Privacy-Preserving User Recruitment Protocol for Mobile Crowdsensing

Mingjun Xiao, *IEEE Member,* Guoju Gao, Jie Wu, *IEEE Fellow,*
Sheng Zhang, *IEEE Member,* and Liusheng Huang, *IEEE Member*

Abstract—Mobile crowdsensing is a new paradigm in which a requester can recruit a group of mobile users via a platform and coordinate them to perform some sensing tasks by using their smartphones. In mobile crowdsensing, each user might perform multiple tasks with different sensing qualities. Meanwhile, the users participating in the crowdsensing will ask for sufficient rewards to compensate for their expenditures. Hence, an important problem is how to recruit the users with minimum cost while achieving a satisfactory sensing quality for each task. Furthermore, in order to ease users' worries about privacy disclosures, the user recruitment process needs to protect each user's sensing quality and recruitment cost information from being revealed to other users or to the platform. In this paper, we propose two secure user recruitment problems for the cases where the recruitment costs of users are homogeneous and heterogeneous. After proving the NP-hardness of the problems, we design two secure user recruitment protocols by using secret sharing scheme. Both of the proposed protocols adopt greedy strategies, which can recruit nearly optimal users while ensuring that the total sensing quality of each task is no less than a given threshold. The difference lies in that the two greedy strategies are based on two unique utility functions. We analyze the approximation ratios of the two protocols and prove the security under the semi-honest model. Finally, we demonstrate the significant performance of the proposed protocols through extensive simulations and executions on real smartphones.

Index Terms—Mobile crowdsensing, privacy, sensing quality, secret sharing, user recruitment.

1 Introduction

Nowadays, smartphones have become extremely prevalent in day-to-day life. Most smartphones have powerful sensing, storage, and computation abilities, which can be seen as powerful mobile sensors with different functionalities. In order to make full use of these sensing resources, a new sensing paradigm called mobile crowdsensing is proposed [4]. Roughly speaking, mobile crowdsensing refers to a group of mobile users being coordinated to perform large-scale sensing tasks over urban environments through their smartphones. Since mobile crowdsensing can perform sensing tasks that individual users cannot cope with, it has stimulated many applications such as urban WiFi characterization, traffic information mapping, noise pollution monitoring, and so on, attracting much attention [4].

A typical mobile crowdsensing system consists of a collection of mobile users and a platform residing on the cloud. The platform accepts sensing tasks from requesters and recruits mobile users to perform these sensing tasks by using their smartphones. After accomplishing the sensing tasks, mobile users will return the corresponding results to requesters. In a mobile crowdsensing system, user recruitment or task allocation is one of the most important components. So far, many user recruitment or task allocation algorithms have been proposed [9, 10, 14]. Also, many incentive mechanisms such as [19, 28, 32–34] have been designed for the user recruitment component.

In this paper, we focus on the privacy-preserving user recruitment problem in sensing-quality-aware mobile crowdsensing systems. Consider that a requester wants to recruit a group of mobile users to perform some sensing tasks via a crowdsensing platform, while ensuring that each task can be accomplished with a satisfactory quality. For example, the sensing tasks might be taking some time-relative photos at many locations for air quality analysis. In general, the sensing quality depends on the heterogeneous smart devices and mobile behaviors, which can be measured mainly by the time of taking photos, the camera configurations of smart devices, and the number of photos taken by users. As a result, each user can determine the values of his sensing quality according to a predetermined criterion. During the user recruitment process, each mobile user needs to tell the platform which tasks he/she can deal with and how many sensing qualities he/she can contribute for each task. Accordingly, the mobile users will ask for variable rewards to compensate for their expenditures. This might reveal some private sensitive information. The reward requested by a mobile user will reveal the tasks that he/she can perform and the relevant sensing quality. Here, the tasks that a user can perform will reveal which locations the user might visit, while the sensing quality will reveal the frequency, time, distance of the visit, and so on. In order to avoid privacy disclosures and to make users willing participate in the

- *M. Xiao, G. Gao and L. Huang are with the School of Computer Science and Technology / Suzhou Institute for Advanced Study, University of Science and Technology of China, Hefei, P. R. China.*
Correspondence to: xiaomj@ustc.edu.cn
- *J. Wu is with the Department of Computer and Information Sciences, Temple University, 1805 N. Broad Street, Philadelphia, PA 19122.*
E-mail: jiewu@temple.edu
- *S. Zhang is with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, P. R. China.*
E-mail: sheng@nju.edu.cn

This paper is an extended version of the conference paper [30] published in IEEE INFOCOM 2017. This research was supported by the National Natural Science Foundation of China (NSFC) (Grant No. 61872330, 61572457, 61379132, 61936015, U1709217), NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1618398, CNS 1651947, CNS 1564128, the NSF of Jiangsu Province in China (Grant No. BK20191194, BK20131174, BK2009150), and Anhui Initiative in Quantum Information Technologies (Grant No. AHY150300).

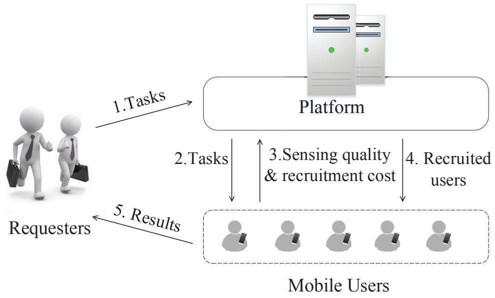


Fig. 1: The crowdsensing model

crowdsensing, it is necessary to protect each user's private sensitive information from being revealed during the user recruitment process.

Existing crowdsensing works rarely discuss privacy issues. Only a few works, such as [3, 13, 27, 36], studied the problem of protecting the privacy of sensing results collected by mobile users. Nevertheless, none of them investigates the privacy-preserving issues in the user recruitment process. To fill this gap, we study the problem of protecting users' input privacy during the process of recruiting users. In the problem, the platform and mobile users need to jointly make the user recruitment decision by conducting computations over their inputs. Meanwhile, each user needs to protect his/her inputs from being revealed to the platform or to other users. Moreover, the recruited users should make all sensing tasks be performed with satisfactory sensing qualities. Here, the differential privacy schemes [27, 36], which only can output the probabilistic results by introducing randomness into the queries, are not competent for this problem, since many precise and complex computations need to be conducted over users' private inputs in our problem. Although the homomorphic encryption and garbled circuit protocols can solve this problem, they will result in a huge computation or communication overhead that is unacceptable to mobile users.

To solve the privacy-preserving user recruitment problem, we design two secure user recruitment protocols for two different scenarios. To ensure the security, we apply secret sharing techniques during the user recruitment procedures. More specifically, the major contributions include:

1) We propose and formalize the homogeneous and heterogeneous secure user recruitment problems for sensing-quality-aware mobile crowdsensing systems. Here, "homogeneous" and "heterogeneous" mean that the recruitment costs of mobile users are uniform and different, respectively. Unlike the existing user recruitment problem, our problems take into consideration the privacy-preserving issues.

2) We first prove the NP-hardness of the problem, and then propose a greedy strategy for the homogeneous user recruitment problem. According to this, we design a hOmogeneous Secure User Recruitment protocol (O-SUR) by using the secret sharing scheme. We not only analyze the performance of the O-SUR protocol, but also prove that O-SUR is secure against any semi-honest adversaries. Furthermore, we demonstrate that as long as the computation function of the total sensing quality is an increasing submodular function, the O-SUR protocol can still produce a solution with a logarithmic approximation ratio.

3) We also propose another greedy strategy based on a new utility function for the heterogeneous user recruitment problem. Based on this, we design a hEterogeneous secret-sharing-based Secure User Recruitment protocol (E-SUR). After analyzing the correctness and approximation ratio of E-SUR, we prove that E-SUR can protect the inputs of each user from being revealed to the platform or to other users, even if they might collude.

4) In addition, we prove that O-SUR and E-SUR are two lightweight secure protocols, which do not depend on encryption/decryption operations and any trusted third-party. To the best of our knowledge, these are the first secure user recruitment protocols designed for mobile crowdsensing.

5) We conduct extensive simulations to verify the significant performance of the proposed protocols. We also implement and run the O-SUR and E-SUR protocols on real smartphones which demonstrates that O-SUR and E-SUR can work well in real applications.

The remainder of the paper is organized as follows: We introduce the models, problem, and preliminary in Section 2. The O-SUR and E-SUR protocols are proposed in 3 and 4, respectively. In Section 5, we evaluate the performances of the two protocols. After reviewing the related work in Section 6, we conclude the paper in Section 7.

2 Models, Problem, and Preliminary

In this section, we first introduce the crowdsensing and security models, and then propose the optimization problems. Additionally, we present the secret sharing scheme.

2.1 Crowdsensing Model

Consider a mobile crowdsensing system, in which a requester has many sensing tasks to deal with, denoted by $\mathcal{S} = \{s_1, s_2, \cdots, s_m\}$. Some mobile users, denoted by $\mathcal{U}=\{u_1, \cdots, u_n\}$, are willing to participate in the crowdsensing. Each user $u_i \in \mathcal{U}$ would determine a series of sensing tasks that he can perform (i.e., a subset of all sensing tasks). Since the sensing tasks that each user intends to perform are different, the consumed resources including local storage, battery, memory of the smart device, etc, are heterogeneous. Moreover, the users participating in the crowdsensing also suffer threats to their privacy [15, 22, 24]. Hence, all mobile users will ask for sufficient rewards to compensate for the expenditures and the risks. Let $c_i \in \mathbb{Z}_p$ denote the recruited cost for the user u_i ($\in \mathcal{U}$), in which $\mathbb{Z}_p$ is a prime field. Actually, c_i is private and known to nobody except for u_i itself. In this paper, we consider that c_i is the true consumed cost, since the truthfulness of mobile users can be ensured by using an incentive mechanism [5, 19, 28, 33, 34].

When users perform sensing tasks, the data collected by them might be of different qualities due to their heterogeneous smart devices and mobile behaviors. In general, multiple users need to be recruited to perform a common task so as to achieve a satisfactory sensing quality. We use $q_{i,j} \in \mathbb{Z}_p$ to indicate the *sensing quality* of user u_i $(1 \le i \le n)$ performing task s_j $(1 \le j \le m)$. Specially, $q_{i,j} = 0$ means that user u_i cannot deal with task s_j. In fact, the worse case for a user is that he cannot perform a sensing task, so the values of users' sensing qualities are non-negative in our crowdsensing system. Here, each user u_i knows his sensing qualities $q_{i,1}, \cdots, q_{i,m}$, since he can determine the value of each sensing quality $q_{i,j}$ by evaluating the corresponding

TABLE 1: Description of major notations

Variable	Description
$\mathcal{U}$, $\mathcal{S}$, Φ	the sets of all users, all tasks, and recruited users, respectively.
u_i, s_j	the i-th user, and the j-th task.
c_i	the recruitment cost of u_i ($\in\mathcal{U}$).
$q_{i,j}$	the sensing quality of user u_i performing task s_j.
$Q_j(\Phi)$	the total sensing quality of s_j based on Φ (Definition 2).
θ_j	the threshold of the required total sensing quality of task s_j ($\in\mathcal{S}$).
$f(\Phi)$, $g(\Phi)$	two utility functions about recruited users, where $g(\Phi)=\varphi f(\Phi)$ in which φ is a constant (Eq. 24).
$\Delta_i f(\Phi)$, $\Delta_i g(\Phi)$	the incremental utility for the functions $f(\Phi)$ (Definition 7) and $g(\Phi)$ (Eq. 25) by adding u_i into Φ.
b_i	a bit number that indicates whether u_i is recruited, i.e., $b_i=1$ is equivalent to $u_i\in\Phi$.
$VIEW_i$, M_i	the view and the set of received messages of u_i in the whole protocol execution process (Definition 1).
$s[i]$, $[s]$	u_i's share of a secret s, and all shares of s (Eq. 11).
$\mathbb{Z}_p$, l	a prime field, and $l=\lceil\log_2 p\rceil$.
κ	a security parameter, i.e., the degree of the random polynomial in Shamir's scheme (Definition 5).

sensing data according to a predetermined criterion. For example, each user can map a sensed image to a sensing quality value in $\mathbb{Z}_p$ according to the clarity and size.

Fig. 1 shows the execution process of the mobile crowdsensing. First, the requester publishes all sensing tasks in $\mathcal{S}$ to the users in $\mathcal{U}$ via a platform. Then, each user u_i determines the values of $q_{i,1}$, $\cdots$, $q_{i,m}$, c_i and sends them to the platform. Next, the platform recruits some users from $\mathcal{U}$ to perform the tasks in $\mathcal{S}$ while ensuring that the total sensing quality of each task is no less than a given threshold. Finally, each recruited user will go to perform the tasks in $\mathcal{S}$ and return the results to the requester. During this process, some incentive mechanisms such as [5, 19, 28, 33, 34] can be adopted to stimulate users to participate in the crowdsensing. In this paper, we will not discuss the detailed incentive mechanism and will only focus on the privacy-preserving user recruitment problem.

2.2 Security Model

When a user u_i participates in the crowdsensing, his/her sensing quality and recruitment cost values might reveal his/her private sensitive information. In order to avoid privacy disclosures, we need to protect each user's sensing qualities and cost from being revealed to the platform or to other users. For this privacy-preserving issue, we consider a typical security model, i.e., the semi-honest model [7]. In this model, each user will follow the whole user recruitment protocol, showing the honest aspect. On the other hand, the user will also try to derive the extra information from the received data, showing the dishonest aspect. The semi-honest model is reasonable, since the user is generally willing to follow and accomplish the secure protocol so as to benefit from participating. Because of this, the semi-honest model is widely-used [6, 7, 16, 17]. The privacy under the semi-honest model can formally be defined as follows:

Definition 1 (Privacy under the Semi-honest Model [7]). Let $\mathcal{F}(x_1,\cdots,x_n)=(\mathcal{F}_1,\cdots,\mathcal{F}_n)$ be an n-ary functionality, where x_i ($\in\mathbb{Z}_p$) and $\mathcal{F}_i$ are the i-th user's input and output ($1\le i\le n$). Consider a protocol for computing $\mathcal{F}$. The view of the i-th party during an execution of this protocol is denoted as $VIEW_i=(x_i,r,M_i)$, in which r represents the outcome of the i-th user's internal coin tosses and M_i represents the messages that this party has received. In other words, $VIEW_i$ is all the data that the i-th party can observe during the execution of the protocol. Now, we suppose that κ ($<n$) parties might collude, denoted as $\mathcal{I}=\{u_{i_1},\cdots,u_{i_\kappa}\}$. Moreover, we let $VIEW_\mathcal{I}$ denote the view of the κ collusion parties, in which $VIEW_\mathcal{I}\triangleq(\mathcal{I},VIEW_{i_1},\cdots,VIEW_{i_\kappa})$. We say that the protocol privately computes $\mathcal{F}$ if there exists a polynomial-time algorithm, denoted as $\mathcal{A}$, such that for every $\mathcal{I}$ above

$$\mathcal{A}(\mathcal{I},(x_{i_1},\cdots,x_{i_\kappa},\mathcal{F}_\mathcal{I}))\equiv VIEW_\mathcal{I}. \tag{1}$$

where $\equiv$ denotes the computational indistinguishability.

Remarks: Eq. 1 asserts that the view of the users in $\mathcal{I}$ can be efficiently simulated based solely on their inputs and outputs. In other words, they cannot derive extra information during the execution of the protocol.

2.3 Problem

We focus on the secure user recruitment problem in the above mobile crowdsensing under the semi-honest model. We use set Φ to denote a user recruitment solution where $u_i\in\Phi$ indicates that user u_i is recruited. The platform needs to recruit some mobile users from $\mathcal{U}$ to perform the sensing tasks while ensuring that the total sensing quality of each task is no less than a given threshold. We use $\theta_j\in\mathbb{Z}_p$ for $\forall s_j\in\mathcal{S}$ to denote the threshold. At the same time, we give the definition of the total sensing quality as follows:

Definition 2. The total obtained sensing quality of task s_j ($\in\mathcal{S}$) based on a user recruitment solution Φ, denoted as $Q_j(\Phi)$, is computed in the following formula:

$$Q_j(\Phi)\triangleq Q(q_{i,j}|u_i\in\Phi), \tag{2}$$

where $Q(\cdot)$ is a general function about $q_{i,j}$.

In many existing applications, the total sensing quality of a task is directly defined as the sum of the sensing quality of each recruited user performing this task, i.e., $Q_j(\Phi)=\sum_{u_i\in\Phi}q_{ij}$. This is a special form of our definition of the total sensing quality. Actually, in addition to the sum of sensing quality, our definition can also be calculated in other ways. For example, if the sensing quality $q_{i,j}$ represents the probability of successful sensing, $Q_j(\cdot)$ may be defined as their joint probability, i.e., $Q_j(\Phi)=1-\prod_{u_i\in\Phi}(1-q_{i,j})$. In fact, so long as $Q_j(\Phi)$ is an increasing submodular function with $Q_j(\Phi=\phi)=0$, our proposed user recruitment protocol can achieve a provably logarithmic approximation. We will present the analysis in Section 3.5 in detail. For better understanding, we directly use $Q_j(\Phi)=\sum_{u_i\in\Phi}q_{ij}$ to denote the total sensing quality in our user recruitment problem.

To solve the secure user recruitment problem, where the sensing qualities and recruitment costs of mobile users need to be protected from being revealed simultaneously, we define two optimization problems which are gradually progressive and in-depth. First, we propose the *homogeneous* secure user recruitment problem in which the recruitment costs of all users are uniform. Since the recruitment costs of users are homogeneous, minimizing the total cost is equivalent to minimize the number of recruited users. Second, we define the *heterogeneous* secure user recruitment problem, where the recruitment costs of users are heterogeneous, involving the privacy protection of sensing quality and recruitment cost simultaneously. More specifically, we have the following definitions.

Definition 3. The *hOmogeneous Secure User Recruitment (**O-SUR**) problem*, in which the recruitment costs of all users are homogeneous, is to privately find a minimum number of recruited users to perform the sensing tasks (i.e., determine a user recruitment solution $\Phi \subseteq \mathcal{U}$) while ensuring that the total sensing quality of each task is no less than a given threshold, i.e.,

$$\begin{aligned} &Minimize: && |\Phi| && (3)\\ &Subject\ to: && \Phi \subseteq \mathcal{U} && (4)\\ & && Q_j(\Phi) \geq \theta_j,\ 1 \leq j \leq m && (5)\\ &Security\quad: && \text{Eq. 1 holds.} && (6)\end{aligned}$$

Definition 4. The *hEterogeneous Secure User Recruitment (**E-SUR**) problem* is to select a set of users Φ from the alternative user set $\mathcal{U}$ with minimum cost under the sensing quality constraints, while protecting the recruitment cost and sensing qualities of each user from being revealed to other users or to the platform. That is,

$$\begin{aligned} &Minimize: && C(\Phi) = \textstyle\sum_{u_i \in \Phi} c_i && (7)\\ &Subject\ to: && \Phi \subseteq \mathcal{U} && (8)\\ & && Q_j(\Phi) \geq \theta_j,\ 1 \leq j \leq m && (9)\\ &Security\quad: && \text{Eq. 1 holds.} && (10)\end{aligned}$$

Here, we assume that there always exists at least one feasible solution for these two optimization problems. This is reasonable because we can expand the alternative user set (i.e., $\mathcal{U}$) by inviting more mobile users to participate in the crowdsensing, until the solutions to the optimization problems appear.

For ease of presentation, we also use an n-bit vector $(b_1, \cdots, b_i, \cdots, b_n)$ to indicate the user recruitment solution where $b_i = 1$ for $u_i \in \Phi$; otherwise, if $u_i \notin \Phi$, we set $b_i = 0$.

2.4 Preliminary

In this paper, we address privacy-preserving issues by using secret sharing schemes. A widely-used secret sharing scheme is Shamir's scheme [21]. Denote the shares of a secret s among n users as

$$[s] \triangleq (s[1], \cdots, s[i], \cdots, s[n]), \tag{11}$$

where $s[i]$ is the i-th user's share. Then, Shamir's secret sharing scheme can be defined as follows:

Definition 5. Let p be an odd prime, and $\mathbb{Z}_p$ is a prime field. To share a secret s ($s \in \mathbb{Z}_p$) among n users ($n < p$), Shamir's scheme determines a random polynomial $g_s(x) = s + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_\kappa x^\kappa \bmod p$ with randomly chosen $\alpha_i \in \mathbb{Z}_p$ for $1 \leq i \leq \kappa$, $\kappa < n$. Then, the share of the i-th user is $s[i] = g_s(i)$.

It has been proved that in Shamir's scheme, any h shares with $h \leq \kappa$ give no information on s (called κ-privacy), while any h shares with $h > \kappa$ can uniquely disclose s (called (κ+1)-reconstruction).

In the following, we will propose the corresponding solutions to the O-SUR and E-SUR problems under the semi-honest model in Sections 3 and 4, respectively. Additionally, we list the main notations in Table 1.

3 The O-SUR Protocol

In this section, we propose a hOmogeneous Secure User Recruitment (O-SUR) protocol by using secret sharing scheme. We first introduce some secure operations in the secret sharing scheme. Then, we analyze the NP-hardness of the O-SUR problem and propose the greedy user selection strategy as the building blocks of the O-SUR protocol. Next, we design the O-SUR protocol and present an example to illustrate the user recruitment procedure. Finally, we analyze the performance and security of the O-SUR protocol.

3.1 Secure Operations

In the O-SUR protocol, each sensing quality is turned to be a secret shared among all users. When the users make the user recruitment decision, they need to jointly conduct some mathematical operations on the shared secrets, which are defined as follows:

Definition 6. Let $x, y \in \mathbb{Z}_p$ be two secrets shared by n users and $[x]$, $[y]$ be the corresponding polynomial shares. Then, the secure mathematical operations are defined as follows:

$$\begin{aligned} &[z_1] \leftarrow SecAdd([x],[y]), && [z_2] \leftarrow SecSub([x],[y]),\\ &[z_3] \leftarrow SecMulti([x],[y]), && [z_4] \leftarrow SecCmp([x],[y]),\\ &[z_5] \leftarrow SecMax([x],[y]), && [z_6] \leftarrow SecMin([x],[y]),\end{aligned} \tag{12}$$

where $z_1 = x + y \bmod p$; $z_2 = x - y \bmod p$; $z_3 = xy \bmod p$; $z_4 = 1$ if $x \leq y$, or $z_4 = 0$ when $x > y$; $z_5 = \max\{x, y\}$, and $z_6 = \min\{x, y\}$.

In Definition 6, the $SecAdd$ and $SecSub$ operations can be conducted efficiently without any communications among n users. For $SecAdd$, each user u_i can locally compute his/her share by letting $z_1[i] = x[i] + y[i]$. For example, assume $x[i] = x + \alpha_1 i + \alpha_2 i^2 + \cdots + \alpha_\kappa i^\kappa \bmod p$ and $y[i] = y + \beta_1 i + \beta_2 i^2 + \cdots + \beta_\kappa i^\kappa \bmod p$, where $\alpha_1, \cdots, \alpha_\kappa, \beta_1, \cdots, \beta_\kappa$ are randomly chosen from $\mathbb{Z}_p$. Then, $z[i] = x + y + (\alpha_1 + \beta_1) i + \cdots + (\alpha_\kappa + \beta_\kappa) i^\kappa \bmod p$. Likewise, the $SecSub$ operation can also be locally conducted by letting each user compute $z_2[i] = x[i] - y[i]$.

In contrast, the $SecMulti$ and $SecCmp$ operations are a bit more complex, and they require users to communicate with one another. In this paper, we realize the two operations by using the secure multi-party multiplication protocol in [16] and the secure multi-party comparison protocol in [17], respectively. The multiplication protocol in [16] is a well-known and efficient protocol built on a verifiable secret sharing scheme. It requires $O(n^2 l)$ bit-operations per user ($l = \lceil \log_2 p \rceil$) and one round of communication. The comparison protocol in [17] is one of the most efficient secure comparison protocols. The computation complexity is dominated by 15 rounds of invocations of the multiplication protocol, and the communication complexity is $279l$+5 times of the multiplication protocol.

Procedure 1 The User Recruitment Strategy of O-SUR

```
Input: $\mathcal{U}$, $\mathcal{S}$, $\{q_{i,j}|u_i \in \mathcal{U}, s_j \in \mathcal{S}\}$, $\{\theta_j|s_j \in \mathcal{S}\}$
Output: $\Phi$
$\Phi = \emptyset$; $f(\Phi) = 0$;
while $f(\Phi) < \sum_{j=1}^{m} \theta_j$ do
    Select a user $u_i \in \mathcal{U} \backslash \Phi$ to maximize $\Delta_i f(\Phi)$;
    $\Phi = \Phi \cup \{u_i\}$;
return $\Phi$
```

The $SecMin$ and $SecMax$ operations can be realized by using $SecMulti$ and $SecCmp$. More specifically, we can let

$$\begin{aligned} SecMax([x],[y]) \triangleq & SecAdd([x], SecMulti(SecCmp([x],[y]), \\ & SecSub([y],[x]))) \quad (13) \\ SecMin([x],[y]) \triangleq & SecAdd([x], SecMulti(SecSub(1- \\ & SecCmp([x],[y])), SecSub([y],[x]))). \quad (14) \end{aligned}$$

Eq. 13 is correct, since the right part will be $SecAdd([x],[0])$ if $x > y$; otherwise, it will be $SecAdd([x], SecSub([y], [x]))$. Likewise, Eq. 14 is also correct. Here, the $SecMax$ and $SecMin$ operations can directly obtain the maximum and minimum values of x and y, respectively, without revealing which is the larger or smaller one. Moreover, the $SecMin$ and $SecMax$ operations can be extended to support more than two operands. For example, $SecMin([x_1],[x_2],[x_3]) \leftarrow SecMin([x_1], SecMin([x_2],[x_3]))$. Additionally, all of these secure operations can support the computation between secret and public values. For example, when the secret x in Definition 6 is replaced by a public value $r \in \mathbb{Z}_p$, the $SecAdd$ operation can be conducted by letting $z_1[i] = r + y[i]$. Moreover, $SecMulti$ can be computed directly by letting $z_3[i] = r \cdot y[i]$ for each user u_i without any communications. The computation complexity of $SecCmp$ becomes 7 rounds of invocations of the multiplication protocol, and the communication complexity becomes $17l$ times of the multiplication protocol [17].

3.2 The Building Blocks

Before the solution, we first prove the NP-hardness of the user recruitment problem in the following theorem.

Theorem 1. The user recruitment problem is NP-hard.

Proof: We consider a special case of the user recruitment problem: given a mobile crowdsensing, where the user set is $\mathcal{U}$, the task set is $\mathcal{S}$, the recruitment costs of users $\{c_i|u_i \in \mathcal{U}\}$ are uniform, each sensing quality $q_{i,j} \in \{0, 1\}$, and the threshold of total sensing quality is $\theta_j = 1$ for $\forall s_j \in \mathcal{S}$; determine a user recruitment solution Φ, such that the platform can minimize $|\Phi|$, while the total sensing quality of each task s_j is no less than θ_j. Here, if a user u_i can perform a task s_j, i.e., $q_{i,j} = 1$, we say that u_i can cover s_j. Moreover, once a task is covered by a user, the total sensing quality of this task must be no less than θ_j. Then, when we replace each u_i in $\mathcal{U}$ by using the set of tasks that u_i can cover, denoted by $\mathcal{S}_i$ ($\subseteq \mathcal{S}$), this problem can be equivalently seen as a set cover problem, a well known NP-hard problem: given a task set $\mathcal{S}$, a collection of subset $\{\mathcal{S}_i | 1 \le i \le n\}$, find a minimum size of subcollection of $\{\mathcal{S}_i | 1 \le i \le n\}$ that covers all tasks in $\mathcal{S}$. Thus, the special user recruitment problem is NP-hard. Consequently, the general user recruitment problem is also at least NP-hard. ■

Protocol 1 The O-SUR Protocol

```
Input: $\mathcal{U}$, $\mathcal{S}$, $\{q_{i,j}|u_i \in \mathcal{U}, s_j \in \mathcal{S}\}$, $\{\theta_j|s_j \in \mathcal{S}\}$
Output: $b_1, \cdots, b_n$
Phase 1: the requester publishes $\mathcal{S}$ to $\mathcal{U}$ via the platform;
Phase 2: users input their sensing quality vectors;
for $i = 1$ to $n$ do
    user $u_i$ determines the sensing qualities$q_{i,1}, \cdots, q_{i,m}$;
    for $j = 1$ to $m$ do
        user $u_i$ generates the polynomial sharing $[q_{i,j}]$;
        user $u_i$ sends the share $q_{i,j}[i']$ to user $u_{i'}$;
Phase 3: users jointly make the decision of user recruitment;
for $i = 1$ to $n$ do
    $[b_i] \leftarrow [0]$;
for $j = 1$ to $m$ do
    $[Q_j] \leftarrow [0]$;
for $round = 1$ to $n$ do
    for $i = 1$ to $n$ do
        $[\Delta_i f] \leftarrow [0]$;
        for $j = 1$ to $m$ do
            $[\delta] \leftarrow SecMin([q_{i,j}], SecSub(\theta_j, [Q_j]))$;
            $[\Delta_i f] \leftarrow SecAdd([\Delta_i f], [\delta])$;
        $[\Delta_i f] \leftarrow SecMulti([\Delta_i f], SecSub([1], [b_i]))$;
    $[\Delta_{max} f] \leftarrow SecMax([\Delta_1 f], \cdots, [\Delta_n f])$;
    for $i = 1$ to $n$ do
        $[z] \leftarrow SecCmp([\Delta_{max} f], [\Delta_i f])$;
        $[b_i] \leftarrow SecAdd([b_i], SecMulti(SecSub([1], [b_i]), [z]))$;
        for $j = 1$ to $m$ do
            $[\delta] \leftarrow SecMin([q_{i,j}], SecSub(\theta_j, [Q_j]))$;
            $[Q_j] \leftarrow SecAdd([Q_j], SecMulti([z], [\delta]))$;
Phase 4: the users reconstruct the results;
for $i = 1$ to $n$ do
    user $u_i$ collects all shares of $[b_i]$;
    user $u_i$ derives $b_i = \sum_{j=1}^{m} b_i[j]$;
```

Since the user recruitment problem is NP-hard, we adopt a greedy strategy to recruit users. The greedy criterion is that the user who can improve the total sensing qualities of all tasks the most well will be recruited first. More precisely, the greedy strategy is based on the following utility function:

Definition 7. *Utility function* $f(\Phi)$ indicates the total sensing qualities of all tasks in $\mathcal{S}$ contributed by the users in set Φ, until each task s_j reaches the corresponding threshold θ_j, defined as follows:

$$f(\Phi) = \sum_{j=1}^{m} \min\{Q_j(\Phi), \theta_j\} = \sum_{j=1}^{m} \min\{\sum_{u_i \in \Phi} q_{i,j}, \theta_j\}. \quad (15)$$

Moreover, for a given user set Φ, we denote the *incremental utility* of recruiting a new user u_i into Φ as

$$\Delta_i f(\Phi) = f(\Phi \cup \{u_i\}) - f(\Phi). \quad (16)$$

The procedure of recruiting users in the O-SUR protocol is based on the above defined utility, which not only contains the optimization objective, but also takes the nonlinear constraints (i.e., the sensing quality constraints) into consideration. Further, the greedy user recruitment strategy is shown in Procedure 1. The whole user recruitment procedure contains multiple rounds of iterations. At the beginning, the set of recruited users is an empty set, i.e.,

$\Phi = \emptyset$. Then, in each round of iteration, the user who can improve the utility $f(\Phi)$ the most, i.e., the user u_i who can maximize the value of $\Delta_i f(\Phi)$, is recruited and added into Φ. The user recruitment process terminates when $f(\Phi) = \sum_{j=1}^{m} \theta_j$. After this process, the user recruitment result Φ is produced.

3.3 The Detailed O-SUR Protocol

Then, we introduce the O-SUR protocol which adopts the same utility function and greedy strategy as Procedure 1 to recruit users. The difference lies in that all inputs and computations are conducted by using the secret sharing techniques. First, each input $q_{i,j}$ is seen as a secret, and it is replaced by its polynomial shares $[q_{i,j}]$ in O-SUR. Second, when users jointly make recruitment decisions, all computations are conducted by using the secure operations in Definition 6, and all intermediate results are produced in the manner of shared secrets. To ensure this, we replace $u_i \in \Phi$ and $\Delta_i f(\Phi)$ by using $[b_i] = [1]$ and $\sum_{j=1}^{m} \min\{q_{i,j}, \theta_j – Q_j(\Phi)\}$. Here, all users can communicate with each other via the platform. Moreover, in order to prevent the selected user from being revealed in each round of iteration, we hide the maximum incremental utility and the selected user in a $SecMax$ operation and a $SecCmp$ operation. Only in the final phase, each user u_i can collect the corresponding shares to reconstruct the value of b_i so as to know whether he/she is recruited. Additionally, the whole process is conducted in a distributed way.

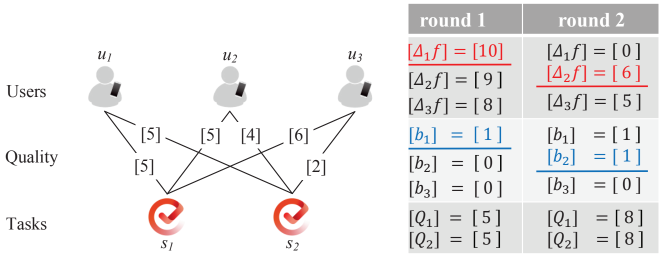
The detailed O-SUR protocol is shown in Protocol 1, which mainly contains four phases. First, the requester generates tasks and publishes them to users via the platform. Second, mobile users determine their sensing qualities. Moreover, all users construct the polynomial secret shares of their sensing quality values as the inputs in Steps 3-5. Third, all mobile users jointly make recruitment decisions by using the secure operations in Definition 6. More specifically, Steps 6-9 initialize for the user recruitment decision process. In Steps 11-17, users jointly find the maximum incremental utility value, i.e., $\Delta_i f$. In Steps 18-23, users determine the recruited user and update the corresponding Q_j. Note that we will omit Φ in $\Delta_i f(\Phi)$ and $Q_j(\Phi)$ in Protocol 1 for simplicity. Fourth (Steps 24-26), all users collect their corresponding shares to reconstruct the recruitment results. After the recruited users perform the sensing tasks and upload the sensing results to the requester, the requester would pay them accordingly. During the process, the platform does not know the recruitment and sensing results.

The computation and communication complexity of the whole protocol is dominated by the $SecMin$ operations in Steps 14 and 22, which are $O(mn^2)$ invocations of secure multiplication operations. Therefore, the protocol will result in $O(mn^4 l)$ bit-operations per user and $O(mn^2 l)$ rounds of communication, where a round of communication means that users communicate with one another once.

3.4 Example

To better understand Protocol 1, we use an example to illustrate the user recruitment procedure. In the example, there are two tasks and three users with six sensing qualities, as shown in Fig. 2. The protocol is conducted as follows:

- First round: The three users jointly compute their incremental utility values, of which $[\Delta_1 f] = [10]$ is the largest. Thus, user u_1 is recruited, i.e., $[b_1] = [1]$. Accordingly, we have $[Q_1] = [Q_2] = 5$.
- Second round: The users jointly compute their incremental utility values again, based on $[Q_1] = [Q_2] = 5$. Since $[b_1] = [1]$, $[\Delta_1 f]$ is set as $[0]$. This time, $[\Delta_2 f] = [6]$ becomes the largest value. Thus, user u_2 is recruited, i.e., $[b_2] = [1]$. Accordingly, $[Q_1] = [Q_2] = \theta_1 = \theta_2 = 8$. No more users will be recruited.



round 1	round 2
$[\varDelta_1 f] = [10]$	$[\varDelta_1 f] = [0]$
$[\varDelta_2 f] = [9]$	$[\varDelta_2 f] = [6]$
$[\varDelta_3 f] = [8]$	$[\varDelta_3 f] = [5]$
$[b_1] = [1]$	$[b_1] = [1]$
$[b_2] = [0]$	$[b_2] = [1]$
$[b_3] = [0]$	$[b_3] = [0]$
$[Q_1] = [5]$	$[Q_1] = [8]$
$[Q_2] = [5]$	$[Q_2] = [8]$

(a) Users, tasks and sensing qualities (b) Intermediate results

Fig. 2: Illustration of the O-SUR protocol ($\theta_1 = \theta_2 = 8$)

3.5 The Performance and Security Analysis

In this section, we first prove the correctness and analyze the approximation ratio of the greedy strategy (i.e., Procedure 1). Based on this, we analyze the performance of the O-SUR protocol (i.e., Protocol 1). Afterwards, we prove the security of O-SUR under the semi-honest model.

First, we prove three important properties of the defined utility function $f(\Phi)$ in the following theorems.

Theorem 2. $f(\Phi)$ is an increasing function with $f(\emptyset) = 0$.

Proof: First, if $\Phi = \emptyset$, then $\min\{\sum_{u_i \in \Phi} q_{i,j}, \theta_j\} = 0$ for $\forall j \in [1, m]$. According to Definition 7, $f(\Phi = \emptyset) = 0$. Second, without loss of generality, we consider two user sets Φ_1 and Φ_2, where $\Phi_1 \subseteq \Phi_2$. Then, we have $\min\{\sum_{u_i \in \Phi_1} q_{i,j}, \theta_j\} \leq \min\{\sum_{u_i \in \Phi_2} q_{i,j}, \theta_j\}$. Consequently, we have $f(\Phi_1) = \sum_{j=1}^{m} \min\{\sum_{u_i \in \Phi_1} q_{i,j}, \theta_j\} \leq \sum_{j=1}^{m} \min\{\sum_{u_i \in \Phi_2} q_{i,j}, \theta_j\} = f(\Phi_2)$. Therefore, $f(\Phi)$ is an increasing function with $f(\emptyset) = 0$. The theorem holds. ■

Theorem 3. $f(\Phi) = \sum_{j=1}^{m} \theta_j$ iff Φ is a feasible solution to the user recruitment problem.

Proof: According to Eq. 15, $f(\Phi) = \sum_{j=1}^{m} \theta_j$ iff $\min\{\sum_{u_i \in \Phi} q_{i,j}, \theta_j\} = \theta_j$ holds for each $j \in [1, m]$. In fact, $\min\{\sum_{u_i \in \Phi} q_{i,j}, \theta_j\} = \theta_j$ and $\sum_{u_i \in \Phi} q_{i,j} \geq \theta_j$ are equivalent. Therefore, we have that $f(\Phi) = \sum_{j=1}^{m} \theta_j$ iff $\sum_{u_i \in \Phi} q_{i,j} \geq \theta_j$ holds for each $j \in [1, m]$. This means that the users in Φ can perform each task s_j in $\mathcal{S}$ with a total sensing quality no less than θ_j. Thus, the theorem is correct. ■

Theorem 4. $f(\Phi)$ is a submodular function. More specifically, for two arbitrary user sets Φ_1 and Φ_2, $\Phi_1 \subseteq \Phi_2$, and $\forall u_h \in \mathcal{U} \backslash \Phi_2$, the submodular property holds, i.e.,

$$f(\Phi_1 \cup \{u_h\}) - f(\Phi_1) \geq f(\Phi_2 \cup \{u_h\}) - f(\Phi_2). \quad (17)$$

Proof: To prove the submodular property of $f(\Phi)$, we consider two cases:

Case 1: user u_h cannot deal with task s_j, i.e., $q_{h,j}$=0. For this case, we have

$$\min\{\sum\nolimits_{u_i \in \Phi_1 \cup \{u_h\}} q_{i,j}, \theta_j\} - \min\{\sum\nolimits_{u_i \in \Phi_1} q_{i,j}, \theta_j\} =$$
$$\min\{\sum\nolimits_{u_i \in \Phi_2 \cup \{u_h\}} q_{i,j}, \theta_j\} - \min\{\sum\nolimits_{u_i \in \Phi_2} q_{i,j}, \theta_j\} = 0. \quad (18)$$

Case 2: user u_h can perform task s_j, i.e., $q_{h,j} > 0$. We divide this case into two sub-cases: $\sum_{u_i \in \Phi_1 \cup \{u_h\}} q_{i,j} \leq \sum_{u_i \in \Phi_2} q_{i,j}$ and $\sum_{u_i \in \Phi_1 \cup \{u_h\}} q_{i,j} > \sum_{u_i \in \Phi_2} q_{i,j}$.

For the first sub-case, since $\Phi_1 \subseteq \Phi_2$, we have $\sum_{u_i \in \Phi_1} q_{i,j} \leq \sum_{u_i \in \Phi_1 \cup \{u_h\}} q_{i,j} \leq \sum_{u_i \in \Phi_2} q_{i,j} \leq \sum_{u_i \in \Phi_2 \cup \{u_h\}} q_{i,j}$. Then, we can get:

$$\min\{\sum_{u_i \in \Phi_1 \cup \{u_h\}} q_{i,j}, \theta_j\} - \min\{\sum_{u_i \in \Phi_1} q_{i,j}, \theta_j\}$$
$$= \begin{cases} q_{h,j} & , \theta_j \geq \sum_{u_i \in \Phi_2 \cup \{u_h\}} q_{i,j}; \\ q_{h,j} & , \sum_{u_i \in \Phi_2 \cup \{u_h\}} q_{i,j} > \theta_j \geq \sum_{u_i \in \Phi_2} q_{i,j}; \\ q_{h,j} & , \sum_{u_i \in \Phi_2} q_{i,j} > \theta_j \geq \sum_{u_i \in \Phi_1 \cup \{u_h\}} q_{i,j}; \\ \theta_j - \sum_{u_i \in \Phi_1} q_{i,j} & , \sum_{u_i \in \Phi_1 \cup \{u_h\}} q_{i,j} > \theta_j \geq \sum_{u_i \in \Phi_1} q_{i,j}; \\ 0 & , \theta_j < \sum_{u_i \in \Phi_1} q_{i,j}. \end{cases} \tag{19}$$

$$\min\{\sum_{u_i \in \Phi_2 \cup \{u_h\}} q_{i,j}, \theta_j\} - \min\{\sum_{u_i \in \Phi_2} q_{i,j}, \theta_j\}$$
$$= \begin{cases} q_{h,j} & , \theta_j \geq \sum_{u_i \in \Phi_2 \cup \{u_h\}} q_{i,j}; \\ \theta_j - \sum_{u_i \in \Phi_2} q_{i,j} & , \sum_{u_i \in \Phi_2 \cup \{u_h\}} q_{i,j} > \theta_j \geq \sum_{u_i \in \Phi_2} q_{i,j}; \\ 0 & , \sum_{u_i \in \Phi_2} q_{i,j} > \theta_j \geq \sum_{u_i \in \Phi_1 \cup \{u_h\}} q_{i,j}; \\ 0 & , \sum_{u_i \in \Phi_1 \cup \{u_h\}} q_{i,j} > \theta_j \geq \sum_{u_i \in \Phi_1} q_{i,j}; \\ 0 & , \theta_j < \sum_{u_i \in \Phi_1} q_{i,j}. \end{cases} \tag{20}$$

Comparing Eqs. 19 and 20, we have:

$$\min\{\sum_{u_i \in \Phi_1 \cup \{u_h\}} q_{i,j}, \theta_j\} - \min\{\sum_{u_i \in \Phi_1} q_{i,j}, \theta_j\} \geq$$
$$\min\{\sum_{u_i \in \Phi_2 \cup \{u_h\}} q_{i,j}, \theta_j\} - \min\{\sum_{u_i \in \Phi_2} q_{i,j}, \theta_j\} \tag{21}$$

Similarly, for the second sub-case, we can still derive Eq. 21. In summary, we can conclude that Eq. 21 holds for all cases. Now, according to Eq. 15, we have:

$$f(\Phi_1 \cup \{u_h\}) - f(\Phi_1) \geq f(\Phi_2 \cup \{u_h\}) - f(\Phi_2). \tag{22}$$

Therefore, $f(\Phi)$ is a submodular function. ■

Second, based on the above properties of the utility function, we can prove the correctness of Procedure 1.

Theorem 5. Procedure 1 is correct. That is, it will produce a feasible solution for the user recruitment problem, as long as the problem is solvable.

Proof: In each round of iteration in Procedure 1, a user will be added into the user set Φ. Moreover, according to Theorem 2, the utility $f(\Phi)$ will increase along with the expansion of the user set Φ. Hence, the iteration processes will certainly terminate. According to Procedure 1, when the iteration processes terminate, there must be $f(\Phi) = \sum_{j=1}^{m} \theta_j$. So, we can conclude that Φ is a feasible solution for the user recruitment problem according to Theorem 3. ■

Furthermore, we derive the approximation ratio of Procedure 1. Before this, we first prove that our user recruitment can be re-formalized as a Minimum Integral Submodular Cover with Submodular Cost (MISC/SC) problem.

Lemma 1. The O-SUR problem can be re-formalized as an MISC/SC problem. Specifically, we have:

1) if the problem is solvable, it can be re-formalized as

$$Minimize\{|\Phi| | f(\Phi) = f(\mathcal{U}), \Phi \subseteq \mathcal{U}\}; \tag{23}$$

2) both $f(\Phi)$ and $|\Phi|$ are polymatroid functions on $2^{\mathcal{U}}$, i.e., both of them are increasing submodular functions, and $f(\Phi) = 0$, $|\Phi| = 0$ when $\Phi = \emptyset$.

Proof: 1) If the user recruitment problem is solvable, the user set $\mathcal{U}$ must be a feasible solution, since this set contains all users. According to Theorem 3, $f(\Phi) = \sum_{j=1}^{m} \theta_j$ iff Φ is a feasible solution. Therefore, if Φ is another feasible solution, we must have $f(\Phi) = f(\mathcal{U}) = \sum_{j=1}^{m} \theta_j$. That is to say, the constraint Eq. 5 can be equivalently replaced by $f(\Phi) = f(\mathcal{U})$. Therefore, the user recruitment problem can be re-formalized as Eq. 23.

2) According to Theorems 2 and 4, $f(\Phi)$ is an increasing submodular function with $f(\emptyset) = 0$. Thus, $f(\Phi)$ is a polymatroid function on $2^{\mathcal{U}}$. On the other hand, for two arbitrary user sets Φ_1 and Φ_2, $|\Phi|$ satisfies the equation: $|\Phi_1| + |\Phi_2| = |\Phi_1 \cap \Phi_2| + |\Phi_1 \cup \Phi_2|$. This means that $|\Phi|$ is a modular function, which also implies the submodular property. Moreover, it is easy to verify that $|\Phi|$ is an increasing function with $|\Phi = \emptyset| = 0$. Thus, $|\Phi|$ is also a polymatroid function. Therefore, the lemma holds. ■

Next, we introduce a lemma about the approximation ratio of MISC/SC problems, which is derived from [25].

Lemma 2. For an MISC/SC problem like $Minimize\{|\Phi| | f(\Phi) = f(\mathcal{U}), \Phi \subseteq \mathcal{U}\}$, if $f(\Phi)$ is a polymatroid integer-valued function on $2^{\mathcal{U}}$ and $|\Phi|$ is a modular function, the greedy strategy in Procedure 1 can achieve a $(1 + \ln \gamma)$-approximation solution, where $\gamma = max_{u_i \in \mathcal{U}} f(\{u_i\})$.

Now, we derive the approximation ratio of the proposed procedure in the following theorem.

Theorem 6. Procedure 1 can produce a $(1 + \ln \gamma)$-approximation solution, where $\gamma = \max_{u_i \in \mathcal{U}} f(\{u_i\})$.

Proof: According to Lemma 1, the O-SUR problem can be re-formalized as an MISC/SC problem. Moreover, according to Theorem 4, we have that $f(\Phi)$ is a polymatroid integer-valued function on $2^{\mathcal{U}}$. Additionally, in the proof of Lemma 1, we have shown that $|\Phi|$ is a modular function. Therefore, according to Lemma 2, the greedy strategy in Procedure 1 can achieve a $(1 + \ln \gamma)$-approximation solution, where $\gamma = max_{u_i \in \mathcal{U}} f(\{u_i\})$. The theorem holds. ■

Theorems 5 and 6 show that if the user recruitment problem is solvable, Procedure 1 will produce a nearly optimal solution. Accordingly, we analyze the performance of the O-SUR protocol. Essentially, Protocol 1 is a distributed version of Procedure 1, combined with secret sharing schemes. Therefore, Protocol 1 can achieve the same user recruitment result as Procedure 1. We can straightforwardly get the following theorem:

Theorem 7. Protocol 1 is correct, and it can also produce a $(1 + \ln \gamma)$-approximation solution, where $\gamma = \max_{u_i \in \mathcal{U}} f(\{u_i\})$.

Next, we prove that Protocol 1 is secure against any semi-honest adversaries in the following theorem.

Theorem 8. Protocol 1 can protect the sensing qualities of each user from being revealed to any κ semi-honest adversaries and the platform, even if they might collude, where κ (i.e., the degree of polynomial sharing) may be any integer less than n.

Proof: First, $SecMulti$ and $SecCmp$ are secure according to [16, 17]. Further, according to Eqs. 13 and 14 and the composition security theorem in [7], $SecMax$ and $SecMin$ are also secure. Thus, we only need to prove that O-SUR is secure by itself. According to Definition 1, we first construct a simulator for an arbitrary user such that its view can be efficiently simulated by the output of the simulator. That is to say, the output of the simulator and the view are computational indistinguishability. Without loss of generality, we consider any κ users, denoted by $\mathcal{I}=\{u_{i_1},\cdots,u_{i_\kappa}\}\subset\mathcal{U}$, and construct the view of each user $u_{i_t}\in\mathcal{I}$, i.e., $VIEW_{i_t}$. Going through the whole protocol, we have $M_{i_t}=\{q_{i,j}[i_t],b_i[i_t],Q_j[i_t],\delta[i_t],\Delta_i f[i_t],\Delta_{max}f[i_t],z[i_t]\}$ and $VIEW_{i_t}=(\{q_{i_t,j},n,m,\theta_j\},r,M_{i_t})$. Then, the simulator for the user u_{i_t} randomly selects a number $q'_{i_t,j}$ from the prime filed $\mathbb{Z}_p$. Consider the received messages M_{i_t} in $VIEW_{i_t}$ where the number of shares of each secret is no larger than κ. Also, since both $q_{i_t,j}$ and $q'_{i_t,j}$ are the numbers randomly selected from $\mathbb{Z}_p$, the output of the simulator and the view are computational indistinguishability. That is to say, each received message can be simulated by a number randomly chosen from $\mathbb{Z}_p$. Thus, Eq. 1 holds for O-SUR. According to the composition security theorem in [7], the whole protocol is secure. Thus, this theorem is correct. ■

In Theorem 8, let $\kappa=n-1$, then no one except the secret holder is able to gather all shares to reconstruct the message. In addition, we prove that when the total sensing quality function $Q_j(\Phi)$ is a trivial function instead of $Q_j(\Phi)=\sum_{u_i\in\Phi}q_{i,j}$, Protocol 1 can still work well. In such case, Eq. 5 becomes a non-linear constraint, and computing the utility function $f(\Phi)$ becomes a little complicated. We have the following theorems.

Theorem 9. When $Q_j(\Phi)$ in Protocol 1 is a trivial function that can be securely computed by using the secure operations in Definition 6, Protocol 1 will still be secure.

Proof: In Theorem 8, all parts, except the process of computing $Q_j(\Phi)$ in Protocol 1, have been proven to be secure. Now, if $Q_j(\Phi)$ can also be securely computed, the whole protocol will be secure according to the composition security theorem in [7]. ■

Theorem 10. When $Q_j(\Phi)$ is an increasing submodular function with $Q_j(\Phi=\emptyset)=0$, we have: 1) the utility function $f(\Phi)$ is still submodular; 2) Protocol 1 can still produce a $(1+\ln\gamma)$-approximation solution where $\gamma{=}max_{u_i\in\mathcal{U}}f(\{u_i\})$.

Proof: 1) Consider two arbitrary user sets Φ_1 and Φ_2, $\Phi_1\subseteq\Phi_2$, and $\forall u_h\in\mathcal{U}\backslash\Phi_2$, we need to prove the submodular property holds, i.e., $f(\Phi_1\cup\{u_h\})-f(\Phi_1)\geq f(\Phi_2\cup\{u_h\})-f(\Phi_2)$. To prove this, we adopt the same method as that in Theorem 4: 1) for the case $q_{h,j}=0$, we have $min\{Q_j(\Phi_1\cup\{u_h\}),\theta_j\}-min\{Q_j(\Phi_1),\theta_j\}=min\{Q_j(\Phi_2\cup\{u_h\}),\theta_j\}-min\{Q_j(\Phi_2),\theta_j\}=0$; 2) for the case $q_{h,j}>0$ and $Q_j(\Phi_1)\leq Q_j(\Phi_1\cup\{u_h\})\leq Q_j(\Phi_2)\leq Q_j(\Phi_2\cup\{u_h\})$, when $\theta_j>Q_j(\Phi_2\cup\{u_h\})$, we have $(min\{Q_j(\Phi_1\cup\{u_h\}),\theta_j\}-min\{Q_j(\Phi_1),\theta_j\})-(min\{Q_j(\Phi_2\cup\{u_h\}),\theta_j\}-min\{Q_j(\Phi_2),\theta_j\})=(Q_j(\Phi_1\cup\{u_h\})-Q_j(\Phi_1))-(Q_j(\Phi_2\cup\{u_h\})-Q_j(\Phi_2))>0$, due to the submodular property of $Q_j(\Phi)$; 3) for other cases, it is straightforward to get a similar result as that in Theorem 4. Thus, we have that $(min\{Q_j(\Phi_1\cup\{u_h\}),\theta_j\}-min\{Q_j(\Phi_1),\theta_j\})-(min\{Q_j(\Phi_2\cup\{u_h\}),\theta_j\}-min\{Q_j(\Phi_2),\theta_j\})\geq 0$ holds for all cases, which implies $f(\Phi_1\cup\{u_h\})-f(\Phi_1)\geq f(\Phi_2\cup\{u_h\})-f(\Phi_2)$. Therefore, $f(\Phi)$ is submodular.

2) Since $Q_j(\Phi)$ is an increasing submodular function with $Q_j(\Phi=\emptyset)=0$, $f(\Phi)$ is also an increasing function with $f(\Phi=\emptyset)=0$ according to Eq. 15. We has proved that $f(\Phi)$ is submodular. Therefore, when we replace $Q_j(\Phi)=\sum_{u_i\in\Phi}q_{i,j}$ by using a trivial increasing submodular function, the problem can still be re-formalized as an MISC/SC problem. Moreover, $f(\Phi)$ is a polymatroid integer-valued function on $2^{\mathcal{U}}$. Further, according to Lemma 2, Protocol 1 can still achieve a $(1+\ln\gamma)$-approximation solution, where $\gamma=max_{u_i\in\mathcal{U}}f(\{u_i\})$. ■

Procedure 2 The User Recruitment Strategy of E-SUR

```
Input: $\mathcal{U}$, $\mathcal{S}$, $\{q_{i,j}|u_i\in\mathcal{U},s_j\in\mathcal{S}\}$, $\{\theta_j|s_j\in\mathcal{S}\}$, $\{c_i|u_i\in\mathcal{U}\}$
Output: $\Phi$
$\Phi=\emptyset$; $g(\Phi)=0$;
while $g(\Phi)<\varphi\sum_{j=1}^{m}\theta_j$ do
    Select a user $u_i\in\mathcal{U}\backslash\Phi$ to maximize $\frac{\Delta_i g(\Phi)}{c_i}$;
    $\Phi=\Phi\cup\{u_i\}$;
return $\Phi$
```

Theorems 9 and 10 show that as long as the quality function is an increasing submodular function which can be secretly computed by using the secure operations in Definition 6, the proposed secure user recruitment protocol can be applied to other existing works.

4 The E-SUR Protocol

In this section, we propose a hEterogeneous Secure User Recruitment (E-SUR) protocol based on the secret sharing scheme, where users' costs are heterogeneous. Here, users' costs and sensing qualities as input privacy need to be protected simultaneously. We first propose the basic user recruitment strategy used in E-SUR protocol. Based on this, we propose the E-SUR protocol, followed by performance and security analysis.

4.1 The Building Blocks

Different from the defined utility function $f(\Phi)$ used in O-SUR, we propose a new utility function $g(\Phi)=\varphi\cdot f(\Phi)$ in E-SUR. That is,

$$g(\Phi)=\varphi\sum_{j=1}^{m}\min\{Q_j(\Phi),\theta_j\}=\varphi\sum_{j=1}^{m}\min\{\sum_{u_i\in\Phi}q_{i,j},\theta_j\}. \quad (24)$$

where $\varphi=\max\{\varphi_1,\varphi_2\}$ is a constant related to the approximation ratio of the E-SUR protocol, in which $\varphi_1=\max\{\frac{c_i|1\leq i\leq n}{\theta_j-Q_j(\Phi)|1\leq j\leq m,Q_j(\Phi)<\theta_j,\Phi\subset\mathcal{U}}\}$ and $\varphi_2=\frac{\sum_{i=1}^{n}c_i}{\sum_{j=1}^{m}\theta_j}$. The derivation of φ_1 and φ_2 is shown in the proof of the approximation ratio (i.e., Theorem 13). Also, we use $\Delta_i g(\Phi)$ to denote the incremental utility $g(\Phi)$ of adding a new user u_i into Φ, i.e.,

$$\Delta_i g(\Phi)=g(\Phi\cup\{u_i\})-g(\Phi). \quad (25)$$

According to this, the procedure of recruiting users adopted in the E-SUR protocol is shown in Procedure 2. Here, the adopted greedy strategy in Procedure 2 is based on $\frac{\Delta_i g(\Phi)}{c_i}$. That is to say, the user who improves the utility $g(\Phi)$ per cost the most, i.e., the user u_i who can maximize the value of $\frac{\Delta_i g(\Phi)}{c_i}$, will be selected first. The user recruitment process terminates when $g(\Phi)=\varphi\sum_{j=1}^{m}\theta_j$.

4.2 The Detailed E-SUR Protocol

According to the building blocks, we propose the E-SUR protocol adopting the same utility function (i.e., $g(\Phi)$) and greedy strategy as Procedure 2 to recruit users. In the E-SUR protocol, the inputs including all sensing qualities (i.e., $\{q_{i,j}|\forall u_i\in\mathcal{U},s_j\in\mathcal{S}\}$) and recruitment costs (i.e., $\{c_i|\forall u_i\in\mathcal{U}\}$), and the computation are conducted by using the secret sharing techniques. That is, each input $q_{i,j}$ (also for c_i) is seen as a secret and it is replaced by its polynomial shares $[q_{i,j}]$ (accordingly $[c_i]$). Similar to the O-SUR protocol, when users jointly make recruitment decisions, all computations are conducted by using the secure operations in Definition 6, and all intermediate results are produced in the manner of shared secrets. To this end, we replace $u_i \in \Phi$ and $\Delta_i g(\Phi)$ with $[b_i]=[1]$ and $\varphi\sum_{j=1}^{m}\min\{q_{i,j}, \theta_j - Q_j(\Phi)\}$, and further hide the maximum incremental utility and the selected user in a $SecMax$ operation and a $SecCmp$ operation to prevent the selected user from being revealed in each round of iteration. Only in the final phase, each user u_i can reconstruct the result of b_i by collecting the corresponding shares. Afterwards, each user u_i knows whether she/he is recruited or not. After the recruited users conduct the sensing tasks and upload the results to the requester, the requester will pay the recruited users.

The detailed E-SUR protocol is shown in Protocol 2, which has the similar structure as the O-SUR protocol. The difference lies in that the recruited cost (i.e., c_i) of each user u_i is also protected from being revealed, and the adopted greedy strategy is to maximize $[\widehat{\Delta_i g}]$ (i.e., $[\Delta_i g][\alpha_i]$ where $\alpha_i = \frac{1}{c_i}$) in Protocol 2 instead of $[\Delta_i f]$ in Protocol 1. More specifically, in Steps 3-4, each user constructs the polynomial secret shares of their cost values as the input in addition to their sensing quality values. In Steps 18-21, the users jointly determine the maximum incremental utility value per cost, i.e., $\frac{\Delta_i g(\Phi)}{c_i}$, which is denoted by $[\widehat{\Delta_{max} g}]$ in Protocol 2. After n rounds of iterations, the protocol terminates and each user reconstructs the final recruited results.

Now, we analyze the computation and communication complexity of Protocol 2. We get that the operations $SecMin$ in Steps 16 and 25, $SecMulti$ in Steps 18-19, and $SecMax$ in Step 20 involve n^2m, n^2 and n invocations of secure multiplication operations, respectively. The complexity of the whole protocol is dominated by the three parts. According to Eqs. 13 and 14, we grasp that the $SecMin$ and $SecMax$ operations require $n^2 l$ ($l = \lceil \log_2 p \rceil$) bit-operations per user and l rounds of communications, respectively. Thus, Protocol 2 will lead to $O(mn^4 l)$ bit-operations per user and $O(mn^2 l)$ rounds of communications.

4.3 The Performance and Security Analysis

We first prove the correctness of Procedure 2, i.e.,

Theorem 11. Procedure 2 is correct. That is, 1) Procedure 2 will terminate; 2) $g(\Phi) = \varphi\sum_{j=1}^{m}\theta_j$ iff Φ is a user set that can execute the tasks in $\mathcal{S}$ so that the total sensing qualities of all tasks are not less than their thresholds.

Proof: 1) In Procedure 2, only one user will be added into the user set Φ in each round of iteration. In the worst case, after all n users are added into Φ, we have $g(\Phi) = \varphi\sum_{j=1}^{m}\theta_j$ and the protocol will terminate.

2) On one hand, $g(\Phi) = \varphi\sum_{j=1}^{m}\theta_j$ only when $\min\{Q_j(\Phi), \theta_j\} = \theta_j$ for $\forall j \in [1, m]$, indicating $\theta_j \le Q_j(\Phi)$ for $\forall j \in [1, m]$. Based on this, the total sensing qualities of tasks are not less than their thresholds. On the other hand, if Φ is a user set which can ensure that the total sensing qualities of tasks are not less than the threshold, i.e., $\theta_j \le Q_j(\Phi)$ for $\forall j \in [1, m]$. We directly have $g(\Phi) = \varphi\sum_{j=1}^{m}\theta_j$. Thus, the theorem is correct. ■

Next, we analyze the performance of Procedure 2. Before this, we explore several features of $g(\Phi)$ and $C(\Phi)$.

Protocol 2 The E-SUR Protocol

```
Input: $\mathcal{U}$, $\mathcal{S}$, $\{q_{i,j}|u_i \in \mathcal{U}, s_j \in \mathcal{S}\}$, $\{\theta_j|s_j \in \mathcal{S}\}$, $\{c_i|u_i \in \mathcal{U}\}$
Output: $b_1, \cdots, b_n$
Phase 1: the requester publishes $\mathcal{S}$ to $\mathcal{U}$ via the platform;
Phase 2: users input their sensing quality vectors;
 1: for $i = 1$ to $n$ do
 2:     user $u_i$ determines the sensing qualities $q_{i,1}, \cdots, q_{i,m}$;
 3:     user $u_i$ generates $\alpha_i = \frac{1}{c_i}$ and polynomial sharing $[\alpha_i]$;
 4:     user $u_i$ sends the share $\alpha_i[i']$ to user $u_{i'}$;
 5:     for $j = 1$ to $m$ do
 6:         user $u_i$ generates the polynomial sharing $[q_{i,j}]$;
 7:         user $u_i$ sends the share $q_{i,j}[i']$ to user $u_{i'}$;
Phase 3: users jointly make the decision of user recruitment;
 8: for $i = 1$ to $n$ do
 9:     $[b_i] \leftarrow [0]$;
10: for $j = 1$ to $m$ do
11:     $[Q_j] \leftarrow [0]$;
12: for $round = 1$ to $n$ do
13:     for $i = 1$ to $n$ do
14:         $[\Delta_i g] \leftarrow [0]$, $[\widehat{\Delta_i g}] \leftarrow [0]$;
15:         for $j = 1$ to $m$ do
16:             $[\delta] \leftarrow SecMin([q_{i,j}], SecSub(\theta_j, [Q_j]))$;
17:             $[\Delta_i g] \leftarrow SecAdd([\Delta_i g], [\delta])$;
18:         $[\Delta_i g] \leftarrow SecMulti([\Delta_i g], SecSub([1], [b_i]))$;
19:         $[\widehat{\Delta_i g}] \leftarrow SecMulti([\Delta_i g], [\alpha_i])$;
20:     $[\widehat{\Delta_{max} g}] \leftarrow SecMax([\widehat{\Delta_1 g}], \cdots, [\widehat{\Delta_n g}])$;
21:     for $i = 1$ to $n$ do
22:         $[z] \leftarrow SecCmp([\widehat{\Delta_{max} g}], [\widehat{\Delta_i g}])$;
23:         $[b_i] \leftarrow SecAdd([b_i], SecMulti(SecSub([1], [b_i]), [z]))$;
24:         for $j = 1$ to $m$ do
25:             $[\delta] \leftarrow SecMin([q_{i,j}], SecSub(\theta_j, [Q_j]))$;
26:             $[Q_j] \leftarrow SecAdd([Q_j], SecMulti([z], [\delta]))$;
Phase 4: the users reconstruct the results;
27: for $i = 1$ to $n$ do
28:     user $u_i$ collects all shares of $[b_i]$;
29:     user $u_i$ derives $b_i = \sum_{j=1}^{m} b_i[j]$;
```

Lemma 3. $g(\Phi)$ and $C(\Phi)$ are submodular functions.

Proof: 1) Since $f(\Phi)$ is a submodular function (Theorem 4) and we let $g(\Phi) = \varphi \cdot f(\Phi)$ where φ is a constant, we get that $g(\Phi)$ is also a submodular function.

2) $C(\Phi) = \sum_{u_i \in \mathcal{U}} c_i$ is submodular iff, for two sets Φ_1 and Φ_2, $\Phi_1 \subseteq \Phi_2$, and $\forall u_h \in \mathcal{U} \setminus \Phi_2$, we have $C(\Phi_1 \cup \{u_h\}) - C(\Phi_1) \ge C(\Phi_2 \cup \{u_h\}) - C(\Phi_2)$. It is straightforward to verify that the equation holds. Actually, $C(\Phi_1 \cup \{u_h\}) - C(\Phi_1)$ is always equal to $C(\Phi_2 \cup \{u_h\}) - C(\Phi_2)$. Hence, $C(\Phi)$ is a submodular function. ■

Theorem 12. $g(\Phi)$ and $C(\Phi)$ are two polymatroid functions on $2^{\mathcal{U}}$.

Proof: According to Theorems 2, 4 and Lemma 3, $g(\Phi)$ and $C(\Phi)$ are two increasing, submodular functions with $g(\phi) = 0$ and $C(\phi) = 0$, we get that $g(\Phi)$ and $C(\Phi)$ are two polymatroid functions on $2^{\mathcal{U}}$. ■

Now, we analyze the performance of Procedure 2.

First, the E-SUR problem can be re-formalized a Minimum Fractional Submodular Cover with Submodular Cost (MFSC/SC) problem by replacing the constraint (i.e., Eq.9) with $g(\Phi) = g(\mathcal{U})$, i.e.,

$$Minimize\ \{C(\Phi)|g(\Phi) = g(\mathcal{U}), \Phi \subseteq \mathcal{U}\}, \qquad (26)$$

where $g(\Phi)$ and $C(\Phi)$ are two increasing submodular and further polymatroid functions with $g(\Phi=\phi)=0$ and $C(\Phi=\phi)=0$ according to Theorems 2 and 12 and Lemma 3. Note that "fractional" here means that $g(\Phi)$ is a polymatroid fraction-valued function on $2^{\mathcal{U}}$. This is because $g(\Phi)=\varphi f(\Phi)$, while $\varphi=\max\{\varphi_1,\varphi_2\}$ is a fraction constant.

Second, we introduce a lemma about the approximation ratio of MFSC/SC problems in [25].

Lemma 4. Consider an MFSC/SC problem: Minimize $\{C(\Phi)|g(\Phi)=g(\mathcal{U}),\Phi\subseteq\mathcal{U}\}$, in which $g(\cdot)$ is a polymatroid function on $2^{\mathcal{U}}$, and $g(\mathcal{U})\geq opt$ where opt is the optimal recruited cost of satisfying the sensing quality threshold constraints. If the selected criterion of a greedy algorithm for this problem always satisfies $\frac{\Delta_i g(\Phi)}{c_i}\geq 1$, then the greedy algorithm can achieve a $(1+\rho\ln\frac{g(\mathcal{U})}{opt})$-approximation solution. Moreover, if $C(\Phi)$ is a modular function, then $\rho=1$.

Based on this, we have the following theorem:

Theorem 13. Procedure 2 can produce a $(1+\ln\frac{\varphi\sum_{j=1}^{m}\theta_j}{opt})$-approximation solution, in which opt is the cost of the optimal solution for the E-SUR problem.

Proof: 1) Since the user set $\mathcal{U}$ must be a feasible solution, we have $g(\mathcal{U})=\varphi\sum_{j=1}^{m}\theta_j$. According to $\varphi=\max\{\varphi_1,\varphi_2\}$ in which $\varphi_1=\max\{\frac{c_i|1\leq i\leq n}{\theta_j-Q_j(\Phi)|1\leq j\leq m,Q_j(\Phi)<\theta_j,\Phi\subset\mathcal{U}}\}$ and $\varphi_2=\frac{\sum_{i=1}^{n}c_i}{\sum_{j=1}^{m}\theta_j}$, we get that $g(\mathcal{U})\geq\varphi_2\sum_{j=1}^{m}\theta_j\geq\sum_{i=1}^{n}c_i\geq opt$.

2) Without loss of generality, we denote the recruited user in the last round of iteration as u_h, and denote the recruited user set of this round as Φ' (excluding u_h). Moreover, we have $\Phi\subseteq\Phi'$. At this moment, there must be at least a task whose obtained total sensing quality is less than its threshold; otherwise, the algorithm would have terminated before. For simplicity, let s_j be such a task. Based on this, we have $Q_j(\Phi')<\theta_j$ while $Q_j(\Phi'\cup\{u_h\})\geq\theta_j$. Thus, we have

$$\frac{g(\Phi\cup\{u_i\})-g(\Phi)}{c_i}\geq\frac{g(\Phi\cup\{u_h\})-g(\Phi)}{c_h} \tag{27}$$

$$\geq\frac{g(\Phi'\cup\{u_h\})-g(\Phi')}{c_h} \tag{28}$$

$$\geq\varphi\frac{(\min\{Q_j(\Phi'\cup\{u_h\}),\theta_j\}-\min\{Q_j(\Phi'),\theta_j\})}{c_h} \tag{29}$$

$$\geq\varphi\frac{\theta_j-Q_j(\Phi')}{c_h}\geq\varphi_1\frac{\theta_j-Q_j(\Phi')}{c_h}\geq 1, \tag{30}$$

where Eq. 27 indicates that user u_i is the optimal selection for user set Φ, while Eq. 28 is based on the submodular property of $g(\Phi)$. Now, we get that our greedy strategy satisfies the property of Lemma 4. Based on this, we get that Procedure 2 is a $(1+\ln\frac{\varphi\sum_{j=1}^{m}\theta_j}{opt})$-approximation solution. ■

Since Protocol 2 is actually a distributed version of Procedure 2 combined with secret sharing schemes, Protocol 2 can achieve the same user recruitment result as Procedure 2. So we have the following theorem:

Theorem 14. Protocol 2 is correct, and it can also produce a $(1+\ln\frac{\varphi\sum_{j=1}^{m}\theta_j}{opt})$-approximation solution, where opt means the cost of the optimal solution for E-SUR problem, and $\varphi=\max\{\varphi_1,\varphi_2\}$ in which $\varphi_1=\max\{\frac{c_i|1\leq i\leq n}{\theta_j-Q_j(\Phi)|1\leq j\leq m,Q_j(\Phi)<\theta_j,\Phi\subset\mathcal{U}}\}$ and $\varphi_2=\frac{\sum_{i=1}^{n}c_i}{\sum_{j=1}^{m}\theta_j}$.

Also, we can prove the security of Protocol 2 against any semi-honest adversaries.

Theorem 15. Protocol 2 can protect the sensing qualities and recruitment cost of each user from being revealed to any κ semi-honest adversaries and the platform, even if they might collude. Here, κ means the degree of polynomial sharing, which may be any integer less than n.

Proof: Compared to Protocol 1, the E-SUR protocol (i.e., Protocol 2) involves the privacy-preserving issue about the recruitment costs of all users during the secure user recruitment process. In the security proof of Protocol 1, we have proved that all mathematical operations used in Protocols 1 and 2 are secure according to [7, 16, 17]. Here, we prove the security of E-SUR from itself. Similar to Protocol 1, we also consider any κ users, denoted by $\mathcal{I}=\{u_{i_1},\cdots,u_{i_\kappa}\}\subset\mathcal{U}$, and construct the view of each user $VIEW_{i_t}$ ($u_{i_t}\in\mathcal{I}$). During the whole user recruitment process of Protocol 2, we get that $M_{i_t}=\{q_{i,j}[i_t],\alpha_i[i_t],b_i[i_t],Q_j[i_t],\delta[i_t],\Delta_i g[i_t],\widehat{\Delta_i g}[i_t],\widehat{\Delta_{max} g}[i_t], z[i_t]\}$ and $VIEW_{i_t}=(\{q_{i_t,j},\alpha_{i_t},n,m,\theta_j\},r,M_{i_t})$. Then, we also construct a simulator for an arbitrary user in $\mathcal{I}$ such that its view can be efficiently simulated by the output of the simulator. The simulator for the user $u_{i_t}\in\mathcal{I}$ randomly selects two numbers $q'_{i_t,j}$ and α'_{i_t} from the prime filed $\mathbb{Z}_p$. Since both $q_{i_t,j}$, α_{i_t} and $q'_{i_t,j}$, α'_{i_t} are the numbers randomly selected from $\mathbb{Z}_p$, the output of the simulator and the view are computational indistinguishability. Thus, we get that Eq. 1 holds for E-SUR, and further conclude that the whole protocol is secure [7]. ■

5 PERFORMANCE EVALUATION

We evaluate the O-SUR and E-SUR protocols from two aspects: the user recruitment and the privacy-preserving mechanism, i.e., the secret sharing technique. When we assess the user recruitment performance, we do not take the privacy-preserving mechanism into consideration since it has no effect on the user recruitment results. When evaluating the performance of the adopted secret sharing technique, we only focus on the time efficiency because the security has been verified by theoretical analysis. Furthermore, in order to demonstrate the advantage of our adopted privacy-preserving mechanism, we also compare it with other theoretically-provable security schemes in terms of time efficiency. More specifically, we first introduce the compared protocols used in our simulations and experiments, and then present the detailed simulation settings as well as the evaluation metrics. At last, we present and analyze the obtained simulation/experiment results.

5.1 Protocols in Comparison

First, to evaluate the user recruitment performance of the O-SUR and E-SUR protocols, we design two other user recruitment protocols adopting different selection strategies for comparison. Existing user recruitment protocols or algorithms involve various crowdsensing models (e.g., competition-based model, probabilistic model, etc.), constraints (e.g., delay constraint, budget constraint, etc.), and optimization objectives (e.g., maximizing spatial/temporal coverage, maximizing sensing qualities, etc.). Most of them adopt the greedy strategy to recruit users (e.g., [10, 11, 14, 20, 26, 31]). In these works, the users who can accomplish all sensing tasks with minimum costs are recruited first. Meanwhile, these users are subject to the constraints of

TABLE 2: Evaluation Settings

parameter name	default	range
number of users n	200	100-500
number of tasks m	100	50-250
average sensing quality $\overline{q}$	30	10-90
variance of sensing qualities σ	0.4	0.2-1.0
average sensing quality threshold $\overline{\theta}$	100	50-250
variance of thresholds μ	0.2	0.1-0.5
largest number of tasks per user ρ	20	15-35
average recruited cost of users $\overline{c}$	30	10-50
variance of recruited costs κ	0.3	0.1-0.5

some mobility models. For comparison, we borrow the basic strategy by ignoring other constraints in these works to design two compared user recruitment protocols, which are applicable to our model. We call the first protocol **MCUR**, in which the user who can perform the most tasks (*per cost* in E-SUR scenario) is recruited first [14, 26], i.e., $u_{i*} = argmax_{u_i \in \mathcal{U}/\Phi} \sum_{j=1}^{m} [q_{i,j}]$. Here, $[q_{i,j}] = 0$ when $q_{i,j} = 0$; otherwise, $[q_{i,j}] = 1$. Another protocol is denoted as **MQUR**, in which the user who performs tasks with the most sensing qualities (*per cost* in E-SUR scenario) is recruited first [10, 11, 20, 31], i.e., $u_{i*} = argmax_{u_i \in \mathcal{U}/\Phi} \sum_{j=1}^{m} q_{i,j}$. Together, the two compared protocols and our proposed protocols constitute the most typical greedy user recruitment strategies in crowdsensing.

Second, to prove that O-SUR and E-SUR can work well in real applications, we realize and run them on real smartphones. Here, to evaluate the time efficiency of the O-SUR and E-SUR protocols which adopt the secret-sharing-based secure user recruitment approach, we realize two other privacy-preserving techniques during the user recruitment process for comparison. Besides the secret sharing schemes, the homomorphic encryption and garbled circuit protocols can also be utilized to solve the privacy-preserving user recruitment problem [7]. Based on this, we implement two compared protocols as follows: Homomorphic-Encryption-based User Recruitment (**HEUR**) protocol [18] and Garbled-Circuit-based User Recruitment (**GCUR**) protocol [12]. In HEUR and GCUR, we turn each secure multi-party multiplication operation among n users to $\frac{n(n-1)}{2}$ secure two-party multiplication operations, and we use the homomorphic encryption and garbled circuit protocols to conduct these secure two-party multiplication operations.

5.2 Simulation Settings and Evaluation Metrics

We first introduce the *simulation settings* in the O-SUR, E-SUR, MCUR and MQUR protocols. For the simulations, synthetical traces are adopted, in which we can evaluate the user recruitment performance with different parameters as needed, while ignoring users' mobility models. Here, since O-SUR and E-SUR are designed for two different crowdsensing settings, we divide the simulations into two parts. The O-SUR protocol and two compared user recruitment protocols are conducted in the same simulation settings, and E-SUR is conducted with the compared recruitment protocols in other settings. Note that the simulation settings in the O-SUR and E-SUR problems are same, except for the recruitment costs of all mobile users. Hence, we first present the same simulation settings in both scenarios and then introduce the unique settings.

More specifically, we consider seven shared parameters, including the number of users n, the number of tasks m, the average sensing quality (denoted by $\overline{q}$), the variance of sensing qualities (denoted by σ), the largest number of tasks performed by each user (denoted by ρ), the average sensing quality threshold $\overline{\theta}$, and the variance of thresholds (denoted by μ). In each simulation, we change one parameter while keeping the other parameters fixed. In all simulations, each user u_i randomly selects a value from $(0, \rho]$ as the number of tasks that he/she can perform. For each selected task s_j, the sensing quality $q_{i,j}$ is set as a value randomly chosen from a range $[(1-\sigma)\overline{q}, (1+\sigma)\overline{q}]$. Moreover, for each sensing task s_j ($\in \mathcal{S}$), its total quality threshold θ_j is randomly generated from a range $[(1-\mu)\overline{\theta}, (1+\mu)\overline{\theta}]$.

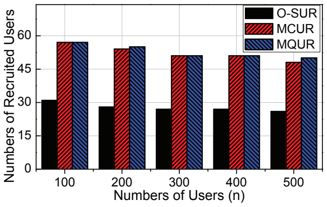


Fig. 3: $|\Phi|$ vs. n

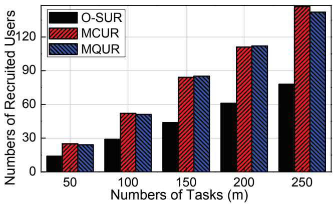


Fig. 4: $|\Phi|$ vs. m

Then, we present the unique setting in the O-SUR and E-SUR simulations. In the O-SUR simulations, the recruited costs of all mobile users are homogeneous and the O-SUR protocol does not involve the values, so the values of cost are not specifically given; while in the E-SUR simulations, the values of recruited cost are heterogeneous. Here, we use $\overline{c}$ and κ to denote the average recruited cost and the variance of cost, respectively. Then, for user u_i ($\in \mathcal{U}$), its recruited cost c_i is randomly generated from $[(1-\kappa)\overline{c}, (1+\kappa)\overline{c}]$. The range and default values of each parameter are illustrated in Table 2. Note that the default settings are used in all simulations unless otherwise specified.

Next, we present the *experiment settings* on real smartphones. To evaluate time efficiency, we realize and run O-SUR, E-SUR, HEUR and GCUR on a real smart phone (Huawei P9: EVA-AL00) with a 2.0GB memory and a processor of 4-core 2.2GHz plus 4-core 1.5GHz. We record the execution time of O-SUR, E-SUR, HEUR, and GCUR in this smartphone, while ignoring the communication time. During the execution, we use another smart phone to simulate the remaining $(n-1)$ users.

At last, we introduce the evaluation metrics in our simulations. We evaluate the performance of the proposed protocols mainly from two aspects: the total recruitment cost and time efficiency. Since the recruitment costs of all users are uniform in the O-SUR scenarios, we evaluate the number of recruited users in the O-SUR problem.

5.3 Simulation Results

First, we present the simulation results about the O-SUR and two compared protocols. Figs. 3 and 4 depict the number of recruited users vs. different numbers of users and tasks. The results show that the number of users recruited by O-SUR is much smaller than MCUR and MQUR. Moreover, when the number of tasks increases, more users are recruited. When we increase the number of users, less users are recruited. This is because when more candidate users emerge, there may be better selections than before, so fewer users are required to accomplish the same tasks. We record the number of recruited users while changing the other parameters (i.e., $\overline{q}$, σ, $\overline{\theta}$ and μ), as shown in

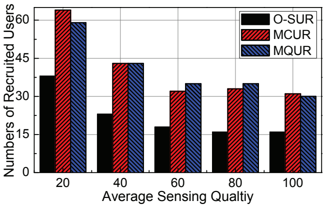


Fig. 5: $|\Phi|$ vs. $\overline{q}$

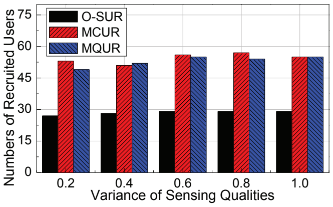


Fig. 6: $|\Phi|$ vs. σ

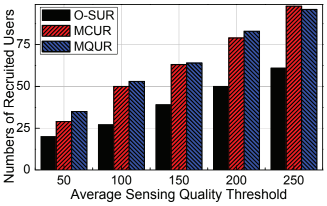


Fig. 7: $|\Phi|$ vs. $\overline{\theta}$

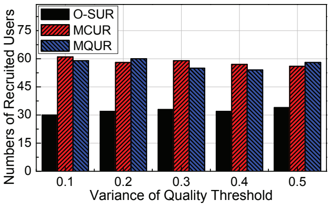


Fig. 8: $|\Phi|$ vs. μ

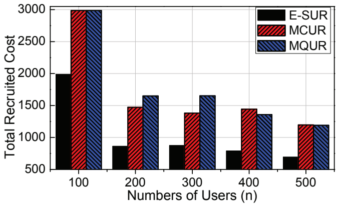


Fig. 9: $C(\Phi)$ vs. n

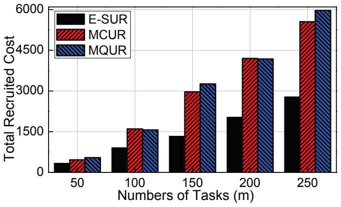


Fig. 10: $C(\Phi)$ vs. m

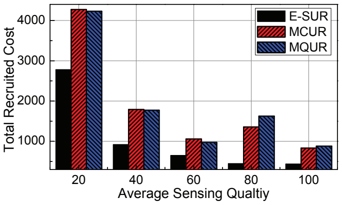


Fig. 11: $C(\Phi)$ vs. $\overline{q}$

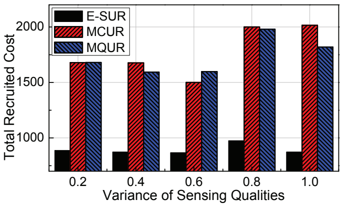


Fig. 12: $C(\Phi)$ vs. σ

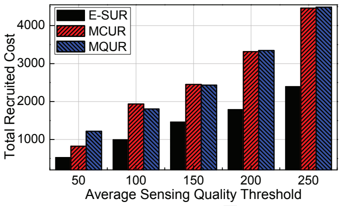


Fig. 13: $C(\Phi)$ vs. $\overline{\theta}$

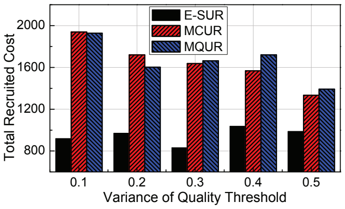


Fig. 14: $C(\Phi)$ vs. μ

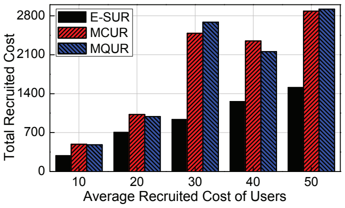


Fig. 15: $C(\Phi)$ vs. $\overline{c}$

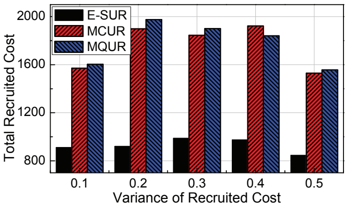


Fig. 16: $C(\Phi)$ vs. κ

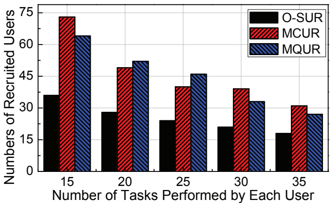


Fig. 17: $|\Phi|$ vs. ρ

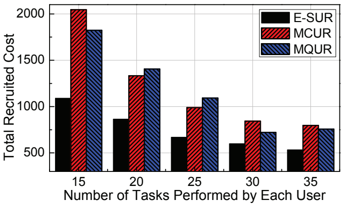


Fig. 18: $C(\Phi)$ vs. ρ

Figs. 5, 6, 7, and 8. Also, the performance results about the largest number of tasks per user is shown in Fig. 17. These results prove that O-SUR has a much better performance than MCUR and MQUR. Moreover, when we increase either the average sensing quality (i.e., $\overline{q}$) or the largest number of tasks performed by each user (i.e., ρ), the number of recruited users decreases. When the average sensing quality threshold (i.e., $\overline{\theta}$) increases, the number of recruited users increases accordingly.

Second, the simulation results about the E-SUR protocol are presented as follows. The performance comparisons in terms of the number of mobile users n, the number of tasks m, the average sensing quality $\overline{q}$ and the variance of sensing quality σ, are shown in Figs. 9, 10, 11, and 12. It also demonstrates the significant performance of E-SUR compared to MCUR and MQUR. Furthermore, we get that E-SUR achieves about 45.4% and 46.8% percent smaller total recruitment costs than MCUR and MQUR, respectively. On the other hand, the simulation results about the average sensing quality threshold $\overline{\theta}$, the variance of quality threshold μ, the average recruited cost of users $\overline{c}$, the variance of cost κ and the largest number of tasks performed by each user ρ are shown in Figs. 13, 14, 15, 16, and 18. By analyzing the results, we conclude that E-SUR achieves about 42.4% and 42.7% percent smaller total recruitment costs than the MCUR and MQUR protocols as a whole, respectively. At the same time, we get that when the number of tasks m, the average sensing quality $\overline{\theta}$ or the average recruitment cost of users $\overline{c}$ increases, the total costs of all protocols increase. However, along with the increase of the number of users n, the average sensing quality of tasks $\overline{q}$ or the largest number of tasks performed by each user ρ, the total costs decrease. These simulations validate our theoretical analysis results.

Third, we present the evaluation results of O-SUR, E-SUR, HEUR and GCUR on smartphones. We run the O-SUR, E-SUR, HEUR, and GCUR protocols in the smartphones by changing the number of users from 5 to 10, while setting $m=6$, $\overline{q}=30$, $\sigma=0.4$, $\overline{\theta}=100$, $\mu=0.2$, $\rho=m$, $\overline{c}=50$ and $\kappa=0.1$. The results are depicted in Fig. 19. When the number of users is larger than 5, HEUR cannot work well in the real smartphone since its run time has exceeded 10^5 ms. GCUR performs even worse than HEUR. Even 5 users can result in a run time of over 10^7 ms. In contrast, the run time of O-SUR is far less than that of HEUR and GCUR in magnitudes. This is because that the GCUR protocol needs to conduct considerable precise and complex Boolean circuit operations while the HEUR protocols requires massive encryption and decryption operations. As shown in Fig. 20, when the number of users is 50 and the number of tasks is 20, the execution time of O-SUR is less than 150s. Compared to the execution time of HEUR and GCUR (dozens of minutes or even hours), our protocol is quite efficient, which means that it can work well in real smartphones. Similarly, the time efficiency of E-SUR is also outstanding compared to HEUR and GCUR, as shown in Figs. 21 and 22. More precisely, the execution time of E-SUR is less than 200s, when the numbers of users and tasks are set as 50 and 20, respectively. The results indicate that both O-SUR and E-SUR can work well in real applications. So, implementing and running the proposed protocols on smartphones in reality is feasible.

6 RELATED WORK

Most works about mobile crowdsensing focus on the user recruitment problem [5, 8, 10, 11, 14, 20, 26, 31] and the task allocation problem [1, 2, 9, 23, 29, 35].

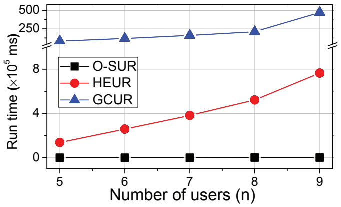


Fig. 19: Run Time of O-SUR, HEUR and GCUR

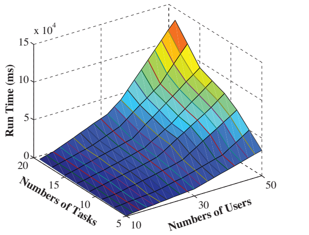


Fig. 20: Run time of O-SUR vs. n and m

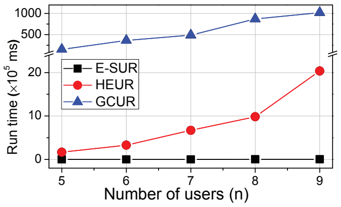


Fig. 21: Run Time of E-SUR, HEUR and GCUR

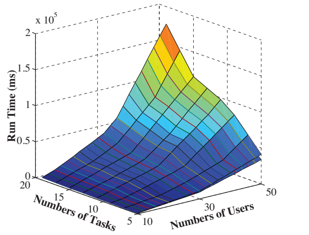


Fig. 22: Run time of E-SUR vs. n and m

On one hand, M. Karaliopoulos *et al.* in [14] propose two greedy algorithms to recruit some mobile users who can perform location-related sensing tasks with a minimum cost; Y. Han *et al.* in [8] propose a dynamic programming algorithm, and further design two distributed algorithms to solve the competition-based participant recruitment problem for delay-sensitive crowdsensing scenarios; Z. He *et al.* in [10] propose a greedy approximation algorithm and a genetic algorithm for the user recruitment problem in vehicle-based crowdsensing, which can achieve nearly optimal spatial and temporal coverage with a limit budget; L. Pu *et al.* in [20] advocate a mobile crowdsourcing paradigm called Crowdlet in which the service quality based on keywords is considered; D. Zhang *et al.* in [31] propose a novel participant recruitment framework, called CrowdRecruiter, for the energy-efficient Piggyback Crowdsensing task model, which focus on minimizing incentive payments under the probabilistic coverage constraint. But none of them has taken the privacy-preserving issues into consideration when conducting user recruitment in crowdsensing.

On the other hand, A. Chatterjee *et al.* in [1] studied the task allocation problem, in which each task might include multiple steps, and each step requires different skills; M. Cheung *et al.* in [2] design an asynchronous and distributed task selection algorithm for the deadline-sensitive and location-dependent task allocation problem in mobile crowdsensing; S. He *et al.* in [9] considered the maximum net reward task allocation problem with the constraint of time budgets; W. Sun *et al.* in [23] propose a fairness-aware distributed approach to maximize the aggregate data utility of heterogeneous sensing tasks within a given budget. None of these studies has taken into consideration the secure user recruitment problem for mobile crowdsensing.

Additionally, many incentive mechanisms such as [19, 28, 32–34] have been designed for stimulating mobile users to participate in mobile crowdsensing. For example, D. Peng *et al.* in [19] design an incentive mechanism to motivate the rational crowdsensing participants to perform data sensing efficiently. In other words, the participants get payments according to their effective contributions in the form of qualities of sensing data. Y. Wei *et al.* in [28] firstly propose the two-sided online interactions among service users and service providers for dynamic mobile crowdsensing. Q. Zhang *et al.* in [33] propose an incentive mechanism to stimulate crowd workers to undertake crowd labeling tasks under a budget constraint. H. Zhang *et al.* in [32] propose a multi-market dynamic double auction mechanism for the proximity-based mobile crowd service systems.

So far, only a few works have studied the privacy issues in mobile crowdsensing systems. For example, Q. Wang *et al.* in [27] investigate the problem of continuous real-time spatiotemporal crowd-sourced data publishing, and design a privacy-preserving online data publishing scheme based on differential privacy. G. Zhuo *et al.* in [36] propose a privacy-preserving verifiable data aggregation and analysis scheme based on homomorphic encryption for cloud-assisted mobile crowdsourcing. X. Jin *et al.* in [13] present a framework for a crowdsourced spectrum sensing service provider to select spectrum-sensing participants, in which the differential privacy is adopted to protect the locations of mobile participants. However, none of these studies investigates the privacy-preserving problem in the user recruitment process. To the best of our knowledge, our proposed protocols are the first privacy-preserving user recruitment protocols designed for mobile crowdsensing systems.

7 CONCLUSION

We propose two secure user recruitment protocols, i.e., O-SUR and E-SUR, for sensing-quality-aware mobile crowdsensing systems. O-SUR applies to the scenario where the recruitment costs of users are homogeneous, while E-SUR is designed for the case in which the recruitment costs are heterogeneous. Both of them adopt greedy strategies to recruit users and use secret sharing schemes to protect users' privacy. The difference lies in that O-SUR and E-SUR adopt two unique utility functions. We prove that both O-SUR and E-SUR can produce a solution with a logarithmic approximation ratio, and they can protect the inputs of each user from being revealed to the platform or to other users, even if they might collude. The simulation results show that O-SUR and E-SUR can work well in real smartphones.

REFERENCES

[1] A. Chatterjee, M. Borokhovich, L. R. Varshney, and S. Vishwanath. Efficient and flexible crowdsourcing of specialized tasks with precedence constraints. In *IEEE INFOCOM*, 2016.

[2] M. H. Cheung, R. Southwell, F. Hou, and J. Huang. Distributed time-sensitive task selection in mobile crowdsensing. In *ACM MobiHoc*, 2015.

[3] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick. A survey on privacy in mobile participatory sensing applications. *Journal of systems and software*, 84(11):1928–1946, 2011.

[4] R. K. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: Current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, 2011.

[5] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu. Truthful incentive mechanism for nondeterministic crowdsensing with vehicles. *IEEE Transactions on Mobile Computing*, 17(12):2982–2997, 2018.

[6] R. Gennaro, M. O. Rabin, and T. Rabin. Simplified vss and fast-track multiparty computations with applications to threshold cryptography. In *ACM PODC*, 1998.

[7] O. Goldreich. *Foundations of Cryptography: Volume 2 - Basic Applications*. Cambridge University Press, 2004.

[8] Y. Han, T. Luo, D. Li, and H. Wu. Competition-based participant recruitment for delay-sensitive crowdsourcing applications in D2D networks. *IEEE Transactions on Mobile Computing*, 15(12):2987–2999, 2016.

[9] S. He, D.-H. Shin, J. Zhang, and J. Chen. Toward optimal allocation of location dependent tasks in crowdsensing. In *IEEE INFOCOM*, 2014.

[10] Z. He, J. Cao, and X. Liu. High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility. In *IEEE INFOCOM*, 2015.

[11] M. Hu, Z. Zhong, Y. Niu, and M. Ni. Duration-variable participant recruitment for urban crowdsourcing with indeterministic trajectories. *IEEE Transactions on Vehicular Technology*, 66(11):10271–10282, 2017.

[12] Y. Huang, D. Evans, J. Katz, and L. Malka. Faster secure two-party computation using garbled circuits. In *USENIX Security Symposium*, volume 201, pages 331–335, 2011.

[13] X. Jin and Y. Zhang. Privacy-preserving crowdsourced spectrum sensing. In *IEEE INFOCOM*, 2016.

[14] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos. User recruitment for mobile crowdsensing over opportunistic networks. In *IEEE INFOCOM*, 2015.

[15] L. Kong, L. He, X. Y. Liu, Y. Gu, M. Y. Wu, and X. Liu. Privacy-preserving compressive sensing for crowdsensing based trajectory recovery. In *IEEE ICDCS*, 2015.

[16] P. Lory. Secure distributed multiplication of two polynomially shared values: Enhancing the efficiency of the protocol. In *SECURWARE*, 2009.

[17] T. Nishide and K. Ohta. Multiparty computation for interval, equality, and comparison without bit-decomposition protocol. In *Public Key Cryptography*, 2007.

[18] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238, 1999.

[19] D. Peng, F. Wu, and G. Chen. Pay as how well you do: a quality based incentive mechanism for crowdsensing. In *ACM MobiHoc*, 2015.

[20] L. Pu, X. Chen, J. Xu, and X. Fu. Crowdlet: Optimal worker recruitment for self-organized mobile crowdsourcing. In *IEEE INFOCOM*, 2016.

[21] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[22] T. Shu, Y. Chen, and J. Yang. Protecting multi-lateral localization privacy in pervasive environments. *IEEE/ACM Transactions on Networking*, 23(5):1688–1701, 2015.

[23] W. Sun, Y. Zhu, L. M. Ni, and B. Li. Crowdsourcing sensing workloads of heterogenous tasks: A distributed fairness-aware approach. In *IEEE ICPP*, 2015.

[24] K. Vu, R. Zheng, and J. Gao. Efficient algorithms for k-anonymous location privacy in participatory sensing. In *IEEE INFOCOM*, 2012.

[25] P. Wan, D. Du, P. Pardalos, and W. Wu. Greedy approximations for minimum submodular cover with submodular cost. *Computer Optimization and Applications*, 45(1):463–474, 2010.

[26] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang. An efficient prediction-based user recruitment for mobile crowdsensing. *IEEE Transactions on Mobile Computing*, 17(1):16–28, 2018.

[27] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren. Rescuedp: Real-time spatio-temporal crowdsourced data publishing with differential privacy. In *IEEE INFOCOM*, 2016.

[28] Y. Wei, Y. Zhu, H. Zhu, Q. Zhang, and G. Xue. Truthful online double auctions for dynamic mobile crowdscouring. In *IEEE INFOCOM*, 2015.

[29] M. Xiao, J. Wu, L. Huang, Y. Wang, and C. Liu. Multi-task assignment for crowdsensing in mobile social networks. In *IEEE INFOCOM*, 2015.

[30] M. Xiao, J. Wu, S. Zhang, and J. Yu. Secret-sharing-based secure user recruitment protocol for mobile crowdsensing. In *IEEE INFOCOM*, 2017.

[31] D. Zhang, H. Xiong, L. Wang, and G. Chen. Crowdrecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In *ACM Ubicomp*, 2014.

[32] H. Zhang, B. Liu, H. Susanto, G. Xue, and T. Sun. Incentive mechanism for proximity-based mobile crowd service systems. In *IEEE INFOCOM*, 2016.

[33] Q. Zhang, Y. Wen, X. Tian, X. Gan, and X. Wang. Incentivize crowd labeling under budget constraint. In *IEEE INFOCOM*, 2015.

[34] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang. Truthful incentive mechanisms for crowdsourcing. In *IEEE INFOCOM*, 2015.

[35] Q. Zhao, Y. Zhu, H. Zhu, J. Cao, G. Xue, and B. Li. Fair energy-efficient sensing task allocation in participatory sensing with smartphones. In *IEEE INFOCOM*, 2014.

[36] G. Zhuo, Q. Jia, L. Guo, M. Li, and P. Li. Privacy-preserving verifiable data aggregation and analysis for cloud-assisted mobile crowdsourcing. In *IEEE INFOCOM*, 2016.

Mingjun Xiao is a professor in the School of Computer Science and Technology at the University of Science and Technology of China (USTC). He received his Ph.D. from USTC in 2004. His research interests include crowdsourcing, mobile social networks, mobile cloud computing, blockchain, data security and privacy. He has published more over 80 papers in referred journals and conferences, including TMC, TC, TPDS, TKDE, TSC, INFOCOM, ICNP, etc. He served as the TPC member of INFOCOM'20, DASFAA'20, INFOCOM'19, ICDCS'19, DASFAA'19, etc. He is on the reviewer board of several top journals such as TMC, TON, TPDS, TSC, TVT, TCC, etc.

Guoju Gao received his B.S. degree in information security from the University of Science and Technology of Beijing, Beijing, China, in 2014. He is currently working toward a PhD degree on computer science and technology with the School of Computer Science and Technology, the University of Science and Technology of China, Hefei, China. His research interests include privacy preservation, mobile crowdsensing and incentive mechanisms.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless net- works, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Service Computing and the Journal of Parallel and Distributed Computing. Dr. Wu was general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

Sheng Zhang received the B.S. and Ph.D. degrees from Nanjing University, in 2008 and 2014, respectively. He is an assistant professor in Nanjing University and a member of the State Key Lab. for Novel Software Technology. His research interests include cloud computing and mobile networks. His publications include those appeared in the IEEE TMC, TPDS, TC, INFOCOM, ICDCS, and ACM MobiHoc. He received the Best Paper Runner-Up Award from IEEE MASS 2012. He is a member of the IEEE.

Liusheng Huang received his MS degree in computer science from University of Science and Technology of China, Anhui, in 1988. He is a professor at the School of Computer Science and Technology, University of Science and Technology of China. His main research interests include delay tolerant networks and Internet of things. He serves on the editorial board of many journals. He has published 6 books and more than 200 papers.