2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) | 978-1-6654-1658-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/TRUSTCOM53373.2021.00066

A Privacy-Enhanced Mobile Crowdsensing Strategy for Blockchain Empowered Internet of Medical Things

Mengyao Peng
College of Computer
and Cyber Security,
Fujian Normal University
Fuzhou, Fujian, China
Fujian Provincial Key
Laboratory of Information
Processing and Intelligent
Control,
Minjiang University
Fuzhou, Fujian, China
e-mail: 18875857995@163.com

Jia Hu
University of Exeter
Exeter, UK
e-mail: j.hu@exeter.ac.uk

Hui Lin
College of Computer
and Cyber Security,
Fujian Normal University
Fuzhou, Fujian, China
Fujian Provincial Key
Laboratory of Information
Processing and Intelligent
Control,
Minjiang University
Fuzhou, Fujian, China
e-mail: linhui@fjnu.edu.cn

Xiaoding Wang
College of Computer
and Cyber Security,
Fujian Normal University
Fuzhou, Fujian, China
e-mail: wangdin1982@fjnu.edu.cn

Wenzhong Lin
Fujian Provincial Key
Laboratory of Information
Processing and Intelligent
Control,
Minjiang University
Fuzhou, Fujian, China
e-mail: lwz@mju.edu.cn

Abstract—The emergence of the Internet of Medical Things (IoMT) brings a huge impact on current medical system in the detection and prevention of medical diseases, as well as the sharing and analysis of medical data. To efficiently collect medical data for disease prevention, the mobile crowdsensing (MCS) is employed. However, the exposure of sensitive information about users and crowdsensing tasks might cause serious privacy leakage in MCS. To solve this problem, in this paper, a Privacy-enhanced Mobile Crowdsensing strategy utilizing Blockchain technology, named PMCB, is proposed. Specifically, we propose to classify the users by spectral clustering based on the social network generated by the social attributes of users. In this way, both crowdsensing tasks and participating users are classified such that task receivers are restricted to receive specific crowdsensing tasks. Furthermore, the blockchain is used to store crowdsensing tasks and smart contract is used for access control. Experiment results show that PMCB can achieve efficient privacy protection in mobile crowdsensing with high system throughput and low transaction latency.

Index Terms—Blockchain, Smart Contract, Privacy Protection, IoMT, Mobile Crowdsensing

I. Introduction

The emergence of the Internet of Medical Things (IoMT) provides a feasible solution to the common problem what the traditional medical system encountered, which is how to obtain user health information in a timely and accurate manner for highly dynamic and distributed medical institutions. In IoMT, users' health data are collected from various mobile terminals, such as tablets, mobile phones, personal computers, etc., for timely diagnosis. Compared with the traditional medical health management model, the medical health management system based on MCS technology has greater advantages, such as saving economic and time costs effectively [1] [2].

However, as an promising data collection method, the mobile crowdsensing technology deals with a large amount of data and this data always contains a lot of sensitive information [3]. Therefore, while MCS provides convenience, it also brings new data privacy security threats, including the following two aspects: one is how to protect the private information in the data provided by the participating users [4], the other one is how to protect the sensitive information contained in the tasks issued by the task releasers. In this paper, how to protect sensitive information in crowdsensing tasks is the major concern. Be ware that MCS needs to recruit users and assign tasks in the process of the data collection. Therefore, from the perspective of the task releasers, it is so vital that the private information in the task is not leaked.

As a popular technology, blockchain has many characteristics. For example, it can solve security and pri-

2324-9013/21/$31.00 ©2021 IEEE
DOI 10.1109/TrustCom53373.2021.00066

vacy issues for a variety of networks or systems. Its decentralized feature ensures that even if one node in the chain is attacked, the private information of other nodes cannot be stolen maliciously [5]. Therefore, by integrating the blockchain and the MCS technology, the privacy leakage problem can be prevented. In this paper, a Privacy-enhanced Mobile Crowdsensing strategy based on Blockchain, named PMCB, is proposed for IoMT. The main contributions of this paper are listed as follows:

1) In order to prevent collusion attacks on sensitive information, this strategy decomposes each task into several subtasks, and then sends them to participating users with different security levels for data collection. The above method of decomposing tasks can split sensitive information in the task to achieve the goal of protecting privacy and it also can prevent participating users of different security levels from directly colluding to obtain private information in the task.

2) To ensure the sensitive information contained in the task can be effectively prevented from being stolen by malicious task receivers. Corresponding to the above-mentioned task decomposition, this strategy also classifies data providers. We use approximating RatioCut based spectral clustering approach to divide data providers into different subgroups. In this method, four indicators are mainly considered, namely node degree, betweenness centrality, local clustering coefficient, and degree-based graph entropy. According to the above method, the tasks and task receivers are divided into the same number of categories, so that the task receivers of each category can only accept tasks of the same category.

3) In order to ensure that the category of the task receivers can match the category of the tasks, this strategy uses the smart contract technology in the blockchain to formulate relevant rules and supervise it to prevent malicious participating users from receiving subtasks of different categories and trying to piece together these subtasks to master a complete task and steal sensitive information in the task.

4) Experimental results show that the strategy proposed in this paper can effectively protect the sensitive information in the task from being leaked, and it can improve system performance in terms of system throughput and transaction latency.

The rest of the paper is organized as follows. We summarized some related work in Section II. We introduced the system model of this strategy in Section III. In Section IV, we introduced the strategy PMCB proposed in this paper in detail. In Section V, we analyze the relevant experimental data and results. Finally, it is summarized in Section VI.

II. REATED WORK

Recently, privacy protection for MCS has attracted more and more attention, and many excellent works have been proposed. In [6], the author propose a strategy for data aggregation. In this strategy, an improved blockchain with a new block header structure and two different block generation rules are designed and introduced, which used to guarantee the sensitive information in the tasks can not be leaked directly or indirectly. In [7], the author used anonymous technologies (such as pseudonyms, group signatures, and k-anonymity e.g.) in the Internet of Vehicles to protect the vehicle privacy. Although tasks may be exposed by using this way, it can effectively protect the user's identity information, which is acceptable to the user. In [8], the author proposed a new mobile crowdsensing system that integrates data aggregation, incentive mechanism and disturbance mechanism. The disturbance mechanism ensures users' privacy protection requirements and data accuracy requirements. In [9], the author proposes a strong privacy protection mobile crowdsensing system with user credit management. The system uses proxy re-encryption and BBS+ signature technology to prevent the user's privacy from leaking. In [10], the author proposes a mobile crowdsensing strategy that does not require a trusted third party for credibility. In addition to anonymous identity verification through group signatures, this strategy also designs a protocol based on blind signatures to achieve user anonymous authorization verification. In [11], the author proposes a combination of key distribution and trust management mechanism. This strategy prevent users' privacy from leaking through the evaluated public key trust. In [12], the author introduced the idea of smart contracts and proposed a two-stage method, including the preregistration stage and the final selection stage, in order to protect the location privacy. In [13], the framework proposed by the author is implemented by two non-colluding cloud platforms and adopting additively homomorphic cryptosystem, which can effectively reduce user cost and protect users' private information. In [14], the author proposed a privacy protection model based on blockchain-based twice verifications and consensuses, using a verifiable anonymity strategy which based on elliptic curve algorithm to protect users' identity privacy. In [15], the user's privacy and payment are protected by using smart contracts as a process executed by the blockchain.

III. SYSTEM MODEL

The system model of PMCB proposed in this paper is shown in Fig. 1. The system mainly includes three parts, namely, task releasers (data users), task receivers (data providers), and server group. The first part, in order to complete a certain data collection work, the task releasers (i.e. data users) need to release related tasks to participating users so that he or she can collect the data needed to complete further analysis work. The second part, task receivers (i.e. data providers) are mobile terminal devices that belong to participating users. In IoMT, these devices mostly provide medical and health data after receiving the tasks. The third part, the server group mainly contains

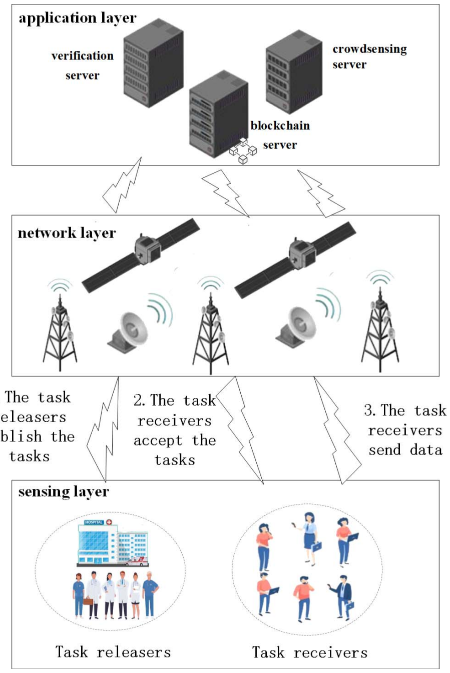


Fig. 1. The system model of PMCB

blockchain servers, verification servers, and crowdsensing servers. The role of the blockchain server is to store the tasks. The role of the verification server is to verify the reliability of the data submitted by the task receivers. And the role of the crowdsensing server is to store valid data that has been verified by reliability. But this article mainly discusses blockchain servers. The task releasers and task receivers at the sensing layer respectively use the network layer to publish tasks or upload data. And these two transmission processes include the privacy of the task on the one hand, and the privacy of the receivers of the task on the other. Therefore, it is particularly important to protect the privacy of these two aspects. And this paper focuses more on the former, that is, task privacy.

In PMCB, in order to achieve privacy protection and prevent collusion attack, this paper will classify tasks and task receivers into different groups. A task releaser needs to divide tasks into different subtasks which can be carried out independently and add task categories to them respectively, and then add them to the blockchain. Task receivers will also be divided into different categories according to their social attributes, and then through the function of smart contract, they can only receive tasks of the same category. On the one hand, the decomposition of sensitive tasks will effectively prevent the leakage of privacy directly. On the other hand, this paper classifies task receivers, which will prevent users of different levels from colluding with each other and causing indirect privacy leakage.

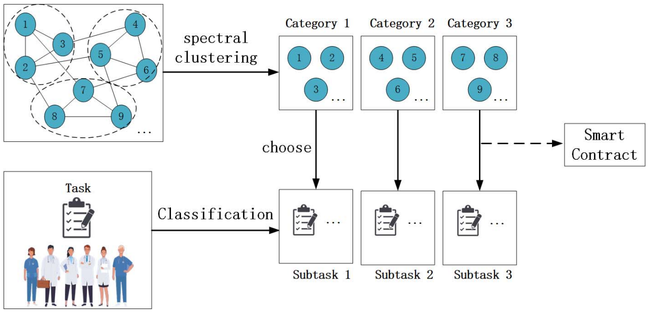


Fig. 2. The workflow of this strategy

IV. THE IMPLEMENTATION DETAILS OF THE PMCB

According to the traditional mobile crowdsensing architecture [16], as shown in Fig. 1, PMCB is mainly divided into three layers: sensing layer, network layer and application layer. The sensing layer is the layer where each terminal device is located, including task releasers and task receivers. The terminal devices publish tasks, receive tasks, and supply data at this layer. The network layer, as the name implies, is used to transmit tasks and data. The application layer stores all kinds of servers, this paper mainly focuses on the blockchain server.

The general process of the implementation of this strategy is as follows(shown as Fig. 2):

1) First of all, based on the social network of the participating users, we use the method of spectral clustering to divide them into different categories according to four metrics (i.e., node degree, betweenness centrality, local clustering coefficient, and degree-based graph entropy), and compute the average of the above four metrics to rank influence levels of different user subgroups. As a result, each user group has a corresponding level.

2) Secondly, after the task releaser classifies the task receivers, it also sorts the sensitivity of its subtasks, and divides them into corresponding levels in order.

3) Finally, in order to allow the task receivers to only receive subtasks of the same level, we use smart contract technology to limit the choices of participating users when the task releaser has completed the above two steps.

A. Task receivers and task classification

- Classification of tasks: Before each task is released, the task releaser needs to divide its task into some types of subtasks (as shown in Fig. 2), which can be selected by task receivers of the same level. Then the data of these divided subtasks can be collected separately. And the reason why the task needs to be divided into several subtasks is to separate some of the private information contained in the task, so that it is impossible for each task receiver to grasp the complete task privacy information. Thereby protecting the sensitive information contained in the task.

- Classification of task receivers: In the above, classifying tasks can effectively prevent the direct disclosure of task privacy, but only classifying tasks cannot resist collusion attacks by task receivers. Therefore, we next classify task receivers, and we use the method of secret sharing to protect the privacy in the subtasks. Whether in the process of dividing task receivers or in the process of evaluating the impact of task receivers, we all choose task receivers with higher influence, so the probability that more than half of the task recivers are colluders is extremely low. The following will introduce the specific method of dividing task receivers in this paper.

  In our previous work [17], we proposed a K-means based privacy-preserving classification mechanism. However, in this paper, we use the social network to classify the level of the influence of the users. Therefore, this strategy will cluster nodes based on the social attributes of task receivers (nodes). We assume that the social network graph of participating users is $G = (V, E)$, and we translate G to similarity graph G^S to achieve node clustering. In order to ensure similar nodes in the same cluster and the clusters are balanced that is the number of nodes in each cluster is approximately the same [18], we use the spectral clustering method based on approximating RatioCut to partition graph G^S. This can effectively avoid the impact caused by the large gap in the number of users in different categories. The algorithm is divided into three steps, as shown in Algorithm 1:

  1. Pairwise similarity computing: we consider some metrics to compute the similarity between pairwise nodes.

  2. Similarity graph constructing: we select the K nearest nodes as node's neighbors, so that, the similarity graph matrix is a sparse matrix.

  3. Similarity graph partition: we use the RatioCut graph partition scheme to partition the similarity graph into T subgraphs, so that the size of subgraphs are approximately equal.

Algorithm 1 Node Clustering

```
Input: $G$
Output: clusters $C_i$, $i = 1, 2, ..., T$
for $i = 1$ to $n$ do
    construct 1-neighborhood graph $G(v_i)$ of each node $v_i$
    compute $x_i$=<$D(v_i)$, $BC(v_i)$, $Lc(v_i)$, $I_f(G(v_i))$>
end for
Similarity graph construction seeing algorithm 2
Similarity graph partition seeing algorithm 3
obtain the clusters $C_i$, $i = 1, 2, ..., T$
return  $C_i$, $i = 1, 2, ..., T$
```

1. Pairwise similarity computing: In order to construct the similarity graph, we consider the following metrics to compute the similarity between each pairwise nodes: node degree, betweenness centrality, local clustering coefficient, and degree-based graph entropy.

Definition 1: Node Degree($D(v_i)$): the degree of node v_i is defined as the number of neighbors of the node.

Definition 2: Local Clustering Coefficient($Lc(v_i)$): $Lc(v_i)$=$\mu_G(v_i)/\omega_G(v_i)$, where $\mu_G(v_i)$ and $\omega_G(v_i)$ are the numbers of triangles and triples in $G(v_i)$, respectively.

Definition 3: Betweeness Centrality($BC(v_i)$): $BC(v_i)$ of node v_i in graph G is the fraction of the shortest paths between all pairs of nodes in the graph that pass through v_i, $BC(v_i)$=$\sum_{s,t \in V, s \neq t \neq v} \dfrac{\sigma_{st}(v_i)}{\sigma(v_i)}$.

Definition 4: Degree Based Graph Entropy($I_f(G(v_i))$) [19]: let $G = (V, E)$ be a connected graph, and we assume that $|V| = n$, $|E| = m$. For a given $v_i \in V$ and an arbitrary real number $\alpha \in R$, the degree-based graph entropy

$$I_f(G(v_i)) = -\sum_{i=1}^{n} \frac{d_i^\alpha}{\sum_{j=1}^{n} d_j^\alpha} \log\left(\frac{d_i^\alpha}{\sum_{j=1}^{n} d_j^\alpha}\right).$$

We suppose $\alpha = 1$, thus,

$$\begin{aligned} I_f(G(v_i)) &= \log(\sum_{i=1}^{n} d_i) - \sum_{i=1}^{n} \frac{d_i}{\sum_{j=1}^{n} d_j} \log d_i \\ &= \log(2m) - \frac{1}{2m} \sum_{i=1}^{n} d_i \log d_i. \end{aligned}$$

For every node $v_i \in G$, we compute $D(v_i)$, $BC(v_i)$, $Lc(v_i)$, $I_f(G(v_i))$, respectively. We call the vector x_i=<$D(v_i)$, $BC(v_i)$, $Lc(v_i)$, $I_f(G(v_i))$> the node vector. Thus, the similarity function of every pair of nodes v_i, v_j can be defined as

$$f_{sim}(v_i, v_j) = e^{-\dfrac{\| x_i - x_j \|^2}{2\sigma^2}}. \tag{1}$$

2. Similarity graph constructing: In this step, we construct a weighted similarity graph named K-nearest neighbor graph, while the neighborhood relationship is symmetric. Suppose G^S= (V^S, E^S, W^S), where V^S represent the nodes,E^S represents the edges,W^S represent the weights on the edges.If v_j^S is the first K similar nodes to v_i^S, $e_{ij}^S \in E^S$. w_{ij}^S represent the similarity of v_i^S and v_j^S, w_{ij}^S= $f_{sim}(v_i, v_j)$, if v_j^S is the K-nearest neighbors of v_i^S, otherwise $w_{ij}^S = 0$.Therefore, the degree of node v_i is defined as the sum of weights of the edges adjacent to v_i, $d_i = \sum_{t=i}^{N_i} d_t$, where N_i is the number of the neighbors of v_i. We use D to represent the degree matrix, $D = (d_{ij})_{n \times n}$, $d_{ij} = d_i$ if $i = j$, else $d_{ij} = 0$. W is the adjacency matrix of G^S, where w_{ij} is the weight of the edge between node v_i and v_j. The detail is shown as algorithm 2.

Algorithm 2 Similarity Graph Construction

```
Input: Node vector $x_i$, $i = 1, ..., n$
Output: $G^S$
for $i = 1$ to $n$ do
    for $j = 1$ to $n$ do
        Compute similarity $f_{sim}(v_i, v_j)$ between node $v_i$ and node $v_j$ according to equation(1)
    end for
end for
for $i = 1$ to $n$ do
    select the first $K$ nearest nodes as the neighbors of node $v_i$
    add edges between $v_i$ and it's neighbors and obtain similarity graph $G^S$
end for
return  $G^S$
```

Algorithm 3 Similarity Graph Partition

```
Input: $G^S$
Output: clusters $C_1, C_2, ..., C_T$
Construct adjacent matrix and degree matrix of graph $G^S$, denoted as $W$ and $D$, respectively
Compute Laplacian matrix $L = D - W$
Compute the eigenvalue of $L$, the first $T$ eigenvalues are denoted as $\lambda_1, \lambda_2, ..., \lambda_T$
Compute the correspondence eigenvectors of the first $T$ eigenvalues denoted as $x_1, x_2, ..., x_T$
let $y_i = \frac{x_i}{\sqrt{N_i}}$, $i = 1, 2, ..., T$, set $Y = (y_i)_{T \times n}$
Consider one column of matrix $Y$ as one node
Utilize $k$-means algorithm to partition these $n$ nodes into $T$ clusters
return  $C_i$,$i = 1, 2, ..., T$
```

3. Similarity Graph partition: In order to cluster nodes in original graph, we translate the original graph into the similarity graph. Thus, the problem is translated into how to partition the similarity graph into T subsets, so that the cut is minimum, $cut(A_1, A_2, ..., A_T) = \frac{1}{2}\sum_i^T W\left(A_i, \overline{A}_i\right)$. In order to make the subset balance, we replace the cut with RatioCut, $RatioCut(A_1, A_2, ..., A_T) = \sum_{i=1}^T \frac{cut(A_i, \overline{A}_i)}{|A_j|}$. For a similarity graph G^S, W is the similarity matrix, its Laplacian matrix can be calculated as $L = D - W$. Suppose that$\lambda_1, \lambda_2, ..., \lambda_T$are the smallest T eigenvalues of L, $\min(RatioCut(A_1, A_2, ..., A_T)) = \sum_{i=1}^K \lambda_i$.Then, compute the eigenvectors $x_i, x_2, ..., x_T$ of λ_i, $i = 1, 2, ..., T$. Then $y_i = \frac{x_i}{\sqrt{N_i}}$, $i = 1, 2, ..., T$, $Y = (y_i)_{T \times n}$. The minimum RatioCut problem can be relaxed as min $tr(Y^T LY)$ subject to $Y^T Y = I$. Consider each column of matrix Y as one node, next, we utilize k-means algorithm to partition these n nodes into T clusters, $C_1, C_2, ..., C_T$. The detail is shown as algorithm 3.

4. After completing the classification of users, we need to rank each group of users to divide them into different security levels. In this paper, we calculate the influence of each group based on four indicators. These four indicators are node degree, local clustering coefficient, betweenness centrality, and degree based graph entropy, and these indicators calculate the influence of nodes according to different definitions. Firstly, a large node degree indicates that the node has more connections with other nodes. Secondly, a high betweenness centrality indicates that the node plays an important role as a bridge in the network. Thirdly, a high node local clustering coefficient means that the 1-neighborhood graph centered on the node has more edges. Fourthly, the higher the degree based graph entropy, the stronger the connectivity of the graph. Therefore, this paper calculate the sum of these four indicators to more accurately rank the influence of the nodes from multiple angles. Moreover, the greater the influence, the higher the security level. Therefore, the influence (I_i) calculation formula is as follows:

1) Average degree(AD): The AD of C_i can be calculated as $\sum_{v \in C_i} D_v / |C_i|$;

2) Average local clustering coefficient(ALC): The ALC of C_i can be calculated as $\sum_{v \in C_i} Lc_v / |C_i|$;

3) Average betweeness centrality(ABC): The ABC of C_i can be calculated as $\sum_{v \in C_i} BC_v / |C_i|$;

4) Average degree based graph entropy(AIF): The AIF of C_i can be calculated as $\sum_{v \in C_i} I_{f_v} / |C_i|$.

Then,

$$\begin{aligned} I_i = &\sum_{v \in C_i} D_v / |C_i| + \sum_{v \in C_i} Lc_v / |C_i| \\ &+ \sum_{v \in C_i} BC_v / |C_i| + \sum_{v \in C_i} I_{f_v} / |C_i|. \end{aligned} \tag{2}$$

B. Use smart contracts to establish access control

Relying on its unique advantages (for example, decentralization, immutability of information, etc.), blockchain technology has been increasingly applied to various platforms, and the smart contract as the core technology of blockchain is also getting more and more attention and use. As the name suggests, the smart contract is based on the immutable data in the blockchain and formulate some rules to control the process of transactions. Smart contract is a kind of agreement, which is characterized by the low cost of making contracts, executing contracts, and verifying contracts. In smart contract transactions, there is no need for a third party to participate. All rules, supervision and decision-making are done by the system itself. Generally, a consensus mechanism is used to determine whether the contract is executed normally according to the regulations. Due to the digital characteristics of smart contracts, data is stored in the blockchain, and encrypted codes are used to enforce the agreement to ensure that transactions are traceable and irreversible.

As shown in the Fig. 3, it is the complete transaction process of Hyperledger. This process can ensure the realization of secure transactions: 1) The client can use the API to construct a transaction proposal request and package the transaction proposal into a correct format; 2) Use the user's encrypted credentials in the transaction proposal to generate a unique signature for this transaction proposal; 3) The endorser peers verify the received transaction proposal request, including whether the format of the proposal is correct, etc.; 4) The transaction request is submitted to the orderer peers. After receiving the transaction request, the orderer peers will sort them by time and create a transaction block; 5) The orderer peers broadcast to the leader peers of all organizations in the same channel, and then the leader peers will verify the received block, and writes the result into the local ledger after passing the verification; 6) The Leader peers broadcast the results to other nodes in the organization, and generally default to 3 nearby nodes.

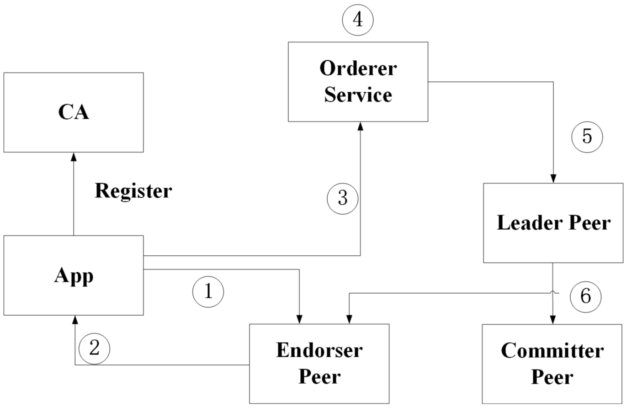


Fig. 3. Transaction process

Based on the above foundation this strategy uses smart contract technology to perform access control, so that the splitting work performed above can protect sensitive information. Specifically, when the task releaser classifies the task into a series of subtasks, each subtask will bring the corresponding category and pack it into a block, and the smart contract will record the relevant information, and then store these on the blockchain server. After the task is released, the task receiver selects the tasks in the contract.

In order to achieve the purpose that the task receiver can only select the tasks of the corresponding level, we designed the smart contract for task selection. As shown in the Fig. 4, when the task receiver wants to receive the task, his personal attributes (i.e. *security level*) will be uploaded. The judgment rules include whether the *security level* of the task receiver is equal to the *task level* of the task which is applied for. If *security level* and *task level* are equal, the task can be successfully received through the monitoring of the smart contract; otherwise, the task receiver will be refused to accept the task. Therefore, it is difficult for malicious users to choose multiple types of tasks to steal sensitive data.

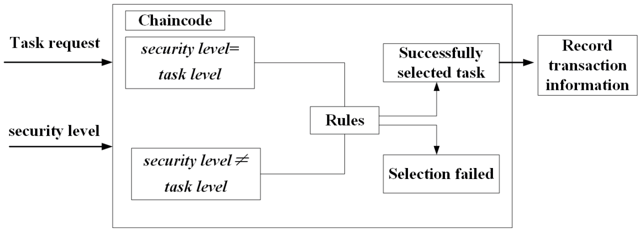


Fig. 4. Smart Contract

V. PERFORMANCE EVALUATION

A. Experiment Setup

The proposed PMCB system is implemented on a Hyperledger Fabric1.2-based simulator. The physical machine runs with 16G of memory, the Intel Core i7 processor with a frequency of 3.2GHZ is equipped with a 64-bit win7 system, and a VMware Workstation 14 Pro of 4GB running memory and 2 processors of Ubuntu system.

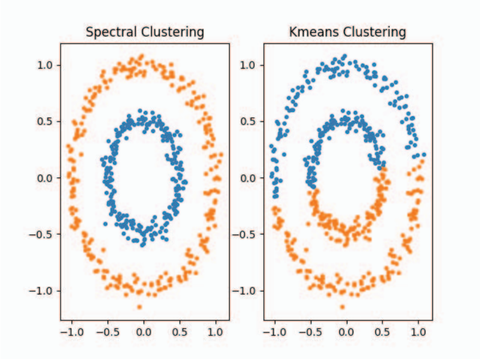
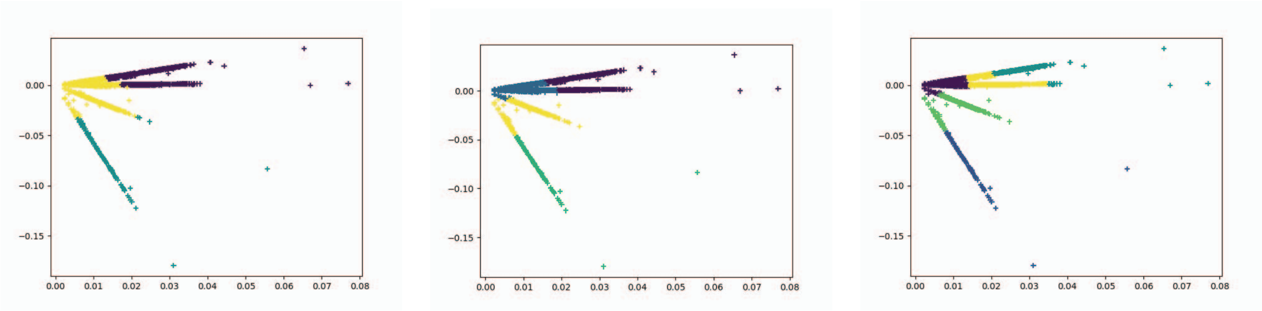


Fig. 5. Comparison of spectral clustering and K-means

B. Performance Metrics and Experiment Results

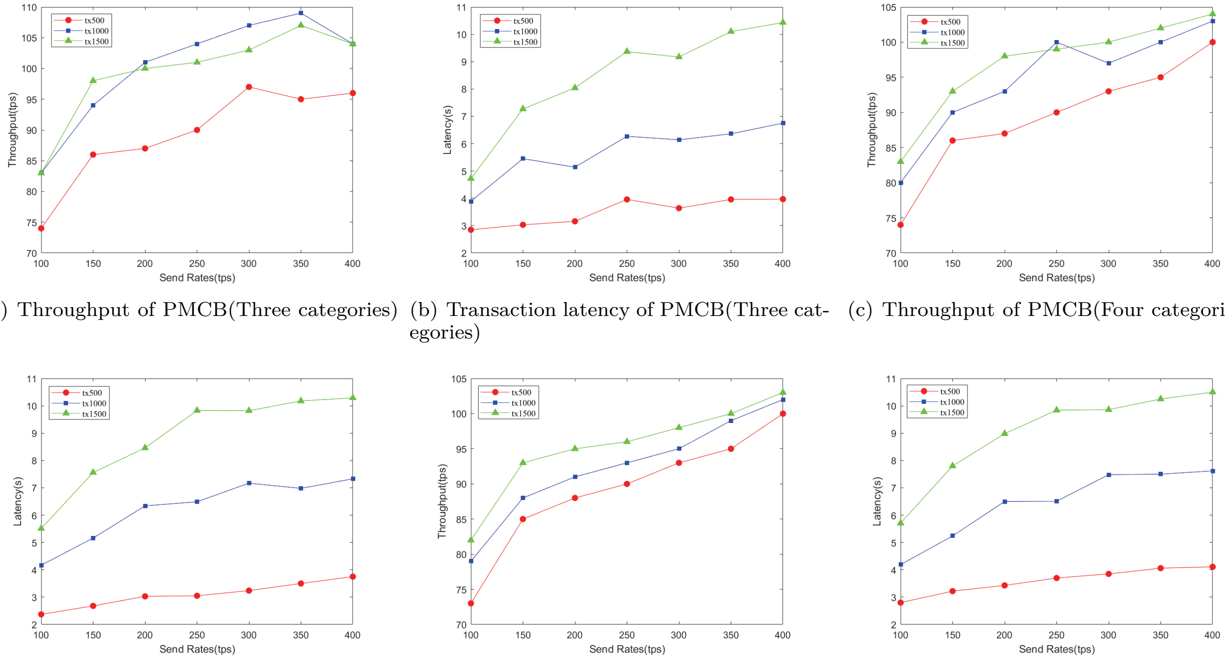
1) Classification of Task Receivers

The system can divide task receivers into different T categories according to actual needs. In this paper, we use the strategy of spectral clustering to replace the K-means strategy proposed in the previous paper. Because, it can be clearly seen from the Fig. 5 that the spectral clustering strategy is better than the K-means strategy. It not only clusters closer nodes together, but also clusters according to the different characteristics of each node to achieve better results. What's more, Fig. 6 is the three classification results on the facebook data set [20]. According to the results of classification, the task receivers of each category can only select the subtasks of the



(a) T = 3 (b) T = 4 (c) T = 5

Fig. 6. The three classification results on the facebook data set



(a) Throughput of PMCB(Three categories) (b) Transaction latency of PMCB(Three categories) (c) Throughput of PMCB(Four categories)

(d) Transaction latency of PMCB(Four categories) (e) Throughput of PMCB(Five categories) (f) Transaction latency of PMCB(Five categories)

Fig. 7. Throughput and transaction latency of PMCB under different classifications

corresponding category, so as to achieve the purpose of protecting task privacy.

2) PMCB Throughput

First, we evaluate the throughput of the system under different transaction sending rates, and then the corresponding throughput under each transaction number *tx* is shown in Fig. 7 (a), (b), (c). As the system throughput trends under different classifications are very similar, this paper will make a detailed analysis of the situation where users are divided into three categories, as follows. It can be seen from the results in Fig. 7 (a), when the number of transactions is 500, the system has the lowest throughput 74tps. When the transaction number exceeds 500, the system throughput at different send rates is relatively close. For example, when the sending rate is 200tps, the throughputs of the transaction number *tx* 1000 and 1500 are 101tps and 100tps, respectively. Furthermore, when the sending rate reaches 300tps, the throughput of the system gradually stabilizes and maintains a high throughput.

3) PMCB Transaction Latency

Then, we tested the transaction latency at different transaction sending rates, and the test results are shown in Fig. 7 (d), (e), (f). As mentioned above, since the system latency trend under different classifications is very close, this paper will make a specific analysis of the cases where users are divided into three categories, as follows. From the result in Fig. 7 (d), we can see that the higher the transaction sending rate, the longer the latency of the system. Meanwhile, as the numbers of transactions continues to increase, the corresponding latency is also increase. For example, when the sending rate is 150tps, the transaction latency of the transaction number tx 500, 1000, and 1500 are 3.03s, 5.45s and 7.27s. According to the experimental results, when the number of transactions is 1500 and the sending rate reaches 400tps, the transaction latency of the system reaches the maximum value of 10.43s.

4) Comparison of different categories

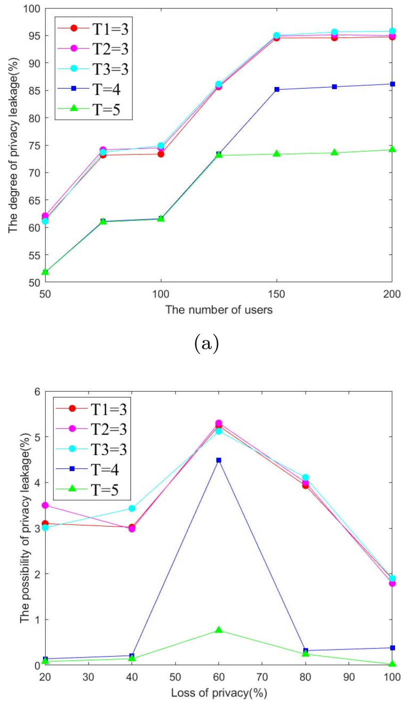
In this paper, we divide users into three, four and five categories, and above, we analyze the relevant performance of different transaction numbers under each category. Next, we will compare and analyze the performance under different classifications. We can see from the results in Fig. 9 (a) that when the $tx = 500$, the system throughputs under different categories are very close. For example, when the transaction sending rate is 250tps and 350tps, the system throughput of the three categories is the same, which is 90tps and 95tps respectively. In addition, regardless of the classification situation, the system throughput always grows with the increase of the transaction sending rate. Furthermore, we can easily see from Fig. 9 (b) and Fig. 9 (c) that the throughput of the system will grow with the increase of the transaction sending rate in each classification case. However, when the users are divided into three types, the throughput of the system will reach the maximum. For example, when $tx = 1000$ and the transaction sending rate reaches 350tps, the system throughput reaches the maximum value, which is 109tps. What's more, when $tx = 1500$ and the transaction sending rate reaches 350tps, the system throughput reaches the maximum value of 107tps.

Next, we will compare and analyze the transaction latency of different classification situations. From Fig. 9 (d), (e), (f), we can clearly see that in either case, the transaction latency will continue to grow as the sending rate increases. Specifically, as shown in Fig. 9 (d), when users are divided into three categories and $tx = 500$, the transaction latency is lower than the other two situations. For example, when the transaction sending rate is 400tps, the transaction latency of the three categories, four categories, and five categories are 3.97s, 6.75s, and 10.43s, respectively. Further, the results in Fig. 9 (e) and Fig. 9 (f) show that no matter when $tx = 1000$ or $tx = 1500$, the transaction latency is still low, when users are divided into three categories. However, different from Fig. 9 (d), the transaction latency in the three categories and four categories are very close. For example, when $tx = 1000$ and the teansaction sending rate is 250tps, the transaction latency of three categories and four categories are 6.49s and 6.51s, respectively. And when $tx = 1500$ and the teansaction sending rate is 300tps, the transaction latency of three categories and four categories are 9.82s and 9.86s, respectively.

5) Comparison of privacy protection under different systems and different classifications

Next, this paper will compare the degree of privacy protection under different classifications and different systems proposed in the past two years [6] [21], and the results are shown in Fig. 8. First, we assign some values in advance. In this paper, we assume that 10% of users belonging to $category1$ may leak the privacy of tasks, $category2$ is 20%, $category3$ is 30%, $category4$ is 40% and $category5$ is 50%. At the same time, we also make assumptions and assignments on the possibility of privacy leakage by users of different levels. For users in $category1$, the probability of leaking private data reaching 20%, 40%, 60%, 80%, and 100% are $1/12, 1/11, 1/9, 1/10, 1/13$, respectively. Similarly, the possibility of users in $category2$ leaking private information are $1/11, 1/10, 1/8, 1/9, 1/12$. The possibility of users in $category3$ leaking private data are $1/10, 1/9, 1/7, 1/8, 1/11$. The possibility of users in $category4$ leaking private data are $1/9, 1/8, 1/6, 1/7, 1/10$. And the possibility of users in $category5$ leaking private data are $1/7, 1/5, 1/3, 1/4, 1/6$. We can easily see from Fig. 8 (a) that, as the number of users continues to grow, the degree of privacy leakage is also increasing and eventually leveling off. In the figure, $T1 = 3$ represents the results of the strategy proposed in this paper, and $T2 = 3$, $T3 = 3$ represents the results of the other two comparative strategies respectively. We can clearly see that whether our scheme or the other two schemes, in the case of $T = 3$, the privacy protection degree of the system is very close. However, the difference between the strategy we proposed and others is that the strategy in this paper can not only divide task receivers into three categories, but also into four categories and five categories, and as shown in the figure, when task receivers are classified into four or five categories, the degree of privacy protection is better than when they are classified into three categories. At the same time, we can also see that the degree of privacy leakage is the highest when users are classified into three categories, and the degree of privacy leakage is the lowest when users are classified into five categories. For example, when the number of users is 150, the degree of privacy leakage corresponding to $T1 = 3$ is 94.55%, and when $T = 4$ and $T = 5$, they are 85.12% and 73.35%, respectively. It can also be clearly seen from the results in Fig. 8 (b) that regardless of our scheme or the other two schemes, the possibility of privacy leakage is very close when task receivers are divided into three categories under different privacy loss situations. However, no matter how much privacy loss is, the probability of privacy leakage of $T = 3$ is higher than that of $T = 4$ and $T = 5$. Furthermore, when the degree of privacy loss is 60%, the corresponding possibility of privacy leakage is the largest among other situations. For example, when the degree of privacy loss is 60%, the probability of privacy leakage is 5.24% in the case of $T = 3$, and the cases of $T = 4$ and $T = 5$ correspond to 4.49% and 0.76%, respectively. As a result, although our scheme is not much different from the other two schemes when $T = 3$, the advantage of our scheme is that we still propose to divide task receviers into four and five categories, and the degree of privacy protection in these two cases is better than that of the three categories.

Based on the analysis of the above experimental results, we can know that no matter in which case, users are classified into 5 categories with the best degree of privacy protection. Therefore, in actual medical applications, we can classify users and tasks into more levels to achieve an ideal privacy protection effect. However, this does not mean that users can be classified infinitely, because too many classifications will also affect other performance of the entire system, such as throughput and latency. Therefore, choosing a reasonable number of categories will not only maintain the performance of the system, but also better protect the private information in the tasks.



(a)

(b)

Fig. 8. Comparison of privacy protection under different systems and different classifications

VI. Conclusion

To achieve privacy protection in mobile crowdsensing, a Privacy-enhanced Mobile Crowdsensing strategy utilizing Blockchain technology (PMCB) is proposed for internet of medical things. Specifically, the proposed PMCB divides the crowdsensing tasks into different categories and store them on the blockchain. Moreover, considering the social attributes of task receivers, the spectral clustering is applied to classify task receivers into different groups. According to task division and task receiver classification, each group of task receivers are restricted to receive a specific category of tasks against collusion attacks. Experiment results show that the PMCB achieves efficient privacy protection with high system throughput and low transaction latency.

References

[1] G. J. Joyia, R. M. Liaqat, A. Farooq, S. Rehman, ”Internet of medical things (IOMT): Applications, benefits and future challenges in healthcare domain,” Journal of Communications, vol. 12, no. 4, pp. 240-247, 2017.

[2] J. C. Cano, J. M. Cecilia, E. Hernandez-Orallo, C. T. Calafate, P. Manzoni, ”Mobile crowdsensing approaches to address the COVID-19 pandemic in Spain,” IET Smart Cities, vol. 2, no. 2, pp. 58-63, 2020.

[3] K. Zhao, S. Tang, B. Zhao and Y. Wu, ”Dynamic and Privacy-Preserving Reputation Management for Blockchain-Based Mobile Crowdsensing,” IEEE Access, vol. 7, pp. 74694-74710, 2019.

[4] L. Pournajaf, D. A. Garcia-Ulloa, L. Xiong, V. Sunderam, ”Participant privacy in mobile crowd sensing task management: A survey of methods and challenges,” Acm Sigmod Record, vol. 44, no. 4, pp. 23-34, 2016.

[5] X. Gu et al., ”Using blockchain to enhance the security of fog-assisted crowdsensing systems,” 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), pp. 1859-1864, 2019, doi: 10.1109/ISIE.2019.8781332.

[6] H. Lin, S. Garg, J. Hu, G. Kaddoum, M. Peng and M. S. Hossain, ”A Blockchain-based Secure Data Aggregation Strategy using 6G-enabled NIB for Industrial Applications,” IEEE Transactions on Industrial Informatics, 2020, doi: 10.1109/TII.2020.3035006.

[7] J. Ni, A. Zhang, X. Lin, X. S. Shen, ”Security, privacy, and fairness in fog-based vehicular crowdsensing,” IEEE Communications Magazine, vol. 55, no. 6, pp. 146-152, 2017.

[8] H. Jin, L. Su, H. Xiao, K. Nahrstedt, ”Incentive mechanism for privacy-aware data aggregation in mobile crowd sensing systems,” IEEE/ACM Transactions on Networking, vol. 26, no. 5, pp. 2019-2032, 2018.

[9] J. Ni, K. Zhang, Q. Xia, X. Lin, X. S. Shen, ”Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing,” IEEE Transactions on Mobile Computing, vol. 19, no. 6, pp. 1317-1331, 2020.

[10] H. Wu, L. Wang, G. Xue, J. Tang, D. Yang, ”Enabling data trustworthiness and user privacy in mobile crowdsensing,” IEEE/ACM Transactions on Networking, vol. 27, no. 6, pp. 2294-2307, 2019.

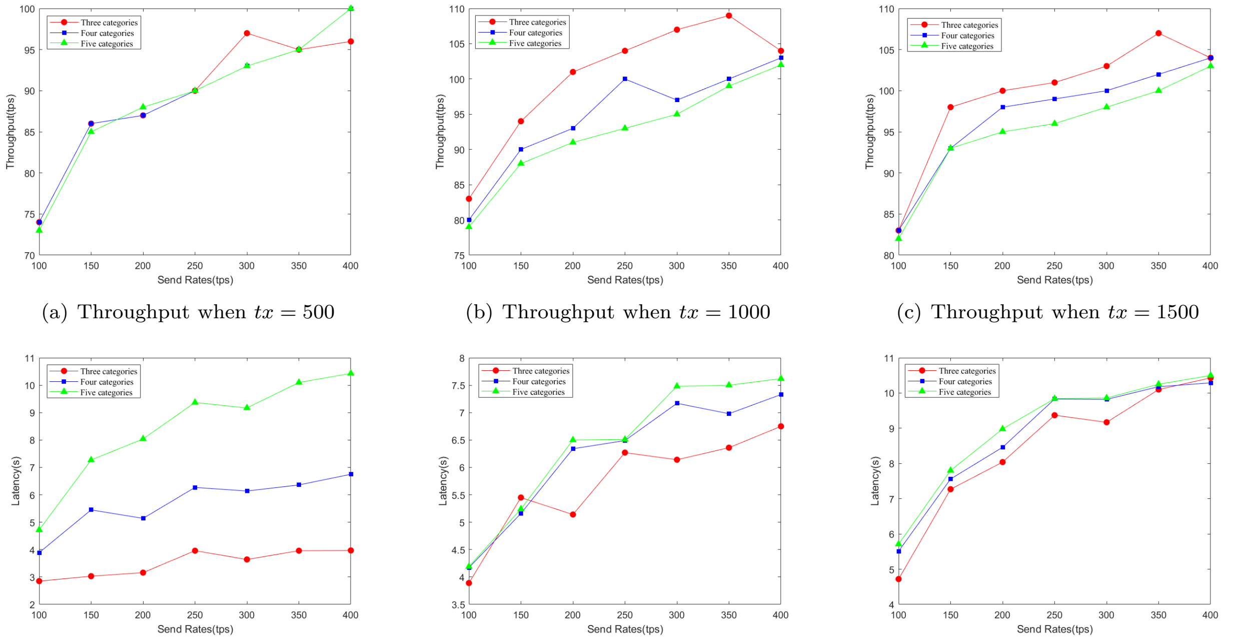
[11] J. B. Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. T. Moreno, A. Skarmeta, ”Privacy-preserving solutions for Blockchain: review and challenges,” IEEE Access, vol. 7, pp. 164908-164940, 2019.

[12] J. Xi, ”CrowdBLPS: A blockchain-based location privacy-preserving mobile crowdsensing system,” IEEE Transactions on Industrial Informatics, vol. 16, no. 6, pp. 4206-4218, 2019.

[13] C. Miao, L. Su, W. Jiang, Y. Li and M. Tian, ”A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems,” IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, pp. 1-9, 2017, doi: 10.1109/INFOCOM.2017.8057114.

[14] J. An, H. Yang, X. Gui, W. Zhang, R. Gui and J. Kang, ”TCNS: Node Selection With Privacy Protection in Crowdsensing Based on Twice Consensuses of Blockchain,” IEEE Transactions on Network and Service Management, vol. 16, no. 3, pp. 1255-1267, 2019.

[15] D. Chatzopoulos, S. Gujar, B. Faltings and P. Hui, ”Privacy Preserving and Cost Optimal Mobile Crowdsensing Using Smart Contracts on Blockchain,” 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), pp. 442-450, 2018, doi: 10.1109/MASS.2018.00068.



(a) Throughput when $tx = 500$

(b) Throughput when $tx = 1000$

(c) Throughput when $tx = 1500$

(d) Transaction latency when $tx = 500$

(e) Transaction latency when $tx = 1000$

(f) Transaction latency when $tx = 1500$

Fig. 9. Throughput and transaction latency comparison of different classifications

[16] D. Tao, R. Gao, H. Sun, ”Sensing-gain constrained participant selection mechanism for mobile crowdsensing,” Personal and Ubiquitous Computing, pp. 1-15, 2020, doi: 10.1007/s00779-020-01470-8.

[17] H. Lin, S. Garg, J. Hu, X. Wang, M. J. Piran and M. S. Hossain, ”Privacy-enhanced Data Fusion for COVID-19 Applications in Intelligent Internet of Medical Things,” IEEE Internet of Things Journal, 2020, doi: 10.1109/JIOT.2020.3033129.

[18] U. V. Luxburg, ”A tutorial on spectral clustering,” Statistics and Computing, vol. 17, no. 4, pp. 395-416, 2007.

[19] S. Cao, M. Dehmer, Y. Shi, ”Extremality of degree-based graph entropies,” Information Sciences, pp. 22-33, 2014, doi: 10.1016/j.ins.2014.03.133.

[20] Stanford Large Network Dataset Collection, http://snap.stanford.edu/data.

[21] X. Wang, S. Garg, H. Lin, G. Kaddoum, J. Hu and M. S. Hossain, ”A Secure Data Aggregation Strategy in Edge Computing and Blockchain empowered Internet of Things,” IEEE Internet of Things Journal, 2020, doi: 10.1109/JIOT.2020.3023588.