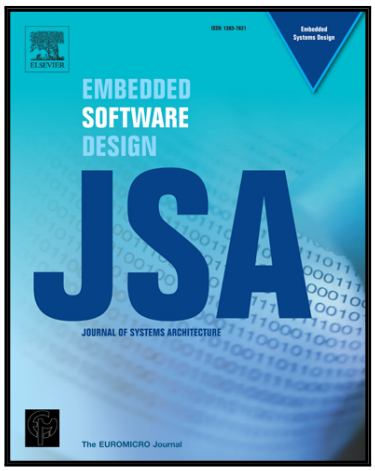


P2TA: Privacy-Preserving Task Allocation for Edge Computing Enhanced Mobile Crowdsensing

Hang Shen, Guangwei Bai, Yujia Hu, Tianjing Wang

PII: S1383-7621(18)30451-X
DOI: https://doi.org/10.1016/j.sysarc.2019.01.005
Reference: SYSARC 1557

To appear in: *Journal of Systems Architecture*

Received date: 1 October 2018
Revised date: 26 December 2018
Accepted date: 16 January 2019

Please cite this article as: Hang Shen, Guangwei Bai, Yujia Hu, Tianjing Wang, P2TA: Privacy-Preserving Task Allocation for Edge Computing Enhanced Mobile Crowdsensing, *Journal of Systems Architecture* (2019), doi: https://doi.org/10.1016/j.sysarc.2019.01.005

P2TA: Privacy-Preserving Task Allocation for Edge Computing Enhanced Mobile Crowdsensing

Hang Shen[a,b], Guangwei Bai[a,*], Yujia Hu[a], Tianjing Wang[a]

[a]*Department of Computer Science and Technology, Nanjing Tech University, 30 South Puzhu Road, Nanjing 211816, China*
[b]*Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo N2L3G1, Canada*

Abstract

In conventional mobile crowdsensing (MCS) applications, the crowdsensing server (CS-server) needs mobile users' precise locations for optimal task allocation, which raises privacy concerns. This paper proposes a privacy-preserving task allocation framework (called P2TA) for edge computing enhanced MCS, focusing on optimize task acceptance rate while protecting participants' privacy by introducing edge nodes. The basic idea is that edge nodes act as task assignment agents with privacy protection that prevents an untrusted CS-server from accessing a user's private data. We begin with a thorough analysis of the limitations of typical task allocation and obfuscation schemes. On this basis, the optimization problem about location obfuscation and task allocation is formulated in consideration of privacy constraints, travel distance and impact of location perturbation. Through problem decomposition, the location obfuscation subproblem is modeled as a leader-follower game between the designer of location obfuscation mechanism and the potential attacker. Against inference attack with background knowledge, a genetic algorithm is introduced to initialize an obfuscation matrix. With the matrix, an edge node makes task allocation decisions that maximize task acceptance rate subject to differential and distortion privacy constraints. The effectiveness and superiority of P2TA compared to exiting task allocation schemes are validated via extensive simulations.

Keywords: Mobile crowdsensing; Task allocation; Edge computing;

*Corresponding author.
Email addresses: hshen@njtech.edu.cn (Hang Shen), bai@njtech.edu.cn (Guangwei Bai)

Obfuscation; Privacy-preserving.

1. Introduction

Mobile crowdsensing (MCS) [1], a type of emerging human-powered sensing paradigm, leverages millions of individual mobile devices to sense, collect, and analyze urban data without deploying a large number of static sensors as sensing infrastructures. Due to its low cost and spatial-temporal coverage, MCS serves as a critical building block for various Internet of Things (IoT) applications [2] from air monitoring [3] to localization [4] and intelligent transportation [5]. On an MCS system, mobile users as task participants are registered as candidate workers who collect and contributes data through sensing devices (e.g. smart phones, smart glasses and smart watches) they carry. Once a new task arrives, the crowdsensing server (referred as CS-server) selects some workers to go to prespecified places to complete certain tasks (i.e. task allocation [6]) by incentives [7].

Because of diverse qualities of participants on different tasks, task allocation is critical to an MCS platform, the efficiency of which depends mostly on location information to calculate the distances between tasks and workers. The longer the distance from a user to the target location of a task, the greater the rewards of completing the task; the shorter the travel distance, the more likely the user is to accept the task and the fewer rewards the CS-server should pay. A number of researchers (e.g. [8, 9]) make an assumption that users' locations are known to the CS-server for better task allocation. However, these location information may fall in the hands of an untrusted CS-server. Even with incentives, concerns about privacy leakage and security threat will discourage users from engaging in MCS. Thus, location privacy preservation should be jointly taken into account in MCS task allocation.

Existing location privacy protection mechanisms (e.g. spatial cloaking [10], dummy [11], perturbation [12] and encryption [13, 14]) designed in the context of location-based services (LBSs) are not directly tailored for MCS task allocation. First, these mechanisms may reduce the availability of data received due to the need to falsify or modify the location or submission time of data collected. Second, in contrast to LBS which focuses mainly on the enhancement of a user's privacy, the optimization of MCS task allocation requires a comprehensive consideration of the interaction between privacy protection and travel distance. Third, due to the large number of users and tasks, privacy-preserving task assignment relies on a platform with sufficient computing and storage capabilities, which differs

from the user-centric privacy protection mechanism which locally runs on a user's mobile device. These constraints and challenges make privacy-preserving MCS task allocation a challenging issue.

Thanks to the emergence of edge computing [15, 16, 17], it is promising to achieve privacy-preserving task allocation by deploying between users and the CS-server. The basic idea behind edge computing enhanced MCS is to perform computations at the edge of the network as an anonymous server and a task allocation agent. The potential advantages mainly include three aspects. First, a user's real location can be replaced by an obfuscated location by an edge node before uploading. Second, the decentralized and switchable nature of edge nodes helps avoid many potential privacy exposure risks. Third, edge nodes are closer to users, which helps improve the real-time performance of task assignment and user response. Last but not least, edge nodes typically do not suffer from computation and storage performance bottlenecks when they undertake privacy protection and task assignment. Yet in spite of these advantages, many private things still can be deduced form the obfuscated locations when a malicious attacker holds certain prior knowledge. If there are no countermeasures against such inference attack, privacy guarantees provided by edge nodes will be downgraded.

This paper proposes a privacy-preserving task allocation framework (P2TA) for edge computing enhanced MCS, where edge nodes act as agents to obfuscate locations uploaded by users and assign tasks to proper users. The focus is on maximizing task acceptance rate while achieving an efficient tradeoff between privacy level and travel distance. The main contributions include:

1. To begin with, we compare user-centric and task-centric task allocation approaches and analyze the impact of location obfuscation on task assignment. On this basis, we formulate the optimization problem regarding maximizing task acceptance rate and providing privacy guarantees.
2. The optimization problem is divided into obfuscation-based privacy game subproblem and task allocation subproblem. The former is constructed as Stackelberg privacy game between an edge node and a potential attacker on the CS-server with adversarial prior knowledge to run inference attack, where distortion and differential privacy constraints are considered. A genetic algorithm is applied to generate an appropriate initial value for location obfuscation. For the latter, a linear programming is built with objective of maximizing task acceptance rate with the obfuscated locations.
3. Through extensive simulations that P2TA outperforms typical task allocation mechanism in terms of privacy protection level and task allocation ef-

ficiency. In particular, our results indicate that when inference error is 1km and differential privacy budget is 0.3, the task acceptance rate reaches its maximum with an appropriate privacy level.

The remainder of this paper is organized as follows. In the next section, we briefly introduce the related works, followed by our motivation in Section 3. Section 4 gives the edge computing assisted system framework. Section 5 introduces key performance metrics and formulates the optimal task allocation problem. Section 6 decomposes the optimal problem into two sub-problems to be solved, followed by performance evaluation in Section 7. Concluding remarks and the research prospect are illustrated at the end.

2. Related Works

While some of research works support privacy-preserving task allocation, they may not applicable in an actual scene because of the lack of consideration of user travel distance. Spatial cloaking (e.g. [18, 19, 20]) is a widely used strategy for protecting location privacy. The main drawback is that tasks are likely to be rejected because of long travel distance. Shokri *et al*. [21] propose to generalize the precise locations of users into a confused region that protects location privacy. As a matter of fact, generalizing area to allocate task is the same as random allocation. While Haze [22] can provide k-anonymous guarantee with statistical information, its task allocation efficiency is limited in precision. In [23], proxy reencryption and BBS+ signature are introduced to prevent privacy leakage. Despite the implementation of anonymous submission, it is possible to make an incentive mechanism difficult to run. LORR [24] is a 2-stage and user-controlled obfuscation scheme, which can improves both privacy gain and objective recommendation quality.

Relatively little research considers task acceptance rate in presence of privacy constraints. Wang *et al.* [25] proposes a user-centric obfuscation method for task allocation, where travel distance and privacy constraints are taken into account. iCrowd [26] is a generic task allocation framework with energy-efficient piggyback, in consideration of different incentive and coverage constraints. The work in [27] addresses the multi-task allocation problem with task-specific minimal sensing quality thresholds, aimed at assigning an appropriate set of tasks to each participant. The work in [25] increases user acceptance rate by minimizing expected overall travel distance, but less overall travel distance is not equivalent to a high task allocation efficiency in some scenarios. Wang *et al.* [28] propose a probabilistic winner selection mechanism (PWSM) to reduce the total travel distance

based on obfuscated locations of users. Because it assigns each task with the user with the largest probability of being closest to it, it is worthwhile to analyze how to balance the acceptance rate and distribution rate.

Few studies have focused on edge/fog computing assisted MSC task allocation. In [29] fog servers are used to assist in generating bus route without the exposure of users travel plans. The authors in [30] propose a road surface condition monitoring system based on vehicular crowdsensing, where a certificateless aggregate signcryption scheme is designed for privacy protection. Fo-SDD [31] uses edge nodes to assist task allocation. Despite much more accurate and secure task allocation for mobile users, it ignores task allocation rate and impact of different privacy constraints. [32] is a privacy preserving reputation management scheme for edge computing enhanced MCS to deal with malicious participants. The work in [33] presents a distributed agent-based privacy-preserving framework (DADP), which provides a valuable reference for designing a fog/edge assisted privacy-preserving mechanism. Unlike DADP where a user can randomly select one agent and upload the check-in data to the untrusted server with anonymous connection, this work protects user privacy through obfuscation-based privacy game.

After studying related works, we realize that there is no comprehensive study concerning maximizing task acceptance rate while reducing travel distance under multiple privacy constraints against inference attacks in an edge computing assisted environment. This drives us to purpose P2TA.

3. Motivation

We motivate this work through three case studies in this section. We first introduce two cases for analyzing and comparing the problems of user-centric and task-centric task allocation methods. Then, we illustrate the impact of location obfuscation on task allocation efficiency.

3.1. Impact of User-Centric Task Allocation

The introduction of edge computing cuts off the opportunity for the CS-Server to directly capture a user's real location. By performing obfuscation mechanism on edge nodes, a user's privacy requirements can be guaranteed. Because users tend to accept tasks with smaller travel distances, a natural step after obtaining privacy security is to assign the closest task to each user. As a matter of fact, such a user-centric policy may result in certain tasks not being assigned to any one user. Fig. 1 gives an irrational task allocation resulting from the pursuit of user acceptance rate only. In this scenario, User A and User B will be assigned

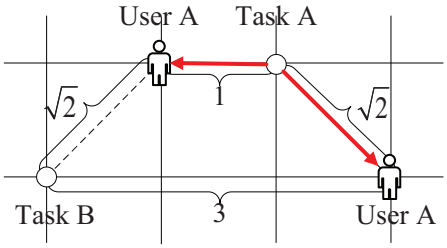


Fig. 1: Case study I (user-centric task allocation).

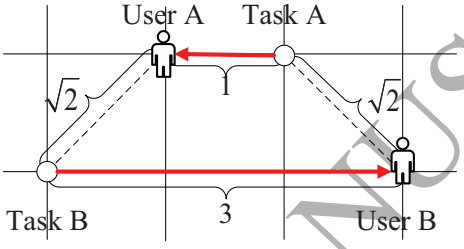


Fig. 2: Case study II (task-centric task allocation).

Task A using the above user-centric policy. Although the distance from User B to Task B is only slightly further than that to Task A, Task B is still not assigned to anyone. In this allocation, the user acceptance rate reaches 100%, but only one task is assigned and the number of accepted tasks is only one. This case prompts us to further study the impact of task allocation rate in the following section.

3.2. Impact of Task-Centric Task Allocation

For comparison, we analyze a task-centric policy that aims to maximize task allocation rate. Take Fig. 2 as an example to illustrate to show whether this policy is beneficial to task allocation. A common step for this case is to choose the closest user for each task. Accordingly, Task A is assigned to User A. The user closest to Task B is still User A. Since each user can only assign one task, a straightforward method is to assign the next closest User B to Task B. However, the distance between User B and Task B is 3, which is likely to cause the task to be rejected. This case has a total of 2 tasks assigned, but only 1 task may be accepted by the user. If Task A is assigned to User B and Task B is assigned to User A, both tasks are likely to be accepted. These two case show that neither the maximum user acceptance rate nor the maximum task allocation rate can achieve the desired

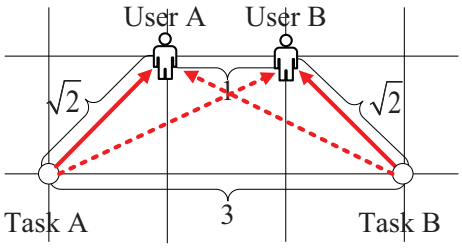


Fig. 3: Case study III (task allocation after obfuscation).

task assignment effect. With this in mind, to design a more reasonable allocation metric is necessary.

3.3. Impact of Location Obfuscation

Users are very likely to refuse to accept tasks if their privacy requirements are not met especially when the CS-server continuously asks participants to upload location data. However, there is a certain mutual restriction between providing privacy guarantees and reducing travel distance. A way to design a location obfuscation mechanism from the perspective of individual privacy (similar to the method in LBSs) is likely to degrade task allocation efficiency. Take Fig. 3 as an example. Without location obfuscation, system will assign Task A (B) to User A (B) according to travel distances (see solid arrows). After obfuscation, system may assign Task A (B) to User B (A) that is further away from the target (according to each users perturbed location) than to User A (B) whose actual location is closer to the target (see dotted arrows). Because a large number of users and tasks coexist, it is very difficult to optimally incorporate obfuscation mechanism into task allocation. This requires us to consider the problem of how to properly decompose the problem, aimed at jointly reducing unnecessary travel distance and maintaining high privacy level.

4. System Overview

In this section, we first define all entities in our system from a real-world perspective, and then describe the connections and interaction processes between different entities.

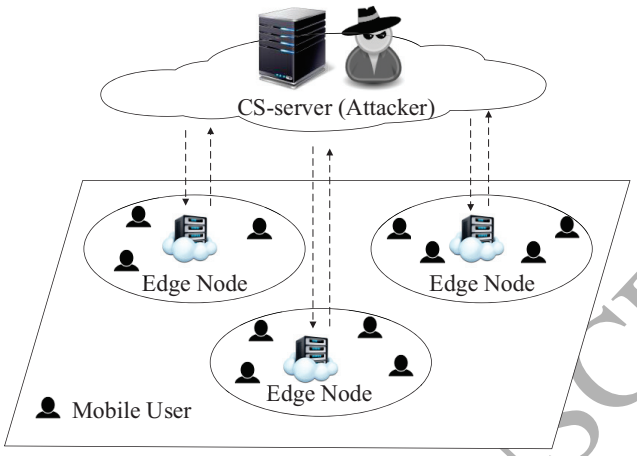


Fig. 4: System Composition.

4.1. Composition

Our system consists of three entities: the CS-server (potential attacker), edge nodes, and mobile users, as shown in Fig. 4. The roles played by different entities are summarized blew.

- **CS-Server**: The CS-Server is an MCS platform operated by a private company (such as the Amazon Mechanical Turk). Unlike the mode that interacts directly with users, the CS-Server in our framework releases tasks to edge nodes that cover the task places. It is assumed that the CS-server is untrusted, who wants to gain access to users' private data.

- **Edge Node**: An edge node exists as both an anonymous server and a task assignment agent. On the one hand, it is in charge of aggregating received locations and obfuscating the locations locally. On the other hand, it is responsible for task allocation and submitting task acceptance result along with obfuscated locations to the CS-server. In this mode, the CS-server cannot peek into original user locations and only observe the obfuscated user locations. In addition, users often switch between different edge nodes during task completion, which spreads possible privacy leakage risks.

- **Users**: Users (workers) As users take privacy seriously, they don't trust the CS-server and think of it as an attacker. Because of security concerns, users

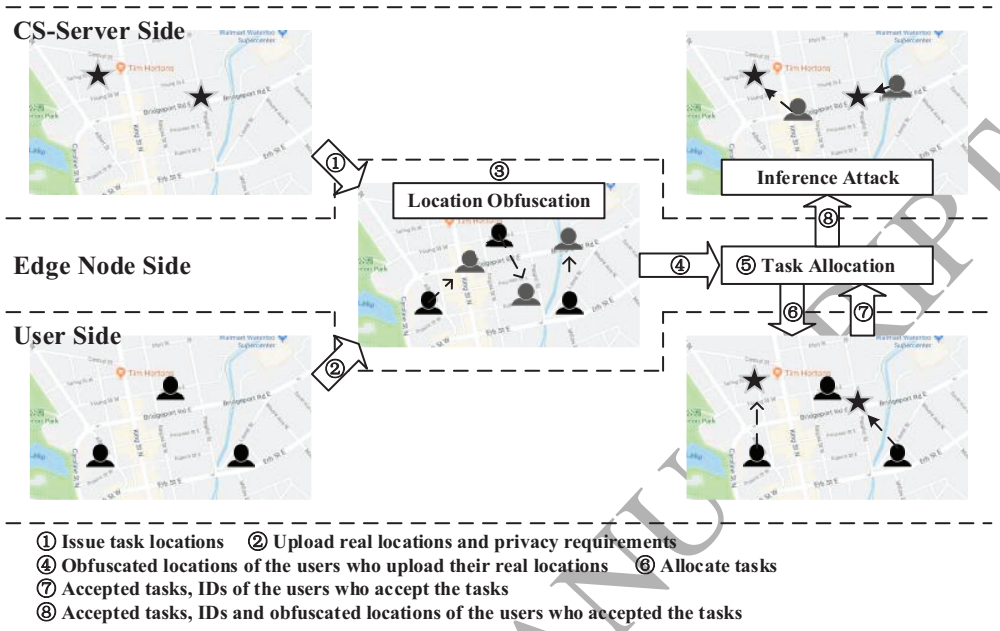


Fig. 5: Running process of privacy-preserving task allocation

uploads their real locations and privacy requirements to edge nodes, instead of the CS-server. Upon receiving task information, they choose to accept or reject it. After accepting tasks, users will inform edge nodes of their IDs along with the accepted tasks. Then, they go to the task place and uses his or her sensing device to collect specific data.

4.2. Workflow

Suppose that the CS-server having various sensing tasks in a certain city which needs to recruit users to conduct. The running process of the whole system can be divided into the following steps (which correspond to the serial number in Fig. 5):

1. **Task location publication on the CS-server side**: The CS-server issues application-specific task locations to edge nodes that cover the task places.
2. **Data submission on the user side**: Users (workers) within the coverage area of an edge node submit their true locations and privacy requirements to the edge node side.
3. **Location obfuscation on the edge node side**: After receiving the task locations from the CS-server, the location obfuscation module on the edge node

side will be activated to collect user data within its coverage area. After obtaining users' real locations, the edge node side performs obfuscation operation that involves in the location obfuscation module. In the case, three black (gray) portraits are the original (perturbed) locations.

4. **Obfuscated location output on the edge node side**: The obfuscated locations of all users are input to a task allocation module.
5. **Task allocation on the edge node side**: The task allocation module assigns tasks to proper users based on the obfuscated locations it receives.
6. **Response on the user side**: After receiving task place information, users who are willing to accept the tasks (go to the task places to complete the tasks) will inform the edge node side of their IDs. In the case diagram we gave, two of the three users agree to accept the tasks.
7. **Feedback on the edge node side**: After collecting the responses from users, the edge node submits the obfuscated locations of the users accepting the task from two of the three users to the server along with the users IDs and the accepted task information. As shown in the case diagram, the edge node side only submits the obfuscated locations for two of the three users accepting the tasks to the CS-server.
8. **Inference attack on the CS-server side**: Since the location obfuscation operation runs on the edge node side, the CS-server does not know users real locations and can only observe the obfuscated locations submitted. With certain prior knowledge, a malicious observer on the CS-server side can run inference attacks optimally tailored against the obfuscation function to minimize inference error.

It should be noted that as for the edge node side, the reason for assigning tasks based on obfuscated locations (rather than real locations) is to prevent real locations from being inferred by the attacker once the task allocation rules are exposed. The specific details will be introduced later.

5. Optimal Task Assignment Problem

As the focus is on location obfuscation and task allocation mechanisms in which the options are decided independently on by each edge node user without knowledge about other edge nodes in the system, we limit our model and analysis in a single edge node in the remainder of the paper, without loss of generality.

In this section, we first describe the problem to be solved. Then we explain the implementation details of related mechanisms based on a probabilistic framework.

Table 1: Main Notations and Variables

Symbols	Definition
η	task acceptance rate
α	user acceptance rate
β	task allocation rate
A	number of task accepted
R_u	set of possible user locations
r	user's actual location
R	set of possible obfuscation output
r'	obfuscation output of r by an edge node
$\hat{r}$	attacker's estimate of actual location r
R_t	set of task locations
r_t	location of task to be allocated
d_u	user's maximum acceptable travel distance
$\pi(r)$	probability distribution over values of r
$p(r'\|r)$	probability of replacing r with r'
$q(\hat{r}\|r')$	probability of estimating $\hat{r}$ as true location with r'
$d(r, r_t)$	travel distance between r and r_t
$d(r, \hat{r})$	estimation error between r and $\hat{r}$
$x(r_t\|r')$	probability of allocating a task at r_t to a user at r'

Finally, we mathematically formulate the proposed problem. The main notations and variables used are listed in Table 1.

5.1. Problem Statement

Consider a scenario where users move in a urban area that is uniformly partitioned into discrete regions which represent the locations where users may stay or receive/perform tasks. Each region represents a location of minimum particle size. When the region size is small enough, the precision requirement of task assignment can be satisfied. It is assumed that each region is covered by one edge node that performs obfuscation-based privacy protection mechanism.

To better understand our problem, we give the following definitions.

Definition 1 (Task Acceptance Rate)**.** *Task acceptance rate η is the proportion of accepted task out of the total task number for each task assignment, defined as*

$$\eta = \frac{A}{T} \tag{1}$$

where A is the number of accepted tasks and T is the total number of tasks determined by MCS perception requirements.

Definition 2 (User Acceptance Rate)**.** *User acceptance rate α is the ratio of the number of users that accept the assigned task to the total number of users for each task assignment, defined as*

$$\alpha = \frac{X}{U} \tag{2}$$

where X is the number of users who accept the allocated tasks and U is the total number of users.

Definition 3 (Task Allocation Rate)**.** *Task allocation rate β is the proportion of tasks assigned to at least one user in the total number of tasks for each task assignment, defined as*

$$\beta = \frac{C}{T} \tag{3}$$

where C is the number of tasks assigned to at least one user and T is the total number of tasks.

Under the constraints of privacy and location perturbation (the metrics of which will be introduced later), the effectiveness of task allocation depends largely on how many tasks are accepted. Let a_t be 1 if a task at target location r_t can be accepted by one user, otherwise be 0, where $r_t \in R_t$. Our goal can be expressed as maximizing

$$A = \sum_{r_t \in R_t} a_t \tag{4}$$

which determines the user acceptance rate that can be achieved.

Fig. 6 shows the dependencies among different metrics to be considered to achieve the above goal, where the arrow direction indicates that one indicator has an effect on the other. For instance, η is jointly determined by α and β. Looking back in the direction of the arrow, we can see that there are two ways to improve α. The first is to protect users' privacy through obfuscation mechanism p, such that the error between estimated location $\hat{r}$ and real region r is ensured under attack function q. The other is to reduce travel distance between r and assigned task location r_t, in which task allocation function x constrains the lower bound of β.

5.2. Location Obfuscation

As a precondition for task acceptance, users want to protect their location information in presence of an untrusted CS-server. After collecting users' locations,

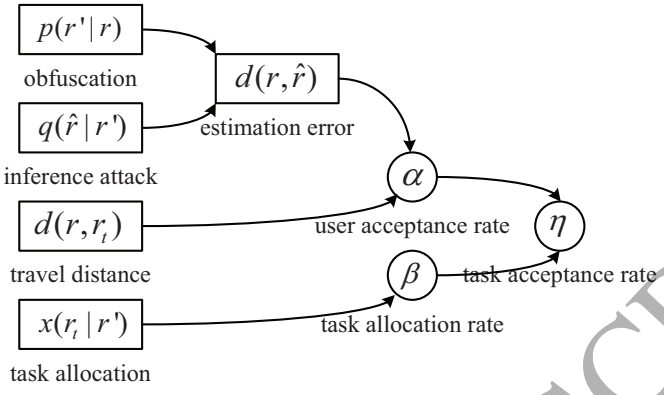


Fig. 6: Dependencies in task assignment problem.

each edge node produces a general obfuscation function that perturbs each true user location $r \in R_u$ by replacing it with an obfuscated location $r' \in R$. Instead of the actual location, the obfuscated location will eventually be uploaded to the CS-server. Given a real location r, the observable r' is chosen according to the following probability distribution.

$$p\left(r' \,|r\right) = \Pr\{R = r' \,|R_u = r\} \tag{5}$$

This function is not only the key to satisfy different privacy requirements, but also partially determines the effect of task allocation.

5.3. Inference Attack

We assume that the attacker aims at inferring a user's real locations that can minimize a user's privacy by observing the output of obfuscation function. Given any observation r', the estimate of true location r is determined by the following probability distribution function over all possible results involved in R_u.

$$q\left(\hat{r} \,|r'\right) = \Pr\left\{R_u = \hat{r} \,|R = r'\right\} \tag{6}$$

It is also assumed that the attacker knows the logic of obfuscation algorithm performed on edge nodes, and he also knows the profile of user location (denoted by $\pi(\cdot)$). When the attacker form the posterior location distribution for a user, it

can perform Bayesian inference attack (similar to [34]) to infer r with r' by

$$q\left(\hat{r}\,|r'\right) = \frac{p\left(r'\,|\hat{r}\right)\cdot\pi\left(\hat{r}\right)}{\sum\limits_{r\in R}\pi\left(r\right)\cdot p\left(r'\,|r\right)} \tag{7}$$

This function estimates a real location r by inverting a given obfuscation mechanism p when r' is released/observed. The estimation error of $\hat{r}$ to r reflects the effectiveness of inference attack, which will be explained latter. Confronted with such an inference attack, a natural step for obfuscation mechanism is to reorient probability distribution p to ensure user's privacy. The two sides form a mutual constraint.

5.4. Privacy Constraints

Differential privacy (reflecting indistinguishability [35]) and distortion privacy (reflecting estimation error [36]) are introduced to measure user privacy gain against posterior inference.

The basic idea behind differential privacy is that suppose the obfuscated location is r', for any two locations r_1 and r_2, their probability of being mapped to r' are similar. For a user u at r', the CS-server cannot distinguish whether a user is at r_1 or r_2 even if the CS-server knows the internal implementation of p. It formally shows such similarity between any two locations r_1 and r_2 for arbitrary r'.

Let ε denote privacy budget associated with the minimum desired privacy of a user. The smaller the value of ε, the higher the privacy. Here we use a generic differential privacy metric. An obfuscation mechanism satisfies ε-differential-privacy, if inequality

$$p\left(r'\,|r_1\right) \leq e^{\varepsilon\cdot d(r_1,r_2)}\cdot p\left(r'\,|r_2\right) \tag{8}$$

holds for all locations $r, r' \in S$ (where $d(r_1, r_2)$ is the distance between r_1 and r_2 reflecting the intuition that if r_1 and r_2 are close to each other), they should be more indistinguishable.

After r' is observed, the attacker can get an estimate value $\hat{r}$ about original location r through inference function q. The distance between r and $\hat{r}$, i.e. $d(r, \hat{r})$, is used to quantify the inference error (referred as distortion) between r and $\hat{r}$ which reflects the attacker's inference error of an obfuscated region. Specifically, the greater the distortion privacy, the lower the attack effect of q, and the lower users worry about $r \sim p(r'|r)$ becoming exposed. The user's distortion privacy metric through obfuscation function p, with respect to a given inference function

q, for a specific location r can be calculated by

$$\sum_{r' \in R} p\left(r' \,|r\right) \sum_{\hat{r} \in R} q\left(\hat{r} \,|r'\right) \cdot d\left(r, \hat{r}\right) \tag{9}$$

With prior leakage $\pi(r)$ which reflects the priori exposed information about r, the expected distortion privacy can be computed by averaging all possible locations as

$$\sum_{r \in R} \pi\left(r\right) \sum_{r' \in R} p\left(r' \,|r\right) \sum_{\hat{r} \in R} q\left(\hat{r} \,|r'\right) \cdot d\left(r, \hat{r}\right) \tag{10}$$

which represents the expected inference error between r and $\hat{r}$. The decision-making of obfuscation mechanism based on distortion and differential privacy can limit possible privacy leaks.

5.5. Task Allocation Rule

Denoted by $d(r, r_t)$ the travel distance between a user's real region r and his or her assigned task location r_t. If travel distance is too long for a user, he or she will probably be unwilling to conduct the task. For an MCS organizer, long travel distance may lead to unsatisfactory conditions such as high incentive to pay and large sensing delay. Consequently, travel distance is inversely proportional to the user acceptance rate, inspired by which the user acceptance rate (defined in (2)) can be computed as

$$\alpha = \frac{k}{d\left(r, r_t\right)} \tag{11}$$

where k is a constant to reflect the relationship between user acceptance rate and travel distance. Because participants in different regions have different sensitivity to travel distance, the value of k can be set via an analysis of user profile.

If obfuscated region r' is released, the assignment of a task toward target place r_t follows probability distribution function x.

$$x\left(r_t \,|r'\right) = \Pr\left\{R_t = r_t \,|R_u = r'\right\} \tag{12}$$

where the principle is to ensure that the CS-server cannot infer a user's real location through x. The upper limit of η is determined by β. A natural way is to improve β by limiting x as

$$\sum_{r' \in R} x\left(r_t \,|r'\right) \cdot U \geq 1, r_t \in R_t \tag{13}$$

such that the condition that each task is allocated to at least one user can be guaranteed.

Depending on allocation probability function x, the expected travel distance of a user at location r can be computed before task assignment by

$$\sum_{r' \in R} \sum_{r_t \in R} p\left(r' \,|r\,\right) \cdot x\left(r_t \,|r'\,\right) \cdot d\left(r, r_t\right) \tag{14}$$

This metric will be used to measure whether the user can accept the given task.

Intuitively, because the CS-server cannot acquire users' real locations, an edge node seems to be able to assign tasks to users according to their real locations, and then submit the obfuscated location to the CS-server. This will not only improve the efficiency of task assignment but also protects user privacy. If we follow this rule, the task allocation function will change from $x(r_t|r')$ to $x(r_t|r)$. The attacker can infer original location r as long as he knows the implementation of $x(r_t|r)$. In view of this situation, an edge node needs to perform task assignment with the locations after treatment.

5.6. Problem Formulation

The optimal task allocation aims to maximize η, equivalent to maximizing the number of accepted tasks. The privacy protection effect with obfuscation function p is constrained by differential privacy, while the accuracy of inference attack with function q is measured by distortion privacy.

The travel travel (cost) of completing a task is taken into account. Let d_u denote user's maximum acceptable travel distance, i.e. the task will be accepted as long as $d(r, r_t) \leq d_u$. Worth noting that if the target is assigned by merely minimizing $d(r, r_t)$, not only will the user's privacy be leaked, but also the task that is easy to complete is assigned multiple times or the task that is difficult to complete cannot be assigned. From a global perspective, we use regional obfuscation function p and assignment probability function x to calculate expected travel distance (defined in (14)) to more objectively reflect the user's expectation of whether the task is acceptable. Let random variable a_r be 1 if the task is expected to be accepted, otherwise be 0. Accordingly, the joint location obfuscation and task allocation (LOTA) problem can be mathematically formalized as:

$$\underset{p,x}{\text{Maximize}} : A = \sum_{r_t \in R_t} a_t \tag{15}$$

Subject to:

$$\sum_{r\in R_u}\sum_{r'\in R} p\left(r'\,|r\right)\cdot x\left(r_t\,|r'\right)\cdot d(r,r_t)\le d_u+\xi\cdot a_t \tag{16}$$

$$\sum_{r\in R_u}\sum_{r'\in R} p\left(r'\,|r\right)\cdot x\left(r_t\,|r'\right)\cdot d(r,r_t)\ge d_u-\xi\cdot(1-a_t) \tag{17}$$

$$a_t\in\{0,1\},\ \forall r_t\in R_t \tag{18}$$

$$p\left(r'\,|r\right)\ge 0,\sum_{r'\in R} p\left(r'\,|r\right)=1,\forall r\in R_u,\forall r'\in R \tag{19}$$

$$p\left(r'\,|r_1\right)\le e^{\varepsilon\cdot d(r_1,r_2)}\cdot p\left(r'\,|r_2\right),\forall r_1,r_2\in R_u,\forall r'\in R \tag{20}$$

$$\sum_{r\in R_u}\pi\left(r\right)\cdot p\left(r'\,|r\right)=\pi\left(r'\right) \tag{21}$$

$$q\left(\hat{r}\,|r'\right)\ge 0,\sum_{\hat{r}\in R} q\left(\hat{r}\,|r'\right)=1,\forall r',\hat{r}\in R \tag{22}$$

$$\sum_{r'\in R} p\left(r'\,|r\right)\sum_{\hat{r}\in R} q\left(\hat{r}\,|r'\right)\cdot d\left(r,\hat{r}\right)\ge d_m \tag{23}$$

$$x\left(r_t\,|r'\right)\ge 0,\sum_{r_t\in R_t} x\left(r_t\,|r'\right)=1,\forall r_t\in R_t,\forall r'\in R \tag{24}$$

$$\sum_{r'\in R} x\left(r_t\,|r'\right)\cdot U\ge 1,\forall r_t\in R_t \tag{25}$$

Before task allocation, an edge node collects all user locations (stored in set R_u) and receives task locations (stored in set R_t) from the CS-server. Then, the edge node attempts to maximize objective (15) i.e. the number of tasks accepted while satisfying constraints. The essence of this problem is to find the solution of obfuscation and task allocation, both of which are mutual-depend as a whole.

By setting constraints (16) and (17), we establish an association between 0-1 variable a_t and decisions, which can reflect the impact of all possible decisions on the objective. The sufficient large constant ξ in these two constraints ensures that a_t is

$$\begin{cases} 1, \sum\limits_{r\in R_u}\sum\limits_{r'\in R} p(r'|r)\cdot x(r_t|r')\cdot d(r_t,r)\ge d_u \\ 0, \sum\limits_{r\in R_u}\sum\limits_{r'\in R} p(r'|r)\cdot x(r_t|r')\cdot d(r_t,r)\le d_u \end{cases}$$

With this rule, the number of task accepted can be updated after each decision, while ensuring that travel distances are within acceptable limits for the user.

Constraint (18) limits a_t to be 0 or 1. Constraint (19) states probability distributions for obfuscation function p. Constraint (20) guarantees user's differential privacy. Constraint (21) aims to limit location perturbation, ensuring that P2TA does not change the overall region distribution of users; in other words, the user location distribution after confusion of p is the same as the original user location distribution. Similar to (19), constraint (22) ensures that inference function q is proper. Constraint (23) guarantees a user's distortion privacy, where d_m is the threshold a user can tolerate. Constraint (24) states probability distribution represented by function x for task assignment. Constraint (25) ensures that any task is assigned at least once.

The LOTA problem consists of two intertwined subproblem regarding privacy protection and task allocation. When the number of participants and the number of tasks are large, it difficult to quickly find optimal solution. If we perform enumeration, the computational complexity of enumerating all possible p, q and x will at least reach $O(n^4)$, where n is the number of regions in which the coverage area of an edge node is divided. To improve practicality, a natural step is to decompose the LOTA problem and solve the different functions separately.

6. Solution: Problem Decomposition

We decompose the LOTA problem to get a suboptimal solution. We first extract location obfuscation subproblem against inference attack (associated with p and q) from the LOTA problem and model it as a privacy game problem with consideration of better task allocation. We then solve the task allocation subproblem (associated with x) according to the obfuscated locations.

6.1. Location Obfuscation Subproblem

Because p and q are with opposite objectives, the location obfuscation subproblem can be formalized as a privacy game between the defender (on the edge node side) and the attacker (on the CS-server side) to quantify privacy-preserving effect. A genetic algorithm based policy is introduced to get a suboptimal solution for location obfuscation.

6.1.1. Privacy Game Model

The privacy-preserving issue can be regarded as a kind of Stackelberg game (as in [37, 38]) where one player, an edge node, commits to a strategy p^* first,

and the other players, the CS-server, selfishly chooses its best response strategy q^* against strategy p^*. The goal of such a game is to find the pair of the best strategies p^* and q^* mutually optimal against each other.

For any user location $r \in R_u$, the decision space is the set of observables R. For any observable $r' \in R$, the strategic space of the attacker is all possible attackers estimates $\hat{s} \in S$. A mixed policy is designed for a given location $r \in R_u$ via a vector $\{p(r_1'|r_i), p(r_2'|r_i), ..., p(r_n'|r_i)\}$, where $\{r_1', r_2', ..., r_n'\} = R$. Correspondingly, for a given observable $r' \in R$, there is a mixed policy for the attacker expressed by a vector $\{q(\hat{r}_1|r_i'), q(\hat{r}_2|r_i'), ..., q(\hat{r}_n|r_i')\}$, where $\{\hat{r}_1, \hat{r}_2, ..., \hat{r}_n\} = R$. Both vectors represent the conditional distribution functions associated with an location obfuscation function for a true location r and an inference attack function for an observable r'. Let P denote the sets of all mixed policies of an edge node for a user. The decision space about obfuscation probability distribution p is expressed as

$$\begin{aligned} P &= \{p(r_1'|r_i), p(r_2'|r_i), ..., p(r_n'|r_i)\}, \\ p(r'|r) &\geq 0, \sum_{r' \in R} p(r'|r) = 1, \forall r \in R_u, \forall r' \in R \end{aligned} \tag{26}$$

Let Q denote the sets of all mixed policies of the attacker, whose decision space about inference probability distribution q is defined as

$$\begin{aligned} Q &= \{q(\hat{r}_1|r_i'), q(\hat{r}_2|r_i'), ..., q(\hat{r}_n|r_i')\}, \\ q(\hat{r}|r') &\geq 0, \sum_{\hat{r} \in R} q(\hat{r}|r') = 1, \forall r', \hat{r} \in R \end{aligned} \tag{27}$$

The optimal strategy for the CS-server is to minimize error of the adversary'inference attack, to which the optimization problem corresponds can be formulated as

$$q^* = \arg\max_{q} \sum_{r, r', \hat{r} \in R} \pi(r) \cdot p^*(r'|r) \cdot q(\hat{r}|r') \cdot d(r, \hat{r}) \tag{28}$$

Subject to:

$$q(\hat{r}|r') \geq 0, \sum_{\hat{r} \in R} q(\hat{r}|r') = 1, \forall r', \hat{r} \in R \tag{29}$$

$$\sum_{r', \hat{r} \in R} \pi(r) \cdot p^*(r'|r) \cdot q(\hat{r}|r') \cdot d(r, \hat{r}) \geq d_m \tag{30}$$

The optimal strategy for an edge node is to maximize the expected inference error of the CS-server. This corresponds to the following formulas

$$p^* = \arg\max_{p} \sum_{r' \in R} p\left(r' \,|r\right) \sum_{\hat{r} \in R} q^*\left(\hat{r} \,|r'\right) \cdot d\left(r, \hat{r}\right) \tag{31}$$

Subject to:

$$p\left(r' \,|r\right) \geq 0, \sum_{r' \in R} p\left(r' \,|r\right) = 1, \forall r \in R_u, \forall r' \in R \tag{32}$$

$$p\left(r' \,|r_1\right) \leq e^{\varepsilon d(r_1, r_2)} p\left(r' \,|r_2\right), \forall r_1, r_2 \in R_u, \forall r' \in R \tag{33}$$

$$\sum_{r \in R_u} \pi\left(r\right) \cdot p\left(r' \,|r\right) = \pi\left(r'\right) \tag{34}$$

$$\sum_{r', \hat{r} \in R} \pi\left(r\right) \cdot p\left(r' \,|r\right) \cdot q^*\left(\hat{r} \,|r'\right) \cdot d\left(r, \hat{r}\right) \geq d_m \tag{35}$$

If we enumerate all possible combinations of p and q, the computational complexity will reach $O(n^3)$. In a real-world scenario, the value of n tends to be larger, which leads to higher complexity. An iteration algorithm is designed to quickly approximate the optimal solution. The basic idea is that the solution to p (or q) can be seen as the input of q (or p), and the p and q are alternatively solved until convergence (or the iteration times exceed a given threshold). Next, we explain the implementation details of the algorithm.

6.1.2. GA based Initialization

To start the iteration of solving p and q, an initial p, denoted as p_0, is needed to be set. The use of iteration algorithm often leads to a local optimal solution, where the selection of p_0 affects how good the local optimal solution can achieve.

In order to make the game not fall into local convergence early, Genetic Algorithm (GA) [39] is introduced for selecting an initial value. The key idea behind GA is to generate a potential solution for utility testing from existing solutions by using either Mutation or Crossover methods under a given probability. The Mutation and Crossover processes are designed for solving p&q-subproblem, mainly consisting of two steps as follows.

Mutation maintains diversity within a population and prevents premature convergence. Take Fig. 7 as an example of a randomly generated regional confusion matrix p_0 mutated into p_1. Given a pre-obtained p_0, a location pair (r_1,r_2) belongs to $\{(r'$,$r)|r', r \in R\}$. A new p_1 can be constructed by setting $p_1(r_1, r_2)$ to

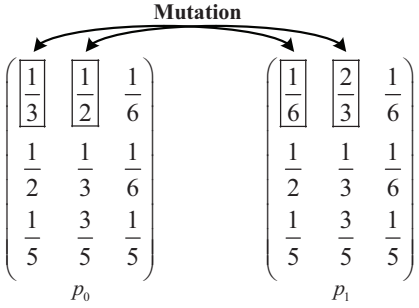


Fig. 7: Mutation example.

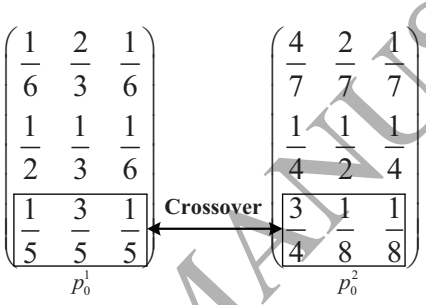


Fig. 8: Crossover example.

$p_0(r_1, r_2)/2$ and by setting $p_1(r_3, r_2)$ to $p_0(r_3 r_2) + p_0(r_1, r_2)/2$. The remaining values are consistent with the values of the same position in p_0.

Crossover selects a pair of matched parents and then randomly selects a set of intersections to exchange genes to produce offspring. We present a case in Fig. 8 where the genes (corresponding to one row of the matrix) of the two randomly generated location confusion matrices p_0^1 and p_0^2 perform such an operation. Given the parents p_0^1 and p_0^2, the crossover function is used to generate two children p_1^1 and p_1^2 by row exchange. More specifically, an edge node randomly selects a location r and then sets p_1^1 (:, r) to p_0^2 (:, r) and p_1^2 (:, r) to p_0^1 (:, r); for the rest values, p_1^1 is set to p_0^1 and p_1^2 is set to p_0^2.

The aim about crossover and mutation is to adjust the probability that an obfuscated location r is assigned to target task location r_t. The sum of the probability of each row is equal to one after the transformation. Different from [25] which uses the GA to adjust the number of tasks assigned, we use the same way to achieve a different goal, i.e., finding $p \in P$ and $q \in Q$ that build an equilibrium point

between the defender and the attacker in such a privacy game.

6.2. Task Allocation Subproblem

After generating the obfuscation function, an edge node next needs to actually allocate tasks in accordance with the obfuscated locations. We refer to the follow-up problem as task allocation decision subproblem aimed at maximizing task acceptance rate. Since only task allocation function x needs to be solved, subsequent calculations are greatly simplified, which corresponds to the following linear program.

$$\underset{x}{\text{Maximize}}\, A = \sum_{r_t \in R_t} a_t \tag{36}$$

Subject to:

$$\sum_{r \in R_u} \sum_{r' \in R} p^* \left(r' \,|r\right) x \left(r_t \,|r'\right) d \left(r_t \,|r\right) \leq d_u + \xi \cdot a_t \tag{37}$$

$$\sum_{r \in R_u} \sum_{r' \in R} p^* \left(r' \,|r\right) x \left(r_t \,|r'\right) d \left(r_t \,|r\right) \geq d_u - \xi \cdot (1 - a_t) \tag{38}$$

$$a_t \in \{0, 1\}, \ \forall r_t \in R_t \tag{39}$$

$$\sum_{r' \in R} x \left(r_t \,|r'\right) \cdot U \geq 1, \forall r_t \in R_t \tag{40}$$

$$x \left(r_t \,|r'\right) \geq 0, \sum_{r_t \in R_t} x \left(r_t \,|r'\right) = 1, \forall r_t \in R_t, \forall r' \in R \tag{41}$$

The above subproblem can be transformed into a simple integer programming problem, where x becomes the only one that needs to be solved. It can be efficiently solved with off-the-shelf linear optimization software. After tasks are assigned according to function x, an edge node collects the IDs of users that accepts the tasks, and then submits it to the CS-server along with the their obfuscated locations. Even if the attacker knows the implementation of task assignment function, he cannot directly guess the true location of a user.

7. Performance Evaluation

This section conducts performance analysis and evaluation of our proposed framework through simulation methodology. In our experiments, an edge node

Table 2: Key parameters in simulation.

Notation	Default	Description
n	36	number of regions in edge node coverage
U	30	number of users
T	10	number of tasks to be assigned
ϵ	0.3	differential privacy requirement
d_m	1	distortion privacy requirement
d_u	1.5km	acceptable travel distance
π	uniform	user spatial distribution
τ	uniform	task spatial distribution

covered area is divided into n regions, the set of which corresponds to R. The default parameter settings for the details are given in Table 2.

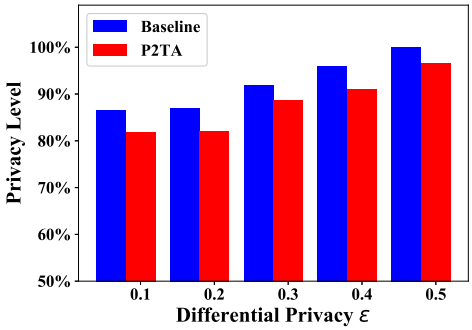
For comparison, a differential obfuscation task allocation mechanism proposed in [25] is chosen as a baseline, where the focus is on minimizing users' total travel distance under differential privacy constraints. A No-Privacy scheme is also introduced as an unconstrained optimal task allocation scheme compared to P2TA and baseline. It is the optimal scheme that uses the real locations for task assignment (equivalent to P2TA without privacy constraints).

Three experiments are designed to study the effect of task assignment from three aspects, including impact of privacy constraints, (acceptable) travel distance and number of participants. In order to improve the accuracy of result data, all the data presented includes the average of multiple random experiments. The performance metrics to be examined includes: 1) privacy level (defined as the proportion of the number of accepted tasks that satisfy privacy constraints among all the accepted tasks), 2) task acceptance rate η, 3) user acceptance rate α and 4) task allocation rate β.

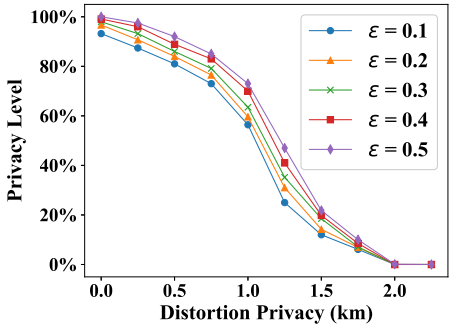
7.1. Impact of Privacy Constraints

Experiment I looks at the effect of privacy constraints on privacy level. The privacy level is defined as the proportion of the number of tasks that satisfy the constraint among all the assigned tasks. We adjust parameters ε and d_m (defined in (8) and (23)) to reflect the change in the rigor of the constraints. The results shown in Fig. 9 are divided into four parts.

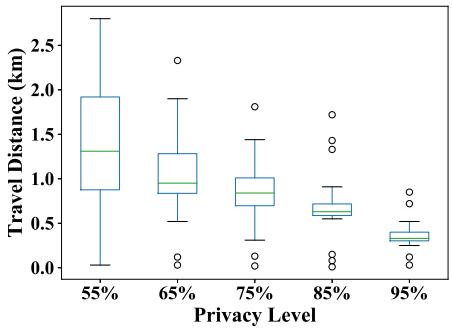
Fig. 9(a) depicts the effect of parameter ε on the average privacy level. It can be seen that the average privacy level of both P2TA and baseline increases with the increase of ε. The higher the value of ε, the higher the degree of distinguishability



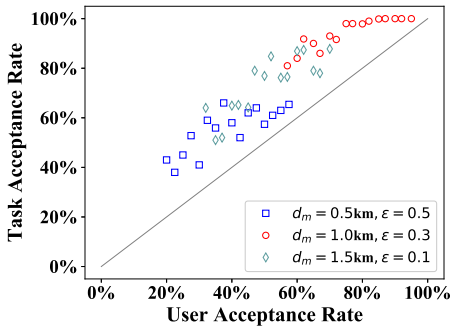
(a) privacy level under differential privacy constraints



(b) privacy level under distortion privacy constraints



(c) travel distance



(d) task acceptance rate

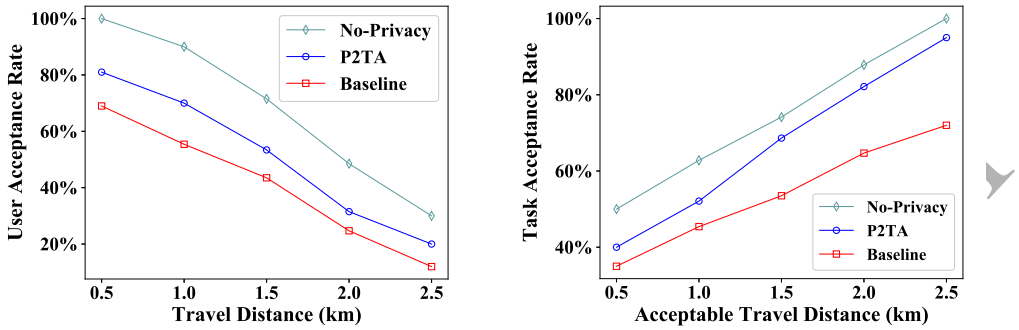
Fig. 9: Impact of Privacy Constraints (Experiment I)

between the user's real locations and the confused locations, and the easier it is to meet the privacy requirement. The baseline's average privacy level is always higher than P2TA because it does not consider distortion privacy; in other words, P2TA's distortion privacy constraint d_m is set to 1km, and such a stricter privacy constraint reduces the proportion of task assignments that satisfy the requirement. Nonetheless, while providing greater privacy protection, P2TA's average privacy level is only about 5% lower than baseline, indicating the adaptability and usability of the mechanism.

The effect of parameter d_m on the average privacy level of P2TA under different ε conditions is shown in Fig. 9(b). The larger the value of d_m, the greater the average privacy level. The reason for this is that the distortion privacy constraint d_m is related to the attacker's expected inference error. The larger the value is, the less accurate the attacker's estimate of a user's real position is, and the higher the privacy protection strength is. In our experiments, the coverage of one edge node is set to 3km $\times$ 3km. Limited by the coverage, the proportion of tasks that can satisfy constraints decreases rapidly especially when d_m is larger than 1km.

In Fig. 9(c) we then show whether different average privacy levels have an impact on the user's travel distance. When privacy level is low, the distribution range of travel distance tends to expand and the average travel distance increases. This is because the lower privacy level (more tasks do not meet privacy constraints) leads to increased randomness of task assignment results. As privacy level increases, the number of abnormal points shows an increasing trend. The main reasons include three aspects. First, because our method reduces the travel distance under the premise of protecting privacy, there is a certain probability that a relatively distant task will be assigned. Second, the optimization goal of this work is not to minimize the travel distance, so an edge node may try to assign some tasks that are further away from users in order to improve the task acceptance rate. Third, the high Privacy level is good for improving the task assignment, and thus the travel distance may be classified as an abnormal point. These factors make the number of abnormal points more frequent when privacy level is high.

Fig. 9(d) compares task acceptance rate η and user acceptance rate α of P2TA under three different strictness levels of privacy constraints. It can be seen that both η and α are maintained at the lowest level under the most relaxed privacy constraints ($d_m = 0.5$ and $\varepsilon = 0.5$). This is because when privacy is threatened, users will refuse to participate in group intelligence perception. Under the most stringent privacy constraints ($d_m = 1.5$ and $= 0.1$), η and α also perform poorly due to strict privacy constraints resulting in lower privacy levels. It is worth noting that when $d_m = 1$ and $\epsilon = 0.3$, η is maintained at 80%–100%, and α is always



(a) user acceptance rate

(b) task acceptance rate

Fig. 10: Impact of distance Factors (Experiment II)

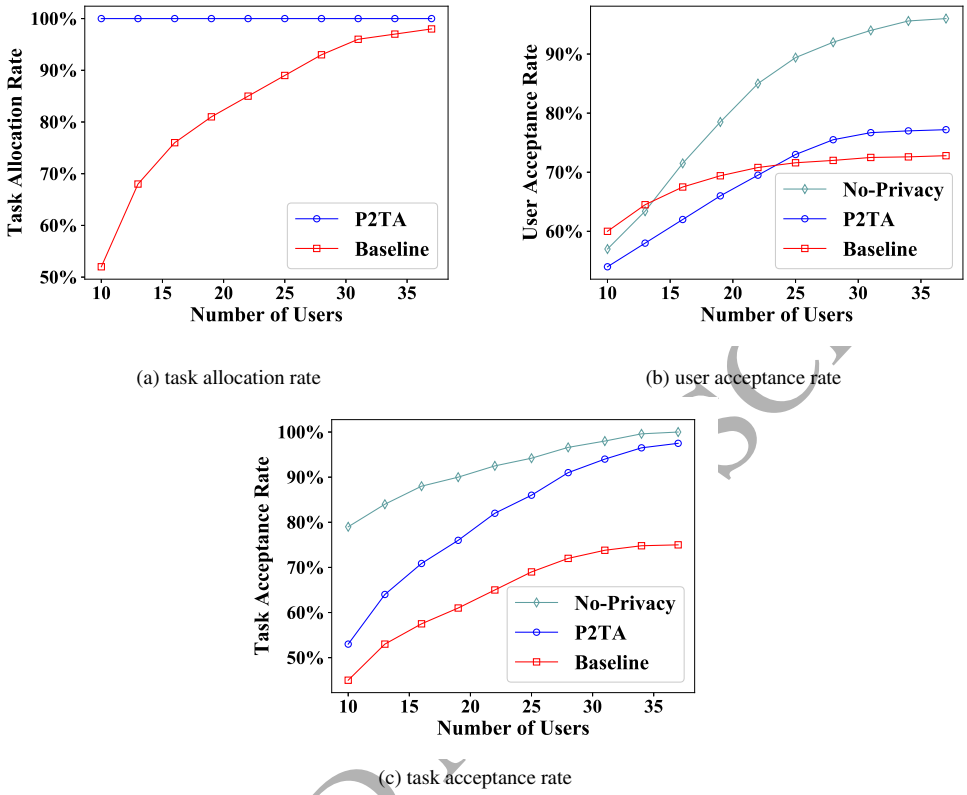
higher than 60%. Such parameter settings are a good balance between privacy protection and task assignment efficiency.

7.2. Impact of Distance Factors

The purpose of next experiment is to analyze the impact on performance exerted by the increase in travel distance (required to complete the task) and acceptable travel distance (from the perspective of a user).

Form Fig. 10(a), as the travel distance increases, user acceptance rate α drops sharply and P2TA always keeps a higher α than baseline. This is because the optimization goal of baseline is minimum overall travel distance which cannot reflect α well in some cases. For instance, there are two users only willing to travel 500m. If it assigns a task with a distance of 600 meters to each of them, no tasks will be accepted. Nevertheless, if a user is assigned a task with a distance of 500m and another user is assigned a task with a distance of 1000m, α will be 50%, even though the overall travel distance is not the minimum.

Unlike the travel distance in Fig. 10(a), the acceptance travel distance in Fig. 10(b) refers to the travel distance acceptable to the user. The longer the acceptable travel distance of a user (the longer the travel distance a user is willing to travel), the greater the task acceptance rate (the more likely he or she is to accept a task). As we can see, there is a rapid growth in task acceptance rate η with the increasing of user's acceptability on travel distance. In particular, η of P2TA is always higher than that of baseline. This is due to the positive correlation between α and η.



(a) task allocation rate

(b) user acceptance rate

(c) task acceptance rate

Fig. 11: Impact of user number (Experiment III)

7.3. Impact of Number of Participants

In Experiment III, the effect of number of participant is explored. The results on η, α and β are shown in Fig. 11.

Fig. 11(a) gives the η with respect to different number of users. As can be concluded from the figure, the β of P2TA remains 100%, which is independent of the number of users. This is because task allocation probability distribution x satisfies a constraint that each task is assigned at least once. Since the number of tasks needs to be assigned remains unchanged, the probability of having an allocable user near each task increases. This makes β achieved by baseline increase gradually and close to P2TA after the number of users reaches 30.

The variation of α in different number of users by each scheme is shown in Fig. 11(b). When the number of users is low, baseline's α is higher than a non-privacy scheme based on P2TA. The reason for this lies in the P2TA's task allo-

cation constraint, which allows P2TA to assign non-nearest tasks to users. The purpose of this allocation strategy is to increase β to improve η. When the number of users increased, α achieved by baseline begins to be lower than that of P2TA. This is because baseline equalizes the overall travel distance to task acceptance rate. When the number of users is large enough, the overall travel distance becomes not able to accurately reflect η.

According to Fig. 11(c), the task acceptance rate under varying number of users with respect to different allocation schemes shows an increase when the user number changes from 10 to 40. P2TA can provide higher η than baseline. It is comparable to No-Privacy scheme after the number of users is exceeds 30. This is because P2TA jointly considers α and β, while baseline only optimizes the overall travel distance. In summary, P2TA can provide a higher task acceptance rate along with better privacy protection effect compared with baseline.

8. Conclusion

A privacy-preserving task allocation (P2TA) framework is designed for edge computing enhanced MCS. The goal is to maximize task acceptance rate while satisfying privacy and location perturbation constraints. While the introduction of edge nodes improve real-time performance of task allocation, it can cut off opportunities for the CS-server to directly obtain users' location data. We solve the joint privacy protection and task allocation problem by means of problem decomposition. A privacy game model is built to optimize defender/attacker objectives against each other to obtain a final obfuscation strategy which can be immune to posterior inference. To address the game problem, a genetic algorithm is performed to choose an initial obfuscation strategy, building upon which an edge node maximizes the number of tasks accepted under constraints of differential privacy and distortion privacy. Simulation results demonstrate that compared with the typical MCS task allocation mechanism, P2TA achieves significant performance improvement of task acceptance rate with higher privacy level.

Acknowledgement

The authors gratefully acknowledge the support and financial assistance provided by the National Natural Science Foundation of China under Grant No. 61502230, 61501224 and 61073197, the Natural Science Foundation of Jiangsu Province under Grant No. BK20150960, the National Key R&D Program of China under Grant No. 2018YFC0808500, the Natural Science Foundation of the

Jiangsu Higher Education Institutions of China under Grant No. 15KJB520015, the Jiangsu provincial government scholarship program, and Nanjing Municipal Science and Technology Plan Project under Grant No. 201608009. The authors thank the anonymous reviewers who provided constructive feedback on earlier pieces of this work appearing at ICA3PP 2018 [40].

References

[1] Jinwei Liu, Haiying Shen, Husnu S. Narman, Wingyan Chung, and Zongfang Lin. A survey of mobile crowdsensing techniques: A critical component for the internet of things. *ACM Transactions on Cyber-Physical Systems*, 2(3):18:1–18:26, 2018.

[2] Qiang Ye and Weihua Zhuang. Distributed and adaptive medium access control for internet-of-things-enabled mobile networks. *IEEE Internet of Things Journal*, 4(2):446–460, 2017.

[3] Liang Liu, Wu Liu, Yu Zheng, Huadong Ma, and Cheng Zhang. Third-eye: A mobilephone-enabled crowdsensing system for air quality monitoring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):20:1–20:26, 2018.

[4] Ruipeng Gao, Bing Zhou, Fan Ye, and Yizhou Wang. Knitter: fast, resilient single-user indoor floor plan construction. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 1–9, 2017.

[5] Moustafa Elhamshary, Moustafa Youssef, Akira Uchiyama, Hirozumi Yamaguchi, and Teruo Higashino. Transitlabel: A crowd-sensing system for automatic labeling of transit stations semantics. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 193–206, 2016.

[6] Bin Guo, Yan Liu, Leye Wang, and Victor O. K. Li. Task allocation in spatial crowdsourcing: Current state and future directions. *IEEE Internet of Things Journal*, 5(3):1749–1764, 2018.

[7] Luis G Jaimes, Idalides J Vergara-Laurens, and Andrew Raij. A survey of incentive techniques for mobile crowd sensing. *IEEE Internet of Things Journal*, 2(5):370–380, 2015.

[8] Bin Guo, Yan Liu, and Wenle Wu. Activecrowd: A framework for optimized multitask allocation in mobile crowdsensing systems. *IEEE Transactions on Human-Machine Systems*, 47(3):392–403, 2017.

[9] Jiangtao Wang, Yasha Wang, Daqing Zhang, Leye Wang, Haoyi Xiong, Abdelsalam Helal, Yuanduo He, and Feng Wang. Fine-grained multitask allocation for participatory sensing with a shared budget. *IEEE Internet of Things Journal*, 3(6):1395–1405, 2016.

[10] Ping Zhao, Jie Li, Fanzi Zeng, Fu Xiao, Chen Wang, and Hongbo Jiang. Illia: Enabling k-anonymity-based privacy preserving against location injection attacks in continuous lbs queries. *IEEE Internet of Things Journal*, 5(2):1033–1042, 2018.

[11] Hang Shen, Guangwei Bai, Mei Yang, and Zhonghui Wang. Protecting trajectory privacy: A user-centric analysis. *Journal of Network and Computer Applications*, 82:128–139, 2017.

[12] Hai Liu, Xinghua Li, Hui Li, Jianfeng Ma, and Xindi Ma. Spatiotemporal correlation-aware dummy-based privacy protection scheme for location-based services. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 1–9, 2017.

[13] Cheng Huang, Rongxing Lu, Xiaodong Lin, and Xuemin Shen. Secure automated valet parking: A privacy-preserving reservation scheme for autonomous vehicles. *IEEE Transactions on Vehicular Technology*, 67(11):11169–11180, 2018.

[14] Jianbing Ni, Kuan Zhang, Yong Yu, Xiaodong Lin, and Xuemin Sherman Shen. Privacy-preserving smart parking navigation supporting efficient driving guidance retrieval. *IEEE Transactions on Vehicular Technology*, 67(7):6504–6517, 2018.

[15] Weisong Shi, Jie Cao, and Quan Zhang. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[16] Miao Du, Kun Wang, Yuanfang Chen, Xiaoyan Wang, and Yanfei Sun. Big data privacy preserving in multi-access edge computing for heterogeneous internet of things. *IEEE Communications Magazine*, 56(8):62–67, 2018.

[17] Jianbing Ni, Aiqing Zhang, Xiaodong Lin, and Xuemin Sherman Shen. Security, privacy, and fairness in fog-based vehicular crowdsensing. *IEEE Communications Magazine*, 55(6):146–152, 2017.

[18] Layla Pournajaf, Li Xiong, and Vaidy Sunderam. Spatial task assignment for crowd sensing with cloaked locations. In *IEEE International Conference on Mobile Data Management (MDM)*, pages 73–82, 2014.

[19] Idalides J. Vergaralaurens, Diego Mendez, and Miguel A. Labrador. Privacy, quality of information, and energy consumption in participatory sensing systems.

In *IEEE International Conference on Pervasive Computing and Communications (PERCOM)*, pages 199–207, 2014.

[20] Shibo He, Dong Hoon Shin, and Junshan Zhang. Toward optimal allocation of location dependent tasks in crowdsensing. In *IEEE International Conference on Computer Communications (INFOCOM)*, pages 745–753, 2014.

[21] Reza Shokri, George Theodorakopoulos, Carmela Troncoso, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. Protecting location privacy: Optimal strategy against localization attacks. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 617–627, 2012.

[22] Joshua W. S. Brown, Olga Ohrimenko, and Roberto Tamassia. Haze: Privacy-preserving real-time traffic statistics. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 540–543, 2013.

[23] Jianbing Ni, Kuan Zhang, Qi Xia, Xiaodong Lin, and Xuemin Shen. Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing. *arXiv preprint arXiv:1806.04057*, 2018.

[24] Thivya Kandappu, Archan Misra, Shih-Fen Cheng, Randy Tandriansyah, and Hoong Chuin Lau. Obfuscation at-source: Privacy in context-aware mobile crowdsourcing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):16:1–16:24, March 2018.

[25] Leye Wang, Dingqi Yang, Xiao Han, Tianben Wang, Daqing Zhang, and Xiaojuan 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)*, pages 627–636, 2017.

[26] Haoyi Xiong, Daqing Zhang, and Guanling Chen. icrowd: Near-optimal task allocation for piggyback crowdsensing. *IEEE Transactions on Mobile Computing*, 15(8):2010–2022, 2016.

[27] Jiangtao Wang and Yasha Wang. Multi-task allocation in mobile crowd sensing with individual task quality assurance. *IEEE Transactions on Mobile Computing*, 17(9):2101–2113, 2018.

[28] Zhibo Wang, Jiahui Hu, Ruizhao Lv, Jian Wei, Qian Wang, Dejun Yang, and Hairong Qi. Personalized privacy-preserving task allocation for mobile crowdsensing. *IEEE Transactions on Mobile Computing*, PP(99):1–1, 2018.

[29] Yuanyuan He, Jianbing Ni, Ben Niu, Fenghua Li, and Xuemin Sherman Shen. Privacy-preserving ride clustering for customized-bus sharing: A fog-assisted approach. In *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 1–8, 2018.

[30] Sultan Basudan, Xiaodong Lin, and Karthik Sankaranarayanan. A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing. *IEEE Internet of Things Journal*, 4(3):772–782, 2017.

[31] Jianbing Ni, Kuan Zhang, Yong Yu, and Xiaodong Lin. Providing task allocation and secure deduplication for mobile crowdsensing via fog computing. *IEEE Transactions on Dependable and Secure Computing*, PP(99):1–1, 2018.

[32] Lichuan Ma, Xuefeng Liu, Qingqi Pei, and Yong Xiang. Privacy-preserving reputation management for edge computing enhanced mobile crowdsensing. *IEEE Transactions on Services Computing*, PP(99):1–1, 2018.

[33] Zhibo Wang, Xiaoyi Pang, Yahong Chen, Huajie Shao, Qian Wang, Libing Wu, Honglong Chen, and Hairong Qi. Privacy-preserving crowd-sourced statistical data publishing with an untrusted server. *IEEE Transactions on Mobile Computing*, PP(99):1–1, 2018.

[34] Xuejun Zhang, Xiaolin Gui, Feng Tian, Si Yu, and Jian An. Privacy quantification model based on the bayes conditional risk in location-based services. *Tsinghua Science and Technology*, 19(5):452–462, 2014.

[35] Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Optimal geo-indistinguishable mechanisms for location privacy. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 251–262, 2014.

[36] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying location privacy. In *IEEE symposium on security and privacy*, pages 247–262, 2011.

[37] Kun Wang, Li Yuan, Toshiaki Miyazaki, Yuanfang Chen, and Yan Zhang. Jamming and eavesdropping defense in green cyber-physical transportation systems using stackelberg game. *IEEE Transactions on Industrial Informatics*, 14(9):4232–4242, 2018.

[38] Reza Shokri. Privacy games: Optimal user-centric data obfuscation. *Proceedings on Privacy Enhancing Technologies*, 2015(2):299–315, 2014.

[39] Melanie Mitchell. Genetic algorithms: An overview. *Complexity*, 1(1):31–39, 2013.

[40] Yujia Hu, Hang Shen, Guangwei Bai, and Tianjing Wang. Privacy-preserving task allocation for edge computing enhanced mobile crowdsensing. In *International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*, pages 431–446, 2018.

Author Biographies

Hang Shen is currently a full-time postdoctoral fellow at Department of Electrical & Computer Engineering, University of Waterloo, Canada. He received the Ph.D. (with Honors) degree in Computer Sciences from the Nanjing University of Science & Technology, Nanjing, China. Since 2015, he has been an Assistant Professor with the Department of Computer Science & Technology, Nanjing Tech University, Nanjing, China. His research interests include mobile crowdsensing, location-based services, data privacy and heterogeneous wireless networks. He is a member of the ACM.

Guangwei Bai received the B.Eng. and M.Eng. degrees in computer engineering from Xi'an Jiaotong University, Xi'an, China, in 1983 and 1986, respectively, and the Ph.D. degree in Computer Science from the University of Hamburg, Hamburg, Germany, in 1999. From 1999 to 2001, he worked at the German National Research Center for Information Technology, Germany, as a Research Scientist. In 2001, he joined the University of Calgary, Canada, as a Research Associate. Since 2005, he has been working at the Nanjing Tech University in China, as a Professor in Computer Science. In 2011, he was a Visiting Professor with the Department of Electrical & Computer Engineering, University of Waterloo, Canada. His research interests are in location-based services, privacy and security and mobile crowdsensing. He is a member of the ACM.

Yujia Hu received his B.Sc. in Computer Science and Technology from the Nanjing Tech University. His is current pursuing his master degree in College of Computer Science and Technology, Nanjing Tech University, China. His research interests include mobile crowdsensing, location-based services and data privacy.

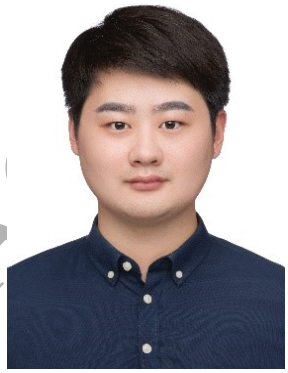
Tianjing Wang received the B.Sc. degree in Mathematics at the Nanjing Normal University in 2000, and the M.Sc. degree in Mathematics at the Nanjing University, Nanjing, China, in 2005, and the Ph.D. degree in Signal and Information System at the Nanjing University of Posts & Telecommunications, Nanjing, China, in 2009. From 2011 to 2013, she was a postdoctoral fellow with the School of Electronic Science and Engineering, Nanjing University of Posts & Telecommunications. From 2013 to 2014, she was a visiting scholar with the Department of Electrical & Computer Engineering, State University of New York at Stony Brook, New York, USA. She is now an Associate Professor in Mathematics at the Nanjing Tech University. Her research interests include location-based services, cognitive wireless networks and privacy protection. She is a member of the ACM.

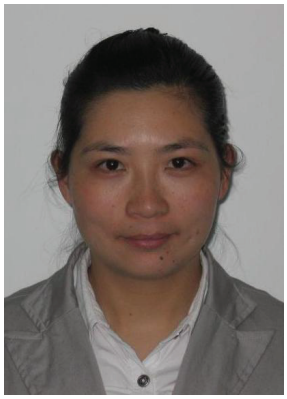
Author Photos

Hang Shen

Guangwei Bai

Yujia Hu





Tianjing Wang