose a novel task hierarchical storage strategy based on the multi-subchain structure.
3) To mitigate the demands of honest workers for mobile crowdsourcing tasks, we propose a cross-subchain mechanism based on root&sub CA and smart contracts, which allow honest workers to obtain tasks from different subchains.

The structure of this article is organized as follows. In the second section, related work is introduced. The third part introduces CBDCS in detail, the fourth part provides the experimental evaluation, and the fifth part gives the conclusion.

II. Related Work

The development of the blockchain technologies [8] has prompted the emergence of blockchain based mobile crowdsourcing. Li et al. [9] designed a distributed mobile crowd-

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61702103 and No. U1905211, and in part by the Fujian Provincial Natural Science Foundation of China under Grant (No. 2020J01167, No. 2020J01169).

978-1-6654-1485-2/20/$31.00 ©2020 IEEE

DOI 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00062

sourcing framework based on the blockchain CrowdBC, CrowdBC using smart contracts to record each step of the mobile crowdsourcing task, but CrowdBC did not propose a task evaluation mechanism. Bhatia et al. [10] proposed a mobile crowdsourcing system based on Ethereum smart contracts, and at the same time implemented credit management for workers, allowing task requesters to select reliable workers to the greatest extent.

Although the above work solved the problem of a single point of failure of the mobile crowdsourcing task server, it did not consider the problem of mobile crowdsourcing privacy. At present, the privacy and security problems in mobile crowdsourcing have received extensive attention and some research results have been obtained. Yang et al. [11] proposed a blockchain-based privacy protection mobile crowdsourcing perception system, which uses the anonymity of the blockchain to protect the true identity of employees. Lin et al. [12] proposed a blockchain-based security and privacy protection mobile crowdsourcing system SecBCS. In SecBCS, each task request or execution uses group signatures for identity verification. Due to the anonymity of group signatures, any real identity of the user will not be revealed.

For the cross-chain access problem studied in this paper, there are also some works. Jiang et al. [13] proposed a cross-subchain interactive distributed access model in the context of data management, and applied IPFS and BigchainDB on notary nodes to design a notarization mechanism for cross-network maintenance to solve the problems of data storage and device labeling. He et al. [14] proposed a joint operation mechanism for cross-subchain transactions, using blockchain technology to design two chains, the main chain, and the side chain, so that the two chains can share data.

Although the above work helps to design a secure distributed mobile crowdsourcing system, there are still following challenges: (i) how to prevent privacy leakage caused by the single point failure; (ii) how to effectively protect task data and prevent malicious workers from obtaining private task data; (iii) how to meet the demands of workers for tasks while protecting task privacy.

III. The Cross-Blockchain Based Distributed Crowdsouring System

The system model of CBDCS mainly includes three subchains, one Repchain, three sub-CAs, and one root CA.

1) *Subchain*: A blockchain used to store mobile crowdsourcing tasks of different levels.
2) *Repchain*: A blockchain used to store the credit of mobile crowdsourcing workers.
3) *Root CA*: It serves the Repchain, generates digital certificates of different levels for all workers based on the credit, and then sends certificates to sub-CAs in different subchains.
4) *Sub-CA*: The three sub-CAs respectively serve the three subchains. In addition to saving certificates sent by the root CA and registering workers in the current subchain, the sub-CA will also generate backup certificates for workers on the current subchain to prevent the root CA malfunction.

In CBDCS, all workers will join the Repchain and a subchain with the same credit level as their own. Workers can only access the data in the Repchain and this subchain. The proposed scheme is discussed in detail below. Table I give the notations used throughout this paper.

TABLE I
Main notations and symbols

Notations	Description
CC_r	The cross-subchain request
Pk_{cu}	The public key of cross-subchain worker
$Credit_c$	The Workers' credit changes count
$Cert_{cc}$	The Cross-subchain certificate
$Credit_i$	The access credit of target subchain
$Credit_w$	The current credit of worker
$Credit_{update}$	The update credit of worker
$FScore$	The task final score
$IScore$	The task integrity score
$QScore$	The task quality score
$TScore$	The task time score
$IScore_a$	The actual task integrity score to operation
$QScore_a$	The actual task quality score to operation
$TScore_a$	The actual task time score to operation
ω_I	The weight of $IScore_a$
ω_Q	The weight of $QScore_a$
ω_T	The weight of $TScore_a$
M	The worker's historical task score matrix
β	The reward of task level
T_{Ascll}	The ASCLL code of task level
$Tlevel$	The task level
n	The number of tasks completed by worker
m	The number of tasks with positive credit growth
H_c	The proportion of tasks with positive credit growth

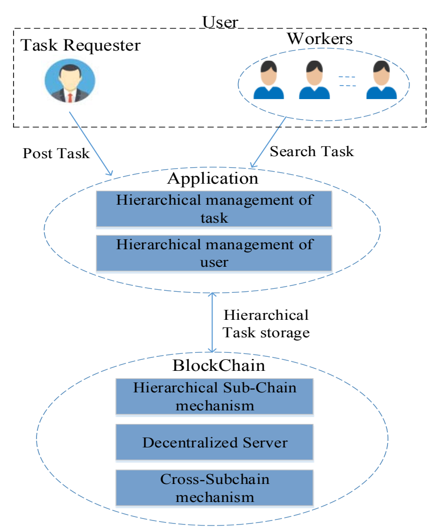


Fig. 1. Architecture of CBDCS.

The architecture of CBDCS, as shown in Figure 1, consists of five modules. In CBDCS, firstly, by using the hierarchical management of the task method, when the task requester posts the task, the mobile crowdsourcing task can be divided into different levels according to the privacy and security requirements of the task. Secondly, based on the Hyperledger Fabric, a multi-blockchain structure is designed in CBDCS and combines with the tasks hierarchical that put different levels of task into different sub-blockchain. Moreover, the subchain architecture is used as a distributed server to overcome the single point of failure problem of traditional mobile crowdsourcing servers. Finally, based on the multi-subchain structure, we designed a cross-subchain mechanism-based the root&sub-CA and smart contract that meets the demands of the mobile crowdsourcing business. While protecting private data, it can provide more tasks for honest workers.

A. Task hierarchical storage strategy

To reduce the possibility that malicious workers obtain private information of task, in this paper, a task hierarchical storage strategy is proposed.

First, in the task post-stage, the task requester can divide the task into specific levels of tasks according to its own needs and the privacy and security requirements of the task. Secondly, in the task storage stage, based on the Hyperledger Fabric, a multi-blockchain structure is designed as distributed servers to overcome the single point of failure problem and implement the hierarchical storage of task by storing the different level of tasks as transactions on the different sub-blockchain.

An example of task hierarchical storage is shown in Figure 2. In CBDCS, tasks can be divided into A, B, and C three levels from high to low. The task requester posts tasks of different levels and stores them in subchains of different levels. When a worker searches for tasks, he can only search for tasks with the same credit level as him. As shown in the figure, a worker with a credit level of C can only search for tasks in the subchain with a level of C. Until his credit level reaches a higher level, he cannot view the tasks in the higher-level subchain.

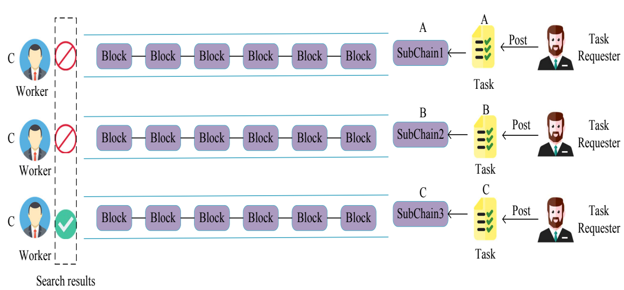


Fig. 2. The example of task hierarchical storage based multi-blockchain.

B. Cross-subchain access mechanism

In CBDCS, to reduce the probability of malicious workers stealing task privacy information, a multi-subchain structure is designed based on the Hyperledger fabric to achieve the hierarchical storage of tasks. If the worker's credit level does not meet the requirements for accessing the subchain, the worker cannot view the tasks stored on the subchain. Only if the worker's credit level is the same as that of the subchain can the worker view the tasks on the subchain. At the same time, workers with a high level of credit can not view tasks in the lower levels of the subchain, ensuring strict data isolation. But in the actual application of mobile crowdsourcing, tasks on a single subchain may not meet the demands of honest workers for tasks. For example, in the subchain of the current level, the number of tasks is less than the number of workers, which may result in some workers not applying for tasks for a long time.

Although the multi-subchain structure realizes strict data isolation and protects the privacy of task data, it also limits the demand for honest workers for tasks to a certain extent. In order to meet the needs of honest workers for the task, CB-DCS proposes a cross-subchain access mechanism to provide honest workers with the opportunity to apply for tasks across the chain. The following details the proposed cross-subchain access mechanism.

1) The Cross-subchain steps: The cross-subchain process mainly involves cross-subchain workers, *Creditjudgment* contracts, the root CA of the Repchain, and the sub-CA of the target subchain.

- *Cross-subchain users*: The worker who submit cross-subchain request;
- *Credit judgment contract*: A smart contract that judges whether the worker's credit meets the cross-subchain requirements;
- *Root CA*: The CA that serves the Repchain and generates certificates of different levels for workers according to their credit conditions. At the same time, when there have a cross-subchain request, generates a temporary cross-subchain certificate for the cross-subchain worker.
- *Sub-CA*: The CA in the target subchain. It only generates certificates for users in the current subchain, and at the same time saves the certificate from the root CA, and registers the cross-subchain workers in the current subchain.

The specific cross-subchain process is described as follows. The worker submits a cross-subchain request CC_r and public key Pk_{cu} in the Repchain, and CC_r is used as an event to trigger the worker's credit judgment contract $CreditJudgment$, which judges whether the worker's current credit $Credit_w$ and the worker's recent credit change count $Credit_c$ meet the cross-subchain requirements. If the conditions meet the requirements, $CreditJudgment$ sends the cross-subchain user's public key Pk_{cu} and the worker's credit $Credit_w$ to the root CA, otherwise it refuses the request. Then, the root CA generates a time-sensitive cross-subchain certificate $Cert_{cc}$ for cross-subchain workers based on the worker's public key Pk_{cu}, and then sends $Cert_{cc}$ to the target subchain CA. Moreover, the target chain sub-CA saves the cross-subchain certificate $Cert_{cc}$ of the cross-subchain workers, and then registers the cross-subchain workers in the current subchain. After the registration is completed, the cross-subchain workers can access the tasks in the target subchain within the validity period of the certificate.

After the smart contract $CreditJudgment$ is triggered, it needs to determine whether the worker's credit meets two conditions: The worker's current credit $Credit_w$ is not lower than the access credit value $Credit_i$ set by the target subchain; the worker's recent credit change count $Credit_c$ must not be less than 18 in the last 20 tasks. Note that $Credit_c$ represents the worker's recent credit change count. The value of $Credit_c$ is 0 or 1, and it is 1 when the credit is positively increased, and it is 0 when the credit is negatively increased. Through the above conditions, it can be ensured to a large extent that cross-subchain workers are not malicious workers, and the quality of tasks completed recently is high.

2) The Cross-subchain certificate: After the smart contract $CreditJudgment$ determines that the worker meets the cross-subchain requirements, it will send the CC_r, Pk_{cu} and $Credit_w$ to the root CA. The root CA generates a time-effective cross-subchain certificate for the worker. Compared with traditional X.509 digital certificates, cross-subchain certificates mainly have the following improvements. Specifically, this scheme adds the original chain and the target chain to the digital certificate. The certificate serves as a cross-subchain trust certificate, recording the source and whereabouts of workers. After the worker passes the cross-subchain judgment, the root CA generates a cross-subchain certificate for the worker requesting the cross-subchain, and records it in the blockchain in the form of a hash value. In addition, this scheme eliminates the signature algorithm module in the digital certificate. In this solution, because the blockchain is inherently non-tamperable, the root CA of the Repchain generates a cross-subchain certificate and records the hash value of the certificate into the blockchain as a cross-subchain trust certificate, instead of the CA's signature process for the certificate.

C. Hierarchical management of workers

In CBDCS, hierarchical management is implemented for workers. CBDCS divides workers and tasks into three levels: A, B, and C. The initial value of each worker's credit is 10. The credit value range corresponding to the credit level is as follows:A $\in [100, +\infty)$, $B \in [50, 100)$, $C \in [10, 50)$. CBDCS considers the task completion time, task completion quality, task completion integrity, and the historical situation of the worker's task completion to comprehensively evaluate the worker's credit. When the task is finished, the task requester will give an evaluation score to the worker. The evaluation score is composed of three parts: task time score $TScore$, task quality score $QScore$, and task integrity score $IScore$, both are in the range [0,100]. The method of worker credit update as:

$$Credit_{update} = Credit_w + FScore * \beta * H_c * m * Tlevel \quad (1)$$

In Equation 1, $Credit_{update}$ represents the credit after update, $Credit_w$ represents the current credit of the worker. $FScore$ represents the final score of the worker completing the task, the calculation method is as follows.

$$FScore = \frac{IScore_a}{\omega_I} + \frac{QScore_a}{\omega_Q} + \frac{TScore_a}{\omega_T} \quad (2)$$

In Equation 2, $IScore_a$, $QScore_a$, $TScore_a$ respectively represent the actual score of the operation, the calculation method is as follows.

$$IScore_a = \begin{cases} IScore - 100, IScore < 50 \\ IScore, IScore > 50 \end{cases} \quad (3)$$

$$QScore_a = \begin{cases} QScore - 100, QScore < 50 \\ QScore, QScore > 50 \end{cases} \quad (4)$$

$$TScore_a = \begin{cases} TScore - 100, TScore < 50 \\ TScore, TScore > 50 \end{cases} \quad (5)$$

In Equation 3, 4 and 5, ω_I, ω_Q, ω_T respectively represent the weight of $IScore_a$, $QScore_a$, $TScore_a$, ω_I, ω_Q, ω_T are calculated from the worker's historical task score matrix M, the calculation method is as follows.

$$M = \begin{pmatrix} IScore_1 & QScore_1 & TScore_1 & \alpha_1 \\ IScore_2 & QScore_2 & TScore_2 & \alpha_2 \\ \vdots & \vdots & \vdots & \vdots \\ IScore_i & QScore_i & TScore_i & \alpha_i \end{pmatrix} \quad (6)$$

$$\omega_I = \begin{pmatrix} IScore_1 & IScore_2 & \cdots & IScore_i \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_i \end{pmatrix} = IScore_1 * \alpha_1 + IScore_2 * \alpha_2 + \cdots + IScore_i * \alpha_i \quad (7)$$

$$\omega_Q = \begin{pmatrix} QScore_1 & QScore_2 & \cdots & QScore_i \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_i \end{pmatrix} = QScore_1 * \alpha_1 + QScore_2 * \alpha_2 + \cdots + QScore_i * \alpha_i \quad (8)$$

$$\omega_T = \begin{pmatrix} TScore_1 & TScore_2 & \cdots & TScore_i \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_i \end{pmatrix} = TScore_1 * \alpha_1 + TScore_2 * \alpha_2 + \cdots + TScore_i * \alpha_i \quad (9)$$

In Equation 6, 7, 8 and 9, $IScore_i$, $QScore_i$, $TScore_i$ represent the score obtained by the worker on the ith task, α_i indicate the positive and negative growth of the worker's credit during the ith task, if the credit is positive growth, α_i=1, otherwise, α_i=0. β represents the bonus of the task level to the score, the value of β corresponding to the task level from high to low is 3,2,1. H_c represents the percentage of positive increase in worker credit, the calculation method is as follows.

$$H_c = \frac{m}{n} \tag{10}$$

In Equation 10, m represents the number of tasks with positive increase in worker credit, n represents the total number of tasks performed by the current worker. $Tlevel$ represents the weight of the task level. In this paper, the task level Ascll code is used to calculate the weight of the task level, the calculation method is:

$$Tlevel = \begin{cases} \frac{T_{Ascll}}{100}, n = 1 \\ 1 - \frac{T_{Ascll}}{\sum_{j=1}^{n} T_{Ascll}^{j}}, n > 1 \end{cases} \tag{11}$$

In Equation 11, $\sum_{j=1}^{n} T_{Ascll}^{j}$ represents the Ascll code sum of all tasks performed by worker.

IV. PERFORMANCE EVALUATION

A. Experiment setup

The experimental platform is built by Hyperledger Fabric1.2. The physical machine is equipped with Intel Core i5 CPU3.2GHZ with 16 GB RAM win7 system, virtual computer software uses the VMware Workstation 14 Pro, the virtual machine is 8GB of memory, allocated 2 processors, 40GB of Ubuntu system.

B. Performance of the proposed multi-blockchain

First, we compare the Triplechain + Repchain multi-subchain structure proposed in CBDCS and the single-chain, double-chain, and triple-chain based on Hyperledger Fabric in terms of throughput.

As shown in Figure 3, through the comparison of the metric, we can find that the throughput performance of the triple-chain is better than that of the single-chain and double-chain. The throughput of the triple-chain structure is 37%-100% higher than the double-chain and single-chain respectively. The triplechain+ Repchain multi-subchain structure used by CBDCS has basically the same performance as the triple-chain in terms of throughput.

The throughput of the triple-chain is higher than that of the single-chain and the double-chain because CBDCS uses the task hierarchical storage strategy to store tasks of different levels in different levels of subchains. Therefore, the three subchains can process transactions in parallel, which can effectively improve throughput. The multi-subchain structure based on hyperledger fabric can prevent users from having multiple identities due to its distributed characteristics and strict Membership Services so that CBDCS can effectively resist Sybil attacks. The performance of the triplechain+ Repchain structure used by CBDCS in terms of latency and throughput is basically the same as that of the three-chain. The reason is that the Repchain only stores the credit of workers and does not participate in the verification and storage of transactions. Therefore, the Repchain has minimal impact throughput.

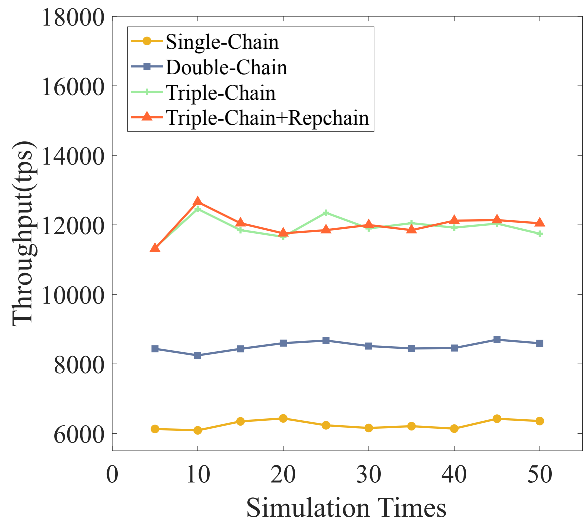


Fig. 3. Performance of the proposed multi-subchain structure based Hyperledger Fabric.

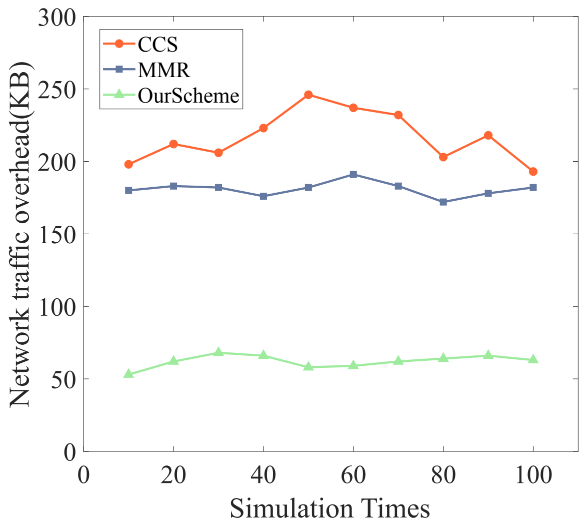
C. Performance of cross-subchain mechanism

In this part, we have compared with other cross-subchain solutions, namely CCS [13] and MMR [15]. The performance metrics of the comparison are as follows: (1) Network Traffic overhead (2) CPU utilization.

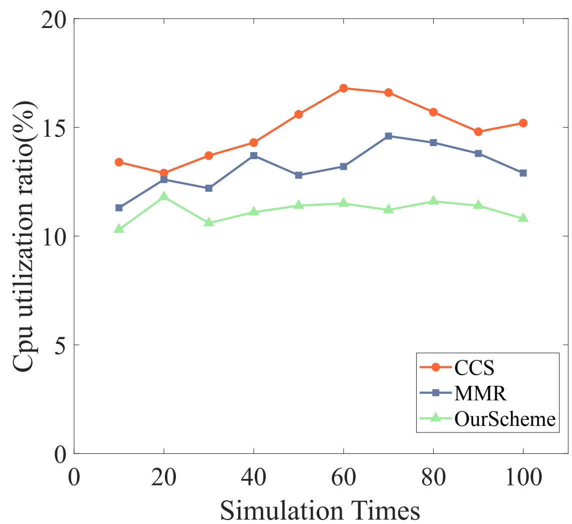
It can be seen from Figure 4 that the cross-subchain solution proposed in this paper is superior to CCS and MMR in both Network Traffic overhead and CPU utilization. In terms of Network Traffic overhead, the average Network Traffic overhead of the cross-subchain solution proposed in this paper is 62.1KB, and the average Network Traffic overhead of CCS and MMR is 216.8KB and 180.9KB, respectively, which are 3.5 times and 2.9 times of the solution in this paper. The scheme in this paper is based on the root&sub-CA and smart contract. The communication in the network only involves the output of the smart contract and the transmission of cross-subchain certificates, so the Network Traffic overhead in the network is extremely low. In terms of CPU utilization, our scheme's average CPU utilization is 11.7%, and the average CPU utilization of CCS and MMR are 14.9% and 13.4%, respectively. It can be seen from the above that, compared with CCS and MMR, the cross-subchain solution proposed in this paper has the best performance.

V. CONCLUSION

This paper proposes a distributed mobile crowdsourcing system called CBDCS to improve the security of mobile crowdsourcing system applications in the Internet of Things. CBDCS integrates two methods. One is based on the Hyperledger Fabric-based multi-subchain architecture and task



(a) Network Traffic overhead



(b) CPU utilization ratio

Fig. 4. Performance of cross-subchain mechanism based on root&sub CA and smart contract.

hierarchical storage strategy. The other is a cross-subchain access mechanism based on the root&sub CA and smart contracts. The multi-subchain structure can effectively improve the privacy and security of mobile crowdsourcing tasks and against the Sybil attacks. The simulation results show that: (i) the proposed multi-subchain structure has good throughput performance; (ii) the cross-subchain access mechanism in CBDCS can provide efficient cross-subchain access according to the actual demands of mobile crowdsourcing.

References

[1] S. Shenbagavalli, T. Priyadharshini, S. Sowntharya, P. Manikandan, and D. S. Saravanan, "Design and implementation of smart traffic controlling system," *International Journal of Engineering Technology Research & Management*, vol. 4, no. 4, pp. 28–36, 2020.

[2] H. Kumar, M. K. Singh, M. Gupta, and J. Madaan, "Moving towards smart cities: solutions that lead to the smart city transformation framework," *Technological forecasting and social change*, vol. 153, p. 119281, 2020.

[3] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in iot," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5986–5994, 2020.

[4] J. Rice and N. Martin, "Smart infrastructure technologies: Crowdsourcing future development and benefits for australian communities," *Technological Forecasting and Social Change*, vol. 153, p. 119256, 2020.

[5] M. C. Lucic, X. Wan, H. Ghazzai, and Y. Massoud, "Leveraging intelligent transportation systems and smart vehicles using crowdsourcing: An overview," *Smart Cities*, vol. 3, no. 2, pp. 341–361, 2020.

[6] L. Tran, H. To, L. Fan, and C. Shahabi, "A real-time framework for task assignment in hyperlocal spatial crowdsourcing," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 3, pp. 1–26, 2018.

[7] Y. Ma, Y. Sun, Y. Lei, N. Qin, and J. Lu, "A survey of blockchain technology on security, privacy, and trust in crowdsourcing services," *World Wide Web*, vol. 23, no. 1, pp. 393–419, 2020.

[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2019.

[9] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang, and R. H. Deng, "Crowdbc: A blockchain-based decentralized framework for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 6, pp. 1251–1266, 2018.

[10] G. K. Bhatia, P. Kumaraguru, A. Dubey, A. B. Buduru, and V. Kaulgud, "Workerrep: building trust on crowdsourcing platform using blockchain," Ph.D. dissertation, IIIT-Delhi, 2018.

[11] M. Yang, T. Zhu, K. Liang, W. Zhou, and R. H. Deng, "A blockchain-based location privacy-preserving crowdsensing system," *Future Generation Computer Systems*, vol. 94, pp. 408–418, 2019.

[12] C. Lin, D. He, S. Zeadally, N. Kumar, and K.-K. R. Choo, "Secbcs: a secure and privacy-preserving blockchain-based crowdsourcing system," *Science China Information Sciences*, vol. 63, no. 3, pp. 1–14, 2020.

[13] Y. Jiang, C. Wang, Y. Huang, S. Long, and Y. Huo, "A cross-chain solution to integration of iot tangle for data access management," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 1035–1041.

[14] H. He, Z. Luo, Q. Wang, M. Chen, H. He, L. Gao, and H. Zhang, "Joint operation mechanism of distributed photovoltaic power generation market and carbon market based on cross-chain trading technology," *IEEE Access*, vol. 8, pp. 66 116–66 130, 2020.

[15] H. Jin, X. Dai, and J. Xiao, "Towards a novel architecture for enabling interoperability amongst multiple blockchains," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 1203–1211.