Secret-Sharing-Based Secure User Recruitment Protocol for Mobile Crowdsensing

Mingjun Xiao*, Jie Wu†, Sheng Zhang‡, and Jiapeng Yu*
*School of Computer Science and Technology / Suzhou Institute for Advanced Study,
University of Science and Technology of China
†Department of Computer and Information Sciences, Temple University
‡Department of Computer Science and Technology, Nanjing University

***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. An important problem is recruiting the minimum number of users while achieving a satisfactory sensing quality for each task. Meanwhile, in order to ease users' worries about privacy disclosures, the user recruitment process needs to protect each user's sensing quality information from being revealed to other users or to the platform. We prove that this problem is NP-hard. To solve this problem, we first propose a Basic User Recruitment (BUR) protocol based on a greedy strategy, which can recruit nearly the minimum amount of users while ensuring that the total sensing quality of each task is no less than a given threshold. Based on BUR, we further propose a Secure User Recruitment (SUR) protocol by using secret sharing schemes. We analyze the approximation ratio and prove the security of the SUR protocol in the semi-honest model. Moreover, we extend SUR to deal with a more general case where the total sensing quality of each task might be an increasing submodular function. Finally, we demonstrate the significant performance of the proposed protocol through extensive simulations and execution in real smartphones.**

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

I. Introduction

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, wireless indoor localization, and so on, attracting much attention [3]. A typical mobile crowdsensing system consists of a collection of mobile users and a platform residing on the cloud. The platform accepts the sensing tasks from some 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 the 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 [6], [7], [9]. Also, many incentive mechanisms such as [12], [18]–[20] 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. Sensing quality can be measured by the number, time, and clarity of the photos. During the user recruitment process, each mobile user needs to tell the platform which tasks he/she can deal with and how well he/she can perform each task. This might reveal some private sensitive information. The tasks that a user can perform will reveal which locations the user might visit. The corresponding sensing quality will reveal the frequency, time, distance of the visit, and so on. In order to avoid privacy disclosures, 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 [2], [8], [17], [21], studied the problem of protecting the privacy of sensing results collected by mobile users. Furthermore, none of them investigate the privacy-preserving problem in the user recruitment process. In the privacy-preserving user recruitment 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. The differential privacy schemes in existing works are not competent for this problem since many complex and precise computations need to be conducted over users' private inputs. Although the homomorphic encryption and garbled circuit protocols can solve this problem, they will result in a huge computational overhead that is unacceptable to mobile users.

To solve the privacy-preserving user recruitment problem, we design a Basic User Recruitment (BUR) protocol based on a greedy strategy and apply secret sharing techniques in BUR to propose a Secure User Recruitment (SUR) protocol. More specifically, our major contributions include:

1) We consider a mobile crowdsensing system in which the total sensing quality of each task is the sum of the

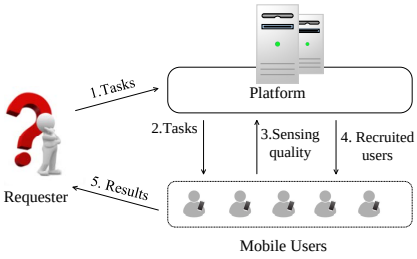


Fig. 1. The crowdsensing model

sensing quality of each user performing this task. We prove that the user recruitment problem in this system is NP-hard. Then, we design the BUR protocol based on a greedy user recruitment strategy for this problem. Moreover, we prove that BUR can produce a solution with a logarithmic approximation ratio.

2) We propose the SUR protocol based on secret sharing. SUR adopts the same user recruitment strategy and can achieve the same result as BUR. Meanwhile, by using secret sharing schemes, SUR can protect the inputs of each user from being revealed to the platform or to other users, even if they might collude. Moreover, SUR is a lightweight secure protocol, which does not depend on encryption/decryption operations and any trusted third-party.
3) We extend the SUR protocol to a more general case, where the total sensing quality of each task is a function about the sensing qualities of the recruited users. Moreover, we prove that if the function is increasing and submodular, SUR can still produce a solution with a logarithmic approximation ratio.
4) We conduct extensive simulations to verify the significant performances of the proposed SUR protocol. We also realize and run the SUR protocol in real smartphones which demonstrates that 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 II. The BUR and SUR protocols are proposed in Sections III and IV. We extend the SUR protocol in Section V. In Section VI, we evaluate the performances of SUR. After reviewing the related work in Section VII, we conclude the paper in Section VIII. *Some complex proofs are moved to the Appendix*.

II. Models, Problem, and Preliminary

A. 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 might perform one or more tasks. When they 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 $(1 \le j \le m)$, where $\mathbb{Z}_p$ is a prime field. Specially, $q_{i,j} = 0$ means that user u_i cannot deal with task s_j. Here, we assume that each user u_i knows his/her sensing qualities $q_{i,1}, \cdots, q_{i,m}$ since he/she can determine the value of each sensing quality $q_{i,j}$ by evaluating the corresponding 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 crowdsensing platform. Then, each user u_i determines the values of $q_{i,1}, \cdots, q_{i,m}$ 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.

B. Security Model

When each user u_i participates in the crowdsensing, his/her sensing quality values might reveal his/her private sensitive information. In order to avoid privacy disclosures, we need to protect each user's sensing qualities 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 [5]. 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 [4], [5], [10], [11]. The privacy under the semi-honest model can formally be defined as follows:

***Definition** 1 (Privacy under the Semi-honest Model [5]):* 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 $(1 \le i \le n)$. For $\mathcal{I} = \{u_{i_1}, \cdots, u_{i_\kappa}\} \subset \mathcal{U}$, we let $\mathcal{F}_\mathcal{I}$ denote the subsequence $\mathcal{F}_{i_1}, \cdots, \mathcal{F}_{i_\kappa}$. Consider an n-party protocol for computing $\mathcal{F}$. The view of the i-th user during an execution of this protocol, denoted as $VIEW_i$, is (x_i, r, m_i) where r represents the outcome of the i-th user's internal coin tosses and m_i represents the messages that the user has received. For $\mathcal{I} = \{u_{i_1}, \cdots, u_{i_\kappa}\}$, we let $VIEW_\mathcal{I} \triangleq (\mathcal{I}, VIEW_{i_1}, \cdots, VIEW_{i_\kappa})$. 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

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

Here, 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.

TABLE I
DESCRIPTION OF MAJOR NOTATIONS

Variable	Description
$\mathcal{U}$, $\mathcal{S}$	the set of all users, and the set of all tasks.
u_i, s_j	the i-th user, and the j-th task.
$q_{i,j}$, Q_j, θ	the sensing quality of u_i performing s_j, the total sensing quality of s_j, and the threshold of the required total sensing quality of each task.
Φ	the set of recruited users.
$f(\Phi)$, $\Delta_i f(\Phi)$	a utility function about recruited users and the incremental utility for adding u_i into Φ. (Definition 3).
b_i	a bit number that indicates whether u_i is recruited.
$VIEW_i$, 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. 7).
$\mathbb{Z}_p$, 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 2).

C. Problem

We focus on the secure user recruitment problem in the above mobile crowdsensing under the semi-honest model. That is, how to privately recruit the users in $\mathcal{U}$ to perform the tasks in $\mathcal{S}$ so that we can minimize the number of recruited users, while ensuring that the total sensing quality of each task is no less than a given threshold, denoted by θ. We use set Φ to denote a user recruitment solution where $u_i \in \Phi$ indicates that user u_i is recruited. Moreover, we use Q_j to denote the *total sensing quality* of task s_j:

$$Q_j = \sum\nolimits_{u_i \in \Phi} q_{i,j}. \tag{2}$$

Then, the problem can be formalized as follows:

$$Minimize: \quad |\Phi| \tag{3}$$

$$Subject\ to: \quad \Phi \subseteq \mathcal{U} \tag{4}$$

$$Q_j \geq \theta,\ 1 \leq j \leq m \tag{5}$$

$$Security \quad : \quad \text{Eq. 1 holds.} \tag{6}$$

Here, in Eq. 2, we define the total sensing quality Q_j as the sum of the sensing quality of each recruited user performing task s_j. In Section V, we will extend it to be a general function. Additionally, for ease of presentation, we 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 $b_i = 0$.

D. Preliminary

In this paper, we address privacy-preserving issues by using secret sharing schemes. A widely-used secret sharing scheme is Shamir's scheme [14]. Denote the shares of a secret s among n users as

$$[s] \triangleq (s[1], \cdots, s[i], \cdots, s[n]), \tag{7}$$

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

***Definition** 2:* Let p be an odd prime and $\mathbb{Z}_p$ be 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 $g_s(x) = s + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_\kappa x^\kappa \mod p$ 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 proven 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 $(\kappa+1)$-reconstruction). Additionally, we list the main notations in Table I.

III. BASIC USER RECRUITMENT PROTOCOL

In this section, we first analyze the complexity of the user recruitment problem. Then, we propose a Basic User Recruitment (BUR) protocol followed by the correctness and approximation ratio analysis.

A. Problem Hardness Analysis

Before the solution, we first prove the NP-hardness of the user recruitment problem, as shown 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 $\mathcal{S}$, each sensing quality is $q_{i,j} \in \{0, 1\}$, and the threshold of the total sensing quality is $\theta = 1$, determine a user recruitment solution Φ, such that the platform can minimize $|\Phi|$, while the total sensing quality of each task is no less than θ. 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. 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 $\{\mathcal{S}_i | 1 \leq i \leq n\}$, find a minimum size of subcollection of $\{\mathcal{S}_i | 1 \leq i \leq n\}$ 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. ■

B. The Greedy User Recruitment Strategy

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 will be recruited first. More precisely, the greedy strategy is based on the following utility function:

***Definition** 3: Utility function* $f(\Phi)$ indicates the total sensing qualities of all tasks in $\mathcal{S}$ contributed by the users in set Φ, until they reach the threshold θ, defined as follows:

$$f(\Phi) = \sum\nolimits_{j=1}^{m} \min\{Q_j, \theta\} = \sum\nolimits_{j=1}^{m} \min\{\sum\nolimits_{u_i \in \Phi} q_{i,j}, \theta\}. \tag{8}$$

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

$$\Delta_i f(\Phi) = f(\Phi \cup \{u_i\}) - f(\Phi). \tag{9}$$

The greedy user recruitment strategy is based on the above defined utility. The whole user recruitment process 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 $f(\Phi) = m\theta$.

Protocol 1 The BUR Protocol

Input: $\mathcal{U}$, $\mathcal{S}$, $\{q_{i,j}|u_i \in \mathcal{U}, s_j \in \mathcal{S}\}$, θ
Output: Φ, 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 values;
1: **for** $i=1$ **to** n **do**
2: user u_i sends $\{q_{i,1}, \cdots, q_{i,m}\}$ to the platform;
Phase 3: the platform makes the decision of user recruitment;
3: $\Phi=\emptyset$; $f(\Phi)=0$;
4: **while** $f(\Phi)<m\theta$ **and** $|\Phi|<n$ **do**
5: Select a user $u_i \in \mathcal{U}\backslash\Phi$ to maximize $\Delta_i f(\Phi)$;
6: $\Phi=\Phi \cup \{u_i\}$;
Phase 4: the platform returns the results to users;
7: **for** $i=1$ **to** n **do**
8: **if** $u_i \in \Phi$ **then**
9: the platform returns $b_i=1$ to user u_i;
10: **else**
11: the platform returns $b_i=0$ to user u_i;

C. The Detailed BUR Protocol

The BUR protocol mainly includes four phases, as shown in Protocol 1. In the first phase, the requester generates tasks and publishes them to mobile users via the platform. Then, mobile users report their sensing qualities to the platform in the second phase. Next, in the third phase, the platform makes the user recruitment decision based on the greedy strategy. In Step 3, the platform first initializes the recruited user set as the empty set, and the corresponding utility value is zero. Then, from Step 4 to Step 6, the platform greedily selects the user who can improve the utility the most and adds it into the recruited user set until the utility value of the recruited users $f(\Phi)$ becomes $m\theta$ or until all users are recruited. After this process, the user recruitment result is produced. In the fourth phase, the platform will notify each user of the result.

D. The Correctness and Approximation Performance

In this subsection, we prove the correctness and analyze the performance of BUR. First, we prove three important properties of the defined utility function $f(\Phi)$.

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

Proof: First, if $\Phi = \emptyset$, then $\min\{\sum_{u_i \in \Phi} q_{i,j}, \theta\} = 0$ for each $j \in [1,m]$. Thus, $f(\Phi = \emptyset) = 0$, according to Definition 3. Second, without loss of generality, we consider two user sets, Φ_1 and Φ_2, where $\Phi_1 \subseteq \Phi_2$. Then, we have $\min\{\sum_{u_i \in \Phi_1} q_{i,j}, \theta\} \leq \min\{\sum_{u_i \in \Phi_2} q_{i,j}, \theta\}$. Consequently, we have $f(\Phi_1) = \sum_{j=1}^{m} \min\{\sum_{u_i \in \Phi_1} q_{i,j}, \theta\} \leq \sum_{j=1}^{m} \min\{\sum_{u_i \in \Phi_2} q_{i,j}, \theta\} = f(\Phi_2)$. Therefore, $f(\Phi)$ is an increasing function with $f(\emptyset)=0$. ■

***Theorem** 3:* $f(\Phi)=m\theta$ iff Φ is a feasible solution to the user recruitment problem.

Proof: According to Eq. 8, $f(\Phi)=m\theta$ iff $\min\{\sum_{u_i \in \Phi} q_{i,j}, \theta\} = \theta$ holds for each $j \in [1,m]$. In fact, $\min\{\sum_{u_i \in \Phi} q_{i,j}, \theta\} = \theta$ and $\sum_{u_i \in \Phi} q_{i,j} \geq \theta$ are equivalent. Therefore, we have that $f(\Phi)=m\theta$ iff $\sum_{u_i \in \Phi} q_{i,j} \geq \theta$ holds for each $j \in [1,m]$. This means that the users in Φ can perform each task in $\mathcal{S}$ with a total sensing quality of no less than θ. 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.,

$$f(\Phi_1 \cup \{u_h\}) - f(\Phi_1) \geq f(\Phi_2 \cup \{u_h\}) - f(\Phi_2). \quad (10)$$

Proof: See Appendix A. ■

Based on the above properties of the utility function, we can prove the correctness of the proposed protocol.

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

Proof: Consider the user recruitment phase in Protocol 1. In each round of iteration, 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 terminate for sure. According to Protocol 1, when the iteration processes terminate, there must be $f(\Phi)=m\theta$ or $\Phi=n$. If the iteration processes terminate for $f(\Phi)=m\theta$, we can conclude that Φ is a feasible solution for the user recruitment problem according to Theorem 3. Otherwise, if the iteration processes terminate for $\Phi=n$, it means that the problem is not solvable, even though all users are recruited. Taking both cases into consideration, we can get that the theorem is correct. ■

Additionally, we can derive the approximation performance of the BUR protocol.

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

Proof: See Appendix B. ■

Theorems 5 and 6 show that if the user recruitment problem is solvable, BUR will produce a nearly optimal solution; otherwise, BUR will recruit all users as the solution.

IV. Secure User Recruitment Protocol

Based on BUR, we propose a Secure User Recruitment (SUR) protocol by using secret sharing schemes. First, we introduce the secret sharing schemes adopted in this paper. Then, we propose the SUR protocol, followed by performance and security analyses.

A. The Building Blocks

In the SUR protocol, each sensing quality is turned into 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** 4:* 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:

$$\begin{array}{ll} [z_1] \leftarrow SecAdd([x],[y]), & [z_2] \leftarrow SecSub([x],[y]), \\ [z_3] \leftarrow SecMulti([x],[y]), & [z_4] \leftarrow SecCmp([x],[y]), \\ [z_5] \leftarrow SecMax([x],[y]), & [z_6] \leftarrow SecMin([x],[y]), \end{array} \quad (11)$$

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

In Definition 4, the $SecAdd$ and $SecSub$ operations can be conducted efficiently without any communication 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, $z_1[i]=x+y+(\alpha_1+\beta_1)i+\cdots+(\alpha_\kappa+\beta_\kappa)i^\kappa \bmod p$. 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 [10] and the secure multi-party comparison protocol in [11], respectively. The multiplication protocol in [10] 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 [11] 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.

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

$$SecMax([x],[y]) \triangleq SecAdd([x], SecMulti(SecCmp([x],[y]), SecSub([y],[x]))) \quad (12)$$

$$SecMin([x],[y]) \triangleq SecAdd([x], SecMulti(SecSub(1-SecCmp([x],[y])), SecSub([y],[x])))). \quad (13)$$

Eq. 12 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. 13 is also correct. 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.

B. The Detailed SUR Protocol

The SUR protocol adopts the same utility function and greedy strategy to recruit users as BUR. 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 SUR. Second, when users jointly make recruitment decisions, all computations are conducted by using the secure operations in Definition 4, 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 $\sum_{j=1}^{m} \min\{q_{i,j}, \theta - Q_j\}$. 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.

Protocol 2 The SUR Protocol

Input: $\mathcal{U}$, $\mathcal{S}$, $\{q_{i,j} | u_i \in \mathcal{U}, s_j \in \mathcal{S}\}$, θ
Output: $b_1,\cdots,b_n$

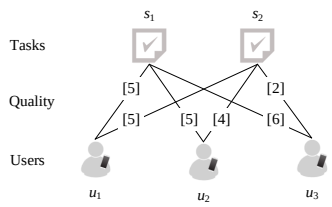
Phase 1: the requester publishes $\mathcal{S}$ to $\mathcal{U}$ via the platform;
Phase 2: users input their sensing quality vectors;
1: **for** $i=1$ **to** n **do**
2: user u_i determines the sensing qualities $q_{i,1},\cdots,q_{i,m}$;
3: **for** $j=1$ **to** m **do**
4: user u_i generates the polynomial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

The detailed SUR protocol is shown in Protocol 2. In Steps 3-5, users construct the polynomial secret shares of their sensing quality values as the inputs. Steps 6-9 initialize for the user recruitment decision process. In Steps 11-17, users jointly find the maximum incremental utility value, i.e., $\Delta_i f(\Phi)$. In Steps 18-23, users determine the recruited user and update the corresponding Q_j. 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. Consequently, 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.

C. Example

To better understand Protocol 2, we use an example to illustrate the secure 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.



(a) Users, tasks and sensing qualities

round 1	round 2
$[\Delta_1 f]=[10]$	$[\Delta_1 f]=[0]$
$[\Delta_2 f]=[9]$	$[\Delta_2 f]=[6]$
$[\Delta_3 f]=[8]$	$[\Delta_3 f]=[5]$
$[b_1]=[1]$	$[b_1]=[1]$
$[b_2]=[0]$	$[b_2]=[1]$
$[b_3]=[0]$	$[b_3]=[0]$
$[Q_1]=[5]$	$[Q_1]=[8]$
$[Q_2]=[5]$	$[Q_2]=[8]$

(b) Intermediate results

Fig. 2. Illustration of the SUR protocol ($\theta=8$)

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, $[Q_1]=[Q_2]=\theta=8$. No more users will be recruited.

D. The Correctness and Security Analysis

Essentially, SUR is the BUR protocol combined with secret sharing schemes to protect users' sensing quality privacy. Therefore, SUR can achieve the same user recruitment result as BUR. We can straightforwardly get the following theorem:

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

Moreover, we can also prove that the SUR protocol is secure against any semi-honest adversaries.

***Theorem** 8:* SUR 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 [10], [11]. Further, according to Eqs. 12 and 13 and the composition security theorem in [5], $SecMax$ and $SecMin$ are also secure. Thus, we only need to prove that SUR is secure by itself. 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 $m_{i_t}=\{q_{i,j}[i_t], b_i[i_t], Q_j[i_t], x[i_t], y[i_t], z[i_t], xz[i_t]\}$ and $VIEW_{i_t}=(\{q_{i_t,j}, n, m, \theta\}, r, m_{i_t})$. Consider all received messages $m_{i_1},\cdots,m_{i_\kappa}$ in $VIEW_{i_1},\cdots,VIEW_{i_\kappa}$ where the number of shares of each secret is no larger than κ. According to Shamir's secret sharing scheme, these shares cannot give any information about the secrets. That is to say, each received message can be simulated by a number randomly chosen from $\mathbb{Z}_p$. Thus, Eq. 1 holds for SUR. Then, according to the composition security theorem in [5], the whole protocol is secure. Thus, this theorem is correct. ■

V. Extension

Although the total sensing quality of a task in many applications can be seen as the sum of the sensing quality of each recruited user performing this task, as calculated in Eq. 2, there still are some cases in which the total sensing quality of a task might be calculated in other ways. For generality, we extend the total sensing quality to be a general function, denoted by $Q_j(\Phi)$:

$$Q_j(\Phi)\triangleq Q(q_{i,j}|_{u_i\in\Phi}), \tag{14}$$

where $Q(\cdot)$ is a general function about $q_{i,j}$. For example, if the sensing quality $q_{i,j}$ represents the probability of successful sensing, $Q(\cdot)$ may be defined as their joint probability, i.e., $Q_j(\Phi)=1-\prod_{u_i\in\Phi}(1-q_{i,j})$.

When we extend the total sensing quality from Q_j to $Q_j(\Phi)$, the second constraint in the problem formalization, i.e., Eq. 5, will become $Q_j(\Phi)\geq\theta$, the utility function will become $f(\Phi)=\sum_{j=1}^{m} min\{Q_j(\Phi),\theta\}$, and $[Q_j]$, $SecMin([q_{ij}], SecSub(\theta,[Q_j]))$ in Protocol 2 will be replaced by $[Q_j(\Phi)]$, $SecMin([Q_j(\Phi\cup\{u_{i,j}\})-Q_j(\Phi)]$, and $SecSub(\theta,[Q_j(\Phi)]))$, respectively. After the extension, Eq. 5 becomes a non-linear constraint, and computing the utility function $f(\Phi)$ becomes a little complicated. Despite this, Protocol 2 can still work well. More specifically, we have:

***Theorem** 9:* When $Q_j(\Phi)$ in Protocol 2 is a trivial function that can be securely computed by using the secure operations in Definition 4, Protocol 2 will still be secure.

Proof: In Theorem 8, all parts except the process of computing $Q_j(\Phi)$ in Protocol 2 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 [5]. ■

***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 2 can still produce a $(1+\ln\gamma)$-approximation solution where $\gamma=max_{u_i\in\mathcal{U}} f(\{u_i\})$.

Proof: See Appendix C. ■

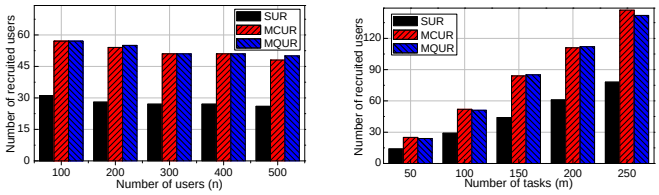
VI. Evaluation

We evaluate the SUR protocol mainly from two aspects. One is the user recruitment performance, i.e., the number of recruited users. Another is the time efficiency. Here, we will not evaluate the security and communication time of SUR since the security has been verified by theoretical analysis, and the communication time depends on the communication delay of wireless networks and communication rounds, which have also been precisely derived by theoretical analysis.

A. Evaluate the User Recruitment Performance

To evaluate the user recruitment performance, we conduct the SUR protocol and two compared protocols on synthetical traces. The compared protocols, simulation settings, and results are presented as follows:

Compared Protocols. Existing user recruitment protocols or algorithms involve various crowdsensing models, constraints, and optimization objectives. Most of them adopt the greedy strategy to recruit users (e.g., [6], [7], [9]). 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 some mobility models. A few works have also discussed the issues of sensing quality (e.g. [15]), however, they are still different from ours. For comparison,



(a) $m=100$ (b) $n=200$

Fig. 3. Number of recruited users vs. number of users and tasks ($p=30$, $\sigma=0.4$, $\theta=100$, and $\rho=20$)

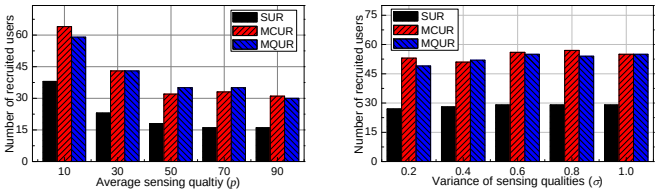
TABLE II
EVALUATION SETTINGS

parameter name	default	range
number of users n	200	100-500
number of tasks m	100	50-250
average sensing quality p	30	10-90
variance of sensing qualities σ	0.4	0.2-1.0
sensing quality threshold θ	100	50-250
largest number of tasks per user ρ	20	15-35

we borrow the basic strategy by ignoring the 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 is recruited first. Another protocol is denoted as MQUR, in which the user who performs tasks with the most sensing qualities is recruited first. Together, the two compared protocols and our SUR protocol constitute the three most typical greedy user recruitment strategies.

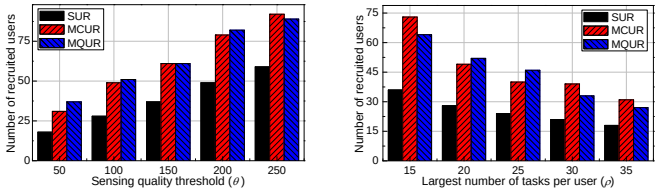
Simulation Settings. For the simulations, synthetical traces are adopted, in which we can evaluate the user recruitment performance with different parameters as needed, while ignoring users' mobility models. More specifically, we consider six parameters, including the number of users n, the number of tasks m, the average sensing quality (denoted by p), the variance of sensing qualities (denoted by σ), the sensing quality threshold θ, and the largest number of tasks performed by each user (denoted by ρ). In each simulation, we change one parameter while keeping the other parameters fixed. The range and default values of each parameter are illustrated in Table II. 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 $[(1-\sigma)p, (1+\sigma)p]$.

Evaluation Results. Fig. 3 depicts the number of recruited users vs. different numbers of users and tasks. The results show that the number of users recruited by 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 four parameters, as shown in Figs. 4 and 5. The results also prove that SUR has a much better performance than MCUR and MQUR. Moreover, when we increase either the average sensing quality or the largest number of tasks performed by each user, the number of recruited users decreases.



(a) $\sigma=0.4$ (b) $p=30$

Fig. 4. Number of recruited users vs. average sensing quality and variance ($n=200$, $m=100$, $\theta=100$, and $\rho=20$)



(a) $\rho=20$ (b) $\theta=100$

Fig. 5. Number of recruited users vs. sensing quality threshold and largest number of tasks performed by each user ($n=200$, $m=100$, $p=30$, and $\sigma=0.4$)

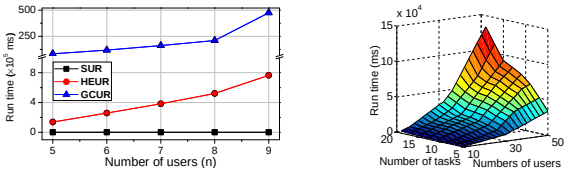
B. Evaluate the Time Efficiency

To evaluate the time efficiency of SUR, we run SUR and two compared protocols on real smartphones. The compared protocols, settings, and the results are presented as follows:

Compared Protocols. Besides the secret sharing schemes, the homomorphic encryption and garbled circuit protocols can also be utilized to solve the privacy-preserving user recruitment problem [5]. Therefore, we design two other secure user recruitment protocols for comparison: Homomorphic-Encryption-based User Recruitment (HEUR) protocol and Garbled-Circuit-based User Recruitment (GCUR) protocol. 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.

Experiment Settings. To evaluate time efficiency, we realize and run SUR and the compared protocols on a real smart phone with a 2.0GB memory and a processor of 4-core 2.2GHz plus 4-core 1.5GHz. We record the execution time of SUR, HEUR, and GCUR in this smart phone, while ignoring the communication time. During the execution, we use another smart phone to simulate the remaining $(n-1)$ users.

Evaluation Results. We run the SUR, HEUR, and GCUR protocols in the smartphones by changing the number of users from 5 to 9, while setting $m=6$, $p=8$, $\sigma=0.4$, $\theta=15$, and $\rho=m$. The results are depicted in Fig. 6(a). When the number of users is larger than 8, 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 10^6 ms. In contrast, the run time of SUR is far less than that of HEUR and GCUR in magnitudes. As shown in Fig. 6(b), when the number of users is 50 and the number of tasks is 20, the execution time of SUR is less than 150s. It can work well in real smartphones.



(a) Run time of three protocols (b) Run time of SUR

Fig. 6. Evaluation: run time vs. the number of users and tasks.

VII. Related Works

Most works about mobile crowdsensing focus on the user recruitment or task allocation problems [6], [7], [9], [13]. For example, M. Karaliopoulos et al. propose two greedy heuristic algorithms to recruit some mobile users who can perform location-related sensing tasks with a minimum cost [9]. Z. He et al. in [7] 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 limited budget. S. He et al. in [6] considered the maximum net reward task allocation problem with the constraint of time budgets. L. Pu et al. in [13] advocate a mobile crowdsourcing paradigm called Crowdlet in which the service quality based on keywords is considered. 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.

So far, only a few works have studied the privacy issues in mobile crowdsensing systems. For example, Q. Wang et al. in [17] investigated 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 [21] propose a privacy-preserving verifiable data aggregation and analysis scheme based on homomorphic encryption for cloud-assisted mobile crowdsourcing. In this scheme, the data is aggregated, encrypted, and stored in the cloud, which can be verified by using homomorphic encryption techniques. X. Jin et al. in [8] present a framework for a crowdsourced spectrum sensing service provider that selects spectrum-sensing participants, in which the differential privacy scheme is adopted to prevent the locations of mobile participants from being revealed. However, none of these investigate the privacy-preserving problem in the user recruitment process.

VIII. Conclusion

We propose a secure user recruitment protocol, called SUR, for sensing-quality-aware mobile crowdsensing systems. SUR adopts a greedy strategy based on a utility function to recruit users and uses secret sharing schemes to protect users' privacy. We prove that SUR can produce a solution with a logarithmic approximation ratio, and it 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 SUR can work well in real smartphones.

Acknowledgment

This research was supported in part by the National Natural Science Foundation of China (NSFC) (Grant No. 61572457, 61379132, U1301256), NSF grants CNS 1629746, CNS 1564128, CNS 1449860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, ECCS 1231461, and the NSF of Jiangsu Province in China (Grant No. BK20131174, BK2009150).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[16] 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.

[17] 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.

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

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

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

[21] 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.

Appendix

A. Proof of Theorem 4

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

$$\min\{\sum_{u_i\in\Phi_1\cup\{u_h\}} q_{i,j},\theta\}-\min\{\sum_{u_i\in\Phi_1} q_{i,j},\theta\}=$$
$$\min\{\sum_{u_i\in\Phi_2\cup\{u_h\}} q_{i,j},\theta\}-\min\{\sum_{u_i\in\Phi_2} q_{i,j},\theta\}=0 \quad (15)$$

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:

$$\min\{\sum_{u_i\in\Phi_1\cup\{u_h\}} q_{i,j},\theta\}-\min\{\sum_{u_i\in\Phi_1} q_{i,j},\theta\}$$
$$=\begin{cases} q_{h,j} & ,\theta\geq\sum_{u_i\in\Phi_2\cup\{u_h\}} q_{i,j};\\ q_{h,j} & ,\sum_{u_i\in\Phi_2\cup\{u_h\}} q_{i,j}>\theta\geq\sum_{u_i\in\Phi_2} q_{i,j};\\ q_{h,j} & ,\sum_{u_i\in\Phi_2} q_{i,j}>\theta\geq\sum_{u_i\in\Phi_1\cup\{u_h\}} q_{i,j};\\ \theta-\sum_{u_i\in\Phi_1} q_{i,j} & ,\sum_{u_i\in\Phi_1\cup\{u_h\}} q_{i,j}>\theta\geq\sum_{u_i\in\Phi_1} q_{i,j};\\ 0 & ,\theta<\sum_{u_i\in\Phi_1} q_{i,j}. \end{cases} \quad (16)$$

$$\min\{\sum_{u_i\in\Phi_2\cup\{u_h\}} q_{i,j},\theta\}-\min\{\sum_{u_i\in\Phi_2} q_{i,j},\theta\}$$
$$=\begin{cases} q_{h,j} & ,\theta\geq\sum_{u_i\in\Phi_2\cup\{u_h\}} q_{i,j};\\ \theta-\sum_{u_i\in\Phi_2} q_{i,j} & ,\sum_{u_i\in\Phi_2\cup\{u_h\}} q_{i,j}>\theta\geq\sum_{u_i\in\Phi_2} q_{i,j};\\ 0 & ,\sum_{u_i\in\Phi_2} q_{i,j}>\theta\geq\sum_{u_i\in\Phi_1\cup\{u_h\}} q_{i,j};\\ 0 & ,\sum_{u_i\in\Phi_1\cup\{u_h\}} q_{i,j}>\theta\geq\sum_{u_i\in\Phi_1} q_{i,j};\\ 0 & ,\theta<\sum_{u_i\in\Phi_1} q_{i,j}. \end{cases} \quad (17)$$

Comparing Eqs. 16 and 17, we have:

$$\min\{\sum_{u_i\in\Phi_1\cup\{u_h\}} q_{i,j},\theta\}-\min\{\sum_{u_i\in\Phi_1} q_{i,j},\theta\}\geq$$
$$\min\{\sum_{u_i\in\Phi_2\cup\{u_h\}} q_{i,j},\theta\}-\min\{\sum_{u_i\in\Phi_2} q_{i,j},\theta\} \quad (18)$$

Similarly, for the second sub-case, we can still derive Eq. 18. In summary, we can conclude that Eq. 18 holds for all cases. Now, according to Eq. 8, we have:

$$f(\Phi_1\cup\{u_h\})-f(\Phi_1)\geq f(\Phi_2\cup\{u_h\})-f(\Phi_2). \quad (19)$$

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

B. Proof of Theorem 6

To analyze the approximation ratio of the proposed protocol, we first prove that our user recruitment can be re-formalized as a Minimum Submodular Cover with Submodular Cost (MSC/SC) problem.

Lemma 1: The user recruitment problem can be re-formalized as an MSC/SC problem. Specifically, we have:

1) if the problem is solvable, it can be re-formalized as
$$Minimize\{|\Phi||f(\Phi)=f(\mathcal{U}),\Phi\subseteq\mathcal{U}\}; \quad (20)$$
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, $f(\Phi)=m\theta$ iff Φ is a feasible solution. Therefore, if Φ is another feasible solution, we must have $f(\Phi)=f(\mathcal{U})=m\theta$. That is to say, the constraint in Eq. 5 can be equivalently replaced by $f(\Phi)=f(\mathcal{U})$. Therefore, the user recruitment problem can be re-formalized as Eq. 20.

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. ■

Second, we introduce a lemma about the approximation ratio of MSC/SC problems, which is derived from [16].

Lemma 2: For an MSC/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 Protocol 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 protocol. According to Lemma 1, our user recruitment problem can be re-formalized as an MSC/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 Protocol 1 can achieve a $(1+\ln\gamma)$-approximation solution, where $\gamma=max_{u_i\in\mathcal{U}}f(\{u_i\})$. The theorem holds.

C. Proof of Theorem 10

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 $min\{Q_j(\Phi_1\cup\{u_h\}),\theta\}-min\{Q_j(\Phi_1),\theta\}=min\{Q_j(\Phi_2\cup\{u_h\}),\theta\}-min\{Q_j(\Phi_2),\theta\}=0$; 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 $\theta>Q_j(\Phi_2\cup\{u_h\})$, we have $(min\{Q_j(\Phi_1\cup\{u_h\}),\theta\}-min\{Q_j(\Phi_1),\theta\})-(min\{Q_j(\Phi_2\cup\{u_h\}),\theta\}-min\{Q_j(\Phi_2),\theta\})=(Q_j(\Phi_1\cup\{u_h\})-Q_j(\Phi_1))-(Q_j(\Phi_2\cup\{u_h\})-Q_j(\Phi_2))>0$, 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 $(min\{Q_j(\Phi_1\cup\{u_h\}),\theta\}-min\{Q_j(\Phi_1),\theta\})-(min\{Q_j(\Phi_2\cup\{u_h\}),\theta\}-min\{Q_j(\Phi_2),\theta\})\geq 0$ 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. 8. Part 1 has proven that $f(\Phi)$ is submodular. Therefore, when we replace Q_j by using $Q_j(\Phi)$, the problem can still be re-formalized as an MSC/SC problem. Moreover, $f(\Phi)$ is a polymatroid integer-valued function on $2^{\mathcal{U}}$. Further, according to Lemma 2, Protocols 1 and 2 can achieve a $(1+\ln\gamma)$-approximation solution, where $\gamma=max_{u_i\in\mathcal{U}}f(\{u_i\})$. The theorem is correct.