TrustWorker: A Trustworthy and Privacy-Preserving Worker Selection Scheme for Blockchain-Based Crowdsensing

Sheng Gao, Xiuhua Chen, Jianming Zhu, Xuewen Dong, and Jianfeng Ma

Abstract—Worker selection in crowdsensing plays an important role in the quality control of sensing services. The majority of existing studies on worker selection were largely dependent on a trusted centralized server, which might suffer from single point of failure, the lack of transparency and so on. Some works recently proposed blockchain-based crowdsensing, which utilized reputation values stored on blockchains to select trusted workers. However, the transparency of blockchains enables attackers to effectively infer private information about workers by the disclosure of their reputation values. In this article, we proposed the TrustWorker, a trustworthy and privacy-preserving worker selection scheme for blockchain-based crowdsensing. By taking the advantages of blockchains such as decentralization, transparency and immutability, our TrustWorker could make the worker selection process trustworthy. To protect workers' reputation privacy in our TrustWorker, we adopted a deterministic encryption algorithm to encrypt reputation values and then selected the top N workers in the light of secret minimum heapsort scheme. Finally, we theoretically analyzed the effectiveness and efficiency of our TrustWorker, and then conducted a series of experiments. The theoretical analysis and experiment results demonstrate that our TrustWorker can achieve trustworthy worker selection, while ensuring the workers' privacy and the high quality of sensing services.

Index Terms—Crowdsensing, worker selection, blockchain, reputation privacy, minimum heapsort

1 Introduction

With the pervasiveness of smart devices and wireless communications, crowdsensing as a new sensing paradigm has emerged [1]. As a subclass of crowdsourcing, it empowers individuals with sensor-rich smart devices to conduct large-scale data collection without deploying static sensors [2]. Because of the advantages on extensive coverage, low deployment cost and so forth, it has attracted researchers from various fields such as traffic [3], environment [4], and healthcare [5]. Typically, there are three entities involved, named a requester, a group of workers and a trusted centralized server (TCS) [6]. Generally, the requester delegates operations such as task allocation, worker selection, results aggregation and disputes arbitration to the TCS [7]. There are many representative examples of crowdsensing. For example, Upwork [8] is currently the world's largest freelance market, which requires requesters to deposit a certain amount of payment into their escrow account before posting tasks. Based on this platform, requesters can hire workers to design or write, and workers compete with each other to obtain opportunities for task execution and rewards. In addition, both the winning workers and requesters need to pay a certain percentage of processing fee to Upwork. In Amazon Mechanical Turk∼(MTurk) [9], requesters can be individuals or companies, and they pay to recruit workers. Workers interested in tasks can submit the request for participation on the platform. MTurk labor market has been widely used by researchers in various fields to recruit workers for data collection and data labeling. Essentially, both Upwork and MTurk are a trusted centralized platform. Apparently, the success or not of crowdsensing rests on the traditional trust-based model, which might suffer from the inevitable issues such as single-point failure, privacy leakage, the lack of transparency in operations and performance bottleneck. There have been some works that use the technical advantages of blockchains such as decentralization, transparency and immutability to build blockchain-based crowdsensing (BBC) to alleviate these issues caused by TCS [10], [11], [12], [13]. However, none of them intensively investigate the worker selection problem in BBC.

Worker selection is the crucial component for crowdsensing to control the quality of sensing services [14], [15], [16], [17]. Due to the openness and low threshold of crowdsensing, any worker can freely participate in and submit the collected data. To get the rewards, some misbehaving workers might intentionally submit poor quality even erroneous data, which would discourage those requesters' enthusiasm for use and hinder the prosperity of crowdsensing [18], [19], [20]. Therefore, how to select reliable workers while guaranteeing a higher data quality is the key factor for the sustainable development of crowdsensing.

- *Sheng Gao, Xiuhua Chen, and Jianming Zhu are with the School of Information, Central University of Finance and Economics, Beijing 100081, China. E-mail: {sgao, zjm}@cufe.edu.cn, xiuhuachen1003@163.com.*
- *Xuewen Dong is with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China. E-mail: xwdong@xidian.edu.cn.*
- *Jianfeng Ma is with the School of Cyber Engineering, Xidian University, Xi'an 710071, China. E-mail: jfma@mail.xidian.edu.cn.*

Manuscript received 17 November 2020; revised 3 June 2021; accepted 4 August 2021. Date of publication 10 August 2021; date of current version 9 December 2022. (Corresponding author: Sheng Gao.)
Digital Object Identifier no. 10.1109/TSC.2021.3103938

There have been some works that concentrate on selecting highly reliable workers to maximize the sensing data quality or minimize the maintenance cost [14], [15], [16], [21], [22]. However, these schemes essentially are centralized and the issues caused by the traditional trust-based model remain. In view of the fact that reputation values have been proved to be an effective measure to differentiate between workers of honest and malicious [23], [24], some works use the blockchain to build BBC and adopt reputation values as a screening indicator of reliability for worker selection [25], [26], [27], [28], [29]. Specifically, they store reputation values of workers on blockchains to ensure the trustability. Due to the transparency of blockchains, all of these works would lead to the privacy leakage by the reveal of reputation values on blockchains. However, there are rarely works concentrate on the privacy issue. The few works only study the data privacy during reputation computation [27], [30], [31]. The main idea of them is to use popular technologies such as blind signature, k-anonymity, and additive secret sharing to confuse the connection between user identity and data reports or the connection between reputation evaluators and reputation feedback. However, the reputation privacy of workers themselves indeed have been ignored. The disclosure of reputation values could have a negative effect on future sensing task allocation, revenues and even personal safety [26], [32], [33]. For example, in cooperative tasks, the disclosure of reputation values helps workers form gangs with each other but damages the fairness of worker selection. Besides, the importance of workers' weight information(that is, reputation values in this paper), which directly affect workers' rewards, is emphasized in [34], [35], but worker selection is not their focus. Although workers can use pseudonyms to complete transactions, they are advised to use real identities that can be authenticated in certified institutions to register, thereby increasing the possibility of accepting tasks [11]. Therefore, the issue of reputation protection is worth studying. However, how to select trustworthy workers for BBC while protecting reputation privacy is an urgent problem to be solved.

In addition, as for reputation-based worker selection, we can analyze it from two different perspectives. One is to select workers based on past profiled reputation. This method considers the influence of workers' past task performance on the current worker selection. The other is to evaluate reputation based on the workers' current task completion, and then select workers based on current reputation. This method directly calculates reputation in time based on workers' current task performance. For the former, we can only think that the selected high-reputation workers have a greater possibility to provide high-quality sensing data for requesters. In fact, this method cannot guarantee that the selected workers will be able to complete the current task well, which even may reduce the task completion quality. However, the latter focuses on improving the completion quality of a certain task, and reputation evaluation results of workers are closely related to the current task completion status. In this way, workers must complete the current sensing task on time and earnestly if they want to be successfully selected, which is conducive to ensuring the data quality in a certain. Although this method seems to favor the requester, it is also a common and reasonable selection method and helps to ensure the quality of task completion. Hence, both types of reputation-based worker selection are reasonable.

In this paper, we propose TrustWorker, a trustworthy and privacy-preserving worker selection scheme for BBC. Since reputation values can be used as a screening indicator of reliability for worker selection, that is high-reputation workers are considered to be able to provide higher data quality, we can select workers with high reputation values for guranteeing the high quality of sensing services. In this paper, to meet the needs of requesters to improve the completion quality of current tasks, TrustWorker uses the current reputation evaluation results as the screening indicator to select workers. Specifically, workers' reputation values are calculated through the results of each data quality evaluation. Compared with relying only on the historical reputation values, Trustworker can more accurately reflects the behavior of workers based on the real-time reputation evaluation, so as to select the high-reputation workers for improving the sensing service quality. This selection method is also reasonable and can be effectively used in practical applications, which is conducive to ensuring the completion quality of current tasks. For example, for design-type tasks, the requester selects excellent workers by evaluating the designated works submitted by workers on the crowdsourcing platform, such as Amazon Mechanical Turk, and then pays rewards for them. Therefore, no matter for workers or requesters, such a worker selection is equally fair.

In terms of reputation privacy, although it is an effective method to hide the connection between the reputation value and the real identity of worker by assigning different random IDs to workers, maintaining the matching relationship between the ID and the identity will also incur additional overhead. In this paper, we choose encryption mechanisms to protect reputation privacy, but how to perform worker selection based on reputation ciphertexts is also a new problem. In TrustWorker, we find that it only needs to determine the top N workers with higher reputation values instead of obtaining the size order of all reputations. At that point, we propose to encrypt the reputation values and then store the ciphertexts on blockchains for guranteeing the reputation privacy. Our TrustWorker can select the workers with higher encrypted reputation values for providing high quality of sensing services in the light of secret minimum heapsort scheme. In addition, we also consider the selection fairness of workers with the same reputation values to improve the enthusiasm of workers. To the best of our knowledge, our TrustWorker is the first work to guarantee trustworthy worker selection operation with reputation privacy protection for BBC. The main contributions in this paper are summarized as follows.

- *Blockchain-based crowdsensing*. To alleviate the issues caused by traditional trust-based model, we propose the TrustWorker, which takes the advantages of blockchain to achieve more trustworthy worker selection while guaranteeing reputation privacy. Different from existing work, TrustWorker displays workers' reputation values in ciphertext and then uses them in worker selection for meeting workers' reputation privacy protection requirements.
- *Reputation privacy protection*. Different from previous works on data privacy protection during reputation

computation, to the best of our knowledge, our TrustWorker is the first work that proposes to store encrypted reputation values on blockchains, which can offer both trustability and reputation privacy.

- *Trustworthy worker selection.* We find that it only needs to determine the top N workers with higher reputation values instead of obtaining the size order of all reputations. Therefore, unlike the previous trustworthy worker selection based on plaintext reputation values in BBC, our TrustWorker selects the top N workers with higher encrypted reputation values for providing high quality of sensing services in the light of secret minimum heapsort scheme.
- *High-quality sensing services.* In TrustWorker, since the selected high-reputation workers provide high-quality sensing data, the requester will eventually receive high-quality sensing services. According to simulation results, our TrustWorker can decrease the data error by adjusting the number of workers and subtasks. This will help ensure that the selected workers in BBC provide higher quality of sensing services.
- *Dynamic reputation update.* According to the workers' performance behaviours in BBC, our TrustWorker can effectively dynamic update the reputation values, which is the criterion for worker selection. To obtain higher reputation values for increasing the probability of being selected, our TrustWorker incents workers to provide high-quality sensing services.

The remainder of this paper is structured as follows. Section 2 introduces related works that focus on worker selection in crowdsensing. We present the system model and problem statement in Section 3. The detailed design of the proposed TrustWorker is elaborated in Section 4, followed by theoretical analysis in Section 5 and performance evaluation in Section 6. Finally, Section 7 summarizes the entire paper.

2 Related Works

Worker selection is an important component in crowdsensing. Apparently, high-reliability workers are more likely to provide high-quality of sensing services [34], [35], [36]. Most of existing works concentrate on how to select reliable workers to maximize the sensing services quality or minimize the maintenance cost.

2.1 Worker Selection in Traditional Trust-Based Model

Some works model the worker selection as an optimization problem. For example, Guo *et al.* [14] proposed two worker selection schemes, where one is to minimize the total moved distance for time-sensitive tasks and the other is to minimize the total number of workers for delay-tolerant tasks. Wang *et al.* [15] proposed a prediction-based user-recruitment strategy for mobile crowdsensing to minimize the sensing data uploading cost. Yang *et al.* [16] extended the user-recruitment strategy in [15] for spatio-temporal sensitive tasks. Although they model the worker selection problem as a target optimization problem for achieving requesters' different requirements, the worker selection is all done by a third-party platform, ignoring the fairness of worker selection results. Some others proposed to use reputation for worker selection. For example, Wang *et al.* [37] proposed to use the reputation level as an influencing factor for worker selection and remuneration payment in mobile crowdsourced sensing. Pouryazdan *et al.* [38] studied two user reputation scoring approaches based on statistical and voting, and then synthesized them to calculate the collaborative reputation score for user recruitment. It combines the decentralized and centralized approach for reputation-aware worker selection to improve the data trustworthiness, but ignoring reputation privacy leakage. Wang *et al.* [39] designed a trustworthy crowdsourcing framework by introducing the social cloud which serves as a service provider and proposed a reputation-based auction mechanism to select the reliable workers, where reputation values store in a database maintained by the social cloud. However, the worker selection operation in these works is conducted by a TCS, which is subject to the issues mentioned above of the traditional trust-based model.

2.2 Worker Selection in Blockchain-Based Crowdsensing

To alleviate the issues in the traditional trust-based model, some recent works proposed to conduct worker selection operation in BBC. For example, Wei *et al.* [25] proposed to use blockchains to store evaluated reputation values for reliable worker selection in BBC. Kadadha *et al.* [26] proposed a decentralized crowdsensing framework for multiple requesters with multiple workers built on Ethereum blockchain (SenseChain). In SenseChain, the Quality Information (QoI) of the tasks is evaluated by integrating three independent metrics of reputation value, time and distance, so as to determine the set of workers with maximum QoI based on the number of workers required by the requester. It can achieve unbiased and trustworthy worker selection, however, it performs all the process in plaintext environment and ignores the privacy of reputation values and sensing data. Chatzopoulos *et al.* [40] considered the influence of the opinions of different requesters on worker selection. It uses a beta distribution to determine the probability distribution that workers will be able to return results in time, and allows different requesters to exchange their opinions about others. By this way, the requester can improve the knowledge about workers as much as possible to make the most appropriate selection, and then update the reputation of workers based on the task completion results. However, privacy protection is not the focus of this work. Zhao *et al.* [27] constructed a blockchain-based mobile crowdsensing system and then presented a privacy-preserving reputation management scheme in order to defend against malicious workers. It can ensure the data privacy during reputation computation based on the additive secret sharing, but neglecting the reputation privacy of workers. Duan *et al.* [41] also only focused on data privacy protection. It uses threshold Paillier cryptosystem and differential privacy to ensure sensing data privacy, introduces SGX technology to ensure the correctness of data aggregation, and uses zero-knowledge proof to resist invalid data provided by data providers. Bhatia *et al.* [42] proposed a blockchain-based crowdsourcing platform

TABLE 1
Symbols and Descriptions

Symbols	Descriptions
r	A requester
w_i	The ith worker
t_z	The zth subtask
$x_{t_z}^{w_i}$	Observed data for subtask t_z of worker w_i
rep_{w_i}	Reputation value of worker w_i
$x_{t_z}^*$	Estimated true data of subtask t_z
$\bar{x}_{t_z}$	Average observed data of subtask t_z
$d(x_{t_z}^{w_i}, x_{t_z}^*)$	Distance between $x_{t_z}^{w_i}$ and $x_{t_z}^*$
std_{t_z}	Standard deviation of all observed data $x_{t_z}^{w_i}$ on t_z
Q	Number of workers submitted data to CS_k
Z	The number of subtasks
K	The number of CS
CN_i	The node on the blockchain
PN	The node selected to be a primary node
CS_k	The kth CS, k=1,2,...,K
num_k	Number of reputation ciphertexts provided by CS_k
p_k	Plaintext set of num_k reputation values
$p_{k,j}$	The jth value in p_k, j=0,1,...,$num_k - 1$
$DETcipher_k$	DET ciphertext set of p_k encrypted by CS_k
$DETcipher_{k,j}$	The jth value in $DETcipher_k$, j=0,1,...,$num_k - 1$
N	Number of selected workers

with a robust reputation management scheme. Specifically, worker reputation values are calculated by aggregating the task data evaluation results from other equivalent participants on the platform for reliable worker selection by the task requester. It needs a sufficiently attractive incentive mechanism to encourage workers who are required to have the corresponding computing skills and conditions to actively participate in task evaluation. Ding *et al.* [43] proposed a reputation mechanism and an arbitration mechanism for worker selection and data evaluator selection respectively, so as to improve the sensing services quality and ensure the fairness of task data evaluation. Chatzopoulos *et al.* [44] introduced Internet service providers(ISPs) to complete location-based task recommendation, and designed a cost-optimal auction to select workers, thereby minimizing the cost of crowdsensing providers. In terms of privacy protection, ISPs need to assign different IDs to participants each time to protect identity and location privacy, which may require additional overhead to maintain the matching between IDs and identities, thereby facilitating operations such as reputation transfer and asset transfer. Unfortunately, most of the above works ignore the privacy of reputation values caused by the transparency of blockchains.

3 System Model and Problem Statement

In this section, we present the system model of our TrustWorker, and then introduce the problem statement. For clarity, we first list some notations used in this paper in Table 1.

3.1 System Model

We introduce the system model of our TrustWorker in Fig. 1, which includes the requester, workers, computing servers and the blockchain.

- *Requester*. A requester exists in the form of an individual or an organization, which publishes sensing tasks with a certain deposit and recruits trustworhty and reliable workers to provide high-quality sensing services.
- *Workers*. Workers can make use of mobile sensing devices (such as smartphones, tablets, wearable devices, etc.) to collect data of the selected sensing task based on their willingness and submit task data to computing servers. Workers are semi-honest, that is, workers will perform operations in accordance with the process, but they are still curious about the sensing data of others. Workers for providing high-quality data will obtain high reputation values and then be selected in worker selection, thereby obtaining corresponding rewards provided by the requester.
- *Computing Servers*. Computing servers are responsible for aggregating task data submitted by workers in a local region to calculate worker reputation values, and then submit corresponding reputation ciphertexts to the blockchain based on worker selection requirements. In the worker selection process, computing servers provide reputation comparison results when necessary by executing the secure two-party reputation comparison protocol. Computing servers can be regarded as a representative of a region and they are assumed to be trusted since they possesses task data of the workers in the local region for the purpose of evaluating reputation values, thereby ensuring the reliability of reputation for using in the worker selection, but the final worker selection result is not determined by them. The final trustworthy worker selection result is ensured by consensus verification performed by nodes on the blockchain. Hence, computing servers can be trusted in terms of its functionalities, but they may still curious about the privacy of workers in other regions. In this paper, the trustability of computing servers means that they can ensure the fairness and reliability of the reputation calculation. This makes sense because these reliable reputation values can be regarded as an effective evaluation indicator in the decision-making or selection process.
- *Blockchain*. A consortium blockchain is adopted in TrustWorker. Workers or requesters who are willing to participate in the sensing tasks in crowdsensing

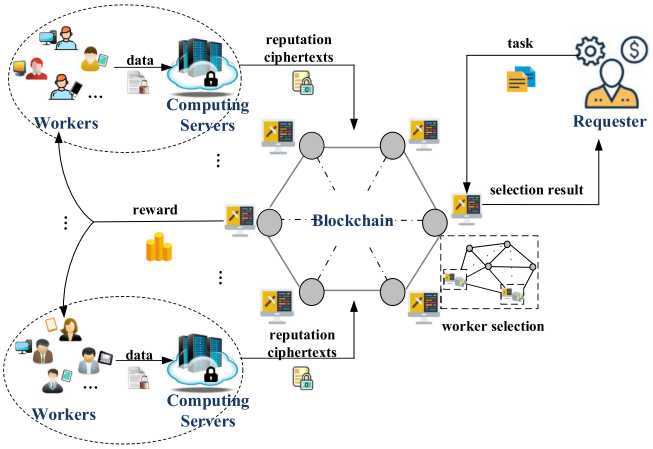


Fig. 1. System model of TrustWorker.

can be nodes on the blockchain. These nodes are responsible for completing the worker selection based on reputation ciphertexts and then perform the consensus verification, which can ensure the trustworthy worker selection results.

3.2 Problem Statement

To perform the trustworthy and privacy-preserving worker selection in BBC, it is necessary to solve some existing problems.

- *How to protect reputation privacy?* Reputation values are regarded as a screening indicator for worker selection, hence storing reputation values on blockchains can ensure the trustability. However, due to the openness and transparency of blockchains, there exists the privacy leakage by the reveal of reputation values, thereby reducing the participant enthusiasm of workers and affecting the completion of tasks.
- *How to ensure trustworthy and reliable worker selection for higher sensing services quality?* Worker selection has an effect on the sensing services quality obtained by the requester. Trustworthy worker selection can ensure high-quality sensing services and enhance the positivity of workers. If the worker selection process is only performed by a single party, it may cause untrustworthy worker selection because a single party can alter the worker selection result to favor some workers over others.
- *How to prevent dishonest behaviors from participants?* Some dishonest behaviors may exist in BBC. For example, the dishonest requester may try to repudiate the payment or pay insufficient rewards after obtaining task data. These dishonest behaviors will hinder the sustainable development of crowdsensing.

Achieving trustworthy worker selection while ensuring reputation privacy becomes an essential process in BBC. Our TrustWorker can solve the above problems for guranteeing the trustworthy and privacy-preserving worker selection, which will be discussed in Section 5.

4 Construction of TrustWorker

In this section, we display the workflow of TrustWorker. With the secret minimum heapsort, we design a novel worker selection scheme for obtaining the worker selection result. TrustWorker not only protects the reputation privacy of workers, but also guarantees the trustability and fairness of worker selection results.

4.1 Overview of TrustWorker

In crowdsensing, the worker selection is a pivotal research issue. To obtain accuracy data results, the requester expects to recruit reliable workers to complete the data collection. In this paper, we assume that the requester distributes a task set $T = \{t_1, t_2, \ldots, t_Z\}$. For each subtask t_z, we use $x^*_{t_z}$ to denote the estimated true data of subtask t_z, and use $x^{\tilde{w}_i}_{t_z}$ to represent the observed data of worker w_i. Due to the possibility of unreliable observed data submitted by workers, TrustWorker takes the reputation value as the screening indicator for worker selection, where the worker reputation

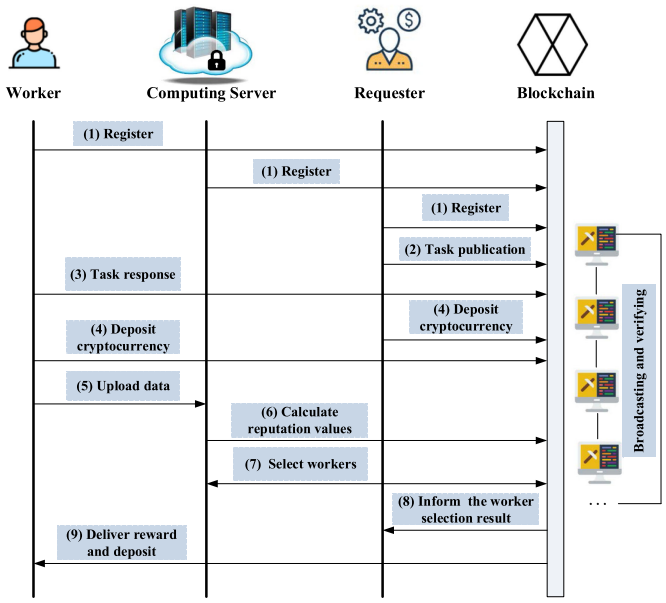


Fig. 2. The overall process of TrustWorker.

value is calculated based on the quality of observed data. Worker reputation ciphertexts will be stored on the blockchain for being used in the worker selection. Furthermore, to ensure the trustability and fairness of selection results, TrustWorker adopts the secret minimum heapsort scheme to select the top N workers. Each valid worker selection result after verification will be recored in the distributed ledger. TrustWorker aims to protect the reputation privacy and guarantee the trustworthy worker selection. We present the overall process of TrustWorker, as shown in Fig. 2.

(1) *Register*. The requester, workers and computing servers all need to register with the blockchain. Each participant is assigned with a public key pair and a certificate. The participant's registered information as a transaction will be recorded in the distributed ledger.

(2) *Task Publication*. The requester r posts the task set to any node CN_i on the blockchain. The task set information includes task set ID, subtask description, execution deadline, task rewards and worker selection requirements. After receiving the task set, CN_i will first verify the registration status of the requester and then broadcast the task through the blockchain if the registration information is valid.

(3) *Task Response*. Workers can personally download the task set which is uncompleted from the blockchain according to their willingness and equipment resources. Then, each task participation request will be submitted to the blockchain. The participation request of workers who have successfully registered will be accepted and then these workers will become candidate participants.

(4) *Deposit Cryptocurrency*. In order to ensure the fairness of transactions, the requester and workers are required to deposit an amount of cryptocurrency. Specifically, the requester is required to deposit an amount of cryptocurrency in advance to the blockchain for the sake of preventing the refusal of payment. The deposit cannot be retrieved unless the requester doesn't receive any task data. Besides, workers are also required to deposit to the blockchain for resisting the refusal of data submittion.

(5) *Upload Data*. After completing tasks, workers can select one CS_k based on the area location and then upload observed data with the timestamp to CS_k which can obtain the task information from the blockchain. If the execution time of the worker exceeds the deadline, the observed task is marked as invalid. Before the execution deadline, the worker w_i uses the public key PK_{CS_k} of CS_k to encrypt observed data and then calculates the hash value of ciphertext. The worker w_i submits the ciphertext, the ciphertext hash value and corresponding signature to CS_k. CS_k will verify the identity of w_i and calculate the hash value of ciphertext to verify the integrity of observed data, and then decrypt the ciphertext data with SK_{CS_k} after receiving them. By evaluating the quality of observed data, CS_k can calculate the reputation value for each worker.

(6) *Calculate Reputation Values*. The reputation value rep_{w_i} is updated according to the data quality provided by the worker w_i in each task. CS_k will evaluate the data quality by calculating the distance between observed data and estimated true data, thereby calculating rep_{w_i} for each worker. These reputation values will be used for the worker selection. Based on the worker selection requirements, that is the setting number of selected workers, CS_k encrypts the top N reputation values in the local group based on the deterministic encryption algorithm(DET) to ensure the privacy security of reputation values, and then submit these DET reputation ciphertexts to the blockchain. In each worker selection, CS_k will update the key, which will be used for the new round of reputation encryption.

(7) *Select Workers*. The requester r submits the task set to the blockchain for the purpose of selecting reliable workers to complete tasks. For the valid task posting request, the worker selection result can be obtained by selecting the final top N workers with higher reputation values based on these local top N DET reputation ciphertexts uploaded by each CS_k. In the case of only knowing DET reputation ciphertexts, the primary node PN selected in advance performs the secret minimum heapsort scheme to select the top N workers and then broadcasts the worker selection result which contains the top N selected workers and the task set ID in the form of a transaction. Each verification node CN_i simulates the implementation of the secret minimum heapsort scheme to verify the transaction. Each valid transaction will eventually be recorded in the distributed ledger.

(8) *Inform the Worker Selection Result*. The requester r can obtain the worker selection result for the specific task set ID from the blockchain. These observed data collected by the selected workers will be submitted to the requester afterwards.

(9) *Deliver Reward and Deposit*. After the valid worker selection result is confirmed, these selected workers can obtain rewards and retrieve their deposit by submitting valid task data.

4.2 Reputation Model

Data quality can be regarded as an important factor affecting the reputation value. Workers who provide high-quality observed data should obtain higher reputation values than those who provide low-quality observed data. To select reliable workers for obtaining high sensing services quality, inspired by [45], TrustWorker adopts the truth discovery mechanism to the reputation calculation. In TrustWorker, each reputation value can be calculated based on the measurement of the distance between observed data and estimated true data, where the reputation value will be used as the criterion in worker selection. For each subtask t_z, CS_k can obtain the estimated true data and the observed data of workers. Specifically, the worker w_i encrypts observed data $\left\{x_{t_1}^{w_i}, x_{t_2}^{w_i}, \ldots, x_{t_Z}^{w_i}\right\}$ by the public key PK_{CS_k} of CS_k and then submits the ciphertexts, the hash value of ciphertexts and the corresponding signature to CS_k after completing tasks in the task set T. CS_k can verify the identity of w_i, the integrity of observed data, and then decrypt the ciphertexts with the private key SK_{CS_k}. For the subtask t_z, TrustWorker uniformly sets a random value as the initial value of the estimated true data $x_{t_z}^*$. CS_k updates the reputation value based on the distance between $\left\{x_{t_1}^{w_i}, x_{t_2}^{w_i}, \ldots, x_{t_Z}^{w_i}\right\}$ and $\left\{x_{t_1}^*, x_{t_2}^*, \ldots, x_{t_Z}^*\right\}$, as shown in formula (1).

$$rep_{w_i} = \log\left(\frac{\sum_{i'=1}^{Q}\sum_{z=1}^{Z} d(x_{t_z}^{w_{i'}}, x_{t_z}^*)}{\sum_{z=1}^{Z} d(x_{t_z}^{w_i}, x_{t_z}^*)}\right). \tag{1}$$

Assuming that observed data is continuous data, $d(x_{t_z}^{w_i}, x_{t_z}^*)$ can be calculated according to formula (2), which represents the distance between $x_{t_z}^{w_i}$ and $x_{t_z}^*$. The smaller the distance $d(x_{t_z}^{w_i}, x_{t_z}^*)$, the better the data quality and the greater the reputation rep_{w_i}. High-reputation workers will be selected first in the worker selection, while low-reputation workers will be eliminated. Furthermore, the selected workers will be rewarded for providing high-quality observed data.

$$d(x_{t_z}^{w_i}, x_{t_z}^*) = \frac{\left(x_{t_z}^{w_i} - x_{t_z}^*\right)^2}{std_{t_z}}. \tag{2}$$

In formula (2), std_{t_z} denotes the standard deviation of all observed data $x_{t_z}^{w_i}$ on t_z, which can be calculated according to formula (3). Here, $\bar{x}_{t_z}$ represents the average of all observed data $x_{t_z}^{w_i}$ on t_z and is calculated by $\bar{x}_{t_z} = \frac{\sum_{i=1}^{Q} x_{t_z}^{w_i}}{Q}$.

$$std_{t_z} = \sqrt{\frac{\sum_{i=1}^{Q}\left(x_{t_z}^{w_i} - \bar{x}_{t_z}\right)^2}{Q}}. \tag{3}$$

These updated reputation values can be used in formula (4) to recalculate the estimated true data $x_{t_z}^*$. The calculation process between formulas (1) and (4) will be executed iteratively until the estimated true data satisfies the convergence criterion.

$$x_{t_z}^* = \frac{\sum_{i=1}^{Q} rep_{w_i} \cdot x_{t_z}^{w_i}}{\sum_{i=1}^{Q} rep_{w_i}}. \tag{4}$$

4.3 Worker Selection With Reputation Privacy Protection

Based on the worker selection requirements, CS_k submits the top N reputation values in the local group to the

blockchain. However, reputation values are private information so that workers are often reluctant to announce them directly on the blockchain. To tackle this issue, CS_k adopts DET to encrypt reputation values and then submits the top N DET reputation ciphertexts in the form of a transaction. Each reputation ciphertexts transaction can be simply denoted as $\{ID_T, ID_{CS_k}, DETcipher_k, timestamp\}$, where ID_T is the ID of task set T, ID_{CS_k} is the ID of CS_k and $DETcipher_k$ is the DET reputation ciphertexts set which contains the local top N reputation ciphertexts provided by CS_k. Since nodes on the blockchain only know DET reputation ciphertexts, they cannot obtain the selection result by comparing DET reputation ciphertexts directly. Hence, TrustWorker adopts the secret minimum heapsort scheme to select the top N workers with higher reputation values. Each worker selection result transaction will be verified by nodes on the blockchain. All valid worker selection results will eventually be recorded in the distributed ledger.

4.3.1 Two-Party Reputation Comparison Protocol

The selection process of the top N workers with higher reputation based on the secret minimum heapsort requires the participation of CS_k. Each CS_k is responsible for assisting nodes on the blockchain to determine the comparison result of reputation values to refresh the DET ciphertexts minimum heap secretly. To protect reputation privacy, TrustWorker uses a secure two-party comparison protocol [46] to perform the reputation comparison between two different CS_k. The secure two-party comparison protocol based on Paillier cryptosystem [47] is shown in Algorithm 1, where both the encryption algorithm $E(\cdot)$ and the decryption algorithm $D(\cdot)$ satisfy additive homomorphic.

Algorithm 1. Two-Party Reputation Comparison Protocol

Input: $CS_{k^*}, CS_k, DETcipher_{k^*,i}, DETcipher_{k,j}$

Output: The comparing result R

Initialization: CS_{k^*} and CS_k decrypt $DETcipher_{k^*,i}$ and $DETcipher_{k,j}$ to obtain reputation plaintexts $p_{k^*,i}$ and $p_{k,j}$, and then set $p_{k^*,i} = \frac{y_1}{x_1}$ and $p_{k,j} = \frac{y_2}{x_2}$ with $gcd(x_1, y_1) = gcd(x_2, y_2) = 1$. CS_k performs KeyGeneration() of Paillier homomorphic encryption and randomly chooses a point (x_3, y_3) with $gcd(x_3, y_3) = 1$;

1: CS_k calculates $a = x_3 - x_2$, $b = y_2 - y_3$, $c = x_2 y_3 - x_3 y_2$ (assuming that $a > 0, b < 0$) and chooses a random number r, where $b_1 = b + r > 0$. CS_k calculates $A = E(a)$, $B = E(b_1)$ and sends A, B, r to CS_{k^*};

2: CS_{k^*} calculates $U = E(u) = A^{y_1} \cdot B^{x_1}$, $L = r \cdot x_1$ and then sends U to CS_k;

3: CS_k decrypts U to obtain $u = a \cdot y_1 + b_1 \cdot x_1$, then calculates $q = u + c$ and sends q to CS_{k^*};

4: CS_{k^*} calculates $H = \frac{1}{2} \cdot (q - L)$ and outputs the comparison result;

$$R = \begin{cases} DETcipher_{k,j} > DETcipher_{k^*,i}, H < 0; \\ DETcipher_{k,j} = DETcipher_{k^*,i}, H = 0; \\ DETcipher_{k,j} < DETcipher_{k^*,i}, H > 0. \end{cases}$$

5: **return** R;

In Algorithm 1, based on the security of homomorphic encryption, CS_{k^*} cannot decrypt A and B without using the private key. Since x_2, y_2, x_3 and y_3 are private input of CS_k, CS_{k^*} cannot infer $p_{k,j}$ from q. In addition, x_1 and y_1 are unknown variables, so that CS_k cannot obtain $p_{k^*,i}$ by decrypting U. Therefore, the secure two-party reputation comparison protocol performed between CS_k and CS_{k^*} enables the comparison without revealing reputation privacy.

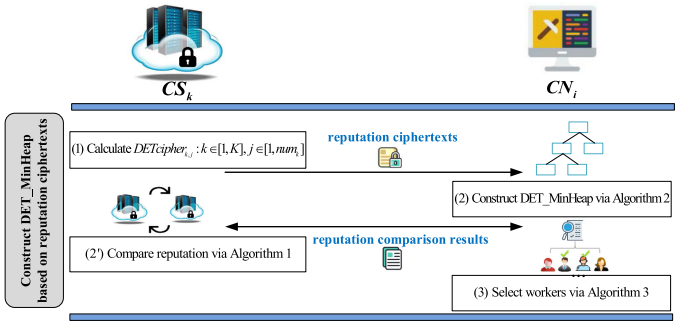


Fig. 3. Workflow of selecting workers based on secret minimum heapsort.

4.3.2 Selecting Workers Based on Secret Minimum Heapsort

To select the top N workers with higher reputation values, TrustWorker constructs a minimum heap based on DET reputation ciphertexts(DET_MinHeap). Based on the worker selection requirements submitted by the requester, each CS_k uploads the local top N DET reputation ciphertexts to the blockchain. With the help of CS_k, CN_i can construct and refresh DET_MinHeap, so as to obtain the worker selection result. Each valid worker selection result after verification will be notified to the requester and recorded in the distributed ledger. The secret minimum heapsort process of selecting the top N workers with higher reputation values is shown in Fig. 3.

(1) Based on the setting number of selected workers, CS_k encrypts the top N reputation values in the local group based on DET and then uploads the ciphertexts $\{DETcipher_{k,1}, DETcipher_{k,2}, \ldots, DETcipher_{k,num_k}\}$ to the blockchain, where the ciphertexts have been arranged in the descending order. If the total number of reputation values in the local group is less than N, CS_k will upload all DET reputation ciphertexts. Besides, to guarantee the fairness of worker selection, some workers with the same higher reputation values will also be selected as one of the local top N workers.

(2) After receiving these DET reputation ciphertexts provided by different CS_k, CN_i invokes Algorithm 2 to construct DET_MinHeap. CN_i is responsible for constructing and refreshing DET_MinHeap based on DET reputation ciphertexts, where DET_MinHeap always maintains N DET reputation ciphertexts according to the setting number of selected workers. Each CS_k executes Algorithm 1 to obtain reputation comparison results, thereby assisting CN_i to determine the corresponding position of $DETcipher_{k,j}$ on DET_MinHeap.

(3) CN_i performs Algorithm 3 to obtain the final worker selection result. One thing needs to be pointed out, in order to ensure the fairness of worker selection, some workers with the same reputation values as the top N workers selected in the process of secret minimum heapsort will also

be chosen. Therefore, the total number of final selected workers may equal to or greater than N.

Algorithm 2. Refreshing the DET_MinHeap

Input: $Heap$, CN_i, CS_k, $DETcipher_k$
Output: New DET_MinHeap $Heap^*$
Initialization: CN_i randomly chooses one $DETcipher_k$ to create the initial $Heap$. For the same $DETcipher$ exists in $DETcipher_k$, perform $sameDET.add(DETcipher)$ and delete it from $DETcipher_k$;

```
for j = 0 to DETcipher_k.length-1 do
  if heap_top is provided by CS_k then
    terminate refresh DET_MinHeap;
  else
    CS_k invokes Algorithm1 with CS_k* to compare DETcipher_{k,j} with heap_top;
  if DETcipher_{k,j} = heap_top then
    sameDET.add(DETcipher);
    terminate refresh DET_MinHeap;
  else if DETcipher_{k,j} < heap_top then
    terminate refresh DET_MinHeap;
  else
    CN_i replaces heap_top with DETcipher_{k,j};
    while leftnode exists do
      if heap_top is not the minimum ciphertext then
        if rightnode = leftnode then
          sameDET.add(DETcipher);
          Swap(heap_top, leftnode);
        else if rightnode < leftnode then
          Swap(heap_top, rightnode);
        else
          Swap(heap_top, leftnode);
      else
        terminate adjust DET_MinHeap;
  returnDET_MinHeap Heap*;
```

Algorithm 3. Outputing the Worker Selection Result

Input: DET_MinHeap $Heap^*$, CN_i, $sameDET$
Output: The worker selection result

```
for i = 0 to N − 1 do
  selection_result.add(Heap[i]);
  if Heap[i] equals to DETcipher in sameDET then
    selection_result.add(DETcipher);
  returnthe selection_result;
```

To display the process of selecting the top N workers with higher reputation values based on the secret minimum heapsort in more detail, Fig. 4 gives one simple case. Assume that N is set as 5 and the number of worker reputation values calculated by CS_1 and CS_2 respectively is equal to 5. CS_1 and CS_2 adopt DET to encrypt the local top N reputation values respectively, and then obtain corresponding DET reputation ciphertexts. The secret minimum heapsort is completed by CS_1, CS_2 and CN_1.

First, suppose that CN_1 constructs an initial DET_MinHeap based on the local top N DET reputation ciphertexts provided by CS_1. Then, CN_1 performs Algorithm 2 to refresh DET_MinHeap based on the DET reputation ciphertexts provided by CS_2, so as to obtain the final worker selection result. In Algorithm 2, only the DET reputation

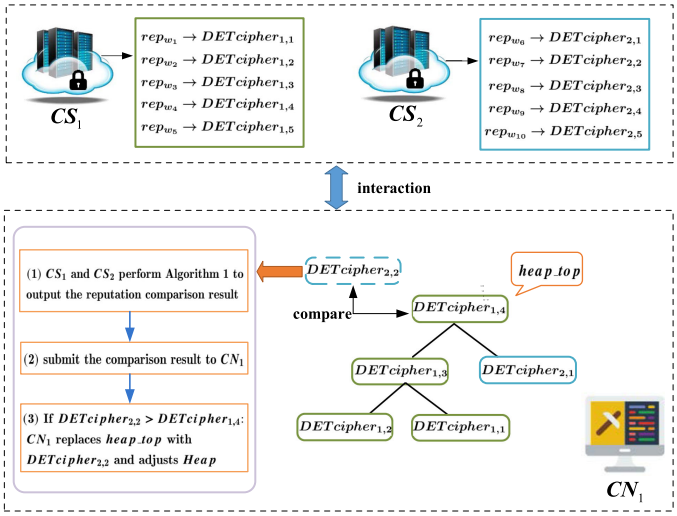


Fig. 4. A simple case of selecting the top N workers with higher reputation values.

ciphertext to be compared is greater than the ciphertext of $heap\_top$, CN_1 will refresh DET_MinHeap. For example, $DETcipher_{2,2}$ is provided by CS_2 and the ciphertext $DETcipher_{1,4}$ on $heap\_top$ is provided by CS_1. When $DETcipher_{2,2}$ is greater than $DETcipher_{1,4}$, CN_1 updates the $heap\_top$ and then adjusts DET_MinHeap.

In the process of adjusting DET_MinHeap, if the two reputation ciphertexts to be compared are provided by different CS_k, then Algorithm 1 will be executed between the two different CS_k. For example, since the $DETcipher_{2,2}$ replaces the $DETcipher_{1,4}$ to be the new $heap\_top$, CN_1 needs to exchange the position of $DETcipher_{2,2}$ with the smaller ciphertext $DETcipher_{1,3}$ or $DETcipher_{2,1}$ if $DETcipher_{2,2}$ is not the minimum ciphertext on DET_MinHeap. As we can see in Fig. 4, $DETcipher_{1,3}$ and $DETcipher_{2,1}$ are provided by CS_1 and CS_2 respectively, hence CS_1 and CS_2 will perform Algorithm 1 to provide the corresponding reputation comparison result for determining the smaller reputation ciphertext. Otherwise, CN_1 can directly obtain the reputation comparison result based on the local top N DET reputation ciphertexts submitted by CS_1 or CS_2. Each adjustment operation for DET_MinHeap is the same execution process.

At last, CN_1 invokes Algorithm 3 to output the worker selection result based on the updated DET_MinHeap.

4.4 Consensus Process in TrustWorker

Based on the secret minimum heapsort, CN_i can calculate the final worker selection result. Each worker selection result will be broadcasted on the blockchain in the form of a transaction. The worker selection result transaction can be represented as $\{ID_T, result_T, timestamp\}$, where ID_T is the ID of task set T and $result_T$ is the corresponding worker selection result. PN selected in advance performs the secret minimum heapsort scheme to select the top N workers with higher reputation values and then broadcasts the worker selection result transaction to each verification node CN_i. After receiving the transaction, CN_i executes the worker selection based on the secret minimum heapsort to verify the validity of each selection result.

Specifically, since some workers with the same reputation values as the top N workers selected in the secret minimum heapsort may also be chosen to gurantee the fairness of worker selection, the total number of final selected workers may be greater than N. Hence, CN_i will verify the number of selected workers in the worker selection result transaction first. If the number of workers is fallacious, the transaction is considered invalid. Then, CN_i will compare the consistency of the selection result. If the corresponding selected workers' numbers are the same, the worker selection result is considered valid, otherwise invalid. PN will receive an approval reply from CN_i when the transaction is valid. Each valid worker selection result will eventually be recorded in the distributed ledger to resist tampering. Afterwards, the requester can obtain the worker selection result for the specific task set ID from the blockchain.

5 Theoretical Analysis

TrustWorker not only considers the reputation privacy leakage issue of worker selection in BBC, but also ensures the trustability of selection result. We will analyze TrustWorker from the five aspects of privacy, trustability, security, fairness and time complexity.

5.1 Privacy

In TrustWorker, reputation privacy can be protected. Different from the existing reputation-based worker selection schemes in BBC, that is reputation values of workers are often published in the form of plaintexts on the blockchain, our TrustWorker adopts a deterministic encryption algorithm to encrypt the top N reputation values and then stores these ciphertexts on the blockchain to protect reputation privacy. Even if the reputation ciphertexts are obtained based on the deterministic encryption, the nodes on the blockchain only know the size relationship between the reputation ciphertexts when performing worker selection. For two identical reputation ciphertexts, the nodes can only infer that the two reputation values are equal, but cannot know the specific reputation plaintext. Hence, even if attackers can access reputation ciphertexts from the blockchain, they cannot obtain original reputation values from them.

5.2 Trustability

In traditional trust-based model, worker selection results are usually verified separately by TCS, which may lead to the untrustworthy worker selection. To achieve trustworthy worker selection, TrustWorker integrates the blockchain technology into crowdsensing, where worker selection results are expanded from single-party verification to multi-party verification. In TrustWorker, each worker selection result will be verified by multiple nodes on the blockchain instead of a single party, where each node performs the secret minimum heapsort to complete transaction verification for each selection result. Hence, our TrustWorker can ensure the validity and trustability of worker selection results, thereby improving the sensing services quality.

5.3 Security

To prevent some dishonest behaviors in BBC, TrustWorker requires the requester and workers to deposit an amount of cryptocurrency in advance to the blockchain. Hence, dishonest participants need to pay a huge price to conduct these dishonest behaviors under mortgage deposits.

5.4 Fairness

In TrustWorker, reputation values of workers are calculated based on the task data quality. Workers with good performance will get higher reputation, while dishonest workers will get extremely lower reputation due to submitting incorrect data. Therefore, reputation values can be used as an fair and valid indicator for worker selection. Furthermore, to ensure the fairness of worker selection, TrustWorker also recognizes these workers, who have the same higher reputation values with the selected top N workers. By this way, each fair worker selection can be obtained, which not only enhances the participation enthusiasm of workers, but also helps improve the sensing services quality.

5.5 Time Complexity

To ensure the privacy-preserving worker selection, TrustWorker adopts the secret minimum heapsort to obtain worker selection results. In TrustWorker, selecting the top N reputation ciphertexts relys on a minimum heap which has the logarithmic height. Since each CS_k provides the local top N DET reputation ciphertexts and then CN_i can construct the initial DET_MinHeap based on them without performing the reputation comparison, the total number of reputation ciphertexts required to be compare in the secret minimum heapsort is $M = (K-1)N$. In Algorithm 2, refreshing DET_MinHeap once which contains N reputation ciphertexts requires $\log N$ complex computations. Each complex computation contains no more than two secure reputation comparisons, where executing a two-party reputation comparison protocol requires eight modular exponentiations. Therefore, for M DET reputation ciphertexts, refreshing DET_MinHeap requires $O(M\log N)$ complex computation.

In TrustWorker, we consider the main information interaction that occurs during the process of selecting workers with higher reputation values based on the secret minimum heapsort. Assume that it takes S bits to transmit a piece of data. For the DET_MinHeap with N reputation ciphertexts, refreshing DET_MinHeap once requires no more than $2\log N$ secure reputation comparisons, and each comparison process needs to transmit eight pieces of data. Therefore, in the secret minimum heapsort of selecting workers, the communication cost of refreshing DET_MinHeap is no more than $16M\log NS$ bits.

6 Performance Evaluation

In this section, we introduce the experiment setup and then perform simulation experiments, thereby analyzing experiment results to evaluate the effectiveness of TrustWorker.

6.1 Experiment Setup

Simulation experiments are performed on an Intel (R) Core (TM) i5-10210U CPU@1.60 GHz 2.11 GHz processor and 16 GB memory. The operating system is Ubuntu 20.04 and the programming language is Java 1.8. The experimental dataset is the simulated data of the distance measurement task,

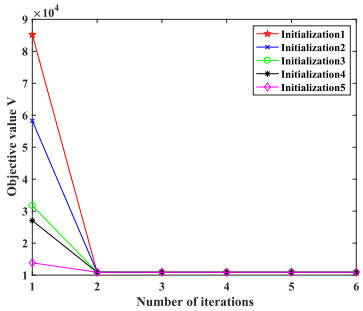
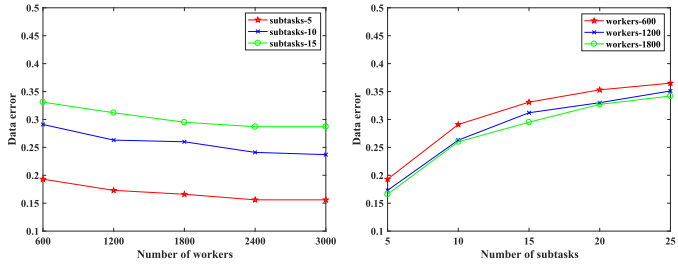


Fig. 5. Convergence evaluation.



(a) (b)

Fig. 6. Data error under different variables: (a) the number of workers, (b) the number of subtasks.

which can be used for indoor floorplan construction *et al.* The content of tasks is to measure the distance of 25 different location points in a building and the true measurement data as the quality evaluation standard for each subtask has been defined. The simulated data is generated by a pseudo-random generator based on the true distance measurement data and the set data fluctuation deviation range (e.g., [-1,1]) to conform to the real collection results from workers as much as possible. In addition, it is assumed that each CS_k is responsible for calculating the same number of reputation values. In the two-party reputation comparison protocol, we set the security parameter of Paillier cryptosystem to 512 bits. (Actually, for higher security demands, this parameter can be set to 1,024 bits or longer.) To test the performance of Trustwoker on the blockchain, we deploy corresponding chaincodes on Hyperledger Fabric v1.4.0 and then adopt Kafka consensus. We execute the performance test by using Hyperledger Caliper, where the test blockchain network runs on a single host. The test blockchain network includes two organizations, where each organization contains two peers.

6.2 Experiment Results Analysis

Based on the experiment results, we analyze the performance of TrustWorker from five aspects: convergence, sensing services quality, reputation changes, computation time cost and average transaction latency and throughput.

6.2.1 Convergence

Reputation values are finally determined by several iterative calculations. Considering the convergence of reputation calculation, we use $V = \sum_{i=1}^{Q} rep_{w_i} \cdot \sum_{z=1}^{Z} d(x_{t_z}^{w_i}, x_{t_z}^{*})$ defined in [34] as an objective value, where rep_{w_i} represents the reputation of worker w_i, $d(x_{t_z}^{w_i}, x_{t_z}^{*})$ represents the distance between the observed data provided by the worker w_i and the estimated true data on the subtask t_z, Q and Z are the number of workers and subtasks respectively. When the difference between two consecutive objective values varies in a small range, the iterative process is considered to have converged. For example, if $V(i+1) - V(i) < 0.1$, where $V(i)(i \geq 1)$ represents the objective value in ith iteration, it can be considered that the reputation calculation has reached convergence after $i+1$ iterations. In this experiment, we randomly select five different initial values for each estimated true data $x_{t_z}^{*}$. In addition, we fix the number of workers and subtasks as 600 and 5, respectively. Observing Fig. 5, we can find that the reputation calculation will converge quickly with a few iterations no matter what the initial values for each estimated true data are.

6.2.2 Sensing Services Quality

To evaluate the impact of worker selection on sensing services quality, we calculate the data error of tasks to assess the sensing services quality, and use the true task data as the standard to measure the deviation of observed data provided by the selected workers. Specifically, we assume that the true data on subtask t_z is $x_{t_z}^{'}$, and adopt the error function MAE to calculate the data error, where the error function can be represented as $\mathrm{MAE} = \frac{1}{N \cdot Z} \sum_{z=0}^{Z} \sum_{i=0}^{N} \left| x_{t_z}^{w_i} - x_{t_z}^{'} \right|$. MAE denotes the mean value of the absolute error between each observed data submitted by selected workers and the true data on each subtask. The lower the data error, the smaller the deviation between the observed data of each selected worker and the true data of each subtask, that is, the better the sensing services quality. To evaluate the sensing services quality, we use the number of workers and the number of subtasks as variables, and repeat this experiment 10 times to calculate corresponding averages. In this experiment, the number of CS is set as 2 and the value of N is set as 10. We evaluate the sensing services quality when the number of subtasks is fixed as 5, 10 and 15 in Fig. 6a, where the number of workers is 600, 1,200, 1,800, 2,400 and 3,000 respectively. Furthermore, to analyze the impact of the number of subtasks on sensing services quality, we also measure the average sensing services quality under 5, 10, 15, 20 and 25 subtasks, when the number of workers is fixed as 600, 1,200 and 1,800 in Fig. 6b.

Observing Fig. 6a, when the number of subtasks is fixed as 5, 10 and 15 respectively, we find that as the number of workers increases, the data error shows an downward trend, that is, the sensing services quality increases. This makes sense because when the number of workers increases, each subtask may be performed by more and more diversified workers, that we can evaluate the reputation values of workers more accurately to select more reliable workers, thereby decreasing the data error to improve the sensing services quality. In Fig. 6b, when the number of workers is fixed as 600, 1,200 and 1,800 respectively, as the number of subtasks increases, the sensing services quality decreases. Since the data error of some subtasks increases affected by the worker selection result, which leads to the increase of overall data error, thereby the sensing services quality provided by the selected workers has declined.

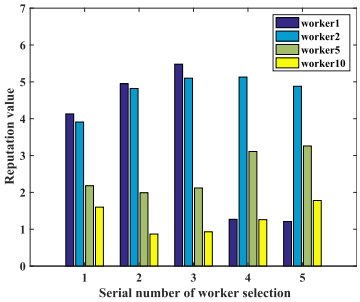


Fig. 7. Reputation changes of workers under different behaviors: worker w_1: from providing high-quality data first to providing low-quality data, worker w_2: continuing to provide high-quality data, worker w_5: from providing low-quality data first to providing high-quality data, worker w_{10}: continuing to provide low-quality data.

6.2.3 Reputation Changes

Reputation values are calculated based on the data quality provided by workers in each task. Therefore, the quality of task data determines the reputation value of the worker. In each worker selection process, workers who provide high-quality data will obtain higher reputation values and be preferentially selected, while workers who provide low-quality data will obtain lower reputation values and be eliminated. In order to analyze the effect of worker performance on reputation changes, we consider four types of possible behavior changes during the five times worker selections. Specifically, we divide the behavior changes of workers into four types: from providing high-quality data to providing low-quality data, continuing to provide high-quality data, from providing low-quality data to providing high-quality data, and continuing to provide low-quality data. In this experiment, we fix the number of workers as 10, consider the third times worker selection as the turning point and then observe reputation changes of worker w_1, w_2, w_5 and w_{10} respectively. The results of reputation changes are shown in Fig. 7.

In Fig. 7, we find that during the previous three times worker selection in TrustWorker, due to the provision of high-quality task data, reputation values of w_1 and w_2 are much higher than those of w_5 and w_{10} who provide low-quality task data. After the third worker selection, w_1 abruptly reduces the task data quality, resulting in a sudden decrease in the reputation value. On the contrary, w_5 who turns to improving the task data quality, gradually promotes his reputation value. In the five times worker selection processes, w_2 who continuously provides high-quality task data always obtains a higher reputation value. While, w_{10} always obtains a lower reputation value due to the continuous provision of low-quality task data. Therefore, the task data quality directly affects the reputation value of the worker. For the performance behavior that provides high-quality data, TrustWorker guarantees the priority of high-reputation workers to be selected. For the performance behavior of providing low-quality task data, TrustWorker penalizes the worker by giving a lower reputation value, thereby making the worker being eliminated in worker selection. As can be seen from Fig. 7, TrustWorker can dynamically update reputation values based on behavior changes and then rely on them to distinguish the priority of workers, so as to obtain worker selection results.

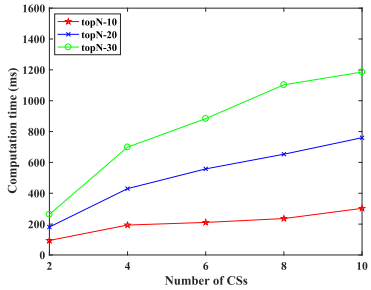


Fig. 8. The computation time of refreshing DET_MinHeap when the number of workers is 600.

6.2.4 Computation Time Cost

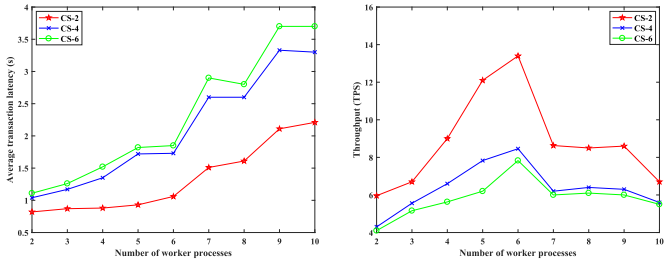
In TrustWorker, the determination of worker selection results depends on the output of DET_MinHeap with N DET reputation ciphertexts. In this experiment, we observe the influence of different numbers of CS and different values of N on the computation time cost of refreshing DET_MinHeap when the number of workers is fixed as 600. We repeat this experiment 10 times and final experimental results are shown in Fig. 8.

Observing Fig. 8, we can find that as the number of CS increases, the computation time cost maintains an upward trend. In the process of refreshing DET_MinHeap, the two-party comparison protocol based on Paillier cryptosystem needs to be implemented between different CS to obtain reputation comparison results. Under the same value of N, as the number of CS rises, the total number of DET reputation ciphertexts provided by different CS increases, resulting in an increase in the total number of two-party comparison protocol executions when refreshing DET_MinHeap. Hence, the computation time cost of refreshing DET_MinHeap increases. Under the same number of CS, we find that as the value of N increases, except the total number of DET reputation ciphertexts increases, the total number of DET reputation ciphertexts maintained by DET_MinHeap also increases, which leads to the increased time on adjusting DET_MinHeap. Therefore, the computation time cost of refreshing DET_MinHeap increases.

6.2.5 Average Transaction Latency and Throughput

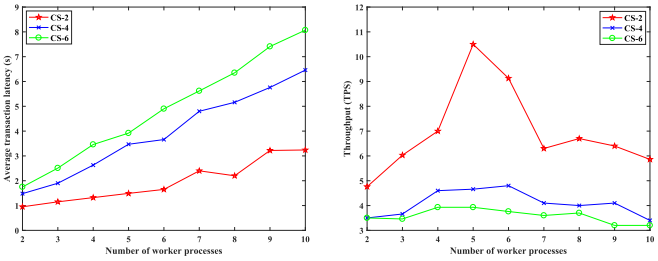
In Trustwoker, worker selection results will be submitted to the blockchain in the form of transactions. To evaluate average transaction latency of worker selection result transactions from submission to response, and throughput within a limited time, we observe and analyze the changes in average transaction latency and throughput under different number of worker processes. We perform experimental tests under the value of N is set to 10 and 20 respectively. The experimental results are shown in Figs. 9 and 10 respectively, where the number of CS is fixed as 2, 4 and 6 respectively. In the specific test, we set the number of worker processes to vary from 2 to 10, and then fix the test duration time as 30 seconds. The number of workers is fixed as 600. In order to reduce the error, we repeatedly run each experiment 10 times and calculate average values.

Observing Figs. 9a and 10a, we find that as the number of worker processes increases, average transaction latency keeps an upward trend. The increase in the number of



(a) (b)

Fig. 9. Test results when $N = 10$: (a) average transaction latency, (b) throughput.



(a) (b)

Fig. 10. Test results when $N = 20$: (a) average transaction latency, (b) throughput.

worker processes means that the total number of transaction submissions increases. Due to the limitation of the system's computing resources, transactions are required to be in a waiting state for a longer time, hence average transaction latency increases overall. In Figs. 9b and 10b, we can find that the system can reach the maximum throughput in a limited time. As the number of worker processes increases, throughput generally maintains a trend of increasing first and then decreasing. When the number of worker processes increases, the transaction sending rate increases, and the total system workload rate reaches the maximum, the system performance can be maximized. For example, the system can obtain the maximum throughput. However, when the total system workload rate reaches saturation, although the number of worker processes increases, limited system resources lead to a decrease in the transaction sending rate, and thus throughput begins to decrease. When the number of CS is 2, 4 and 6, as the number of CS increases, we can find that average transaction latency increases, while throughput decreases. Some corresponding reasonable explanations can be obtained from Fig. 8. Observing Fig. 8, we find that the computation time cost of refreshing DET_MinHeap rises as the increasing number of CS or the increasing value of N. The verification of each worker selection result transaction relys on the output result of DET_MinHeap. The computation time cost of refreshing DET_MinHeap directly affects average transaction latency and throughput. The more the computation time cost of refreshing DET_MinHeap, the higher the average transaction latency and the lower the throughput. Therefore, as the number of CS increases, average transaction latency tends to rise, while throughput decreases. In addition, comparing experimental results of Figs. 9 and 10, we can find that the computation time cost of refreshing DET_MinHeap rises due to the increasing value of N, which leads to an increase in average transaction latency and a decrease in throughput.

7 Conclusion

Considering the reputation privacy and the selection trustability which be ignored in existing worker selection schemes, in this paper, we propose a trustworthy and privacy-preserving worker selection scheme for BBC. In terms of reputation calculation, TrustWorker dynamically updates reputation values of workers by evaluating data quality and then adopts the reputation values to evaluate the reliability of workers. To protect reputation privacy of workers, TrustWorker encrypts reputation values based on a deterministic encryption algorithm, and then stores corresponding reputation ciphertexts to the blockchain based on the worker selection requirements. In addition, to ensure the trustability and fairness of worker selection, TrustWorker adopts the secret minimum heapsort scheme to select the top N workers with higher reputation values. Simulation results show that TrustWorker can achieve high sensing services quality while protecting worker reputation privacy. For the limitation of TrustWorker, it mainly considers the influence of workers' reputation on worker selection under the reputation protection. However, the reputation of requesters may also affect the task selection of workers. This issue is also worth studying to achieve better worker selection. Our future work includes, further keeping the balance between privacy and efficiency in worker selection.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62072487, 61602537, U1509214, and 61972310, in part by Beijing Natural Science Foundation under Grant M21036, in part by the National Statistical Science Foundation of China under Grant 2020LD01, and in part by the Key R&D Program of Shaanxi Province under Grant 2019ZDLGY13-06.

References

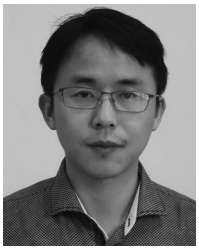
[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.
[2] B. Guo *et al.*, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 1–31, 2015.
[3] X. Wang, Z. Ning, X. Hu, H. Ma, and C. Zhang, "A city-wide real-time traffic management system: Enabling crowdsensing in social internet of vehicles," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 19–25, Sep. 2018.
[4] L. Liu, W. Liu, Y. Zheng *et al.*, "Third-Eye: A mobilephone-enabled crowdsensing system for air quality monitoring," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–26, 2018.
[5] S. H. Marakkalage *et al.*, "Understanding the lifestyle of older population: Mobile crowdsensing approach," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 1, pp. 82–95, Feb. 2018.
[6] B. Zhao, S. Tang, X. Liu, and X. Zhang, "PACE: Privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 1924–1939, May 2021.
[7] S. Gao, J. Ma, W. Shi, G. Zhan, and C. Sun, "TrPF: A trajectory privacy-preserving framework for participatory sensing," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 6, pp. 874–887, Jun. 2013.
[8] Upwork, 2020. [Online]. Available: https://www.upwork.com/
[9] Amazon mechanical turk, 2020. [Online]. Available: https://www.mturk.com/

[10] Y. Lu, Q. Tang, and G. Wang, "ZebraLancer: Private and anonymous crowdsourcing system atop open blockchain," in *Proc. 38th IEEE Int. Conf. Distrib. Comput. Syst.*, 2018, pp. 853–865.
[11] M. Li *et al.*, "CrowdBC: A blockchain-based decentralized framework for crowdsourcing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 6, pp. 1251–1266, Jun. 2019.
[12] C. Lin, D. He, S. ZeadallyN. Kumar, and K.-K. R. Choo, "SecBCS: A secure and privacy-preserving blockchain-based crowdsourcing system," *Sci. China Inf. Sci.*, vol. 63, no. 3, pp. 20–33, 2020.
[13] M. Yang *et al.*, "A blockchain-based location privacy-preserving crowdsensing system," *Future Gener. Comput. Syst.*, vol. 94, pp. 408–418, 2019.
[14] B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han, "ActiveCrowd: A framework for optimized multitask allocation in mobile crowdsensing systems," *IEEE Trans. Hum. Mach. Syst.*, vol. 47, no. 3, pp. 392–403, Jun. 2017.
[15] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 16–28, Jan. 2018.
[16] Y. Yang, W. Liu, E. Wang, and J. Wu, "A prediction-based user selection framework for heterogeneous mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2460–2473, Nov. 2019.
[17] J. An, D. Liang, X. Gui, H. Yang, R. Gui, and X. He, "Crowdsensing quality control and grading evaluation based on a two-consensus blockchain," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4711–4718, Jun. 2019.
[18] K. H. Kawajiri, R. Shimosaka M, "Steered crowdsensing: Incentive design towards quality-oriented place-centric crowdsensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 691–701.
[19] Y. Wen *et al.*, "Quality-driven auction-based incentive mechanism for mobile crowd sensing," *IEEE Trans. Veh. Technol.*, vol. 64, no. 9, pp. 4203–4214, Sep. 2015.
[20] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," in *Proc. 16th ACM Int. Symp. Mobile Ad Hoc Network. Comput.*, 2015, pp. 177–186.
[21] Y. Bin, W. Yan, and L. Ling, "Crowdtrust: A context-aware trust model for worker selection in crowdsourcing environments," in *Proc. IEEE Int. Confe. Web Serv.*, 2015, pp. 121–128.
[22] N. B. Truong, G. M. Lee, T. Um, and M. Mackay, "Trust evaluation mechanism for user recruitment in mobile crowd-sensing in the Internet of Things," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 10, pp. 2705–2719, Oct. 2019.
[23] F. Hendrikx, K. Bubendorfer, and R. Chard, "Reputation systems: A survey and taxonomy," *J. Parallel Distrib. Comput.*, vol. 75, pp. 184–197, 2015.
[24] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *Proc. IEEE Conf. Comput. Commun.*, 2012, pp. 2140–2148.
[25] F. Wei and Y. Zheng, "Mcs-chain: Decentralized and trustworthy mobile crowdsourcing based on blockchain," *Future Gener. Comput. Syst.*, vol. 95, pp. 649–666, 2019.
[26] M. Kadadha, H. Otrok, R. MizouniS. Singh, and A. Quali, "SenseChain: A blockchain-based crowdsensing framework for multiple requesters and multiple workers," *Future Gener. Comput. Syst.*, vol. 105, pp. 650–664, 2020.
[27] 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.
[28] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wirel. Commun.*, vol. 27, no. 2, pp. 72–80, Apr. 2020.
[29] B. Luo, X. Li, J. Weng, J. Guo, and J. Ma, "Blockchain enabled trust-based location privacy protection scheme in VANET," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2034–2048, Feb. 2020.
[30] X. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "Enabling reputation and trust in privacy-preserving mobile sensing," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2777–2790, Dec. 2014.
[31] L. Ma, X. Liu, Q. Pei, and Y. Xiang, "Privacy-preserving reputation management for edge computing enhanced mobile crowdsensing," *IEEE Trans. Serv. Comput.*, vol. 12, no. 5, pp. 786–799, Sep./Oct. 2019.
[32] O. Hasan, L. Brunie, E. Bertino, and N. Shang, "A decentralized privacy preserving reputation protocol for the malicious adversarial model," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 6, pp. 949–962, Jun. 2013.
[33] M. R. Clark, K. Stewart, and K. M. Hopkinson, "Dynamic, privacy-preserving decentralized reputation systems," *IEEE Trans. Mobile Comput.*, vol. 16, no. 9, pp. 2506–2517, Sep. 2017.
[34] J. Tang, S. Fu, M. Xu, Y. Lou, and K. Huang, "Achieve privacy-preserving truth discovery in crowdsensing systems," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 1301–1310.
[35] G. Xu, H. Li, S. Liu, M. Wen, and R. Lu, "Efficient and privacy-preserving truth discovery in mobile crowd sensing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3854–3865, Apr. 2019.
[36] C. Miao *et al.*, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, 2015, pp. 183–196.
[37] Y. Wang, X. Jia, Q. Jin, and J. Ma, "QuaCentive: A quality-aware incentive mechanism in mobile crowdsourced sensing (MCS)," *J. Supercomput.*, vol. 72, no. 8, pp. 2924–2941, 2016.
[38] M. Pouryazdan, B. Kantarci, T. Soyata, L. Foschini, and H. Song, "Quantifying user reputation scores, data trustworthiness, and user incentives in mobile crowd-sensing," *IEEE Access*, vol. 5, pp. 1382–1397, 2017.
[39] K. Wang, X. Qi, L. Shu, D. Deng, and J. J. P. C. Rodrigues, "Toward trustworthy crowdsourcing in the social Internet of Things," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 30–36, Oct. 2016.
[40] D. Chatzopoulos, M. Ahmadi, S. Kosta, and P. Hui, "FlopCoin: A cryptocurrency for computation offloading," *IEEE Trans. Mobile Comput.*, vol. 17, no. 5, pp. 1062–1075, 2018.
[41] H. Duan, Y. Zheng, Y. Du, A. Zhou, C. Wang, and M. H. Au, "Aggregating crowd wisdom via blockchain: A private, correct, and robust realization," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2019, pp. 1–10.
[42] G. K. Bhatia, S. Gupta, A. Dubey, and P. Kumaraguru, "WorkerRep: Immutable reputation system for crowdsourcing platform based on blockchain," 2020, *arXiv:2006.14782*.
[43] Y. Ding, Z. Chen, F. Lin, and C. Tang, "Blockchain-based credit and arbitration mechanisms in crowdsourcing," in *Proc. 3rd Int. Symp. Auton. Syst.*, 2019, pp. 490–495.
[44] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui, "Privacy preserving and cost optimal mobile crowdsensing using smart contracts on blockchain," in *Proc. 15th Int. Conf. Mobile Ad-hocSensor Syst.*, 2018, pp. 442–450.
[45] Q. Li, Y. Li, J. Gao, B. Zhao, and W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *Proc. ACM SIGMOD Int. Conf.*, 2014, pp. 1187–1198.
[46] S. Li, Y. Guo, S. Zhou, D. Wang, and J. Dou, "Efficient protocols for the general millionaires' problem," *Chinese J. Electron.*, vol. 26, no. 4, pp. 696–702, 2017.
[47] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Adv. Cryptol.-EUROCRYPT'99.*, 1999, pp. 223–238.

Sheng Gao received the BS degree in information and computation science from the Xi'an University of Posts and Telecommunications, Xi'an, China, in 2009 and the PhD degree in computer science and technology from Xidian University, Xi'an, China, in 2014. He is currently an associate professor with the School of Information, Central University of Finance and Economics, Beijing, China. He has authored or coauthored more than five books and 40 papers in refereed international journals and conferences. His current research interests include blockchain, federated learning, data security, and privacy computing.

Xiuhua Chen received the BEng degree in software engineering from the College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China, in 2019 and the MS degree from the School of Information, Central University of Finance and Economics, Beijing, China, in 2021. Her research interests include blockchain, crowdsensing, and privacy computing.

Jianming Zhu received the PhD degree in computer application technology from Xidian University, Xi'an, China, in 2004. From 2008 to 2009, he was a research fellow with the University of Texas at Dallas, Richardson, TX, USA. He is currently a professor with the School of Information, Central University of Finance and Economics, Beijing, China. He has authored or coauthored more than ten books 100 research papers in refereed international journals and conferences. His current research interests include blockchain, financial technology, and data security.



Xuewen Dong received the BEng, MS, and the PhD degrees in computer science and technology from Xidian University, Xi'an, China, in 2003, 2006, and 2011, respectively. From 2016 to 2017, he was a visiting scholar with Oklahoma State University, Stillwater, OK, USA. He is currently a professor with the School of Computer Science and Technology, Xidian University. His research interests include wireless network security, cognitive radio network, and blockchain.

Jianfeng Ma received the BS degree in mathematics from Shaanxi Normal University, Xi'an, China, in 1985 and the MS and PhD degrees in computer software and communications engineering from Xidian University, Xi'an, China, in 1988 and 1995, respectively. He is currently a professor with the School of Cyber Engineering, Xidian University, Xi'an, China. From 1999 to 2001, he was a research fellow with the Nanyang Technological University of Singapore. He has authored or coauthored more than 400 papers in refereed international journals and conferences. His current research interests include cryptography, wireless network security, and data security.