5-2019

A blockchain-based location privacy-preserving crowdsensing system

Mengmeng YANG
Deakin University

Tianqing ZHU
China University of Geosciences Wuhan

Kaitai LIANG
University of Surrey

Wanlei ZHOU
University of Technology Sydney

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Published in Future Generation Computer Systems,
Volume 94, Pages 408-418, May 2018
https://doi.org/10.1016/j.future.2018.11.046

A blockchain-based location privacy-preserving crowdsensing system

Mengmeng Yang [a], Tianqing Zhu [b,c,*], Kaitai Liang [d], Wanlei Zhou [c], Robert H. Deng [e]

[a] School of Information Technology, Deakin University, Australia
[b] School of Computer Science, China University of Geoscience, Wuhan, PR China
[c] School of Software, University of Technology Sydney, Australia
[d] University of Surrey, UK
[e] School of Information Systems, Singapore Management University, Singapore

HIGHLIGHTS

- A new type of location privacy attack as a result of the payment process in the crowdsensing system.
- A blockchain-based privacy preservation framework for protecting worker locations.
- Prevent re-identifications attack by leveraging a private blockchain network.

ARTICLE INFO

Article history:
Received 31 August 2018
Received in revised form 17 October 2018
Accepted 26 November 2018
Available online 10 December 2018

Keywords:
Crowdsensing
Privacy-preserving
Location privacy
Blockchain

ABSTRACT

With the support of portable electronic devices and crowdsensing, a new class of mobile applications based on the Internet of Things (IoT) application is emerging. Crowdsensing enables workers with mobile devices to travel to specified locations and collect data, then send it back to the requester for rewards. However, the majority of the existing crowdsensing systems are based on centralized servers, which are prone to a high chance of attack, intrusion, and manipulation. Further, during the process of transmitting information to and from the service server, the worker's location is usually exposed. This raises the potential risk of a privacy infringement. In this paper, we first identify three ways locations can be disclosed in traditional crowdsensing systems. Then, we propose a novel solution, dubbed a blockchain privacy-preservation crowdsensing system, to address these privacy problems. The proposed system not only protects the privacy of worker locations but also increases the success rate of completing the assigned task. Specifically, the system entails a rewards-based task assignment process that, essentially, markets the given assignment and uses the anonymized characteristics of blockchain technology to hide the identity information of users. To prevent attacks through re-identification, we have introduced a private blockchain to distribute the worker's transaction records.

1. Introduction

As a new emerging application of the IoT, crowdsensing takes advantage of sensor-equipped mobile devices to collect and share data [1]. Users are registered as candidate workers in the crowdsensing platform. This allows the server to select workers to complete data collection tasks for a reward. To complete a spatial crowdsensing task, workers physically travel to a pre-defined location, collect the required data, and transmit it back to a server. This type of data-collection process has been used in many large-scale real-world applications, such as environmental monitoring [2], traffic detection [3], and point of interest identification [4]. However, spatial crowdsensing may expose the worker's location and their travel history to a would-be attacker, which raises serious privacy concerns to the point where it affects worker uptake of the system. Therefore, ensuring the privacy of the workers' locations is highly desirable.

There are three ways in which a worker's location privacy might be disclosed to an untrusted server. First, workers need to submit their exact location to the server to be allocated tasks more efficiently. Second, when a worker accepts an assigned task, the server knows that worker's future location, i.e., their final destination. Third, after completing the task, the server processes their payment so it knows the task the worker completed. As such, completing a task reveals the worker's previous locations, which might be used to form a precise travel history.

Numerous techniques have been proposed to protect the privacy of a user's location, such as dummy locations [5], k-anonymity

* Corresponding author at: School of Software, University of Technology Sydney, Australia.
E-mail addresses: mengmeng.yang@deakin.edu.au (M. Yang), tianqing.zhu@uts.edu.au (T. Zhu), k.liang@surrey.ac.uk (K. Liang), wanlei@deakin.edu.au (W. Zhou), robertdeng@smu.edu.sg (R.H. Deng).

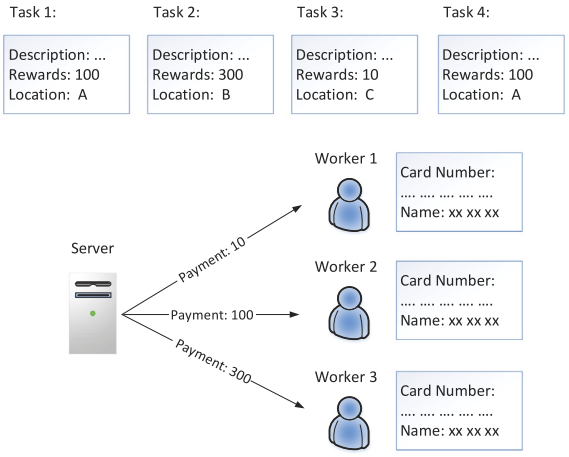

Fig. 1. Payment process.

[6], the obfuscation method [7], and differential privacy [8,9]. Most of these countermeasures only consider the first way of privacy disclosure that the workers upload their location information to the crowdsensor's server. For instance, Kazemi et al. [10] proposed protecting the worker locations by cloaking the region around the exact location. Only a few approaches hide the tasks assigned to workers [11–13]. For example, Bin et al. [12] presented a clustering-based approach in which the server assigns tasks to a cluster head instead of the cluster members. But these types of countermeasures are based on a strong assumption that the cluster head is fully trusted. Unfortunately, even if the location information has the strongest protection during the task assignment process, very few methods prevent exposure of the workers' locations during the payment process.

Fig. 1 shows an example payment process between an untrusted server and anonymized workers where the server has no idea about which tasks the workers completed. This process can be used to successfully prevent worker locations from being disclosed to the server or other network users in the task assignment phase. However, the payment information is still associated with the real identity of the worker. By observing the payment, the server can infer which task a particular worker actually completed. For instance, suppose the server paid 10 to Worker 1 and the only task in the pool that has a reward of 10 is Task 3. It would be very easy for the server to infer that Worker 1 has been to location C.

The above privacy problem can be tackled by involving a trustworthy third party in the payment process. For example, the server pays the total reward for all tasks to a third party, who forwards individual rewards to particulars worker. However, it is a challenging undertaking to guarantee that the third party's payment process is precise and secure. In addition, instilling worker trust in the third party is also challenging. The advanced features of blockchain technology (e.g., anonymity, immutability) provide some promise for a better solution to the above privacy challenges. In a blockchain network, users trade for services using cryptocurrencies, and each user is associated with anonymous account information. As the account address is a public key, it is hard for other users of the system to determine the real identity of the account owner. Therefore, blockchains can be leveraged to solve both the second and third ways of location privacy disclosure.

Although blockchain seems to be an ideal solution for protecting worker privacy, currently, it cannot be directly applied to a crowdsensing system. Transparency is one of the renowned features of the blockchain, which may present the risk of disclosing an individual's privacy. Lu et al. [14] point out that a "considerable amount of information about the workers will be leaked to the public through their participation history". Participants' identities might be revealed by observing a large amount of transactional information. To prevent a re-identification attack, we make use of a private blockchain to disperse the participants' transaction records. Specifically, we arrange some miners to create multiple private chains. Workers who do not want to disclose their location information can choose tasks from various private blockchains for each time slot, which makes it hard for attackers to compromise the participants' transaction history

The main contributions of this paper are summarized as follows:

- We identify a new type of location privacy attack as a result of the payment process. A server is able to infer where a worker has been by linking the amount of the payment to the task rewards. Also, the real identity of the worker cannot be protected using traditional payment methods.
- We propose a blockchain-based privacy preservation framework for protecting worker locations in crowdsensing systems. The framework, not only protects location information but also guarantees fair trading without the need for a trusted third party.
- The framework also prevents re-identifications attack by leveraging a private blockchain network, which distributes worker transaction records across many different networks. Hence, attackers cannot infer a worker's identity by observing their corresponding transaction history.
- We further systematically analyze the efficiency, accuracy, and security of the proposed system.

The rest of the paper is organized as follows. In Section 2, we introduce the preliminaries. Section 3 defines the problem and presents the proposed system framework. We outline the crowdsensing system in Section 4. Section 5 presents the privacy analysis. Section 6 details the results of the performance evaluation. Section 7 discusses related work, and Section 8 concludes the paper.

2. Preliminaries

2.1. Crowdsensing

Crowdsensing is a technique where a large group of individuals with mobile devices equipped with sensors collectively share sensory data to measure, analyze, or infer any issue of common interest. Traditional crowdsensing systems contain three entities: the requester, the server, and the workers. The requester posts the sensing tasks, the server assigns the tasks, and the workers complete the task and send the associated data to the requester for a reward. Crowdsensing has two models of task assignment, *worker selected tasks* (WST) and *server assigned tasks* (SAT).

In WST, the server publishes the tasks and the workers autonomously select the ones they prefer. The advantage of this model is that the workers do not need to reveal their exact location and the server does not know which tasks the workers have chosen to do. That is, the server has no idea where the worker is (the worker's exact location) and where the worker is going to (the location of the assigned task). The drawback of this model is that the server does not have any control over the allocation of the tasks, and workers often choose tasks based on their own objectives (e.g., the k closest tasks to reduce travel costs) [15], which may result in a low assignment success rate.

In SAT, the worker first reports their location information to the server, and the server assigns tasks according to the worker's location. In this model, the server takes control of assigning nearby tasks to the workers, while maximizing the assignment success rates. Nonetheless, both the worker's location and the task assignment information are revealed to the server, which may raise privacy concerns.

2.2. Blockchain

Blockchain was invented by Satoshi Nakamoto in 2008 to serve as the public transaction ledger of the Bitcoin cryptocurrency [16]. The ledger records a continuously growing list of transaction records, called blocks, which are linked by the cryptographic hash of the previous block. A blockchain is typically managed by a peer-to-peer network collectively following a pre-defined consensus protocol [17]. The public blockchain is permissionless and is open to everybody without exception.

A private blockchain is a blockchain that has an access control layer built into the protocol. The owner of the blockchain is a single entity who has the control over who can join the network, and who can participate in the consensus process of the blockchain. Therefore, only the participants who get the invitation and permission can join the private network.

A smart contract is a tiny executable program that is stored inside a blockchain. Once certain conditions are triggered, the program can run automatically. Smart contracts permit trusted transactions and agreements to be carried out among disparate, anonymous parties without the need for a trusted third party.

The characteristics of the blockchain are listed as follows:

- *Decentralized systems*. The blockchain network is a peer-to-peer network which, by its nature, is decentralized. All participating nodes have the same copy of the blockchain ledger, which stores all the encrypted transaction information. The more people that join the blockchain, the more secure it is. The more people join the blockchain, the more secure it is.
- *Immutability*. Blockchains are designed to be immutable. Once a block is written to a blockchain, the information cannot be altered.
- *Process integrity*. Users can trust that transactions will be executed exactly as the protocol commands, removing the need for a trusted third party.
- *Anonymous*. Blockchains use a pseudo-identity mechanism, i.e., public keys are used as identifying information, and each user can generate as many pseudo-identities as he/she likes to increase privacy.

3. Problem definition and system model

3.1. Notations

Let $\mathcal{W}\{w_1, w_2, \ldots, w_m\}$ be the set of workers, and $\mathcal{T}\{t_i, t_2, \ldots, t_n\}$ be the set of tasks. Each worker and task has a unique location denoted by the coordinates (x_i, y_i). $d_{w,t}$ represents the distance between the worker and the task, and p_w is the worker's acceptance rate. Each worker has a preferred working region $R^g_{w_i}$. TL_{w_i} represents the task list for a specific worker w_i. The worker w_i's task list TL_{w_i} includes the tasks in R_{w_i} and the isolated tasks I_t. $Pool_u$ represents the domain of all the users who registered to the public blockchain and $Pool_w$ represents the domain of workers who have uploaded their preferred working regions to the blockchain respectively. More notations are shown in Table 1.

3.2. Problem definition

In this paper, we consider a location privacy problem in a crowdsensing system where the worker chooses tasks released by the requester and, in turn, completes each task for a reward. However, the server may be untrusted. The problem is formally defined as follows.

Table 1
Notations.

Notations	Description
ID_u	The ID of the user (a worker or a requester)
ID_w	The ID of the worker
ID_r	The ID of the requester
ID_a	The ID of the agent
I_t	A set of isolate tasks
$Pool_w$	A set of workers' ID
$Pool_u$	A set of user's ID
$R^g_{w_i}$	Worker w_i's preferred working region
$R^w_{t_i}$	Rewards assigned to task t_i
$d_{w,t}$	Distance between the worker and the task
$p^d_{w_i}$	Worker's acceptance probability based on distance
p_w	Worker's acceptance probability
$p^d_{t_i}$	Task assignment probability based on distance
$maxD$	Worker's maximum travel distance
TL_{w_i}	Worker w_i's task list
$t^{ID_{t_i}}_{due}$	The due time of the task t_i

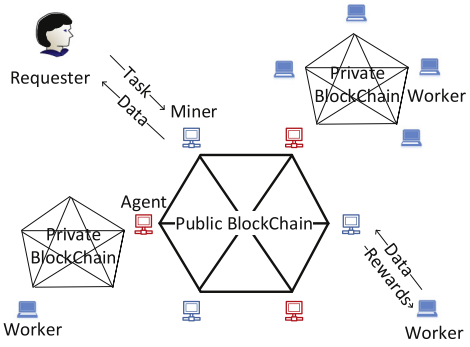

Fig. 2. Framework of the proposed crowdsensing system.

Problem 1. Take a set of workers $\mathcal{W}\{w_1, w_2, \ldots, w_m\}$, each of which has a location $w_i(x, y)$. $\mathcal{T}\{t_i, t_2, \ldots, t_n\}$ are the tasks released by the requester. Design a location privacy protection method, through which, the tasks are assigned to workers with an enhanced task assignment success rate, while the workers' exact locations and assigned tasks need to be protected.

That is, to protect the privacy of a worker's location, the worker's exact location and the task assigned to the worker both need to be protected.

3.3. System model

To solve the privacy problems mentioned in the Introduction, we have developed a new crowdsensing system framework, as shown in Fig. 2.

Four parties are involved in the proposed framework.

- *Miner*. The miners are in charge of validating new transactions and recording them on the global ledger. Because they contribute computing resources, the miners have the opportunity to gain transaction fees and rewards. In addition, a miner can also be a requester that posts a task or a worker that accepts a task.
- *Agent*. The agent works as a miner in the public blockchain and organizes a private blockchain. The agent downloads all the tasks from the public blockchain and publishes them to the agent's private blockchain network. Authorized workers

can choose a task from a private blockchain to avoid disclosing all their transaction records to the public. The agent can charge a fee to the workers for providing this service.

- *Requester.* The requester releases the tasks to the blockchain for the purpose of collecting information from certain locations. The requester can acquire adequate sensory data by taking advantage of the smart contracts in the blockchain.
- *Worker.* The workers are participants who undertake the tasks published by the requester on the blockchain. They can choose to take tasks from either the public or the private blockchains according to their own privacy concerns. Once the task is complete and the requested data has been uploaded, the workers receive their rewards. All workers are anonymous.

Within the proposed system, the requester releases the tasks to the public blockchain, and the agents release a copy of the tasks to their own private blockchains. The workers take the tasks from either a private blockchain or the public blockchain and complete the tasks in return for rewards. Agents submit the data collected from workers to the public blockchain for rewards.

3.4. Adversary model

Assumptions. We assume that all the participants on the blockchain network are untrusted except few agents. There could be more than hundreds of agents in the public blockchain. We believe that at least one or more agents would like to hold the moral bottom and unwilling to disclose the worker's information, which is practical and reasonable.

Attackers. The attacker can be any participant in the blockchain network.

- *Participants in the public blockchain.* Anybody can join the public blockchain, it also can be an attacker. But the participants in the public blockchain can only access the worker's transaction records recorded on the public blockchain.
- *Workers.* The attacker can also be a worker. And the workers can get the other worker's transaction records from the public ledger. Also, the worker can join some of the private blockchains and get the other worker's transaction records on some agent's private blockchain networks. But it is not possible for a worker to join all the agent's private blockchain network due to the private blockchain' member control protocol.
- *Agents.* The agents are possible to be malicious as well. Each agent holds the whole transaction records on his own network. But it is not possible for all of the agents to collude with each other.

4. Privacy-preserving crowdsensing system

In this section, we present the proposed blockchain-based crowdsensing system. The proposed system can solve the three aforementioned privacy disclosure issues during both the task assignment and payment process (i.e., the worker's current location, previous location, and future location). Further, the tasks can be crowdsourced to workers without relying on any trusted parties.

4.1. Overview

Using blockchain's advantages of anonymization and decentralization, we propose a blockchain-based distributed crowdsensing system. In the system, blockchains allow the requester and the worker to reach an agreement for services. The blockchain plays a third-party role but overcomes the weaknesses of using a third party, such as a single point of failure in trust. Additionally, the use of private blockchains prevent re-identification attacks based on the vast transaction histories. The general idea is that a private blockchain distributes the workers' transaction records across many networks held by different agents. Therefore, it is almost impossible for the attackers to collect all the transaction records to infer the worker's real identity, without corrupting all the agents in the network.

The general executive process of the proposed blockchain based system is shown in Fig. 3.

- *Register (Public Blockchain).* Both requesters and workers need to register with the blockchain system. Each registered user is assigned a pair of keys, and their identity is stored in the user pool. The user's registered information as a transaction is recorded in the public ledger of the blockchain.
- *Task Assignment.* The requester releases the tasks to the blockchain and assigns the rewards for each task according to the worker's uploaded location information.
- *Smart Contract Creation in the Public Blockchain.* To ensure fair trade, the requester creates a smart contract, which runs automatically according to a predefined protocol. Both the requester and the worker are required to deposit an amount of cryptocurrency in advance to the blockchain. In addition, the requester needs to define several rules for workers to ensure tasks are assigned appropriately and the quality of the uploaded sensory data.
- *Load tasks to the Private Blockchain.* The agent downloads all the information related to the tasks from the public blockchain and posts all the information on their private blockchain network. The agent is responsible for maintaining the consistency of the task information on the two blockchain networks (the public chain and their own private chain).
- *Register (Private Blockchain).* The workers choose an agent and register with the agent's private blockchain using the public keys assigned by the public blockchain. The worker's identity is validated, and only the workers who have uploaded their location information to the public blockchain for the specific requester are able to gain access.
- *Smart Contract Creation in the Private Blockchain.* As the agent needs to take the tasks from the public blockchain, the agent needs to ensure a task is still available before assigning it to a worker. Therefore, a new smart contract is created to ensure each task can be assigned successfully without losing the cryptocurrency deposit.
- *Upload Sensory Data.* Workers upload the sensory data to the blockchain. The miners validate the quality of the uploaded data. The qualified data is accepted and recorded, and the corresponding workers receive their rewards. If the sensory data is unqualified, the worker loses their deposit.
- *Payment.* If the uploaded data is qualified, the smart contract automatically executes the payment process.

4.2. Implementation of the proposed system

4.2.1. Register (public blockchain)

A user does not need to register with the blockchain using his/her real identity. Instead, each user is assigned a pair of keys. The user registers with the platform using a public key, which does not contain any information about the user. The public key is then used as the user's ID and address for transactions. This anonymous registration process increases the privacy level of users compared to traditional crowdsensing systems.

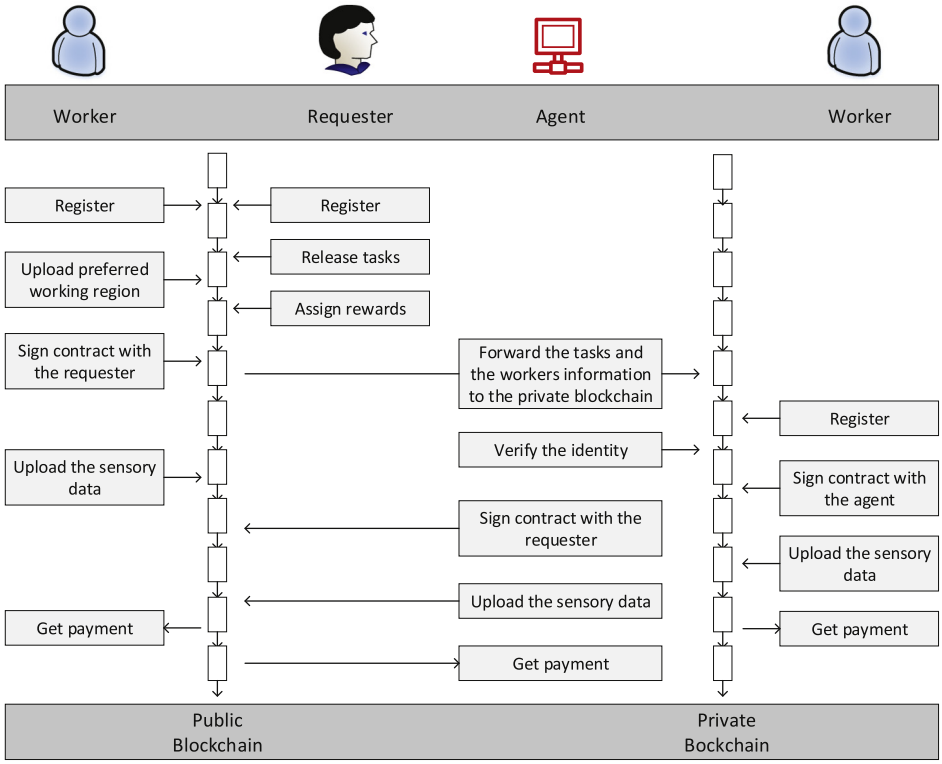

Fig. 3. Overview of the executive process in the proposed crowdsensing system.

Algorithm 1 Register in the Public Blockchain

Require: U_{type}
Ensure: pk, sk, U_{id}, $RegisterSucess$
1: $RegisterSuccess = False$;
2: $\{pk, sk\} \leftarrow keyGenerator()$;
3: $ID_{u_i} \leftarrow pk$;
4: $U_{type} \in$ {Worker, Requester};
5: **if** $ID_{u_i} \in Pool_u$ **then**
6: **return** $RegisterSuccess$
7: **end if**
8: $Pool_u \leftarrow U_{pool} \cup \{ID_{u_i}\}$;
9: $RegisterSucess = True$;
10: **return** $RegisterSucess$

Algorithm 1 provides the details of the registration process. Variable *RegisterSuccess* is an indicator that reflects whether or not the registration has been successful. The pair of secret keys is generated in Step 2. The public key is used as the user's ID in Step 3. U_{type} indicates the type of the registered user, which could be a worker or a requester. If the user ID exists in the user ID pool $Pool_u$, the ID cannot be re-registered and the registration process fails (Step 5 to Step 7). If the user ID is not in the $Pool_u$, it is added to the user pool $Pool_u$ in Step 8, and the *RegisterSuccess* status is changed to *True* in Step 9, which indicates that the user has successfully registered with the public blockchain network.

4.2.2. Task assignment

The requester releases the task information to the blockchain. The task information includes the description of the task, the location of the task, the finish time, and the status.

Here, we have combined the two different kinds of task assignment models. That is, workers upload their location information to the blockchain and choose the tasks they prefer. To guarantee a high task assignment success rate, different rewards are assigned to the tasks according to the workers' location. The details of the task assignment process are shown in Algorithm 2.

Algorithm 2 Task assignment

Require: Reward budget W_{rb}, Worker's preferred working regions $\mathcal{R}$, A set of tasks $\mathcal{T}$.
Ensure: Tasks with rewards
1: The server release all the tasks $\mathcal{T} < t_1, t_2, ..., t_{tn} >$ to the blockchain.
2: The workers who would like to take the tasks upload they preferred working region $\mathcal{R}$ to the blockchain.
3: The server assigns rewards to the tasks according to the uploaded working region $\mathcal{R}$.
4: The workers choose the tasks according to their distances to the task and the rewards they can get after finish the task.

- *Release tasks.* The requester releases tasks to the public blockchain by creating a series of transaction records. The format of the task is $t_i\langle\mathcal{L}, \mathcal{I}, Time\rangle$. $\mathcal{L}$ refers to the task's location coordinates, and $\mathcal{I}$ is a file that contains the task's description. *Time* indicates when the task needs to be finished by.

- *Upload working region* $\mathcal{R}$. . Workers are required to upload their location information to guarantee a success rate. However, to prevent their location information from being disclosed to attackers, we propose that the workers only need to upload their preferred working regions $\mathcal{R}\langle o, r\rangle$ by observing the distribution of the released tasks instead of their exact locations, where o is the coordinates of the center of the $\mathcal{R}$ and r is the radius. If the worker uploads their preferred task regions, the worker's ID is added to the worker pool $Pool_w$.

- *Assign rewards.* Workers usually prefer tasks with short travel distances assuming all the tasks have the same rewards, which reduces task success rates. Therefore, we have enhanced success rates by assigning tasks different rewards. Observing the uploaded working regions $\mathcal{R}$, we find that some tasks cover many working regions, while some tasks are not included in any working region. Obviously, the probability of a task being accepted is proportional to the number of regions the task covers. Hence, higher rewards are assigned to edge tasks (tasks that cover few regions) to increase the acceptance rate. Specifically, $R_{t_i}^w \sim \frac{1}{p_t^d}$, where p_t^d is the probability of the task being accepted based on the distance. The minimum probability of acceptance is calculated as the distance between the task and worker is $2r$. Let n be the number of regions that task t covers, we have $p_{w_i}^d = \frac{maxD-2r}{maxD}$, and $p_t^d = 1-(1-p_{w_i}^d)^n$. For the isolated tasks (tasks that have few interested workers), the acceptance probability is calculated by counting the number of region centers within the worker's maximum travel distance to the task, and let the average distance to the task be equal to $maxD - r$. Therefore, the probability of edge tasks and isolated tasks being accepted is increased, which should improve the task's success rate.

- *Task selection.* Distance and rewards are two important factors affecting a worker's choice. Therefore, we model the worker's acceptance probability p_w is modeled as a function of both distance $d_{w,t}$ and reward R_t^w, as follows:

$$p_w = f(d_{w,t}, R_t^w) = \begin{cases} y(\alpha \frac{R_t^w}{d_{w,t}}) & d_{w,t} \leq maxD \\ 0, & d_{w,t} > maxD, \end{cases} \quad (1)$$

  where $y(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ is a hyperbolic tangent function [18] that is used to map the probability to the range of [0, 1]. The tasks with lower distances and higher rewards are selected with high probability, and the tasks with longer distances and lower rewards are be selected with low probability. If the distance to the task exceeds the worker's maximum travel distance, it will not be selected even if the reward is high. In addition, according to the rules defined by the requester, workers can only choose tasks within their preferred working region or isolated tasks.

The process of task assignment does not require the worker's exact location information. Instead, workers only need to report the cloaked regions they prefer. Therefore, exact worker locations are hidden within cloaked regions and, thus, cannot be disclosed. In addition, in the proposed task assignment process, the server improves task success rates by assigning rewards instead of appointing tasks to workers, and the workers choose tasks themselves according to the rules defined by the server. Therefore, the server has no idea which tasks a worker has taken. In other words, the server has no idea where a worker will go. In short, the proposed task assignment process not only protects the worker's location information but also ensures a high task assignment success rate.

4.2.3. Smart contract creation in the public blockchain

To allow for fair trading, a smart contract is created that contains the agreement between requester and workers is created. Fig. 4 shows the components included in the smart contract.

The requester's ID and the worker's ID are included in the contract. The requester ID indicates the owner of the task, and the worker ID indicates the worker who has accepted the task. The user ID is also used as the transaction address for transferring cryptocurrency. In addition, information about the task is included, such as the task ID and a description of the work to be completed. *STATUS* indicates the status of the task, i.e., whether the task is still available or has been accepted by enough workers. *REWARDS* shows the cryptocurrency the workers receive on completion. To prevent cheating by the requester, a deposit is required. The worker also needs to make a deposit, in case a malicious worker takes the task but refuses to submit appropriate data. A validation function in the smart contract verifies whether the worker is eligible to accept a task according to the rules defined by the requester. Here, workers can only choose the tasks on their task list (tasks within the reported region and isolated tasks), while the agent can choose any tasks for a worker. The purpose of this is to guarantee consistency and task success rates. The evaluation function in the smart contract is used to verify whether the submitted sensory data is appropriate. The evaluation standards are defined by the requester. We only consider two levels of quality for the sensory data, qualified and unqualified. Algorithm 3 shows the details of the executive process.

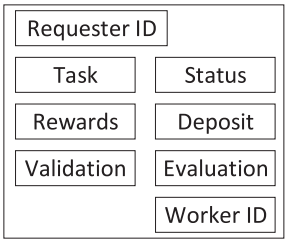

Fig. 4. Components of a smart contract created by the requester.

4.2.4. Task transfer

To prevent great quantities of personal transaction data from being visible to the public, which leads to identity disclosure, the published task information and the worker's registration information are forwarded to the private blockchain by agents. The workers can choose to accept tasks from a private blockchain instead of the public blockchain. Each agent provides a private blockchain with a copy of the tasks published in the public blockchain as well as the workers' information (e.g., ID and location information). Workers can choose to join any private blockchain with the corresponding agent's permission. The agent must follow the rules defined by the requester and can only create new contracts with the workers that have joined their private blockchain.

4.2.5. Registration (private blockchain)

Unlike the public blockchain, workers must be validated by either the network owner or by a set of rules put in place by the owner in the private blockchain. In the proposed system, the agent verifies whether the workers are eligible to join the network by examining three conditions:

- Whether the number of workers is under the private blockchain researches the upper limit?
- Whether the worker's ID is n the worker pool $Pool_w$?
- Whether all the tasks in worker's task list have been assigned?

Specifically, the worker submits their request to the agent to show their interest in joining the private blockchain network. If the worker passes an identify authentication, and the worker's ID is in the worker pool $Pool_w$, the request should be approved. Algorithm 4 shows the details of the authentication process.

Algorithm 3 Contract Creation

Require: Worker's ID ID_w, Requester's ID ID_r, Worker's pool $Pool_w$, Due time $t_{due}^{ID_{t_i}}$
Ensure: Tasks with rewards

Task $\leftarrow ID_t$;
Owner $\leftarrow ID_r$;
Status $\leftarrow$ *Available*;
Reject $=$ *False*;
legal $=$ *False*;
Rewards $\leftarrow$ *WithdrawMoney*(ID_r);
if *Rewards* $< R_{t_i}$ **then**
  return *ContractCreate* $=$ *False*;
end if
if *Validation*(ID_w) $=$ *True* **then**
  legal $=$ *True*;
  Deposite $\leftarrow$ *WithdrawMoney*(ID_w);
  Sign the contract with ID_w and publish it to the blockchain;
  Status $=$ *UnAvailable*;
end if
if *legal* $=$ *False* **then**
  return *Reject* $=$ *True*;
end if
SensoryData $=$ *WaitDataUpLoading*();
if current time $> t_{due}^{ID_{t_i}}$ **then**
  Status $=$ *Fail*;
  $ID_r \leftarrow$ *Transfer*(*Rewards*, *Deposite*, *Owner*);
  return *Status*;
end if
if *Evaluation*(*SensoryData*) $=$ *True* **then**
  $ID_w \leftarrow$ *Transfer*(*Rewards*, *Deposite*, ID_w);
  State $=$ *Done*;
else
  $ID_r \leftarrow$ *Transfer*(*Deposite*, *Owner*);
  State $=$ *Availabe*;
end if

Algorithm 4 Authentication Algorithm

Require: *pk sk* Worker pool $Pool_w$
Ensure: *Authentication*

Authentication $=$ *False*;
IsThePerson $=$ *UserAuthentication*(*pk*, En_{sk}(*pk*));
if *IsThePerson* $=$ *False* **then**
  return *Authentication*
end if
if $W_{id} \in Pool_w$ **then**
  return *Authentication* $=$ *True*;
end if
return *Authentication*

4.2.6. Smart contract creation in private blockchains

Because a task's status does not automatically synchronize the public blockchain with the private blockchains, the agent needs to check whether or not the task is available in the public blockchain before assigning it to workers to avoid problems with a null smart contract. For example, assigning a contract to a worker for a task that is no longer available would mean both the agent and the worker lose their deposit. Or if an agent assigns a contract to a requester but the worker changes their mind and subsequently refuses to accept the task, the agent will lose their deposit. Therefore, the agent must create a separate smart contract with the workers to ensure the agent does not lose their deposit. The details of this contract are shown in Algorithm 5.

Algorithm 5 The Smart contract in the Private blockchain

Require: User ID W_{id}, Task list W_{tlist}, Task ID T_{id}, Due time $t_{due}^{T_{id}}$
Ensure: Tasks with rewards

Task $\leftarrow ID_t$;
Owner $\leftarrow ID_r$;
Status $\leftarrow$ *UnDistributed*;
Reject $=$ *False*;
legal $=$ *False*;
Rewards $\leftarrow$ *WithdrawMoney*(ID_a);
if *Rewards* $< R_{t_i}$ **then**
  return *ContractCreate* $=$ *False*
end if
if *Validation*(ID_w) $=$ *True* **then**
  legal $=$ *True*;
  Deposite $\leftarrow$ *WithdrawMoney*(ID_w);
  Sign the contract with ID_w and publish it to the blockchain.
  Status $=$ *Undetermined*;
end if
if *legal* $=$ *False* **then**
  return *Reject* $=$ *True*
end if
The agent sign contract with requester in the public blockchain.

if The agent gets the task **then**
  Status $=$ *UnAvailable*;
end if
if The agent does not get the task **then**
  $ID_w \leftarrow$ *transfer*(*Deposite*);
  $ID_a \leftarrow$ *transfer*(*Rewards*);
  Status $=$ *Done*;
end if
SensoryData $=$ *WaitDataUpLoading*();
if current time $> t_{due}^{ID_{t_i}}$ **then**
  Status $=$ *Fail*;
  $ID_r \leftarrow$ *Transfer*(*Rewards*, *Deposite*, *Owner*)
  return *Status*
end if
if *Evaluation*(*SensoryData*) $=$ *Ture* **then**
  $ID_w \leftarrow$ *Transfer*(*Rewards*, *Deposite*, ID_w);
  State $=$ *Done*;
else
  $ID_r \leftarrow$ *Transfer*(*Deposite*, *Owner*)
  State $=$ *UnDistributed*;
end if

Unlike contracts on the public blockchain, an agent cannot be sure whether or not a task is still available. Therefore, the initial status of the task is *UnDistributed* (Step 3), which means the task has not yet been completed. Once a worker signs a contract with the agent, the task's status is changed to *Undecided* (Step 13), which means the task has been assigned to at least one worker, but it is not clear whether the task is still available. The agent must continue to check the status of tasks in the public blockchain and update the status of each one. If the relevant task is still available, the agent signs the contract with the requester. The status of the task is then changed to *UnAvailable* in the private blockchain (Steps 20 to 22). Steps 23 to 27 shows that if the task is not available, the deposits are paid back to the agent and the worker. The status of the task changes to *Done* when the task is successfully completed. The public blockchain uses the same process.

4.2.7. Upload sensory data

Once each worker finishes their task, they upload the sensory data to the blockchain. If the task takes longer than the required time, the contract is terminated, and the task fails. If the worker finishes and uploads the data on time, the miners validate the quality of the uploaded data. How quality is evaluated is beyond the scope of this paper. Hence, we have only included two levels of quality — qualified and unqualified. Interested readers can refer to paper [19] for quality estimation. If the uploaded sensory data satisfies the requirements defined by the requester (i.e., the data is qualified), the worker receives their rewards and their deposit is refunded. If the uploaded sensory data does not satisfy the requirements (i.e., the data is unqualified) the worker does not receive their reward, and the deposit is transferred to the agent.

4.2.8. Payment

Contra to traditional payment methods, in a blockchain network, a worker's account information is not linked to any personal information. The smart contract automatically pays workers in a cryptocurrency through their public keys without knowing their real identity.

5. Privacy ans security analysis

5.1. Privacy analysis

In this paper, we consider three forms of location privacy disclosure common to traditional crowdsensing systems and analyze how our proposed system tackles these attacks.

- The server knows a worker's current location when they submit their interest in a task. This type of privacy disclosure occurs when the workers upload their exact location to the server so that tasks can be assigned appropriately. In the proposed system, the workers upload their preferred working regions instead of their exact location. Workers in the blockchain network are anonymous, so even a worker's identity is re-identified, the attacker has no idea of the worker's exact location. In addition, the workers can choose to cloak their preferred regions to suit their own privacy needs. More cautious workers can submit larger areas with a shifted center to hide their location.
- The server knows a worker's future location from the task assigned. In the proposed system, tasks are not assigned by the requester or the server. Instead, workers choose the tasks they would like to undertake. Additionally, the workers are anonymized in the blockchain system. To prevent re-identification attacks, a private blockchain is introduced to distribute transaction records. As the private blockchain has the membership control, not all the participants can access. Therefore, with the proposed crowdsensing system, workers can protect their location information by registering multiple accounts and choosing tasks through different agents.
- The server knows a worker's previous location from the payment process. Traditional crowdsensing systems cannot avoid this type of location privacy disclosure. In a traditional payment method, the user's bank account information must be authenticated with their real name. Hence, it is inevitable that a server will discover a worker's real identity. Also, as mentioned, payments can be linked to tasks. The proposed system uses cryptocurrency and workers are anonymized, so payments can be made without knowing the worker's real identity.

Based on the above analysis, we claim that the proposed method can effectively prevent the location privacy disclosure in spatial crowdsensing systems.

5.2. Security analysis

As previously discussed, a traditional crowdsensing system inevitably discloses private information during the payment process. Currently, the only way to solve this problem is to introduce a trustworthy third party. However, it is difficult to guarantee proper and safe payments through a third party. For example, a worker completes a task on time and the data is qualified, but the trusted third party underpays the worker or even refuses to pay the reward.

The proposed blockchain-based system takes advantage of blockchain's inherent characteristics to eliminate the security threats brought by trusted third parties. The requester and the worker reach an agreement by signing a smart contract on the blockchain. The requester defines the rules of payment, including the evaluation criteria of the sensory data, the corresponding rewards, and so on. Once the smart contract is triggered, the predefined program runs automatically, and the corresponding rewards are paid.

Another security threat in traditional crowdsensing systems comes from malicious workers who accept tasks but do not subsequently submit qualified data or even undertake the task. These malicious acts often mean conscientious workers do not receive their rewards because the requester fails to collect enough data. The usual practice in traditional systems is to build a reputation management system, and only workers who with good reputations are allocated tasks. However, this system penalizes new workers, as they have no established reputation.

The proposed system effectively solves this type of security problem. Both the requester and the workers need to deposit some amount of cryptocurrency before the task is assigned. If the worker does not submit qualified data on time, the deposit is paid to the requester (or agent) according to the program predefined in the smart contract. The consensus protocol of the blockchain ensures that the smart contracts are executed correctly. The combination of a smart contract and a deposit-based mechanism ensures fair trading between the workers and the requesters.

A possible security problem in the proposed system is that the uploaded data may be downloaded and reused by other malicious workers. This problem can be solved by encrypting the sensory data using the requester's or agent's public key before uploading it to the blockchain network. And the new method need to be proposed to solve the data quality evaluation problem. Therefore, we take this problem as an future work.

6. Performance evaluation

We analyzed the performance of the proposed system in two respects: the task assignment success rate, and the execution efficiency using blockchain.

6.1. Task assignment

The task assignment success rate (TASR) was evaluated through a series of experiments using both a real-world dataset and a synthetic dataset. We used the Yelp business dataset [20] as the real-world data, which includes information about local businesses in four countries. Gyms located in Australia were used as the task locations, and post offices were used as workers. Further, we compared the proposed method with three baselines. The proposed method is denoted as TA+p+r, which means workers upload their preferred working region and the tasks have different rewards.

- Baseline 1: The workers upload their preferred working region and the tasks have the same rewards (TA + p − r).

Table 2
TASR by varying maxD in the Yelp dataset.

	3 km	5 km	7 km	9 km	11 km	13 km
TA+p+r	0.474	0.604	0.684	0.738	0.782	0.823
TA+p+r	0.447	0.564	0.633	0.693	0.748	0.784
TA+p−r	0.455	0.620	0.691	0.742	0.790	0.817
TA−p−r	0.440	0.584	0.660	0.703	0.741	0.771

Table 3
TASR by varying maxD in the synthetic dataset.

	15 km	25 km	35 km	45 km	55 km	65 km
TA+p+r	0.433	0.653	0.792	0.877	0.93	0.965
TA−p+r	0.421	0.643	0.773	0.856	0.909	0.953
TA+p−r	0.43	0.659	0.798	0.873	0.913	0.957
TA−p−r	0.428	0.638	0.773	0.862	0.907	0.954

Table 4
TASR when varying the number of workers in the synthetic dataset.

	10	40	70	100	130	160
TA+p+r	0.24	0.519	0.714	0.826	0.92	0.939
TA−p+r	0.174	0.517	0.705	0.819	0.909	0.93
TA+p−r	0.264	0.496	0.693	0.830	0.925	0.946
TA−p−r	0.163	0.483	0.672	0.820	0.913	0.941

- Baseline 2: The workers upload their exact location and the tasks have different rewards (TA − p + r).
- Baseline 3: The workers upload their exact locations and the tasks have the same rewards (TA − p − r).

Table 2 shows the results on the Yelp dataset at different maximum travel distances. Obviously, the TASR increased with an increase in *maxD* because there are more tasks that cannot be reached if the value of *maxD* is small. In addition, we observed that the methods with different rewards (TA + r + p, TA + r − p) performed better than methods with the same rewards for all tasks (TA + r + p, TA − r − p) regardless of the value of the maximum travel distance. This is because higher rewards prompt the workers to choose tasks with a greater travel distance, which contributes to the TASR. Also, we find that the methods using cloaking regions do not reduce the TASR compared to the methods that use exact location information; they perform similarly. This means that using a cloaking region does not affect the assigned rewards.

We also tested all the methods using a synthetic dataset. Specifically, we randomly generated 1000 tasks and 30 workers with both x and y coordinates within the range [1, 1000].

Table 3 shows the results on the synthetic dataset with different maximum travel distances. Similar results can be observed.

Table 4 shows the results when varying the number of workers from 10 to 160 in steps of 30 at a $maxD = 15$ km.

We observe that the proposed method outperformed all methods that assign the same rewards to each task in all settings. Also, with a small number of workers, the TASR was much lower because a limited number of workers means fewer tasks can be completed. The TA + r + p method significantly outperformed the TA − r + p method with only ten workers. However, the performance difference between the two methods decreased as the number of workers increased. But the TA + r + p method still outperformed the TA − r + p method. Similarly, using cloaked regions did not affect task assignment, no matter how many workers there were.

6.2. Efficiency analysis

Theoretically, blockchain's design provides a security environment that is easily adapted to a crowdsensing framework. Due to its distributed and decentralized nature, blockchain eliminates a potential single point of failure associated with traditional crowdsensing systems. Its anonymous nature allows the workers to complete tasks without disclosing their real identity. Executing of the smart contract allows for safe and fair trading, which has positive impacts on the efficiency of transactions between workers and requesters.

The biggest concern with using blockchain is the latency of the confirmation time because blockchain requires a strict verification process to create new transaction records. In the proposed system, the requester calculates assigned rewards off the chain to reduce the number of computing resources needed on the chain. The time complexity for executing a smart contract is less than $O(n)$. Therefore, there are no burdensome computing tasks on the blockchain. A pioneering work, [21], has shown that only conducting lightweight processing tasks on the blockchain improves efficiency. Therefore, the efficiency of the proposed system is acceptable.

7. Related work

Preserving privacy in crowdsensing. Various technologies have been proposed to protect a worker's location information [22–31]. For example, dummy locations [22] protects user locations by adding false positions to the true location information before sending it to the server. Cloaking regions [25] transform an exact location into a region large enough to thwart attacks. Differential privacy-based methods [26] add controlled random noise to a user's location, making it indistinguishable from other locations within a predefined radius r. The transformation method [27] performs basic geometric operations on the user's location, and private information retrieval [28] protects the user's location by applying encryption.

Most of these technologies have been used in crowdsourcing systems to protect the workers' location privacy. Hu et al. [32] employed a peer-to-peer cloaking technique to cloak worker locations among $k - 1$ other workers. Shen et al. [12] applied an encryption technique and proposed a privacy framework that performs worker task matching in an encrypted domain. Wang et al. [33] proposed a location privacy-preserving task allocation framework with geo-obfuscation to protect users' locations during task assignments, which means make participants obfuscate their reported locations under the guarantee of differential privacy. Zhu et al. [34] proposed a location privacy-preserving mechanism CKD for a mobile crowdsensing system. The proposed method combines k-anonymity and differential privacy-preserving technologies. Bin et al. [35] presented a clustering method in which the location of the virtual cluster center is reported to the server by a cluster head. Once the cluster head receives the task from the server, tasks are assigned to the chosen cluster member according to their exact location.

Most of these methods only considered the first situation of location privacy disclosure during the process of task assignment. That is, they hide the workers' exact location to prevent attackers from inferring where the workers are. Only a few are able to hide the tasks the worker accepts, and none consider privacy disclosure during the payment process. Therefore, a worker's location information can be disclosed once they complete the task and receive the payment.

Blockchain-based crowdsensing systems. Blockchain technology has been well studied, but there have been only a few efforts to combine a blockchain with a crowdsensing system. Li et al. [36] presented a design for CrowdBC, which is a blockchain-based decentralized framework for crowdsensing systems. A series of algorithms based on smart contracts have also been developed. The main purpose of paper [36] is to address the problem of a single point of failure in the traditional crowdsensing system.

Wang et al. [37] proposed a secure incentive mechanism for crowdsensing applications, where cryptocurrency built on a blockchain is used to prevent the security issues caused by a trustful center. Lu et al. [17] presented a private and anonymous crowdsensing system (ZebraLancer) built atop Open Blockchain. ZebraLance allows a fair exchange between crowdsourced data and its corresponding rewards. To guarantee anonymity while preserving accountability, they proposed an anonymous authentication scheme that supports a delicate linkability but only for authenticated messages sharing a common prefix. Federico et al. [38] proposed a blockchain-based crowdsensing application for process court adjudications. The mentioned workers solved the security problem, the privacy problem of Blockchain, and the specific application problem. However, the location privacy problems in existing crowdsensing systems were not discussed.

8. Conclusion

This paper analyzes the current privacy problems in the existing spatial crowdsensing system where worker locations are inevitably disclosed during the payment process. To prevent breaches of privacy, we proposed a novel blockchain-based privacy-preserving crowdsensing system. We use the anonymous nature of blockchains to protect the real identity of workers. To prevent re-identification attacks, private blockchains are hosted by agents. The public blockchain is transparent, but a worker's transaction history is distributed across the private blockchain networks, which effectively prevents re-identification attacks. In addition, during the task assignment stage, we propose a new task assignment pattern. We combine the WST and SAT models together to reduce the ways location privacy can be disclosed and enhance the task assignment success rates through a reward assignment system. The experiments show that the proposed method performs well while protecting the privacy of worker locations.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No. 61502362.

References

[1] Y. Wang, W. Meng, W. Li, J. Li, W.X. Liu, Y. Xiang, A fog-based privacy-preserving approach for distributed signature-based intrusion detection, J. Parallel Distrib. Comput. 122 (2018) 26–35.
[2] R.K. Rana, C.T. Chou, S.S. Kanhere, N. Bulusu, W. Hu, Ear-phone: An end-to-end participatory urban noise mapping system, in: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, ACM, 2010, pp. 105–116.
[3] P. Mohan, V.N. Padmanabhan, R. Ramjee, Nericell: Rich monitoring of road and traffic conditions using mobile smartphones, in: Proceedings of the 6th International Conference on Embedded Networked Sensor Systems, SenSys 2008, Raleigh, NC, USA, November 5-7, 2008, 2008, pp. 323–336.
[4] Y. Chon, N.D. Lane, F. Li, H. Cha, F. Zhao, Automatically characterizing places with opportunistic crowdsensing using smartphones, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, ACM, 2012, pp. 481–490.
[5] T. Hara, A. Suzuki, M. Iwata, Y. Arase, X. Xie, Dummy-based user location anonymization under real-world constraints, IEEE Access 4 (2016) 673–687.
[6] Y. Xiao, L. Xiong, Protecting locations with differential privacy under temporal correlations, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM, 2015, pp. 1298–1309.
[7] C.A. Ardagna, M. Cremonini, S.D.C. di Vimercati, P. Samarati, An obfuscation-based approach for protecting location privacy, IEEE Trans. Dependable Secure Comput. 8 (1) (2011) 13–27.
[8] M.E. Andrés, N.E. Bordenabe, K. Chatzikokolakis, C. Palamidessi, Geo-indistinguishability: Differential privacy for location-based systems, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, ACM, 2013, pp. 901–914.
[9] T. Zhu, G. Li, W. Zhou, P.S. Yu, Differentially private data publishing and analysis: A survey, IEEE Trans. Knowl. Data Eng. 29 (8) (2017) 1619–1638.
[10] L. Kazemi, C. Shahabi, A privacy-aware framework for participatory sensing, SIGKDD Explor. 13 (1) (2011) 43–51.
[11] M. Yang, T. Zhu, Y. Xiang, W. Zhou, Density-based location preservation for mobile crowdsensing with differential privacy, IEEE Access 6 (2018) 14779–14789.
[12] Y. Shen, L. Huang, L. Li, X. Lu, S. Wang, W. Yang, Towards preserving worker location privacy in spatial crowdsourcing, in: 2015 IEEE Global Communications Conference, GLOBECOM 2015, San Diego, CA, USA, December 6-10, 2015, 2015, pp. 1–6.
[13] P. Xiong, L. Zhang, T. Zhu, Reward-based spatial crowdsourcing with differential privacy preservation, Enterprise IS 11 (10) (2017) 1500–1517.
[14] Y. Lu, Q. Tang, G. Wang, ZebraLancer: Private and anonymous crowdsourcing system atop open blockchain, CoRR abs/1803.01256 (2018) arXiv:1803.01256.
[15] H. To, C. Shahabi, Location privacy in spatial crowdsourcing, CoRR abs/1704.06860 (2017) arXiv:1704.06860.
[16] E. Staff, Blockchains: The great chain of being sure about things, Econom. Retriev. 18 (2016).
[17] Y. Lu, Q. Tang, G. Wang, ZebraLancer: Private and anonymous crowdsourcing system atop open blockchain, CoRR abs/1803.01256 (2018) arXiv:1803.01256.
[18] J. Anderson, Hyperbolic Geometry, Springer Science & Business Media, 2006.
[19] J. Wang, M. Li, Y. He, H. Li, K. Xiao, C. Wang, A blockchain based privacy-preserving incentive mechanism in crowdsensing applications, IEEE Access 6 (2018) 17545–17556.
[20] Yelp dataset challenge, https://www.yelp.com/dataset_challenge.
[21] G. Zyskind, O. Nathan, A. Pentland, Enigma: Decentralized computation platform with guaranteed privacy, CoRR abs/1506.03471 (2015) arXiv:1506.03471.
[22] T. Hara, A. Suzuki, M. Iwata, Y. Arase, X. Xie, Dummy-Based user location anonymization under real-world constraints, IEEE Access 4 (2016) 673–687.
[23] Q. Zhang, L.T. Yang, Z. Chen, P. Li, F. Bu, An adaptive droupout deep computation model for industrial IoT big data learning with crowdsourcing to cloud computing, IEEE Trans. Ind. Inf. (2018).
[24] X. Liu, R. Choo, R. Deng, R. Lu, J. Weng, Efficient and privacy-preserving outsourced calculation of rational numbers, IEEE Trans. Dependable Secure Comput. (2016).
[25] C.A. Ardagna, M. Cremonini, S.D.C. di Vimercati, P. Samarati, An obfuscation-based approach for protecting location privacy, IEEE Trans. Dependable Sec. Comput. 8 (1) (2011) 13–27.
[26] M.E. Andrés, N.E. Bordenabe, K. Chatzikokolakis, C. Palamidessi, Geo-indistinguishability: differential privacy for location-based systems, in: 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013, 2013, pp. 901–914.
[27] A. Gutscher, Coordinate transformation - A solution for the privacy problem of location based services? in: 20th International Parallel and Distributed Processing Symposium, IPDPS 2006, Proceedings, 25-29 April 2006, Rhodes Island, Greece, 2006.
[28] X. Yi, R. Paulet, E. Bertino, V. Varadharajan, Practical approximate k nearest neighbor queries with location and query privacy, IEEE Trans. Knowl. Data Eng. 28 (6) (2016) 1546–1559.
[29] Q. Zhang, L.T. Yang, Z. Chen, P. Li, M.J. Deen, Privacy-preserving double-projection deep computation model with crowdsourcing on cloud for big data feature learning, IEEE Internet Things J. 5 (4) (2018) 2896–2903.
[30] S. Zeng, X. Wang, H. Cui, C. Zheng, D. Feng, A unified collaborative multikernel fuzzy clustering for multiview data, IEEE Trans. Fuzzy Syst. 26 (3) (2018) 1671–1687.
[31] W. Meng, L. Jiang, Y. Wang, J. Li, J. Zhang, Y. Xiang, JFCGuard: Detecting juice filming charging attack via processor usage analysis on smartphones, Comput. Secur. 76 (2018) 252–264, http://www.sciencedirect.com/science/article/pii/S0167404817302493.
[32] J. Hu, L. Huang, L. Li, M. Qi, W. Yang, Protecting location privacy in spatial crowdsourcing, in: Web Technologies and Applications - APWeb 2015 Workshops, BSD, WDMA, and BDAT, Guangzhou, China, September 18, 2015, Revised Selected Papers, 2015, pp. 113–124.
[33] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, X. Ma, Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation, in: Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017, 2017, pp. 627–636.
[34] Z. Chi, Y. Wang, Y. Huang, X. Tong, The novel location privacy-preserving ckd for Mobile Crowdsourcing Systems, IEEE Access 6 (2018) 5678–5687.
[35] B. Zhu, S. Zhu, X. Liu, Y. Zhong, H. Wu, A novel location privacy preserving scheme for spatial crowdsourcing, in: Electronics Information and Emergency Communication, ICEIEC, 2016 6th International Conference on, IEEE, 2016, pp. 34–37.
[36] M. Li, J. Weng, A. Yang, W. Lu, CrowdBC: A blockchain-based decentralized framework for crowdsourcing, IACR Cryptol. ePrint Arch. 2017 (2017) 444.

[37] J. Wang, M. Li, Y. He, H. Li, K. Xiao, C. Wang, A blockchain based privacy-preserving incentive mechanism in crowdsensing applications, IEEE Access 6 (2018) 17545–17556.
[38] A. Ast, A. Sewrjugin, The Crowdjury, a Crowdsourced Justice System for the Collaboration Era, 2015.

Mengmeng Yang received the BEng degree from Qingdao Agricultural University, China, in 2011 and the MEng degree from Shenyang Normal University, China, in 2014.

She is currently a PhD student in the School of Information Technology, Deakin University, Australia. Her research interests include privacy preserving, machine learning, and network security.

Tianqing Zhu received the BEng and MEng degrees from Wuhan University, China, in 2000 and 2004, respectively, and the PhD degree in computer science from Deakin University, Australia, in 2014.

Dr Tianqing Zhu is currently a senior lecturer in the school of software in University of Technology Sydney, Australia. Before that, she was a lecture in the School of Information Technology, Deakin University, Australia, from 2014 to 2018. Her research interests include privacy preserving, data mining and network security.

Dr. Tianqing has won the best student paper award in PAKDD 2014. She is a member of the IEEE.

Kaitai Liang received the PhD degree from the Department of Computer Science, City University of Hong Kong in 2014. He is currently an assistant professor with the Department of Computer Science, University of Surrey, U.K. His research interests are applied cryptography and information security in particular, encryption, network security, blockchain, post-quantum cryptography, privacy-enhancing technology and security in cloud computing.

Professor Wanlei Zhou received the B.Eng and M.Eng degrees from Harbin Institute of Technology, Harbin, China in 1982 and 1984, respectively, and the PhD degree from The Australian National University, Canberra, Australia, in 1991, all in Computer Science and Engineering. He also received a DSc degree from Deakin University in 2002.

He is currently the Head of School of school of software in University of Technology Sydney, Australia. He was an Alfred Deakin Professor and Chair of Information Technology in Deakin University. Professor Zhou has published more than 300 papers in refereed international journals and refereed international conferences proceedings. Prof Zhou's research interests include distributed systems, network security, and privacy preserving.

Prof. Wanlei has chaired many international conferences and has been invited to deliver keynote address in many international conferences. He is a Senior Member of the IEEE.

Robert Deng is AXA Chair Professor of Cybersecurity and Director of the Secure Mobile Centre, School of Information Systems, Singapore Management University (SMU). His research interests are in the areas of data security and privacy, cloud security and Internet of Things security. He received the Outstanding University Researcher Award from National University of Singapore, Lee Kuan Yew Fellowship for Research Excellence from SMU, and Asia-Pacific Information Security Leadership Achievements Community Service Star from International Information Systems Security Certification Consortium. He serves/served on many editorial boards and conference committees. Including the editorial boards of IEEE Security & Privacy Magazine, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, Journal of Computer Science and Technology, and Steering Committee Chair of the ACM Asia Conference on Computer and Communications Security. He is an IEEE Fellow.