Blockchain-assisted D2D Data Sharing in Fog Computing

Yi Peng*, Taiping Cui*, Bin Shen*, Feng Lin†, Xiaoge Huang*, Qianbin Chen‡
*School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications
†School of Automation, Chongqing University of Posts and Telecommunications
‡Chongqing Key Laboratory of Mobile Communications Technology, Chongqing, 400065, China
Email:{pengyi0719@163.com, cuitp@cqupt.edu.cn, shenbin@cqupt.edu.cn, linfeng@cqupt.edu.cn, huangxg@cqupt.edu.cn, chenqb@cqupt.edu.cn}

2022 IEEE 95th Vehicular Technology Conference (VTC2022-Spring) | 978-1-6654-8243-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/VTC2022-Spring54318.2022.9861033

***Abstract*—In fog network, device-to-device (D2D) sharing is an important way to obtain data. However, due to an untrusted environment, it is difficult for a device to assess the reliability of the received data. What's more, devices may be reluctant to share data because of selfish, resulting in data supply and demand imbalances. In this regard, a data sharing scheme assisted by blockchain and matching algorithm is proposed. In order to ensure the authenticity of the data, the Bayesian inference model is employed to predict quality of the data, and a multi-factor data evaluation method is presented to make accurate judgments. Furthermore, different utility functions for data requesters and providers are defined, and a two-way matching game is introduced to balance of data supply and demand. To reduce the blockchain consensus delay and ensure the activeness of fog nodes, a practical byzantine fault tolerates (PBFT) consensus mechanism based on the frequency of interaction is investigated. The simulation results verify the effectiveness of the algorithm. The proposed data sharing scheme promotes the interaction of information in the fog computing network.**

***Index Terms*—fog computing, blockchain, D2D data sharing, consensus mechanism**

I. Introduction

In order to improve the various problems of traditional cloud networks, Cisco proposed the concept of "fog computing" in 2014 [1]. Fog computing is characterized by low time delay and mobility. The fog network provides a platform for data sharing between mobile devices. Data sharing enables users to obtain important information around them in time and provides great convenience for users' lives. However, due to the mobility and variability of the devices, the devices do not fully trust each other [2]. When there are malicious users on the network, they will deliberately spread false data, causing confusion to other users' judgments. Therefore, how to effectively evaluate the credibility of data is an important issue in data sharing. In addition, for selfish purposes, data owners are unwilling to participate in sharing; data requesters compete with each other for better data services. This causes an imbalance in data supply and demand between devices [3]. The problem of matching between the provider and the requester is essential to achieve data sharing.

In recent years, blockchain has received more and more attention due to its decentralization and anonymity. It is considered to be one of the effective means to solve privacy and trust issues [4]. Thanks to the distributed consensus algorithm, the blockchain enables all nodes to work together to maintain a consistent database [5]. In [6], the author proposed a data sharing scheme based on the subjective logic of the three rights to ensure safe data sharing between vehicles. However, the author only considers the reputation value of the vehicle, without actually judging the authenticity of the data. If there is an attacker, the correctness of the data is still uncertain. In [2], the author proposed an announcement scheme based on reputation system. The vehicle broadcasts the sensed data to neighbors. Neighbors evaluate the credibility of these messages and upload the feedback to a centralized entity to update the reputation value. As the number of vehicles increases, this may cause a broadcast storm and waste network resources.

Based on the above opportunities and challenges, in order to achieve high-quality data sharing in the fog network, we propose a data sharing scheme empowered by blockchain. First, a two-layer network architecture is proposed. And then, in order to prevent malicious nodes from spreading false information, a Bayesian inference model is used to predict the data, and a multi-factor data evaluation method is proposed to accurately determine whether the data is true or false. We define different utility functions for data requesters and providers, so as to maximize the value of social benefits by one-to-one match. To further reduce the blockchain consensus delay and ensure reliability, a practical byzantine fault tolerates (PBFT) consensus mechanism based on interaction frequency is proposed.

The remainder of this paper is organized as follows: system model are described in Section II; Section III describes data sharing empowered by blockchain; Section IV is numerical results and Section V is the conclusion.

II. System Model

This section introduces the system model of data sharing, including network model and data sharing process.

A. Network Model

The network model is composed of fog nodes (FNs) and Internet of Things (IoT) users, as shown in Fig. 1.

Fog nodes: FNs are usually deployed on routers, switches or smart edge nodes near IoT devices [7]. IoT devices use

978-1-6654-8243-1/22/$31.00 ©2022 IEEE

FNs for data processing, data management and data storage services. A consortium blockchain is established on FNs for data management.

Users: they have wireless communication capability and collects local data through sensing equipment. Each user is associated with FN with the closest communication distance. The user plays different roles according to its different needs. The user that collects and shares data acts as data provider $\mathbb{Q} = \{Q_1, Q_2, \cdots, Q_j\}$; the user that requests data acts as data requester $\mathbb{R} = \{R_1, R_2, \cdots, R_i\}$.

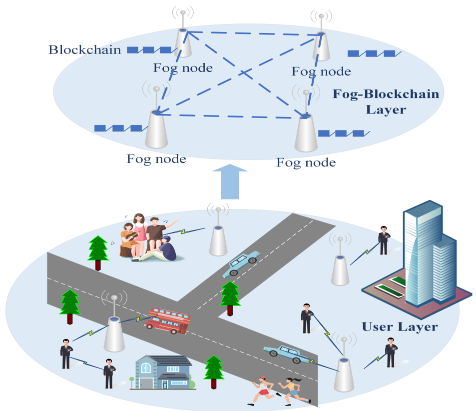


Fig. 1. System model.

B. Data Sharing Process

If the user requests data, it will send an interest packet to FN. Interest packets include timestamps, signatures and data request information. The provider sends a data packet to FN. The data packet includes a timestamp, signature, and information summary, as shown in Fig. 2, step 1. Based on

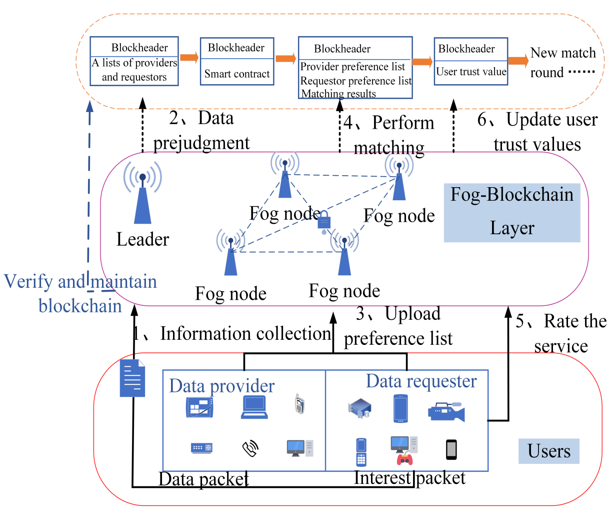


Fig. 2. Data sharing process.

the data information uploaded by each user, the FN predicts the authenticity of the data through Bayesian inference, filters out the correct data and eliminates the data providers who upload false information. Next, the fog node will package the relevant information of the data requester and provider as a transaction. It is written into the blockchain after successful verification by the fog-blockchain layer as shown in Fig. 2, step 2. The user makes his own preference list and uploads it to FN by querying the information on the blockchain, as shownin Fig. 2, step 3. In order to realize the automation of data management, the operation data from provider and the requester are received and stored through the smart contract. The designed matching algorithm is automatically executed, as shown in Fig. 2, step 4. After the data sharing is completed, the participants score the sharing results and upload them to FN, as shown in Fig. 2, step 5. The fog node updates the trust value of the participants as shown in Fig. 2, step 6. Next, the above steps will be repeated to perform the next round of data sharing.

III. Data Sharing Empowered by Blockchain

A. Multi-factor Data Evaluation Method

In the process of data sharing, due to virus infection or selfish reasons, the device may spread wrong information [8]. In order to reduce the impact of these malicious messages on other devices, we designs a method to quantify the authenticity of the data.

1) Pre-judgment Period of data

The pre-judgment of the data is equivalent to the credibility assessment of the data itself. This step is performed on FN. First, FN groups the data uploaded by users $\{E_1, E_2, \cdots, E_{l,\ldots}\}$, E_l represents the data group related to the event e_l. The credibility of the data uploaded by user j is defined as:

$$c_j^l = b + e^{-\alpha d_j^l - \beta t_j^l} \tag{1}$$

where d_j^l is the distance between user j and the location of event e_l, t_j^l is expressed as the time difference between the time t_j when the user j learned the event and the time t when the event occurred, $t_j^l = t_j - t$. b is the lower limit of data credibility, α and β control the rate of change of credibility, $\alpha + \beta = 1$. The shorter the distance between user j and the event occurred, the earlier the time to know the event occurred and the more trustworthy the data is.

The credibility set C^l of the data e_l can be obtained, $C^l = \{c_1^l, c_2^l, c_3^l \cdots\}$. On the basis of obtaining the set of credibility, FN uses the Bayesian model [9] to infer the aggregate credibility P of the event:

$$P(e/C) = \frac{P(e)\prod_{j=1}^{N} P(c_j/e)}{P(e)\prod_{j=1}^{N} P(c_j/e) + P(\bar{e})\prod_{j=1}^{N} P(c_j/\bar{e})} \tag{2}$$

where $\bar{e}$ is the complementary event of e, $P(c_j/e) = c_j$, $P(c_j/\bar{e}) = 1 - c_j$. $P(e)$ is expressed as the prior probability of event e. $P(e/C) \in [0, 1]$. Once $P(e/C)$ exceeds the preset threshold Thr, FN considers the data related to the event to be true; if $P(e/C)$ does not exceed the set threshold, the data is considered unreliable. Users who upload unreliable data will be kicked out of the sharing list in this round of data sharing.

2) Trust Value Based on Experience

The experience-based trust value is the use of the user's past behavior to update the user's trust value and indirectly judge the authenticity of the data. After the data sharing is completed, the requester will score the provider based on the data quality $T_{i,j}$, $T_{i,j} \in (-1, 1)$. FN averages the scores of requesters $T_j^{ave} = \frac{1}{L} \sum_{i \in L} T_{i,j}$, L is the number of requesters interacting with provider j this time. Let s_j denote the trust degree of user j based on experience, $s_j \in (-1, 1)$. The update criteria are as follows:

If $T_{i,j}^{ave} > 0$, s_j is increased to:

$$s_j^{'} = \begin{cases} \lambda^t (1-\eta) s_j + \eta, s_j \geq 0 \\ \lambda^{-t} (1+\eta) s_j + \eta, s_j \leq 0 \end{cases} \quad (3)$$

If $T_{i,j}^{ave} < 0$, s_j is reduced to:

$$s_j^{'} = \begin{cases} \lambda^t (1-\mu) s_j + \mu, s_j \geq 0 \\ \lambda^{-t} (1+\mu) s_j + \mu, s_j \leq 0 \end{cases} \quad (4)$$

where s_k is experience-based trust at the moment, $s_k \in (-1, 1)$, s_k' represents the updated trust. η is a positive increment factor, $0 < \eta < 1$. μ is a negative decrement factor, $-1 < \mu < 0$. We set $\mid \mu \mid > \mid \eta \mid$, once the vehicle has cheated, trust is easily broken and it is difficult to build trust. λ is forgetting factor, $0 < \lambda < 1$. t is the time difference the current interaction time and the previous interaction time. In order to make the accumulated trust value of previous behaviors have less influence on the current moment, discount the previous trust value λ^t or λ^{-t}. So this can slow the rate of increase or decrease experience-based trust.

3) Historical Interaction

When the requester initiates a sharing request to the provider, the satisfaction of the provider's previous services will be measured. This satisfaction is related to the historical interaction between the two. p_{ij} represents the level of satisfaction with the current service, $p_{ij} \in [0, 1]$. The cumulative value of historical interaction is:

$$h_{ij} = \frac{\partial_{ij}}{N} = \frac{\sum_{t_n \in \{t_1, \cdots t_N\}} p_{ij}(t_n)}{N} \quad (5)$$

where ∂_{ij} represents the accumulation of satisfaction, $\partial_{ij} = \partial_{ij} + p_{ij}(t_n)$. The moment when the service is requested is denoted as $t_n = t_N > \cdots > t_2 > t_1$; N is the number of requests for data. A higher N means that the requester has more prior knowledge about the provider, so that it can judge the provider more accurately.

In the provider selection stage, the requester R_i evaluates the provider based on the above three indicators, and establishes the provider score matrix as follows:

$$W_{n \times m} = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & w_{n3} \end{pmatrix} = \begin{pmatrix} c_1 & s_1 & h_1 \\ c_2 & s_2 & h_2 \\ \vdots & \vdots & \vdots \\ c_n & s_n & h_n \end{pmatrix} \quad (6)$$

where n is the number of providers, c_n represents the data l credibility score of Q_n; s_n represents the trust value of Q_n based on experience; h_n represents the historical interaction score of the requester R_i to the provider Q_n. Then, the entropy weight [10] method is used to evaluate the scoring weights of the above three scoring indicators. In order to obtain the standardized evaluation matrix $\hat{W}$, the normalization method is used to normalize the matrix elements:

$$\hat{w}_{jm} = \frac{w_{jm} - \min_j w_{jm}}{\max_j w_{jm} - \min_j w_{jm}} \quad (7)$$

Calculate the weight of each evaluation index m of the provider:

$$v_{jm} = \frac{\hat{w}_{jm}}{\sum_{j=1}^{n} \hat{w}_{jm}} \quad (8)$$

The information entropy H_m of the evaluation index m is:

$$H_m = -\frac{\sum_{j=1}^{n} v_{jm} \ln v_{jm}}{\ln n} \quad (9)$$

Normalize the information entropy of the rating index m:

$$a_m = \frac{1 - H_m}{\sum_{m=1}^{3} (1 - H_m)} \quad (10)$$

Get the current satisfaction score of requester R_i for each provider Q_n with respect to data j:

$$G_j = \sum_{m=1}^{3} \hat{w}_{jm} a_m \quad (11)$$

B. *Problem Definition and Solution*

In this section, we define different utility functions for the requester and the provider, study the balance of data supply and demand from the perspective of one-to-one matching.

1) Utility Function Definition

For a requester, the quality of the data not only depends on the accuracy of the data, but also depends on the timeliness of the data obtained [10]. We define the requester's satisfaction with the transmission delay $S_{R_i Q_j}^l$:

$$S_{R_i Q_j}^l = d_{R_i}^{\exp} - d_{R_i Q_j}^{tra} \quad (12)$$

We use the difference between the expected data transmission delay and the actual service delay to reflect the requester's true transmission delay satisfaction. $d_{R_i}^{\exp}$ is the expected transmission delay of l. $d_{R_i Q_j}^{tra}$ is the actual transmission delay of data l from Q_j to R_i. $d_{R_i Q_j}^{tra} = f_j^l / R_{j,i}$, f_j^l is the data size, $R_{j,i}$ is the transmission rate. The requester always hopes to obtain high-quality data in the shortest time, so the requester's utility is defined as:

$$UR_i^l = \varphi G_j + (1 - \varphi) S_{R_i Q_j}^l \quad (13)$$

where $\varphi \in (01)$ is used as a weight to dynamically adjust the proportion of data satisfaction and latency satisfaction.

In the transaction, the data provider is concerned about the cost and benefits of sharing data, so the benefit of the provider is defined as:

$$UQ_j^k = \rho\mu d_{R_iQ_j}^{tra}\left(1+\frac{\exp(s_j-\xi)}{M}\right) \tag{14}$$

where ρ, μ represents the cost per unit energy and the transmission power of D2D respectively. Suppose there are M competing providers, $\xi \in (0,1)$ is an adjustable parameter. The profit of the provider increases with the increase of s_j and the decrease of M. For potential providers, a higher s_j will get a higher return, which encourages users to truly participate in long-term data sharing.

This paper transforms the data sharing problem into a matching problem between the requester and the provider. We define the optimal variable matching decision as $w_{R_iQ_j}$, $w_{R_iQ_j}=1$ indicates that the provider Q_j shares the data with the requester R_i, otherwise $w_{R_iQ_j}=0$. In order to maximize the overall welfare of data sharing, the objective function of the trading system is:

$$\begin{aligned}
&\max_{w_{R_iQ_j}} \sum_{i\in N}\sum_{j\in M} w_{R_iQ_j}UR_i^k + \beta\sum_{j\in M}\sum_{i\in N} w_{R_iQ_j}UQ_j^k\\
&\text{s.t. } C1: w_{R_iQ_j}UR_i^k > 0, \forall i\in N, \forall j \in M\\
&\quad C2: w_{R_iQ_j} = \{0,1\}, \forall i\in N, \forall j\in M\\
&\quad C3: \sum_{j\in\mathbb{Q}} w_{R_iQ_j} \le 1, \forall i \in \mathbb{R}\\
&\quad C4: \sum_{i\in\mathbb{R}} w_{R_iQ_j} \le 1, \forall j \in \mathbb{Q}
\end{aligned} \tag{15}$$

among them, the constraint $C1$ guarantees that the utility of the requester cannot be negative. $C2$ means that the value of $w_{R_iQ_j}$ is either 0 or 1; $C3$ means that the requester can only request data from one provider; $C4$ means that the provider provides services to one requester.

2) The Matching Algorithm Design

In order to solve the above optimization problems, we base the delayed acceptance algorithm [12] to design a data supply-demand algorithm. Transform the optimization function into a stable marriage problem with a preference list.

$Definition 1 (One\text{-}to\text{-}One\, Matching)$: Define a one-to-one function f: $\mathbb{Q} \to \mathbb{R} \bigcup \{\varnothing\}$, such that

1) $\forall Q_j \in \mathbb{Q}, f(Q_j) \in \mathbb{R}$, and $|f(Q_j)| \in \{0,1\}$;

2) $\forall R_i \in \mathbb{R}, f(R_i) \in \mathbb{Q}$, and$|f(R_i)| \in \{0,1\}$; where Q_j is the j-th provide, R_i is the i-th requester. $f(Q_j) = R_i \Leftrightarrow f(R_i) = Q_j$ means that if provider Q_j matches requester R_i, requester R_i also matches provider Q_j.

At the initialization step, the requester and the provider construct a preference list (P_{R_i} and P_{Q_j}) according to their utility (UR_i^l and UQ_j^l) in descending order. Define a candidate list of providers V_{Q_j} which is continuously updated with each round of iteration. Requester R_i makes a matching request to the most preferred provider Q_{j^*}. If R_i is not in Q_{j^*}'s preference list, R_i will delete Q_{j^*} from his preference list P_{R_i} and send an invitation to the next most preferred provider. If R_i is in Q_{j^*}'s preference list, Q_{j^*} will add R_i to the candidate list V_{Q_j}. Next, provider Q_{j^*} will choose the most preferred R_{i^*} in V_{Q_j} for matching, $w_{R_i{}^*Q_j^*} = 1$. The remaining requesters (except R_{i^*}) are rejected by Q_{j^*}. Similarly, they delete Q_{j^*} from the preference list and invite the next more preferred provider. If the matching result of this round is consistent with the previous round, then the match ends.

C. Consensus Mechanism Based on Interaction Frequency

In order to avoid consuming too much energy and incentivize FNs to actively participate in the network, we adopt a practical byzantine fault tolerants (PBFT) [11] consensus protocol based on the frequency of interaction. The leader is responsible for the generation of blocks and gets corresponding rewards. FN with the highest interaction frequency with users during T period is selected as the leader. The interaction frequency is defined as follows:

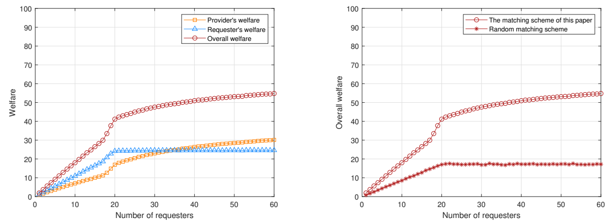
$$Fr_y = \frac{\sum_{x=1}^{X} f_y^x}{\sum_{y=1}^{F}\sum_{x=1}^{X} f_y^x} \tag{16}$$

where f_y^x is the number of interactions between the FN y and the user x (requester, provider) within the communication range. The larger the ratio is, the content processed by FN is at most within T; it will obtain the right to package and win block rewards.

The total number of consensus nodes is n, and the number of abnormal nodes allowed by the byzantine fault tolerance mechanism is f, $f = (n-1)/3$. In time T, the leader packs the data collected in the consensus layer into a block, broadcasts the data block with timestamp and signature to other nodes for verification. After each consensus node receives the block content from the leader, it starts to audit the correctness of the content. After the audit is completed, their signatures will be added to the audit results and broadcast to other FNs. If the audit result collected by the consensus node from other nodes is greater than $2f$, the node sends a confirmation message to other nodes, indicating that the node's preparation phase has been completed. If the consensus node collects $2f+1$ confirmation messages, indicating that a consensus has been reached, the block is written to the blockchain.

IV. NUMERICAL RESULTS

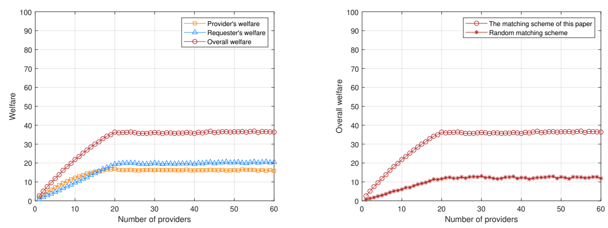
This section uses the MATLAB simulation platform to verify the matching algorithm in the blockchain-enabled data sharing and the performance of the blockchain. FN radius is 300m, Thr is 0.5. The size of requested data follows a logarithmic distribution between 0M Bytes and 100M Bytes. Fig. 3 shows matching results is evaluated by changing the number of requester. The number of providers is set to 20. As shown in Fig. 3 (a), when the number of requester is less than the number of providers, the utility of both requester and providers will increase with the number of requester. Since the requester chooses to match the most preferred provider, his utility grows faster than the provider. When the number of requester increases, there may be more requester competing



(a) Welfare (b) Overall welfare

Fig. 3. Matching results as the number of requester increasesing.

for the same provider, which will lead to better choices for providers and increase their utility.



(a) Welfare (b) Overall welfare

Fig. 4. Matching results as the number of provider increases.

Fig. 4 shows matching results is evaluated by changing the number of provider. The number of requester is set to 20. As shown in Fig. 4 (a), when the number of providers is less than the number of requester, the providers have more options, so the utility grows faster. When the number of providers is greater than the number of requesters, the utility of providers will decrease slightly and basically stabilize. This is because the requestor can basically choose the provider he likes.

In order to highlight the superiority of the matching method, we compared the method with random matching, as shown in Fig. 3 (b), Fig. 4 (b). In the beginning, the total benefit value is proportional to the number of requesters(providers). The matching scheme in this paper has a larger increase in the total benefit value than the random matching scheme, which shows that the matching algorithm proposed in this paper can find the best partner for users. When the ratio of supply to demand reaches 1, the total utility continues to remain unchanged.

Fig. 5 shows the impact of different block sizes, the number of consensus nodes, and different consensus algorithms on latency. The traditional DPoS consensus algorithm and the joint PoW and PoS consensus algorithm [2] have longer delays than our consensus algorithm. The consensus algorithm we proposed only performs consensus processing on FNs, rather than all nodes in the network, which greatly saves consensus delay. As the block size increases, the delay also increases. This is because the more content contained in a block, the greater the transmission and network delay.

V. CONCLUSION

In this paper, a D2D data sharing method empowered by blockchain was proposed. A multi-factor data evaluation method was employed to make the authenticity judgment

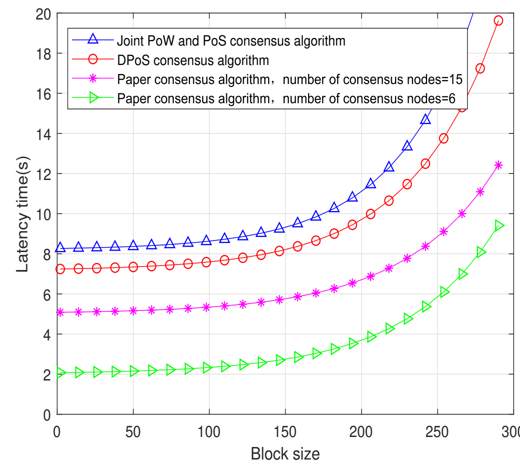


Fig. 5. Delay of consensus mechanism PBFT based on interaction frequency.

of data more accurate. We employed a one-to-one matching algorithm to optimize the correlation between data supply and demand. Blockchain technology further enhanced the authenticity of data. More importantly, PBFT consensus mechanism based on interaction frequency was adopted to encourage fog nodes to actively participate in data sharing and reduce consensus delay.

REFERENCES

[1] Ni J, Zhang K, Lin X, et al. Securing Fog Computing for Internet of Things Applications: Challenges and Solutions[J]. IEEE Communications Surveys & Tutorials, 2018, 20(99):601-628.

[2] Yang Z, Yang K, Lei L, et al. Blockchain-based Decentralized Trust Management in Vehicular Networks[J]. IEEE Internet of Things Journal, 2018:1-1.

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

[4] H. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A survey," IEEE Internet Things J., vol. 6, no. 5, pp. 8076–8094, Oct. 2019

[5] S. Seng, C. Luo, X. Li, H. Zhang and H. Ji, User Matching on Blockchain for Computation Offloading in Ultra-Dense Wireless Networks, IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 1167-1177, 1 April-June 2021.

[6] Kang J, Yu R, Huang X, et al. Blockchain for Secure and Efficient Data Sharing in Vehicular Edge Computing and Networks[J]. IEEE Internet of Things Journal, 2019, 6(3):4660-4670.

[7] Xiao T, Cui T, Islam S et al, "Joint Content Placement and Storage Allocation Based on Federated Learning in F-RANs," Sensors, 2020, 21(1):215.

[8] Chen C, Wang C, Qiu T, et al. A Secure Content Sharing Scheme based on Consortium Bolckchain in Vehicular Named Data Networks[J]. IEEE Transactions on Industrial Informatics, 2019, PP(99):1-1.

[9] Raya M, Papadimitratos P, Gligor V D, et al. On Data-Centric Trust Establishment in Ephemeral Ad Hoc Networks[C]// Infocom the Conference on Computer Communications IEEE. IEEE, 2008.

[10] Yu Y, Liu S, Guo L, et al. CrowdR-FBC: A Distributed Fog-Blockchains for Mobile Crowdsourcing Reputation Management[J]. IEEE Internet of Things Journal, 2020, PP(99):1-1.

[11] Kouicem D E, Y Imine, Bouabdallah A , et al. A Decentralized Blockchain-Based Trust Management Protocol for the Internet of Things[J]. IEEE Transactions on Dependable and Secure Computing, 2020, PP(99):1-1.

[12] S. Bayat, Y. Li, L. Song, and Z. Han. Matching theory: Applications in wireless communications, IEEE Signal Process. Mag., vol. 33, no. 6, pp. 103–122, Nov. 2016.