When Crowdsourcing Meets Social IoT: An Efficient Privacy-Preserving Incentive Mechanism

Xiaoying Gan, *Member, IEEE*, Yuqing Li, Yixuan Huang, Luoyi Fu, *Member, IEEE*, and Xinbing Wang, *Senior Member, IEEE*

***Abstract*—Crowdsourcing is an effective paradigm in human centric computing for addressing problems by utilizing human computation power, especially in booming social Internet of Things (IoT). By leveraging mutual friendship between computing entities (i.e., workers), collaborative tasks can thus be routed and finally fulfilled by multihop friends with high expertise. However, crowdsourcing in social IoT may reveal the privacy of task requesters which results in a large dilemma. In this paper, we focus on designing a multihop routing incentive mechanism which can also preserve task requester's privacy. Specifically, a utility maximization problem under privacy and budget feasibility constraints is formulated. Defining the conditions for privacy insurance, we give guidelines on how many subtasks should an entire task be divided into, and analyze the tradeoff between privacy and task accuracy. To enable efficient crowdsourcing task routing in social IoT, we first consider 1-hop myopic routing case and propose a near-optimal task assignment algorithm with $1/2$ approximation ratio for an arbitrary prior knowledge. We further design multihop payment policy to establish an equilibrium where workers are motivated to forward subtasks to their friends with the best expertise. The extensive simulations validate that our mechanism achieves a high level of average information gain with modest privacy guarantee.**

***Index Terms*—Crowdsourcing, incentive mechanism, privacy preserving, social Internet of Things (IoT), task assignment.**

I. Introduction

CROWDSOURCING, emerging as a new paradigm to solve tasks which are difficult for computers, has gain its popularity at an incredible rate. Traditional crowdsourcing systems (e.g., Amazon Mechanical Turk [1] and Yahoo! Answers [2]) allocate tasks to the public. Because of specific platforms required, however, such systems are not flexible and robust enough. If there is something wrong with platforms, task requesters cannot allocate tasks to workers. Furthermore, workers may not be reliable due to their unfamiliarity with task requesters.

Through the social Internet of Things (IoT) paradigm [3], [4], the social relationship among computing entities (i.e., users or devices) is fortified for efficient resource utilization and task completion. By introducing social IoT, an alternative solution is to spread the task to someone in social IoT who excels in this task, thus expanding traditional crowdsourcing networks and achieving broader crowdsourcing application. Crowdsourcing in social IoT has the following advantages.

1) The advent of social IoT services [5] has aroused people's enthusiasm in making social connections online. According to the statistics of Facebook [6], the number of users remaining monthly active on Facebook has been tremendously growing, reaching 2.2 billion in 2017. This could be one of the largest networks in history, which implies great computing capacities if people share idle resources with friends. For example, the average number of friends for a Facebook user is approximately 200, indicating a large of pool of worker candidates.
2) The social IoT paradigm is organized according to mutual friendship, where friends have enough incentives to offer their helps in a voluntary manner. Further, due to workers' friendship and the large scale of social IoT, one can easily obtain helps from multiple friends, without worrying too much about their complicated economic incentives of participation (the incentive issues are discussed in the previous literatures [7], [8], [9]), so it is also suitable for handling collaborative computing tasks.

However, how to motivate friends to route tasks efficiently within fixed budget remains to be a big problem. Moreover, as it always to be, the task requester may reveal private information to the crowd or violate copyright while carrying out crowdsourcing. For example, if task requester plans to ask workers to translate a passage, what can he do without violating reserved copyright hold by the author. If task requester plans to identify a strange in a picture, how can he do without revealing personal information? Or task requester

Manuscript received May 11, 2018; revised May 25, 2019 and July 11, 2019; accepted July 15, 2019. Date of publication July 23, 2019; date of current version December 11, 2019. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1003000, Grant 2018YFB1004705, and Grant 2018YFB2100302, in part by NSFC China under Grant 61672342, Grant 61671478, Grant 61602303, Grant 61532012, Grant 61822206, and Grant 61829201, in part by the Science and Technology Innovation Program of Shanghai under Grant 17511105103 and Grant 18510761200, in part by the Open Research fund of National Mobile Communications Research Laboratory, Southeast University under Grant 2018D06, and in part by the Shanghai Key Laboratory of Scalable Computing and Systems. *(Corresponding author: Xiaoying Gan.)*

X. Gan is with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: ganxiaoying@sjtu.edu.cn).

Y. Li, Y. Huang, L. Fu, and X. Wang are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: liyuqing@sjtu.edu.cn; huangyixuan@sjtu.edu.cn; yiluofu@sjtu.edu.cn; xwang8@sjtu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2019.2930659

may allow his 1-hop[1] friends to get access to tasks but do not want multihop friends to see task information, how can he do? In summary, crowdsourcing and privacy-preserving conflict inherently, especially in fast-growing social IoT.

To tackle privacy protection in crowdsourcing, there are mainly two methods, i.e., transformation and division. Transformation suggests alterations in task content and style [10]. For example, if we want to judge whether a person in a picture is over 50 years old, we would better not ask others so directly since it is impolite. To dissemble what we really want to know, we could ask how old is this person. Division indicates task segregation [11]. If the task is divided into small pieces, workers can only get partial information, which facilitates preserving task requester's privacy. For instance, in order to identify the age of a person in a picture but not to reveal personal portrait, we could partition the entire picture into several parts, and then allocate parts of the image to workers to identify specific features, such as hair color, skin corrugation, etc. We can evaluate his age from answers submitted by workers.

In this paper, we mainly focus on the task assignment and privacy preserving in social IoT, where division is adopted for task requester's privacy protection. Previous work like [12] focused on the tradeoffs between privacy, reliability, and crowdsourcing cost. But when it comes to social IoT, task requester's privacy becomes more susceptible since friends can get contact with each other. Due to the attention we should be given to a number of subtasks and it should be divided into entire task. If the entire task is divided into few subtasks indicating that each subtask contains more information about the entire task, there is a high chance that task requester's 1-hop friends can know the entire task by connections. Yet, if we divide the entire task into more subtasks, the information included in all of subtasks may not help you judge the answer of the entire task. For example, if we only know the hair color of a person, we cannot judge his age precisely since sickness can also result in white hair. On the other hand, although more subtasks indicate high reliability in privacy preserving, the number of workers assigned to each subtask will be reduced, further decreasing the accuracy of the entire task.

Our main contributions are highlighted as follows.

1) We propose a novel privacy-preserving incentive mechanism for social crowdsourcing, providing a theoretical basis to settle the dilemma between privacy and task accuracy. By leveraging mutual friendship, collaborative computing tasks can be routed and finally fulfilled by multihop friends with high expertise. Specifically, a utility maximization (UM) problem under requester's privacy and budget feasibility constraints is formulated, where overall service utility is defined as the expected information gain from observing collected answers.
2) Defining the conditions for privacy insurance, we give guidelines on how many subtasks should an entire task be divided into. To make sure requester's privacy, we give the upper bound of the probability of privacy reveal. The relationship between privacy and task accuracy are also analyzed.
3) We design 1-hop myopic routing policy and multihop payment policy for efficient crowdsourcing task routing in social IoT. In 1-hop routing case, by exploring monotone submodular property, we propose a near-optimal task assignment algorithm with 1/2 approximation ratio for an arbitrary prior knowledge. In multihop routing case, we prove that task requester's cost will increase with the length of task forwarding chain, which can motivate workers to allocate tasks to the friend with the best expertise in his ego network.
4) With extensive simulation results, we verify that the proposed mechanism corroborates the theoretical analysis, and achieves a high level of average information gain with modest privacy guarantee.

The rest of this paper is organized as follows. We provide the details of system model in Section II. In Section III, problem formulation for privacy-preserving incentive mechanism is given. To proceed, we first design multihop routing incentive policy and then conduct privacy-preserving analysis in Section IV. Finally, simulation, related work, and conclusion are shown in Sections V–VII, respectively.

II. System Model

In this section, we will first give an overview of our privacy-preserving incentive mechanism and then illustrate the details in the following sections.

A. Service and Network Model

In this paper, we mainly focus on utilizing the social IoT paradigm to tackle labeling tasks, each of which has one true answer from L possible labels belonging to the label set $\mathcal{L} = \{1, 2, \ldots, L\}$. To begin crowdsourcing, the task requester will first divide a labeling task T into N subtasks in the task set[2] $\mathcal{T}_{\text{sub}} = \{t_1, t_2, \ldots, t_N\}$. Each subtask represents one feature that characterizes the original labeling task T. The answer of subtask t_n ($n \in \mathcal{N} = \{1, 2, \ldots, N\}$) is one of M values belonging to the feature set $\mathcal{M} = \{1, 2, \ldots, M\}$. Each label in $\mathcal{L}$ can be mapped to a N-dimensional feature vector, indicating the feature vector can include all information of every label.

The social IoT paradigm is organized according to mutual friendship, where friends have enough incentives to offer their helps. Workers (i.e., computing entities) in social IoT are represented by nodes on a undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with the source node denoting task requester. Edges in the graph represent workers' friendship, indicating whether a particular worker can directly route the task to another worker. In other words, workers are willing to offer helps to others only if the two corresponding nodes are directly connected

[1]In social IoT, 1-hop friends refer to the people who have the direct connection with task requester. 2-hop friends refer to the people who have the direct connection with 1-hop friends of task requester but do not connect to task requester directly. By analogy, we can get multihop friends.

[2]The division here is adopted for task requester's privacy protection, just as above-mentioned. There have been researches on how to automatically divide a problem using recursive crowdsourcing [13]. This line of research is orthogonal to this paper. Here, we assume the task division is already provided.

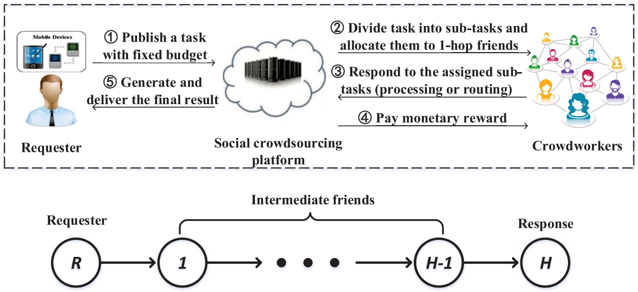


Fig. 1. General privacy-preserving incentive mechanism. Top: process of social crowdsourcing service. Bottom: illustration of a single multihop path for subtask routing.

in $\mathcal{G}$. Let $\mathcal{V}_0$ denote the set of 1-hop friends[3] of task requester, which is defined as task requester's ego network. The ego network of 2-hop friends (or equivalently, people who belongs to task requester's ego network) is denoted by $\mathcal{V}_{1,i}$, where $i \in \{1, 2, \dots, |\mathcal{V}_0|\}$. By the definition above, the ego network of k-hop friends can be represented by $\mathcal{V}_{k,i}$, where $i \in \{1, 2, \dots, |\mathcal{V}_{k-1,i}|\}$.

The process of social crowdsourcing service is described as shown in Fig. 1.

1) The task requester first publishes a task with a fixed budget on social crowdsourcing platform.[4]
2) The platform divides the task into subtasks and allocates them to the selected potential 1-hop friends (by worker ability) through social IoT paradigm.[5]
3) Each worker assigned the subtask decides whether to process it directly or to forward it to a friend in his own ego network.
4) The platform pays monetary reward to involved workers according to their efforts.
5) The platform generates the final result and then delivers it to the requester. In this way, such collaborative computing task, by exploiting workers' mutual friendship, can be routed and finally fulfilled by multihop friends who are capable of subtasks. Here, multihop friends may be strangers to the requester. Even if this were the case, multihop routing is essentially performed based on a set of 1-hop routings, which rely on mutual friendship among two directly connected workers.

In what follows, we will elaborate on how to enable privacy preserving and multihop incentive through the implementation of our mechanism.

B. Crowd Worker Model

Crowd workers vary in *ability*[6] that can be characterized by a confusion matrix [15], one for each worker in terms of each subtask. Each entry π_{lm} in confusion matrix is the probability that the corresponding worker, when given a subtask with true ans0wer $l \in \mathcal{M}$, provides an answer $m \in \mathcal{M}$. Obviously, $\sum_{m \in \mathcal{M}} \pi_{lm} = 1$ for all $l \in \mathcal{M}$. We call $\pi = [\pi_{lm}] \in [0, 1]^{M \times M}$ as the confusion matrix of the worker for that subtask. It is assumed to be drawn from a distribution $\mathcal{D}$ on space of confusion matrices. A simplified version of confusion matrix is defined in [16]. Specifically, each worker i is given an ability parameter $\gamma_i \in (0, 1)$ denoting the probability of providing the correct answer, and the error probability is assumed to be evenly distributed. Formally, let y_n denote the correct answer and r_{in} denote the answer from worker i on subtask t_n, then

$$\begin{aligned} &\mathbb{P}(r_{\text{in}} = l | y_n = l) = \gamma_i, \forall l \in \mathcal{M} \\ &\mathbb{P}\big(r_{\text{in}} = l' | y_n = l\big) = \frac{1 - \gamma_i}{M - 1}, \forall l, l' \in \mathcal{M}, l' \neq l. \end{aligned} \tag{1}$$

It is obvious that responses from the worker with greater γ_i will lead to larger information gain. For each worker i, we introduce worker effect denoted by $e(\gamma_i)$ to capture the relationship between his ability parameter γ_i and submitted answer's quality. Intuitively, one desirable effect should effectively reflect how good or how useful a worker's answer is for the final prediction of true answers. The simplest measure of worker effect is directly using the ability parameter, or its logistic form. That is, $e(\gamma_i) = \gamma_i$ or $e(\gamma_i) = \log[(\gamma_i)/(1 - \gamma_i)]$.

C. Privacy-Preserving Model

Assume that only people who are 1-hop friends can share information with each other. In general, assigning a subtask, which are only one fraction of entire task, to multihop friends will not reveal privacy of task requester. However, 1-hop friends may also have connections with each other. If task requester and one of his 1-hop friends share too many friends, this 1-hop friend is more likely to know the entire task by asking other friends, thereby potentially compromising task requester's privacy. Let b denote the minimum number of subtasks which can include the information of entire task. One desirable property we aim to achieve is privacy insurance defined in Definition 1.

Definition 1 (Privacy Insurance): The crowdsourcing system can guarantee privacy of task requester if $|\mathcal{V}_0 \cap \mathcal{V}_{1,i}| < b$ holds for $\forall i \in \{1, 2, \dots, |\mathcal{V}_0|\}$, where $|\cdot|$ is the cardinality.

In particular, the above privacy insurance constraint is equivalent to that

$$\max_{1 \le i \le |\mathcal{V}_0|} \big|\mathcal{V}_0 \cap \mathcal{V}_{1,i}\big| < b. \tag{2}$$

D. Multihop Routing Incentive Model

After task allocation, a worker can either directly finish the assigned subtask, or forward it to another worker in his ego network. Repeat this process and then a routing path is formed finally. Fig. 1 presents an example of such routing path with the length of K, where the subtask of interest is fulfilled by Kth worker. Along the routing path, the subtask can thus be routed to a more capable worker that lies beyond task request's ego network.

Our mechanism aims to arrive at an accurate answer inference, which highly depends on the quality of task finishing.

[3]Existing researches mostly rely on social graph $\mathcal{G}$ to capture interactions among friends [3]. It is widely considered that 1-hop friends (directly connected in $\mathcal{G}$) are apart in a distance of 1, and the distance between multihop friends (indirectly connected in $\mathcal{G}$) is hop number.

[4]The platform is assumed not to be involved in the reward transfer process between task requester and workers.

[5]Suppose friends can be assigned with one and only one subtask initially.

[6]Here, worker ability is actually given and does not require any evaluation, which is similar to task accuracy [14]. If there is such a thing to collect these prerequisite information, that is one-time information gathering before task allocation begins. Accordingly, this information gathering just adds a constant time to the entire process, without affecting task routing and payment later.

Under multihop routing, each worker is expected to submit inference truthfully or route the task to other worker that can best refine the inference. In crowdsourcing systems, however, there is usually no intrinsic value for processing or routing a particular task. As a result, workers especially for strategic and self-interested ones, may reluctantly contribute to task finishing or even if doing so, they may reduce their processing efforts strategically.

Taking into account workers' strategic and self-interested behaviors, we incentivize workers to provide high-quality task finishing or accurate answer inference. In particular, there are mainly two ways to motivate workers to contribute to completion of subtasks. First, workers can get payment by directly finishing subtasks. Second, workers can still obtain payment if they forward subtasks to other highly capable workers. Let $\mathcal{W}_n$ denote the set of all workers who contribute to finishing subtask t_n in one of two ways above and deserve to get payment. To formalize the problem, we introduce payment policy defined in Definition 2.

Definition 2 (Payment Policy): A payment policy denoted as $\rho^n : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ denotes the set containing all possible sets of associated workers' efforts toward subtask t_n, calculates the payments to workers based on the collected effort set $\boldsymbol{x}_n = \{x_n^i | i \in \mathcal{W}_n\}$. We use $\rho_i^n \geq 0$ to denote the payment to worker i under effort set $\boldsymbol{x}_n$.

The platform announces to workers the payment policy ρ^n, which then induces a routing game. In this game, each worker acts as a player to decide how to respond to the assigned subtask (processing or routing) by evaluating his own utility or payment. In practice, the crowdsourcing procedure usually works under an estimated incentive budget for task requester. Another desirable property considered is budget feasibility defined in Definition 2.

Definition 3 (Budget Feasibility): For any subtask $t_n, n \in \mathcal{N}$, a payment policy ρ^n is budget feasible under collected effort set $\boldsymbol{x}_n$, if and only if the total payment to workers does not exceed budget B, i.e.,

$$\sum_{i \in \mathcal{W}_n} \rho_i^n \leq B, \forall n \in \mathcal{N}. \tag{3}$$

III. Problem Formulation

A. Information Gain

Before we continue, let us shed some light on information gain by undertaking a preliminary step toward developing a measure of the utility of social crowdsourcing service.

The task requester holds a prior knowledge about subtasks. Let $p_{nm}^0 = \mathbb{P}^0(t_n = m), n \in \mathcal{N}, m \in \mathcal{M}$ denote the probability initially held by task requester that the nth subtask is characterized by the mth feature. For subtask t_n, task requester's *prior knowledge* is denoted by $p_n^0 = (p_{n1}^0, p_{n2}^0, \ldots, p_{nM}^0)$, where $\sum_{m=1}^M p_{nm}^0 = 1$. When task requester is unbiased, the prior knowledge can be simply set as the uniform distribution

$$\mathbb{P}^0(t_n = m) = \frac{1}{M}, \forall n \in \mathcal{N}, m \in \mathcal{M}. \tag{4}$$

We use R_n to denote the received responses to subtask t_n. Task requester's *posterior knowledge* $p_n = \mathbb{P}^0(t_n = m|R_n)$ can be calculated according to Bayes decision rule as follows:

$$\mathbb{P}^0(t_n = m|R_n) = \frac{\mathbb{P}^0(t_n = m, R_n)}{\mathbb{P}(R_n)}. \tag{5}$$

Suppose worker responses for any subtask are independent given the true answer. We obtain $\mathbb{P}^0(t_n = m, R_n) = \mathbb{P}^0(t_n = m) \prod_{r \in R_n} \mathbb{P}(r|t_n = m)$, where $\mathbb{P}(r|t_n = m)$ is determined by the confusion matrix or for simplicity, (1). In particular, the true label of subtask t_n can be inferred as

$$\hat{y}_n = \arg\max_{m \in \mathcal{M}} p_n. \tag{6}$$

Formally, every time the platform asks a worker for the new answer, its goal is to select a set of answers that will result in the greatest expected decrease in the uncertainty of label inference. Information theory provides us with a useful criterion for measuring the amount of uncertainty in the distribution of label predictions, i.e., joint entropy $H(p_n) = -\sum_{m=1}^M \mathbb{P}(t_n = m|R_n) \log \mathbb{P}(t_n = m|R_n)$. We model the contributions of each worker as a noisy channel, and evaluate the quality of submitted responses using this metric. When submitting an answer R_n, the worker contributes $IG(R_n)$ bits in support of that answer, that is

$$IG(R_n) = H\left(p_n^0\right) - H(p_n). \tag{7}$$

As we receive more contributions from workers, $\mathbb{P}(t_n = m|R_n)$ is updated and the joint entropy of subtask t_n changes. The lower t_n's entropy is, the more confident we are in our prediction of the correct answer.

With these definitions, we can measure the improvement due to crowdsourcing in social IoT via the information gain over all subtasks, which can be characterized as

$$IG = \sum_{n=1}^N IG(R_n). \tag{8}$$

Absolute information gain is significantly affected by the number of workers who submit answers. Hence, we leverage the expected information gain per worker (to be specified later) as a metric for evaluating service utility of social crowdsourcing.

Intuitively, a larger information gain leads to a smaller uncertainty, thus increasing inference accuracy of true labels (i.e., the social crowdsourcing performance is improved), and vice versa. To illustrate this further, let us consider a simple case of social crowdsourcing service with task set $\mathcal{T}_{\text{sub}} = \{t_1\}$ and feature set $\mathcal{M} = \{0, 1\}$. Given the true answer $y_n = 1$, the confusion matrix of 1-hop friend (i.e., worker 1) is $\mathbb{P}(r_{11} = 1|y_n = 1) = 0.5$ and $\mathbb{P}(r_{11} = 0|y_n = 1) = 0.5$. Suppose worker 1 only has two friends (i.e., workers 2 and 3) with confusion matrix $\mathbb{P}(r_{21} = 1|y_n = 1) = 0.9$ and $\mathbb{P}(r_{21} = 0|y_n = 1) = 0.1$, and $\mathbb{P}(r_{31} = 1|y_n = 1) = 0.6$ and $\mathbb{P}(r_{31} = 0|y_n = 1) = 0.4$, respectively. We adopt the simple form of $\mathbb{P}^0(t_n = m)$ in (4). If worker 2 is assigned the task, then we have $\mathbb{P}^0(t_1 = 1, R_1) = \mathbb{P}^0(t_1 = 1) \prod_{r \in R_1} \mathbb{P}(r|t_1 = 1) = 0.225$, and $\mathbb{P}^0(t_1 = 0, R_1) = \mathbb{P}^0(t_1 = 0) \prod_{r \in R_1} \mathbb{P}(r|t_1 = 0) = 0.025$. The information gain that worker 2 contributes is $IG = H(p_1^0) - H(p_1) = 0.314$. Similarly, we can obtain worker 3's contribution $IG = 0.276$. As a consequence, worker 2 who contributes more to improving information gain has high priority (or service utility) since high inference accuracy of true labels will be achieved.

B. Utility Maximization Problem

The objective of our mechanism is to improve overall service performance, by reasonably dividing the task into subtasks and routing them to workers. We assumed that the social crowdsourcing platform is authorized to manage task division/allocation and payment allocation. In this process, any worker serves as either the final finisher to process the assigned subtask directly or an intermediate router to publicize the subtask in his own ego network. Under the proposed mechanism, we try to address the following problems.

1) How to divide task requester's task into subtasks and allocate them to 1-hop friends for a guaranteed privacy?

2) From the perspective of workers in multihop routing process, how to respond to assigned subtasks (i.e., processing or routing) with what payment allocation for high task accuracy?

In the context of social crowdsourcing, the key to solving the above problems is to motivate workers to leverage friendship for task completion with high accuracy and privacy insurance. Next, we will formally characterize these problems.

To evaluate performance of social crowdsourcing service subscribed by task requester (or equivalently, task-worker assignment), the mechanism actually allows the use of various utility function to optimize for. One natural utility function choice is the expected information gain from a set of collected worker responses or efforts, which is defined as

$$U(\mathcal{W}) = \mathbb{E}[IG] = \sum_{n=1}^{N} H\left(p_n^0\right) - H\left(p_n^0 | \mathcal{W}_n\right) \tag{9}$$

where $\mathcal{W}_n$ consists of all workers assigned to subtask t_n and $\{\mathcal{W}_n\}_{n=1}^N = \mathcal{W}$. Let $R_{\mathcal{W}_n}$ denote all responses received from worker set $\mathcal{W}_n$. The conditional entropy $H(p_n^0|\mathcal{W}_n)$ illustrates the expected uncertainty of the distribution of label predictions after receiving workers' answers. It can be calculated as $H(p_n^0|\mathcal{W}_n) = -\sum_{R_{\mathcal{W}_n} \in \mathcal{R}_n} \sum_{m=1}^{M} \mathbb{P}(t_n = m, R_{\mathcal{W}_n}) \log \mathbb{P}(t_n = m | R_{\mathcal{W}_n})$, where $\mathcal{R}_n$ is the domain of $R_{\mathcal{W}_n}$ that consists of all possibilities.

In our proposed mechanism, the privacy-preserving and multihop incentive are enabled based on "assignment decision," maximizing the overall service utility while guaranteeing privacy insurance and budget feasibility constraints. Accordingly, the UM problem is to find an optimal task-worker assignment $\mathcal{W}^* = \{\mathcal{W}_n^* | n \in \{1, 2, \ldots, N^*\}\}$ that

$$\max_{\mathcal{W}} \quad U(\mathcal{W}) \tag{10}$$

$$\text{s.t.} \quad \bigcup_{n=1}^{N} \mathcal{W}_n = \mathcal{W} \tag{11}$$

$$\mathcal{W}_n \bigcap \mathcal{W}_{n'} = \emptyset, \forall n \neq n' \tag{12}$$

$$\max_{1 \le i \le |\mathcal{V}_0|} \left| \mathcal{V}_0 \cap \mathcal{V}_{1,i} \right| < b \tag{13}$$

$$\sum_{i \in \mathcal{W}_n} \rho_i^n \le B, \forall n \in \mathcal{N}. \tag{14}$$

Specifically, constraints (11) and (12) ensure that each worker can be assigned to one and only one subtask, which is of vital importance for privacy preserving. Intuitively, the probability that requester's privacy can be revealed increases with the amount of available information of entire task captured from one subtask. If these two constraints focus on individual perspective, then constraint (13) in essence, highlights the risk that privacy is revealed through information sharing between friends. Constraint (14) indicates that the total payment to workers must be within requester's incentive budget, guaranteeing the mechanism can be implemented in practice.

Remark: While our explicit objective is to maximize the expected information gain from collected worker efforts, the formulation equivalently minimizes the expected uncertainty of the distribution of label predictions, which serves as the basis for inference accuracy of true labels. Besides, we would like to notice that the assignment decision $\mathcal{W}$ actually involves both *task division* N and 1-*hop allocation/multihop routing* $\{\mathcal{W}_n\}_{n=1}^N$, where the latter determines how payment policy ρ^n shall be applied in terms of subtask t_n under budget constraint. All of these result in accordance with two aforementioned primal problems of interest and manifest themselves in the conflict between task accuracy and requester's privacy.

C. Analysis

Recall the objective of UM problem in (10). It is obvious that the entropy of prior knowledge $H(p_n^0)$ is a fixed value and the service utility depends solely on $H(p_n^0|\mathcal{W}_n)$. Under the assumption of unbiased prior knowledge of subtasks and uniformly distributed error probabilities, Lemma 1 states that different assignment policies vary in expected information gain of one specific subtask, but lead to the same expectation of overall information gain.

Lemma 1: Given the independence of workers' responses and unbiased prior knowledge of subtasks, we can obtain

$$U(\mathcal{W}) = \sum_{n=1}^{N} \sum_{i_n \in \mathcal{W}_n} e\left(\gamma_{i_n}\right) \tag{15}$$

where to best measure worker effects on information gain $U(\mathcal{W})$, $e(\cdot)$ takes the following form:

$$e\left(\gamma_{i_n}\right) = \log m + \frac{\gamma_{i_n}}{M} \log \gamma_{i_n} + \frac{1-\gamma_{i_n}}{M} \log \frac{1-\gamma_{i_n}}{M-1}. \tag{16}$$

Proof: See Appendix A. ∎

Things become different and more complicated when either the prior knowledge is nonuniformly distributed, or the general confusion matrix is used to characterize worker ability. As a result, different assignments always lead to different expectations of overall information gain.

In fact, finding the optimal assignment $\mathcal{W}^* = \{\mathcal{W}_n^*\}_{n=1}^{N^*}$ that maximizes an arbitrary $U(\mathcal{W})$ is more like a known Partition Problem which has been proved to be NP-hard [16]. The Partition Problem asks whether a given set $\mathcal{Z}$ of positive integers can be divided into two subsets Z_1 and Z_2 such that $\sum_{z \in Z_1} z = \sum_{z \in Z_2} z$. We can construct an instance of our assignment problem to solve such Partition Problem. For every integer $z \in \mathcal{Z}$, we define a corresponding worker $i_z \in \mathcal{V}$ with

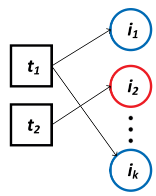


Fig. 2. Proof of NP-hardness.

ability parameter $\gamma_{i_z} = z/\max\{z : z \in \mathcal{Z}\}$. We also define two subtasks t_1 and t_2, and assign them to the predefined workers. We introduce a binary decision variable $I_{n,i}$ to characterize task assignment strategy. Specifically, $I_{n,i} = 1$ represents that worker i has been assigned subtask t_n, and $I_{n,i} = 0$ otherwise. Under such task assignment strategy, the assignment set can thus be characterized by $\mathcal{W}_n = \{i | I_{n,i} = 1, \forall i\}$. As shown in Fig. 2, two subtasks t_1 and t_2 are assigned to the corresponding workers with allocation $W_1 = \{i_1, i_k\}$ and $W_2 = \{i_2\}$. Let $u(\mathcal{W}_n) = \log(\sum_{i \in \mathcal{W}_n} \gamma_i + 1)$ and define service utility as $U(\mathcal{W}) = \sum_n u(\mathcal{W}_n)$. With what has been described above, we may safely conclude that the answer to the original Partition Problem is positive if and only if $u(W_1^*) = u(W_2^*)$, where $\{W_n^*\}$ denotes the optimal assignment for each subtask t_n in assignment problem constructed above. We consider it from the following two aspects.

1) Suppose $\mathcal{W}^*$ is an optimal solution to assignment problem such that $u(W_1^*) = u(W_2^*)$. Then the sums of worker ability for each subtask are equal. Since $\gamma_{i_z} = z/\max\{z : z \in \mathcal{Z}\}$, we have $\sum_{x \in Z_1} z = \sum_{z \in Z_2} z$;
2) Suppose there exist two subsets of $\mathcal{Z}$, Z_1 and Z_2, such that $\sum_{z \in Z_1} z = \sum_{z \in Z_2} z$. Consider the contradiction that under optimal solution $\mathcal{W}^*$, these two subtasks are allocated to two sets of workers with different sums of ability. According to the monotonicity and submodularity (to be discussed later), $U(\mathcal{W})$ can be improved by making sums of worker ability equal, i.e., $\mathcal{W}^*$ is suboptimal, which further results in a contradiction. Therefore, finding the optimal assignment $\mathcal{W}^*$ that maximizes $U(\mathcal{W})$ is NP-hard as well.

D. Our Solution

Given the NP-hardness of UM problem, we turn to approximation algorithm to achieve computation-efficiency balance. Before proceeding further, let us grasp the basics of monotone submodular function.

Definition 3 (Monotone Submodular Function): Let Ω be a finite set. For any $X \subseteq Y \subseteq \Omega$ and $x \in \Omega/Y$, a function: $f : 2^{\Omega} \mapsto \mathbb{R}$ is submodular if and only if $f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y)$, and it is monotonic if and only if $f(X) \leq f(Y)$.

The utility function $U(\mathcal{W})$ defined in (11) falls in the family of monotone submodular functions. Please refer to [17] for details of the proof that expected information gain (corresponds to $U(\mathcal{W})$ here) is submodular and nondecreasing. With such diminishing returns property in place, we naturally set out to explore an efficient approximate algorithm to find the optimal assignment for UM problem with a theoretically acceptable approximation rate.

Recall the optimal task-worker assignment $\mathcal{W}^* = \{\mathcal{W}_n^* | n \in \{1, 2, \ldots, N^*\}\}$. Actually, to find such assignment involves a combination of task division, 1-hop task allocation, and multihop incentive processes. The term N^* suggests the optimal number of subtasks should be divided to, i.e., task-subtask assignment. While the term $\mathcal{W}_n^*$ captures the optimal allocation among workers for each subtask t_n, i.e., subtask-worker assignment, which manifests itself in two ways. One is how to allocate subtasks to 1-hop friends in requester's ego network, which together with task division, contributes to the promise of privacy protection for task requester. Denote the optimal set of selected 1-hop friends as $\mathcal{W}_{\text{1-hop}} = \{i | i \in \mathcal{W}^* \cap \mathcal{V}_0\}$, where each friend can be assigned to one and only one subtask. The other is that in multihop routing originated from 1-hop friends, how to forward subtasks to high-ability workers for maximal service efficiency, whose realization benefits from the payment policy and leveraged mutual friendship potentials.

In the proposed mechanism, multihop routing incentive policy actually depends on privacy-preserving policy. Specifically, task division determines how many multihop routing paths exist and 1-hop allocation determines which 1-hop friend each routing path originates from. In essence, our privacy-preserving incentive mechanism is based on a two-stage decision problem. In the first stage, the platform establishes task division and 1-hop allocation strategies to guarantee requester's privacy, and with this information, then makes multihop payment strategy to stimulate workers' willingness to forward subtasks to high-ability ones under incentive budget. Therefore, one feasible solution is backward induction, i.e., first to convert it to a pure payment allocation problem, and on that basis, to address the combinatorial problem involving task division and allocation.

IV. Privacy-Preserving Incentive Mechanism

A. Multihop Routing Incentive Policy

We first consider multihop payment allocation in the second stage, where total number of subtasks N and the set of selected 1-hop friends $\mathcal{W}_{\text{1-hop}}$ are taken as given parameters. Moreover, equilibrium strategy of workers are studied as well.

As mentioned previously, the true answer of subtask t_n are inferred as $\hat{y}_n = \arg\max_{m \in \mathcal{M}} p_n$. Enabling multihop task routing with high quality and efficiency requires mechanisms that will incentivize workers to both truthfully report posterior probabilities and to route subtasks to multihop friends who can best improve answer inference. Due to the attention it should be given to the worker heterogeneity in ability and sociability, especially under the limited incentive budget. Intuitively, workers with high ability or strong sociability are more likely to receive target subtasks. Suppose the confusion matrices are common knowledge between requester's 1-hop friends. With payment allocation in place, 1-hop friends would like to forward subtasks to workers with high ability parameters who may exist outside requester's ego network. In this way, both

Algorithm 1: 1-HopGreedy

Input: 1-hop friends $\mathcal{V}_0 = \{1, 2, \ldots, |\mathcal{V}_0|\}$, ability parameters $\{\gamma_i\}_{i=1}^{|\mathcal{V}_0|}$, prior $\mathbb{P}^0(t_n)$ over subtasks $\{t_n\}_{n=1}^{N}$

Output: Assignment $\mathcal{W} \leftarrow \{\mathcal{W}_n\}_{n=1}^{N}$

Initialize $\mathcal{W}_n \leftarrow \emptyset, n = 1, 2 \ldots, N$, $\mathcal{W} \leftarrow \{\mathcal{W}_n\}_{n=1}^{N}$

for $i \in \mathcal{V}_0$ *sorted by key=*$e(\gamma_i)$ **do**

  for $n = 1 : N$ **do**

    $\Delta H_n \leftarrow H(\mathbb{P}^0(t_n)|\mathcal{W}) - H(\mathbb{P}^0(t_n)|\mathcal{W}, i \bigcup \mathcal{W}_n)$

  $n^* \leftarrow \text{argmax} \Delta H_n$

  $\mathcal{W}_{n^*} \leftarrow \mathcal{W}_{n^*} \bigcup i$

  $\mathcal{W} \leftarrow \{\mathcal{W}_n\}_{n=1}^{N}$

workers who are experts and workers who are knowledgeable about the expertise of others will get rewarded.

1) 1-Hop Myopic Routing: Before addressing multihop routing problem, let us look at how to efficiently derive the optimal assignment in the case of allocation horizon $K = 1$, i.e., 1-hop myopic routing. Different from multihop routing of interest, task requester allocates subtasks to 1-hop friends in the set $\mathcal{W}_{\text{1-hop}}$ and requires their direct responses instead of routing subtasks to other workers. In this setting, the expected overall information gain varies if the prior knowledge is nonuniform distribution. We propose a greedy algorithm in Algorithm 1-HopGreedy that is, guaranteed with a approximation ratio of (1/2) within polynomial time complexity.

Theorem 1: Let $\mathcal{W}^*$ denote the optimal assignment that maximizes expected information gain, and $\mathcal{W}$ denote the assignment found by Algorithm 1-HopGreedy. Then $U(\mathcal{W}) \geq (1/2)U(\mathcal{W}^*)$.

Proof: See Appendix B. ■

As for time complexity of Algorithm 1-HopGreedy, the sorting operation takes $|\mathcal{V}_0| \log |\mathcal{V}_0|$ time to work out the 1-hop friend sequence in order of descending ability parameter. The subsequent outer loop runs $|\mathcal{V}_0|$ times for every friend and the inner loop runs N times for every subtask. Therefore, the overall time complexity of Algorithm 1-HopGreedy is $|\mathcal{V}_0| \log |\mathcal{V}_0| + N|\mathcal{V}_0| = |\mathcal{V}_0|(\log |\mathcal{V}_0| + N)$.

2) Multihop Payment Allocation: In multihop task routing, any worker can choose either to finish the assigned subtask directly or to forward it to friends in his ego-network. If choosing the latter, he will use $[\alpha/(1+\alpha)]$ ratio of total money as his rewards and use the left $[1/(1+\alpha)]$ as the rewards to motivate his friends, where $\alpha \in (0, 1)$ denoting the payment allocation ratio, is a constant set by the platform.

Inspired by [18] and [19], we adopt the routing scoring rule in multihop routing incentive policy to incentivize accurate inference, along with the effective routing decisions. Consider a routing path of length K (excluding the requester) for any subtask t_n. Under such routing scoring rule, the ith worker gets the payment of

$$\rho_i^n = (1-\alpha)e(\gamma_i) + \alpha e(\gamma_{i+1}) - e(\gamma_{i-1}) \tag{17}$$

where $i \in \{1, 2, \ldots, K-1\}$ and γ_i captures the ability parameter of the ith worker. To facilitate illustration, we assume $\gamma_0 = 0$. We would like to notice that the ith worker's payment is based on the incremental value he provides for refining answer inference (or equivalently, the increment of task finishing's quality he contributes to), which is measured by his report and the report of the worker he routes the subtask to. The final worker on the path does not route, and is paid $c_K = e(\gamma_K) - e(\gamma_{K-1})$.

Along any single multihop routing path, the total payment for all involved workers can be characterized as

$$\begin{aligned} & e(\gamma_K) - e(\gamma_{K-1}) + \sum_{i=1}^{K-1} (1-\alpha)e(\gamma_i) + \alpha e(\gamma_{i+1}) - e(\gamma_{i-1}) \\ &= e(\gamma_K) - e(\gamma_{K-1}) + \alpha(e(\gamma_K) - e(\gamma_1)) + e(\gamma_{K-1}) \\ &= (1+\alpha)e(\gamma_K) - \alpha e(\gamma_1). \end{aligned} \tag{18}$$

According to such payment rule, a worker with ability parameter γ will be paid directly with $e(\gamma)$ in 1-hop routing case. While in multihop routing, task requester needs to totally pay $(1+\alpha)e(\gamma_K = \gamma) - \alpha e(\gamma_1)$ for all workers involved in the task-forwarding chain, to reach the worker who has the same ability parameter γ but lies beyond his ego network. Hence, compared with 1-hop routing, task requester has an extra payment $(1+\alpha)e(\gamma_K = \gamma) - \alpha e(\gamma_1) - e(\gamma) = \alpha(e(\gamma_K) - e(\gamma_1))$ for one single multihop routing path.

Let us take a closer look at how multihop payment allocation and total payment vary with routing path length. Imagine that in one task-forwarding chain, the k-hop friend finishes the subtask by himself and get paid by $c_k = e(\gamma_k) - e(\gamma_{k-1})$. Then the $(k-1)$-hop friend who forward this subtask to him can get the reward αc_k. For the $(k-2)$-hop friend, he can get reward as $\alpha(\alpha c_k + c_k) = (1+\alpha)\alpha c_k$. Using mathematical induction, it is easy for us to derive that for any i-hop friend in the task-forwarding chain, the reward is $\alpha c_k (1+\alpha)^{k-1-i}$. The specific proof details are omitted for simplicity. Obviously, for any two workers (the i-hop and j-hop friends) who contribute to such multihop routing, the i-hop friend will be allocated more payment than the j-hop one if $i < j \leq k$.

With the above payment allocation in place, we can derive the total payment for task requester to motivate his corresponding 1-hop friend.

Lemma 2: If one of task requester's 1-hop friends plans to find the k-hop friend with payment c_k who can finish the subtask, the least total payment is $c_k(1+\alpha)^{k-1}$, where $c_k = e(\gamma_k) - e(\gamma_{k-1})$.

Proof: See Appendix C. ■

Remark: Intuitively, for any i-hop friend in the task-forwarding chain, the assigned subtask will be forwarded further only if there exists one friend with higher ability in his ego network, i.e., $e(\gamma_{i+1}) > e(\gamma_I), \forall i \in \{1, 2, \ldots, k\}$. According to Lemma 2, we observe that the total payment will increase exponentially with increase of routing path length since $c_k > 0$. It is suggested that the subtask cannot be routed by many friends, especially under budget feasibility constraint, i.e., $c_k(1+\alpha)^{k-1} \leq B$. Actually, in real social IoT, the average degree of separation between two random Twitter users is 3.43 [20]. It indicates that in our multihop task routing, the term K for K-hop friend who finishes the subtask by himself is generally less than 4. On the other hand, the forwarding process may be infinite if no worker is willing to fulfill the subtask, making no one can get payment or reward. Hence, in order to get rewards, workers will always to seek friends with the best expertise in their ego networks to finish subtasks since the payment paid to potential experts decreases exponentially.

3) Equilibrium Strategy of Workers: Having introduced the multihop routing scoring rule of interest, we conduct equilibrium analysis of the associated routing game. Consider the case where network structure is common knowledge and actual answer realizations are still assumed private. Intuitively, the routing scoring rule has an effect on workers' routing decisions in equilibrium, which in turn affect how much information can be aggregated. Since the strictly proper scoring rule aims to incentivize accurate reports, a worker's payment is positive if and only if he can improve inference accuracy. To formally establish the connection between worker effect and routing decision, we have the following equilibrium result.

Theorem 2: Consider a routing game in which workers are risk neutral. For the kth worker on the routing path of subtask t_n, let V_n^k denote the set of workers in this ego network who have not yet been assigned to t_n. Under the proposed multihop routing rule, it is a Bayesian Nash equilibrium for the kth worker to finish t_n by himself if $e(\gamma_k) \geq \max_{w\in V_n^k} e(\gamma_w)$, or to forward t_n to the next worker in his ego network who has the maximum effect $e(\gamma_{k+1}) = \max_{w\in V_n^k} e(\gamma_w)$ if $e(\gamma_k) < \max_{w\in V_n^k} e(\gamma_w)$

Proof: See Appendix D. ■

Given task division and allocation among 1-hop friends, the optimal multihop routing can route subtasks through ego network topology, and seek for potential workers with an ability as high as possible for each subtask, enabling high-quality task finishing. Accordingly, the expected worker effect, or rather inference accuracy, is strictly increasing in the length of a routing path.

B. Privacy-Preserving Policy

With the payment allocation information, the goal of the first stage is to determine the optimal task division and 1-hop friend selection to best ensure requester's privacy. Specifically, we first show how many subtasks should be divided into so as to avoid mapping conflict, and then illustrate how to allocate subtasks to 1-hop friends in requester's ego network while guaranteeing his privacy. After that we analyze the tradeoff between answer accuracy and individual privacy.

1) Task Division: Consider a typical crowdsourcing system with N subtasks and M features, where each subtask has one true answer from L labels. The probability of no mapping conflict can be formally characterized as

$$\bar{\mathbb{P}}(M, N; L) = \frac{C^L_{M^N}}{M^{NL}}. \tag{19}$$

Intuitively speaking, the more number of workers assigned to each subtask is, the more accurate each subtask would be. However, the increase in number of workers to one subtask indicates the decrease in number of subtasks, which comes the issue of mapping conflict. For example, if we only use color of eyes and shape of face in facial recognition, it may cause plenty of mapping conflicts since many people can share two same features. Consequently, it may result in inefficient task completion due to small information gain. On the other hand, with the decreased number of subtasks, each subtask is likely to hold much more information of the entire task, causing the leakage of requester's privacy.

In addition to task division, the occurrence of mapping conflicts highly depends on how task answer is distributed. To capture the impact of task answer distribution, we use $F^u(x)$ to denote the cumulative distribution function for number of subtasks when the conflict first occurs in uniform distribution and use $F^a(x)$ to denote that in asymmetrical distribution. We first show a useful lemma for no mapping conflict analysis in case of uniform task answer distribution.

Lemma 3: When task answer is uniformly mapped to the feature space, the probability of no mapping conflict satisfies

$$\bar{\mathbb{P}}(M, N; L) > e^{-L^2/M^N} \frac{1 + O\big(1/M^N\big)}{1 + O\big(1/M^N - L\big)}. \tag{20}$$

Proof: See Appendix E. ■

According to Lemma 3, to guarantee no mapping conflict, the number of subtasks should be larger than $2\log_M L$ if the answer is uniformly distributed in feature space. We would like to notice that the issue of mapping conflicts in uniform distribution is called as *Birthday Problem*. According to the conclusion in [21], we obtain the probability of mapping conflicts in uniform distribution (i.e., birthday surprise) is larger than that in asymmetrical distribution. That is, for $\forall x \in \mathbb{N}^+$, $F^u(x) < F^a(x)$. Therefore, if task answer is asymmetrically mapped to feature space, the number of subtasks should be even larger than $2\log_M L$.

2) 1-Hop Allocation: Given the number of divided subtasks N, the next is to determine how to allocate subtasks among requester's 1-hop friends. Such 1-hop allocation decision together with task division constitutes the major privacy-preserving strategies. In particular, if task division specifies the lower bound of number of subtasks (i.e., worst case for privacy insurance), then surely 1-hop allocation decision captures the upper bound of privacy-preserving probability given subtask number N (i.e., best-case for privacy insurance).

Recall that privacy insurance constraint is defined on b, i.e., the minimum number of subtasks which can include the information of entire task. 1-hop task allocation is, in essence, similar to *coupon collector's problem* [22]. Let $\mathbf{S}$ denote the number of friends who have b different subtasks, and $\mathbf{s_i}$ denote the number of friends needed to have the ith subtask after $i-1$ subtasks. The probability to have the ith subtask after $i-1$ subtasks is $[(n-i+1)/n]$ with expectation $\mathbb{E}(\mathbf{s_i}) = [n/(n-i+1)]$. Thus, the expectation number of friends who have b different subtasks in 1-hop friend's ego network is

$$\begin{aligned}\mathbb{E}(\mathbf{S}) &= \sum_{i=1}^{b} \mathbb{E}(\mathbf{s_i}) \\ &= \sum_{i=1}^{b} \frac{n}{n-i+1}\end{aligned}$$

$$= n\sum_{i=1}^{b}\frac{1}{n-i+1}$$
$$= n\ln\frac{n}{n-b}. \tag{21}$$

Consider a special case. If we have to possess N subtasks that we can know the information of entire task, then the expectation number is $N\ln N$. It is obvious that task requester's privacy cannot be guaranteed if his ego network is the subset of one 1-hop friend's ego network.

By Definition 1, to calculate the probability of privacy preserving of task requester, we need to calculate the probability that the number of types of accessible subtasks is less than b for 1-hop friends who share the most friends with task requester. On the basis of this, we can derive the following privacy-preserving probability for the proposed mechanism.

Lemma 4: If the following condition satisfies:

$$\max_{1\le i\le|\mathcal{V}_0|}\left|\mathcal{V}_0\cap\mathcal{V}_{1,i}\right| = cn\ln\frac{n}{n-b} \tag{22}$$

where c is a constant, and the probability of privacy preserving of task requester is less than $1-(n/b)\ln(n/n-b)$.

Proof: See Appendix F. ■

Using Chernoff Bound, we will get a tighter upper bound.

Theorem 3: For any $\varepsilon > 0$

$$\mathbb{P}\left(|\mathbf{S}-\mu| > \frac{n-b+1}{n}\varepsilon\mu\right) \le \exp\left(-\frac{\mu\varepsilon^2}{2+\varepsilon}\right) \tag{23}$$

where $\mu = n\ln(n/n-b)$.

Proof: See Appendix G. ■

Such exponentially decreasing bound is much tighter than the bound derived by Lemma 4. By Theorem 3, in order to increase the probability of requester's privacy preserving, we need to divide more types of subtasks to 1-hop friends.

3) Tradeoff Between Accuracy and Privacy: Theorem 3 states that to ensure task requester's privacy, large N should be utilized. Intuitively speaking, however, large N will decrease the average number of 1-hop friends assigned to the same subtask, which we call as task redundancy. The reduce in task redundancy will have a negative influence on answer accuracy. Therefore, there exists a tradeoff between individual privacy and answer accuracy. The following theorem shows the relationship between accuracy and redundancy.

Theorem 4: To achieve error probability less than ε, the task redundancy should be scaled as

$$\frac{|\mathcal{V}_0|}{N} = O\left(M\log\frac{M}{\varepsilon}\right). \tag{24}$$

Proof: See Appendix H. ■

According to Theorem 4, if the number of types of subtask N is large, to guarantee error probability is less than ε, the number of labels for each subtask should be decreased. Since $M \ge 2$, N cannot be increased infinitely.

C. Discussions

1) Discussions About Varying Ability Case: This paper focuses on utilizing the social IoT paradigm to tackle labeling tasks. Existing researches to labeling tasks mostly make a simplifying assumption that worker ability (or confusion matrix) is given [15]. Chances are, however, that worker ability may experience changes due to some realistic factors, such as limited resources. Such dynamic changes manifest in task accuracy that it can provide. A large body of literature was dedicated to the interactions between ability and accuracy [23]. In particular, workers with high ability usually provide more accurate results, and high accuracy of service offering can also contribute to worker ability. It is worth in-depth study on routing incentives with varying worker ability. Much attention should be given to how to model the positive feedback between ability and accuracy, and how to develop privacy-preserving incentive mechanism taking varying ability into account.

2) Discussions About Multirequester Case: In our mechanism, the entire task is divided into multiple subtasks and friends are motivated to route each of them efficiently, thus achieving a tradeoff between privacy and accuracy. Actually, our analysis is based on the single requester case, where each worker is assumed to finish at most one subtask.[7] However, our model can be applied to study the multirequester case. On the one hand, when each worker responses at most one subtask, the routing process for all subtasks is independent of each other. Essentially, the multirequester scenario can be regarded as the single requester case with much more subtasks routed among social IoT. On the other hand, consider the case where each worker can response multiple subtasks. Given infinite resources for each worker, the multirequester scenario is actually a set of single requester cases, each of which is independent of each other. But when workers are limited by finite resources, they can only choose part of subtasks to response, i.e., different subtasks are tightly coupled together. Notice that a full analysis of task coupling effect under limited worker resources is out of the scope of this paper. It surely will be an interesting future work to study routing incentives with coupling task response.

V. Performance Evaluation

In this section, we conduct extensive simulations to validate the theoretical analysis above. We consolidate the effectiveness of multihop routing and show the expected number of friends which will reveal the privacy of task requester.

A. Simulation Setup

Consider the realistic data set from Facebook [24] to represent the social IoT paradigm. The data we obtained are composed by ten worker's ego networks which partially overlap with each other. Notice that the network model of interest is an indirected graph, where the edge represents the social relationship between two workers.

As for possible answers to subtasks, we set the fixed number of values in the feature set as $M = 4$. Unless otherwise specified, we suppose the number of subtasks $N = 10$ and the minimum number of subtasks including the entire task information b are uniformly distributed in $[5, 8]$. There

[7]The allocation issue regarding one-to-one mapping between workers and tasks has been intensively studied in the literatures [15], [16]. Hence, it is reasonable to assume each worker can finish at most one subtask.

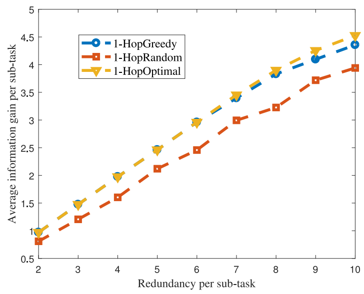


Fig. 3. Average information gain with a varying redundancy.

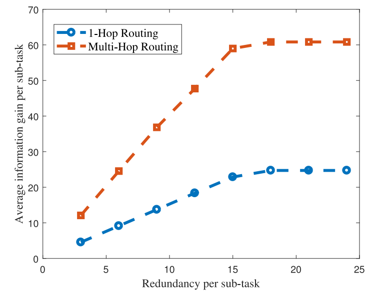


Fig. 4. Average information gain with and without multihop incentive.

TABLE I
RUNNING TIME FOR THE PROPOSED ASSIGNMENT VERSUS BASELINE ALGORITHMS

Algorithms	Average Running Time
1-HopOptimal	481.65 sec
1-HopGreedy	0.06 sec
1-HopRandom	0.05 sec

are $|\mathcal{V}_0|$ workers in the task requester's ego network. The ratio $[(|\mathcal{V}_0|)/N]$ is referred to as *task redundancy*. We generate worker ability parameters γ_i from a hyper-parameter $a_i \in (0, \infty)$ according to the following function adapted from [17]:

$$\gamma_i = \frac{M-1}{M}\left(\frac{1}{M-1} + \left(\frac{1}{2}\right)^{\frac{1}{a_i}}\right) \tag{25}$$

where $\gamma_i \in ((1/M), 1)$ monotonically increases with a_i. The hyper-parameters are drawn from exponential distribution $f(x; \lambda) = \lambda e^{-\lambda x}, x \geq 0$. The expectation $\mu = (1/\lambda)$ can be viewed as the *overall* quality of workers. Besides, each task requester will randomly impose a budget constraint within $[3, 6]$. The payment allocation ratio α follows the uniform distribution within $[0.4, 0.6]$.

We compare the proposed privacy-preserving incentive mechanism with two baseline algorithms.

1) *Random Routing [25]:* Workers are randomly selected into the target task-worker assignment $\mathcal{W}$ under the constraints in 1-hop routing (denoted by 1-HopRandom).
2) *Brute-Force Routing [26]:* Tasks are assigned to workers by brute-force search under the constraints in 1-hop routing (denoted by 1-HopOptimal).

B. Results for Multihop Routing Incentive

1) 1-Hop Myopic Routing: In the scenario of 1-hop myopic routing, we illustrate the performance comparison of average information gain and running time per subtask.

We first plot the average information gain per subtask with task redundancy $[(|\mathcal{V}_0|)/N]$ varying from 3 to 24 with an interval 3 in Fig. 3. Intuitively, the larger task redundancy is, the more accurate each subtask would be. We can see that both our proposed assignment 1-HopGreedy and 1-HopRandom achieve higher information gain as task redundancy increases, and ours is more superior. For better comparison, we also show the results for 1-HopOptimal as an upper bound of average information gain. The difference between the average information gain achieved by ours and the best achievable level is small, which is in accordance with our theoretical analysis.

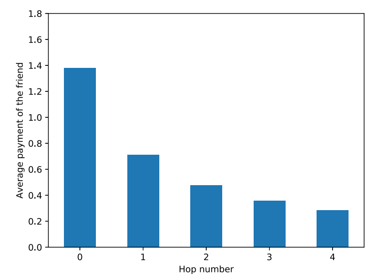


Fig. 5. Payment allocation with respect to routing path length.

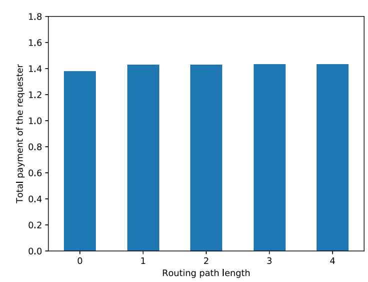


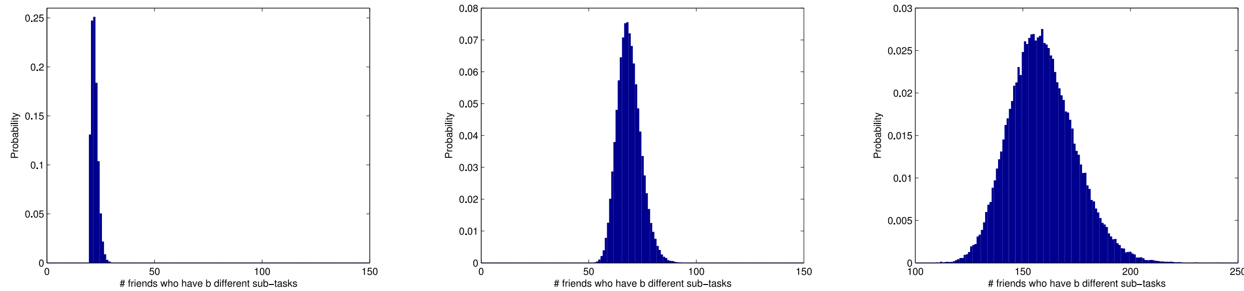
Fig. 6. Total payment with respect to routing path length.

Table I presents that these algorithms share different average running time. While 1-HopOptimal is very slow due to the NP-hardness of the problem, the other two solutions are relatively fast. Combining with Fig. 3, we observe that 1-HopGreedy provides a good tradeoff between performance and complexity, which serves as a near-optimal average information gain with very low running time.

2) Multihop Routing: By exploiting mutual friendship between workers, our multihop routing incentive mechanism is developed under the strictly multihop routing scoring rule. To testify the performance of our multihop routing incentive policy, we next compare the average information gain in 1-hop routing and multihop routing, where workers' ability parameters are generated in the same way for the former case.

1) *1-hop Routing:* Task requester's 1-hop friends finish subtasks by directly providing their responses.
2) *Multihop Routing:* Each worker forwards the assigned subtask to the friend with the highest effort in his ego network so that his effort is the highest.

The number of total workers in the graph $|\mathcal{V}|$ is set as $[(|\mathcal{V}|)/(|\mathcal{V}_0|)] = [(|\mathcal{V}_0|)/N]$. We generate a random graph with



(a) (b) (c)

Fig. 7. Distribution of number of friends who have b different subtasks when the total number of subtasks $N = 100$. (a) $b = 20$. (b) $b = 50$. (c) $b = 80$.

each node representing a worker and each edge representing a friendship. From Fig. 4, we observe that as task redundancy increases, the average information gain under these two mechanisms increases and multihop routing achieves the highest information gain. Compared to 1-hop routing, multihop routing would be more promising in exploiting collaborative ability and sociability, in that it can route tasks to more capable workers lying beyond requester's ego network. Especially when high answer accuracy is guaranteed, the benefits of multihop routing are more likely to achieve. This explains why the gap between them becomes larger as $[(|\mathcal{V}_0|)/N]$ increases.

Multihop routing cannot completely exert its functions without effective payment-based incentive policy. Fig. 5 illustrates how payment allocation changes with respect to routing path length. Actually, the i-hop friend's payment is based on the incremental quality he provides for refining answer inference. Hence, the payment paid to potential experts decreases exponentially as the routing path length increases, which is the premise of guaranteeing the feasibility of multihop routing. With such payment allocation in place, we go further to investigate the influence of routing path length on the total payment for task requester to motivate his 1-hop friends. From Fig. 6, we can see that the overall trend in total payment is increasing slightly in terms of routing path length. Combining the performance comparison results in Fig. 4, Fig. 6 further demonstrates the potential benefits of multihop routing incentive policy. That is, multihop routing can greatly enhance the utility of social crowdsourcing service, but does not make much difference in total payment.

C. Results for Privacy Preserving

In the section, we first evaluate the distribution of number of friends who have b different subtasks when b varies. From Fig. 7, we observe that when b is larger, the distribution is more divergent. Intuitively speaking, since collecting each different subtask is an independent process, so the total variance is the sum of that of each process. Hence, larger b indicates larger variance. Furthermore, the distribution of number of friends is around the expectation number which validates the correctness of Lemma 4.

We further conduct experiments to consolidate the conclusion of Lemma 4 as shown in Fig. 8. We set the total number of subtasks $N = 100$. Each data point in the following simulations is derived from 10 000 experiments. It can be obviously

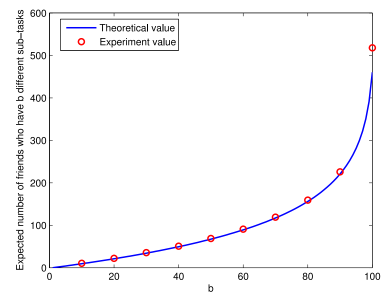


Fig. 8. Average (expected) number of friends who have b different subtasks when the total number of subtasks $N = 100$.

observed that the average (expected) number of friends who have b different subtasks increases faster as b becomes larger because it becomes more difficult to collect the last few subtasks. As illustrated in Fig. 8, the experimental value fits well with the theoretical value.

VI. Related Work

In this section, we briefly highlight three key desirable properties of social crowdsourcing.

A. Task Assignment

In the seminal paper of crowdsourcing paradigm [15], [27], Karger *et al.* constructed a bipartite graph to allocate labeling tasks and infer true labels with iterative learning algorithm and low-rank matrix approximation. Later, some adaptive assignment approaches were put forward with the assumption that requester can observe the answer and allocate tasks to one worker at a time [28], [29], or wait for a worker to complete all his tasks before moving on to the next worker [30]. Li *et al.* [16] first discovered the best workers and leveraged them exclusively. With the aim of maximizing task coverage while minimizing incentive cost, Wang *et al.* [31] considered how to assign crowdsourcing tasks with multiple spatio-temporal constraints to workers. Wang *et al.* [32] integrated participatory-mode and opportunistic-mode crowdsourcing in a two-phased hybrid task allocation framework called HyTasker, which jointly optimizes them under incentive budget constraint. Cosley *et al.* [33] proposed SuggestBot to perform task routing in Wikipedia, whose goal is to identify a single target node quickly through local routing decisions.

While such single-target task differs from the task routing problem we seek to solve, its results provide theoretical and experimental supports for the prospect that local routing decisions may have positive effect on global performance. However, these approaches have limited efficiency to address the task assignment issue in social IoT. In particular, mutual friendship among workers can often be exploited to seek workers with the highest ability, which is vital to guaranteeing high-quality task finishing.

B. Privacy

In this paper, we built a connection between privacy preserving and task assignment [12]. By exploring the heterogeneity of user privacy degree requirement, Yang *et al.* [34] designed incentive mechanisms to motivate users to assist others achieving *k*-anonymity location privacy. Koh *et al.* [35] proposed a privacy-aware incentive scheme that allows workers to specify their location privacy requirements and improves the spatial coverage of collected dataset. Yuan *et al.* [36] developed a grid-based privacy-preserving framework for spatial crowdsourcing, taking into account both location privacy and content privacy of tasks. Han *et al.* [37] first proposed differentially private and budget-limited mechanisms for mobile crowdsourcing with provable performance bounds. However, currently, existing privacy protection works typically separate privacy preserving from task assignment decisions, which are closely coherent with each other. On the one hand, there exists a tradeoff between privacy and accuracy. On the other hand, as the key to achieving privacy preserving, social decisions involving task division, and 1-hop allocation have direct effects on task assignment later. To break this barrier, we propose a privacy-preserving incentive mechanism, solving the dilemma between task assignment of crowdsourcing and privacy preserving in social IoT.

C. Incentive Mechanism

Recently there are many promising efforts underway to address the incentive issues of crowdsourcing in numerous contexts [38], [39], [40]. For example, Yang *et al.* [41] designed two incentive mechanisms for user centric model and platform centric model. Wang *et al.* [42] focused on how to motivate teams of socially tight-knit workers to truthfully report skill and working cost information. Zhang and van der Schaar [43] developed a nonmonetary incentive mechanism based on participants' reputation. Wang *et al.* [8] presented a novel inference and learning approach in quality-aware incentive mechanism to estimate workers' long-term dynamic quality. Considering online situation where workers may arrive in a random order, Zhao *et al.* [9] designed an online auction mechanism to help task requester select a subset of workers to maximize service value under budget constraint. These works typically focus on single-hop case, which is in the "burden" of identifying expertise. The boom in social networking services, however, offers unique potentials to mutual friendship exploitation, enabling multihop incentive mechanism. In this case, individuals are incentivized to forward tasks to workers who may only have limited expertise but get along with other workers capable of task finishing. The generalized task markets (GTMs) framework developed by Shahaf and Horvitz [44] has much in common with our problem. Its goal is to seek for a coalition of workers whose multiattribute skills meet task requirements. However, GTM approach assumes a binary utility model (i.e., tasks are completed or not). Such practice leads to tractable analysis, but miss key inherent conditional voluntary of workers, especially for those only contributing to task routing.

VII. Conclusion

In this paper, we focus on solving the dilemma between multihop incentive of crowdsourcing and privacy preserving in social IoT, where division is adopted as the method for preserving task requester's privacy. Defining the conditions to ensure privacy, we give guidelines on how many subtasks should an entire task be divided into. To make sure the probability of revealing requester's privacy, we give the upper bound of the probability of privacy reveal. We also illustrate the tradeoff between privacy and task accuracy. Furthermore, we design 1-hop myopic routing policy and multihop payment policy for efficient crowdsourcing task routing in social IoT. By leveraging monotone submodular property, we first propose an assignment algorithm with $1/2$ approximation ratio for 1-hop routing. In multihop routing case, we prove that requester's cost will increase with the length of task forwarding chain, which motivates workers to forward subtasks to friends with the best expertise in their ego networks. In the future, we will design an incentive mechanism based on auction model while considering privacy preserving. We may generalize task type, not limited to labeling problems.

Appendix A
Proof of Lemma 1

For the conditional entropy

$$\begin{aligned} H(p_n^0|\mathcal{W}_n) &= -\sum_{R_{\mathcal{W}_n}\in\mathcal{R}_n}\sum_{m=1}^{M}\mathbb{P}\big(t_n=m,R_{\mathcal{W}_n}\big)\log\mathbb{P}\big(t_n=m|R_{\mathcal{W}_n}\big)\\ &= -\sum_{R_{\mathcal{W}_n}\in\mathcal{R}_n}\sum_{m=1}^{M}\mathbb{P}\big(t_n=m,R_{\mathcal{W}_n}\big)\log\frac{\mathbb{P}\big(t_n=m,R_{\mathcal{W}_n}\big)}{\mathbb{P}\big(R_{\mathcal{W}_n}\big)} \end{aligned} \tag{26}$$

where $\mathbb{P}(t_n = m, R_{\mathcal{W}_n}) = \mathbb{P}(t_n = m)\mathbb{P}(R_{\mathcal{W}_n}|t_n = m)$. Given the independence of responses $R_{\mathcal{W}_n}$, we have

$$\mathbb{P}\big(R_{\mathcal{W}_n}|t_n=m\big)=\prod_{r_n\in R_{\mathcal{W}_n}}\mathbb{P}(r_n|t_n=m). \tag{27}$$

Given further the unbiased prior knowledge, we have

$$\begin{aligned} \mathbb{P}^0(t_n=m) &= \frac{1}{M}\\ \mathbb{P}\big(R_{\mathcal{W}_n}\big) &= \prod_{r_n\in R_{\mathcal{W}_n}}\mathbb{P}(r_n)=\left(\frac{1}{M}\right)^{|\mathcal{W}_n|}. \end{aligned} \tag{28}$$

Then based on the above equations

$$H\left(p_n^0|\mathcal{W}_n\right) = -\sum_{R_{\mathcal{W}_n}\in\mathcal{R}_n}\sum_{m=1}^{M}\frac{1}{M}\prod_{r_n\in R_{\mathcal{W}_n}}\mathbb{P}(r_n|t_n=m) \times \left(\log\frac{1}{M}\prod_{r_n\in R_{\mathcal{W}_n}}\mathbb{P}(r_n|t_n=m) - \log\left(\frac{1}{M}\right)^{|\mathcal{W}_n|}\right). \tag{29}$$

Submitting (27) and (28), we get

$$\begin{aligned} H\left(p_n^0|\mathcal{W}_n\right) &= (1-|\mathcal{W}_n|)\log M - \sum_{i_n\in\mathcal{W}_n}\frac{\gamma_{i_n}}{M}\left[\log\frac{\gamma_{i_n}}{M}-\log\frac{1}{M}\right] \\ &\quad + (M-1)\frac{1-\gamma_{i_n}}{M(M-1)}\left[\log\frac{1-\gamma_{i_n}}{M(M-1)}-\log\frac{1}{M}\right] \\ &= (1-|\mathcal{W}_n|)\log M - \sum_{\mathcal{W}_n\in\mathcal{W}_n}\frac{\gamma_{i_n}}{M}\log\gamma_{i_n} \\ &\quad + \frac{1-\gamma_{i_n}}{M}\log\frac{1-\gamma_{i_n}}{M-1} \\ &= \log M - \sum_{i_n\in\mathcal{W}_n} e(\gamma_{i_n}). \end{aligned} \tag{30}$$

The entropy of the prior knowledge is simply

$$\begin{aligned} H\left(p_n^0\right) &= -\sum_{m=1}^{M}\mathbb{P}^0(t_n=m)\log\mathbb{P}^0(t_n=m) \\ &= -M\times\frac{1}{M}\log\frac{1}{M} \\ &= \log M. \end{aligned} \tag{31}$$

Thus,

$$\begin{aligned} U(\mathcal{W}) = \mathbb{E}[IG] &= \sum_{n=1}^{N} H\left(p_n^0\right) - H\left(p_n^0|\mathcal{W}_n\right) \\ &= \sum_{n=1}^{N}\sum_{i_n\in\mathcal{W}_n} e(\gamma_{i_n}). \end{aligned} \tag{32}$$

This finishes the proof of Lemma 1.

Appendix B
Proof of Theorem 1

In our mechanism, in order to avoid privacy reveal and guarantee high-quality task finishing, the assignment allocates exactly one subtask to each 1-hop friend in requester's ego network. We encode such constraint in a partition matroid.

A matroid is a structure that captures and generalizes the notion of linear independence in vector space [44]. It can be specified as $O = (\mathcal{J}, \mathcal{I})$, where $\mathcal{J}$ is the ground set and $\mathcal{I} \in 2^{\mathcal{J}}$ is the collection of subsets of elements in $\mathcal{J}$ that are independent. In 1-hop routing, the ground set is $\mathcal{J} = N \times |\mathcal{V}_0|$. If we partition $\mathcal{J}$ into disjoint sets $B_i = \{i\} \times N$ associated with possible assignments for each 1-hop friend $i \in \mathcal{V}_0$, we can construct the desired partition matroid by defining an independent set to include no more than one element from each of these sets.

As above-mentioned, such 1-hop routing problem falls into monotone submodular optimization with cardinality constraints. To this end, we are inspired by [45] to propose the greedy selection process, shown in Algorithm 1, which sorts by increasing ability of 1-hop friends and prioritizes assigning subtasks to high-ability friends. Such algorithm starts with the empty set $\mathcal{W}_n$ for each subtask, and adds the element maximizing the derivative ΔH_n (ties broken arbitrarily) until there is no element which can be added to create a feasible solution. Using the results of [45], we can conclude that the proposed greedy 1-hop routing algorithm is guaranteed to produce a solution $\mathcal{W} \geq (1/2)\max_{W\in\mathcal{I}} U(W)$.

Appendix C
Proof of Lemma 2

Using the formula to calculate the sum of geometric series, we have

$$\begin{aligned} & c_k + \sum_{i=1}^{k-1}\alpha c_k(1+\alpha)^{k-1-i} \\ &= c_k + \alpha c_k\frac{(1+\alpha)^{k-1}-1}{(1+\alpha)-1} \\ &= c_k(1+\alpha)^{k-1}. \end{aligned} \tag{33}$$

This finishes the proof of Lemma 2.

Appendix D
Proof of Theorem 2

Consider a multihop routing path for any subtask t_n. Under strictly proper routing scoring rule, for any $k \in \{1, 2, \ldots, K-1\}$, the kth worker truthfully updates the answer inference in two ways. One is that truthful processing maximizes inference accuracy, and thus the expected effect $e(\gamma_k)$. The other is that for any worker w who may be routed the subtask, truthful processing by the kth worker maximizes the effect $e(\gamma_w)$. In this case, the expected effect of $e(\gamma_w)$ (from the perspective of the kth worker) is strictly greater when the kth worker processes truthfully.

According to the proposed payment policy, the kth worker will get an extra payment $\alpha(e(\gamma_{k+1}) - e(\gamma_k))$ if he chooses to forward the subtask. In order to maximize the payment, the kth worker will forward the subtask to the worker with the highest effect or ends the routing if his effect is the highest in his ego network. Above all, we can conclude that no one on the routing path wants to deviate from this equilibrium strategy, given the belief that all other workers follow it.

Appendix E
Proof of Lemma 3

Using *Stirling's approximation*, the probability of no mapping conflicts (19) can be developed as

$$\bar{\mathbb{P}}(M,N;L) = \frac{\sqrt{2\pi M^N}\left(M^N/e\right)^{M^N}}{\sqrt{2\pi\left(M^N-L\right)}\left(M^N-L/e\right)^{M^N-L}}$$

$$\times \frac{1+O\big(1/M^N\big)}{1+O\big(1/M^N-L\big)}$$
$$= \sqrt{\frac{M^N}{M^N-L}} \cdot \left(1-\frac{L}{M^N}\right)^{L-M^N} \cdot e^{-L}$$
$$\times \frac{1+O\big(1/M^N\big)}{1+O\big(1/M^N-L\big)}. \tag{34}$$

Noting that $(1-x)^y < e^{-xy} (y > 0)$, we have

$$\bar{\mathbb{P}}(M,N;L) > \sqrt{\frac{M^N}{M^N-L}} \cdot e^{L-L^2/M^N} \cdot e^{-L}$$
$$\times \frac{1+O\big(1/M^N\big)}{1+O\big(1/M^N-L\big)}$$
$$= \sqrt{\frac{M^N}{M^N-L}} \cdot e^{-L^2/M^N}$$
$$\times \frac{1+O\big(1/M^N\big)}{1+O\big(1/M^N-L\big)}$$
$$> e^{-L^2/M^N} \cdot \frac{1+O\big(1/M^N\big)}{1+O\big(1/M^N-L\big)}. \tag{35}$$

This finishes the proof of Lemma 3.

Appendix F
Proof of Lemma 4

According to the Markov Inequality, we have

$$\mathbb{P}(\mathbf{S} > b) < \frac{\mathbb{E}(\mathbf{S})}{b} = \frac{n}{b} \ln \frac{n}{n-b}. \tag{36}$$

Hence, we can derive that

$$\mathbb{P}(\mathbf{S} < b) < 1 - \frac{n}{b} \ln \frac{n}{n-b}. \tag{37}$$

We have completed the proof.

Appendix G
Proof of Theorem 3

Since $\mathbf{s_i}$ are all independent random variables and $0 \leq [(n-b+1)/n]\mathbf{s_i} \leq 1$ for $\forall i \in \mathcal{N}$, using Chernoff Bound, we have

$$\mathbb{P}\left(|\mathbf{S}-\mu| > \frac{n-b+1}{n}\varepsilon\mu\right) \leq \exp\left(-\frac{\mu\varepsilon^2}{2+\varepsilon}\right). \tag{38}$$

This finishes the proof of Theorem 3.

Appendix H
Proof of Theorem 4

It is obvious that task redundancy denoting by the average number of 1-hop friends assigned to each type of subtasks is $|\mathcal{V}_0|/N$. In our mechanism, each subtask has one true answer from M possible labels. According to Karger's result in [15], it is possible to obtain an answer to each task correctly with probability $1-\varepsilon$ as long as the redundancy per task is $O((M/q)\log(M/\varepsilon))$, where q is the crowd-quality parameter that is, related to friend reliability. Considering the stability of pool of 1-hop friends, q can be regarded as a constant. Hence, task redundancy can be scaled as $O(M\log(M/\varepsilon))$. We have completed the proof.

References

[1] J. Heer and M. Bostock, "Crowdsourcing graphical perception: Using mechanical turk to assess visualization design," in *Proc. ACM CHI*, Apr. 2010, pp. 203–212.

[2] H. Shen, Z. Li, J. Liu, and J. E. Grant, "Knowledge sharing in the online social network of Yahoo! Answers and its implications," *IEEE Trans. Comput.*, vol. 64, no. 6, pp. 1715–1728, Jun. 2015.

[3] L. Wang, H. Wu, Z. Han, P. Zhang, and H. V. Poor, "Multi-hop cooperative caching in social IoT using matching theory," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2127–2145, Apr. 2018.

[4] V. Sharma, I. You, D. N. K. Jayakody, and M. Atiquzzaman, "Cooperative trust relaying and privacy preservation via edge-crowdsourcing in social Internet of Things," *Future Gener. Comput. Syst.*, vol. 92, pp. 758–776, Mar. 2019.

[5] K. Wang, X. Qi, L. Shu, D.-J. Deng, and J. J. P. C. Rodrigues, "Toward trustworthy crowdsourcing in the social Internet of Things," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 30–36, Oct. 2016.

[6] *Statistics of Facebook*. Accessed: Aug. 7, 2019. [Online]. Available: http://www.socialbakers.com

[7] X. Gan, Y. Li, W. Wang, L. Fu, and X. Wang, "Social crowdsourcing to friends: An incentive mechanism for multi-resource sharing," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 795–808, Mar. 2017.

[8] H. Wang, S. Guo, J. Cao, and M. Guo, "MeLoDY: A long-term dynamic quality-aware incentive mechanism for crowdsourcing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 4, pp. 901–914, Apr. 2018.

[9] D. Zhao, X.-Y. Li, and H. Ma, "Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 647–661, Apr. 2016.

[10] P. Korshunov, S. Cai, and T. Ebrahimi, "Crowdsourcing approach for evaluation of privacy filters in video surveillance," in *Proc. ACM CrowdMM*, Oct. 2012, pp. 1–6.

[11] P. Dai, Mausam, and D. S. Weld, "Decision-theoretic control of crowd-sourced workflows," in *Proc. AAAI*, Jul. 2010, pp. 1168–1174.

[12] L. R. Varshney, "Privacy and reliability in crowdsourcing service delivery," in *Proc. SRII*, Jul. 2012, pp. 55–60.

[13] A. Kulkarni, M. Can, and B. Hartmann, "Turkomatic: Automatic, recursive task and workflow design for mechanical turk," in *Proc. AAAI Workshop*, Aug. 2011, pp. 2053–2058.

[14] Y. Baba and H. Kashima, "Statistical quality estimation for general crowdsourcing tasks," in *Proc. ACM KDD*, Aug. 2013, pp. 554–562.

[15] D. R. Karger, S. Oh, and D. Shah, "Efficient crowdsourcing for multi-class labeling," in *Proc. ACM SIGMETRICS*, Jun. 2013, pp. 81–92.

[16] H. Li, B. Zhao, and A. Fuxman, "The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing," in *Proc. ACM WWW*, Apr. 2014, pp. 165–176.

[17] J. Bragg, A. Kolobov, and D. S. Weld, "Parallel task routing crowd-sourcing," in *Proc. AAAI*, Jul. 2014, pp. 1–11.

[18] R. Hanson, "Logarithmic markets coring rules for modular combinatorial information aggregation," *J. Prediction Markets*, vol. 1, no. 1, pp. 3–15, 2012.

[19] H. Zhang, E. Horvitz, Y. Chen, and D. C. Parkes, "Task routing for prediction tasks," in *Proc. AAMAS*, Jun. 2012, pp. 889–896.

[20] R. Bakhshandeh, M. Samadi, Z. Azimifar, and J. Schaeffer, "Degrees of separation in social networks," in *Proc. 4th Annu. Symp. Comb. Search*, Jul. 2011, pp. 18–23.

[21] M. S. Klamkin and D. J. Newman, "Extensions of the birthday surprise," *J. Comb. Theory*, vol. 3, no. 3, pp. 279–282, Oct. 1967.

[22] *Coupon Collector's Problem*, Wikipedia, San Francisco, CA, USA, 2014.

[23] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *J. Roy. Stat. Soc. C (Appl. Statist.)*, vol. 28, no. 1, pp. 20–28, Jan. 1979.

[24] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[25] A. Papoutsaki *et al.*, "Crowdsourcing from scratch: A pragmatic experiment in data collection by novice requesters," in *Proc. AAAI*, Jul. 2015, pp. 140–149.

[26] X. Fan, P. Yang, and Q. Li, "Fairness counts: Simple task allocation scheme for balanced crowdsourcing networks," in *Proc. IEEE MSN*, Dec. 2015, pp. 258–263.

[27] D. R. Karger, S. Oh, and D. Shah, "Budget-optimal crowdsourcing using low-rank matrix approximations," in *Proc. IEEE Allerton*, Sep. 2011, pp. 284–291.

[28] X. Zhang, Y. Wu, L. Huang, H. Ji, and G. Cao, "Expertise-aware truth analysis and task allocation in mobile crowdsourcing," in *Proc. IEEE ICDCS*, Jul. 2017, pp. 922–932.

[29] C.-J. Ho and J. W. Vaughan, "Online task assignment in crowdsourcing markets," in *Proc. AAAI*, Jul. 2012, pp. 45–51.
[30] C.-J. Ho, S. Jabbari, and J. W. Vaughan, "Adaptive task assignment for crowdsourced classification," in *Proc. ICML*, Jun. 2013, pp. 534–542.
[31] L. Wang, Z. Yu, Q. Han, B. Guo, and H. Xiong, "Multi-objective optimization based allocation of heterogeneous spatial crowdsourcing tasks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 7, pp. 1637–1650, Jul. 2018.
[32] J. Wang *et al.*, "HyTasker: Hybrid task allocation in mobile crowd sensing," *IEEE Trans. Mobile Comput.*, to be published.
[33] D. Cosley, D. Frankowski, L. G. Terveen, and J. Riedl, "SuggestBot: Using intelligent task routing to help people find work in Wikipedia," in *Proc. ACM IUI*, Jun. 2007, pp. 32–41.
[34] D. Yang, X. Fang, and G. Xue, "Truthful incentive mechanisms for *k*-anonymity location privacy," in *Proc. IEEE INFOCOM*, Mar. 2013, pp. 2994–3002.
[35] Y. J. Koh, W. G. Peters, D. Leong, I. Nevat, and W. C. Wong, "Privacy-aware incentive mechanism for mobile crowd sensing," in *Proc. IEEE ICC*, May 2017, pp. 1–6.
[36] D. Yuan, Q. Li, G. Li, Q. Wang, and K. Ren, "PriRadar: A privacy-preserving framework for spatial crowdsourcing," *IEEE Trans. Inf. Forensics Security*, to be published.
[37] K. Han, H. Liu, S. Tang, M. Xiao, and J. Luo, "Differentially private mechanisms for budget limited mobile crowdsourcing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 934–946, Apr. 2019.
[38] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Robust incentive tree design for mobile crowdsensing," in *Proc. IEEE ICDCS*, Jul. 2017, pp. 458–468.
[39] W. Wu *et al.*, "Incentive mechanism design to meet task criteria in crowdsourcing: How to determine your budget," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 2, pp. 502–516, Feb. 2017.
[40] F. Tian, B. Liu, X. Sun, X. Zhang, G. Cao, and L. Gui, "Movement-based incentive for crowdsourcing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7223–7233, Aug. 2017.
[41] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1732–1744, Jun. 2016.
[42] W. Wang *et al.*, "Strategic social team crowdsourcing: Forming a team of truthful workers for crowdsourcing in social networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1419–1432, Jun. 2019.
[43] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2140–2148.
[44] D. Shahaf and E. Horvitz, "Generalized task markets for human and machine computation," in *Proc. AAAI*, Jul. 2010.
[45] G. L. Nemhauser and L. A. Wolsey, "Maximizing submodular set functions: Formulations and analysis of algorithms," *North Holland Math. Stud.*, vol. 59, no. 1, pp. 279–301, Jan. 1981.

Xiaoying Gan (M'09) received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006.

She is currently an Associate Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University. From 2017 to 2019, she has been supporting the open research fund of National Mobile Communications Research Laboratory, Southeast University, Nanjing, China. From 2009 to 2010, she was a Visiting Researcher with the California Institute for Telecommunications and Information Technology, University of California at San Diego, San Diego, CA, USA. Her current research interests include network economics and wireless resource management.

Yuqing Li received the B.S. degree in communication engineering from Xidian University, Xi'an, China, in 2014. She is currently pursuing the Ph.D. degree in electronic engineering with Shanghai Jiao Tong University, Shanghai, China.

Her current research interests include edge/mobile computing, social aware networks, privacy/security, and network economics.

Yixuan Huang is currently pursuing the B.S. degree with the School of School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China.

His current research interests include crowdsourcing and social network.

Luoyi Fu (M'17) received the B.E. degree in electronic engineering and the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009 and 2015, respectively.

She is currently an Assistant Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. Her current research interests include social networking and big data, scaling laws analysis in wireless networks, connectivity analysis, and random graphs.

Xinbing Wang (SM'12) received the B.S. degree in automation from Shanghai Jiao Tong University, Shanghai, China, in 1998, the M.S. degree in computer science and technology from Tsinghua University, Beijing, China, in 2001, and the Ph.D. degree with a major in electrical and computer engineering and minor in mathematics from North Carolina State University, Raleigh, NC, USA, in 2006.

He is currently a Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University. His current research interests include resource allocation and management in mobile and wireless networks, cross-layer call admission control, asymptotics analysis of hybrid systems, and congestion control over wireless ad hoc and sensor networks.

Prof. Wang has been a member of the Technical Program Committees of several conferences, including ACM MobiCom in 2012, ACM MobiHoc in 2012, and IEEE INFOCOM from 2009 to 2013.