On Designing Collusion-Resistant Incentive Mechanisms for Mobile Crowdsensing Systems

***Abstract*—With the rise of smartphones, crowdsensing applications have tremendously increasing trending. As a result, many incentive issues are worth researching. In this paper, we give detailed discussions on collusion resistant incentive mechanisms for crowdsensing applications. Two fundamental criteria are found: one is to judge whether a truthful incentive mechanism can resist any collusion without profit trading by achieving group strategyproof equilibrium, and the other is to judge whether a truthful mechanism can defend against any collusion even with profit trading by achieving t-truthful equilibrium. Furthermore, we also propose our solution which can resist any form of collusion attack, even including profit trading among the attackers. Extensive simulations verify our results. To the best of our knowledge, we are the first to investigate the collusion resistant incentive mechanisms for crowdsensing applications.**

I. Introduction

With the rapidly increasing use of portable devices (e.g. smartphones), there are plenty of research works [1], [2], [3] and numerous applications [4], [5], [6], [7] on mobile sensing. Sensing apps on smartphones allow people to sense and collect different kinds of information ubiquitously. To sufficiently explore the sensing capability of each sensor within the network, crowdsensing [8] [1] techniques can be applied. For instance, by using the crowdsensing platform, we can access the up-to-date parking slots information on the neighboring community [4], [6], or monitor weather changes [7].

The incentive issues in crowdsensing systems have been extensively studied [9], [10], [11], [12], [13]. Crowdsensing work embodies costs in nature (e.g. service fees, battery usage, etc). Hence we need to stimulate the participating users to contribute their sensed data, by giving them some incentives (e.g., in monetary forms) to cover the cost incurred by individual sensing activities. Existing incentive mechanisms (e.g., [9], [10], [13]) use game theoretical approaches to achieve different equilibria. For instance, strategyproof crowdsensing incentive mechanisms[14], [15], [16] can guarantee that it is to the best interest of each participating user to honestly report their information such as sensing cost. However, strategyproofness cannot eliminate the possibility that a group of users collaborate to cheat in the system and harm the system or other users' utility. Unfortunately, there has not been any existing work that systematically investigates the collusion attack problem in crowdsensing incentive mechanisms.

In real-world crowdsensing applications, a single entity may simultaneously own a number of mobile devices, e.g. smartphones, laptops, tablets, watches, etc. These devices may have diverse sensing costs and but can participate the same crowdsensing task. In the crowdsensing procedure, since the payment that each user receives is usually correlated to its sensing contribution and the involved cost, the same owner can manipulate a group of devices' sensing strategies at the same time and make them collaboratively cheat (e.g., reporting carefully-calculated fake cost information to rule out other potential participants) for better payoffs as a collusion. It creates serious problems for the platform, such as losing system-wide utility and deteriorating other users' enthusiasm to participate.

Collusion attack is an important but difficult problem to address in general [17], [18], [19]. The difficulty lies in two aspects: **1)** A variety of collusion strategies (e.g., with or without profit trading) make the mitigation challenging. It means that in practice it requires security expertise to consider as many collusion scenarios as possible for a specific setting and provide technical solutions to mitigate the known collusion attacks. But like other penetration analysis, there is no guarantee of guard against unexamined cases. **2)** Difficulty exists when trying to provide theoretical guarantee of guard in the incentive mechanism design. As we mentioned earlier, since only strategyproofness cannot provide collusion resistance [17], people try to use other game theoretical solution concepts such as *Group Strategyproof Equilibrium* [20] to achieve collusion resistance. However, within the scope of wireless networks, sometimes it is not possible at all to achieve any group strategyproof equilibrium in some games. For example, one can never achieve any group strategyproofness in theory for noncooperative routing games in wireless ad hoc networks [17]. When it is possible that colluding nodes can trade profit among them, this problem becomes even harder. For the same routing problem, we cannot obtain any strategyproofness if the colluding users can trade profit among them [18].

In this paper we systematically study the collusion-resistance problem in crowdsensing incentive mechanism design. Our goals are not only to provide theoretical results, for example, on the possibility of achieving group strategyproofness, but also to make the theoretical results have direct benefits in practice. In particular, we notice that even with the formal definition of group-strategyproofness and *t-truthfulness* (defense against profit trading), it is very challenging for mobile crowdsensing practitioners to map the theoretical definitions to their concrete systems that they have or are about to design. In this paper, we will provide much more straightforward criteria/techniques to judge whether a given incentive mechanism is group-strategyproof or t-truthful, so that the designer or practitioners can easily apply. This effort has significant benefits in reducing unnecessary overhead and meanwhile maintains the security guarantee. For instance,

if we can easily determine that the incentive mechanism can intrinsically defend against profit trading by achieving t-truthfulness, then we do not have to use extra cryptographic protocols such as Designated Verifier Signatures [21] or Restricted Verifier Signatures [17] that are used to restrict profit trading. Thus the computation and communication overhead are reduced.

Our major contributions are summarized as follows:

(1) We investigate a popular type of crowdsensing games, in which each contributing user takes some sensing time to finish the assigned tasks and earns the promised payment from the platform. We not only find it is feasible to design group strategyproof incentive mechanisms for this type of crowdsensing game, but also obtain the straightforward criteria to judge whether a given strategy-proof incentive mechanism can achieve group strategyproofness. Moreover, we rigorously derive the criteria to judge whether a truthful mechanism is t-truthful to guard against profit trading.

(2) For another group of typical crowdsensing games, in which each contributing user finishes the assigned discrete sensing tasks to earn corresponding payment, we propose a concrete collusion-resistant incentive mechanism. Differently from the case in which the sensing time can take any positive value, assigning discrete sensing tasks may let each user not have to be completely truthful, since usually a small perturbation on truthful strategy will not change the user utility. Hence it can be more difficult to establish a truthful incentive mechanism that can resist collusion, or even profit trading among the colluding nodes. However, we can show that our incentive mechanism can achieve group strategyproofness and t-truthfulness. In other words, our solution can defend against colluded cheating, even with profit trading.

(3) To verify our theoretic results, we also perform extensive simulations, and we show that all the experimental results support our findings.

The remainder of this paper is organized as follows. Section II gives the necessary technical preliminaries including equilibria, collusion attack and profit trading. Section III discusses the basic settings of the crowdsensing game model. Section IV gives the criteria whether a truthful crowdsensing incentive mechanism can be group strategyproof. Section V gives the criteria whether a truthful mechanism can be t-truthful. In Section VI we propose the discrete collusion resistant incentive mechanism which can achieve both group strategyproofness and t-truthfulness. Section VII gives the simulation results and Section VIII summarizes the related literatures. Section IX concludes this paper.

II. Technical Preliminaries

This section describes the technical preliminaries including the basics of crowdsensing game, the concept of equilibrium in game theory, and the idea of profit trading. We also give some useful local properties of bivariate functions in the appendix. Table I presents the frequently used symbols in this paper.

[1]Similarly, by removing subscripts, t denotes the sensing time profile, p denotes the payment profile, u denotes the user utility profile, n denotes the profile of number of tasks, etc.

TABLE I
Frequently used notations

Notation	Description
U	the set of users
n	the number of users: $\|U\|$
s_i	the claimed cost unit (strategy) of user i
s	the strategy profile of all users [1]
s_{-i}	the strategy profile excluding user i's strategy
κ_i	the cost unit of user i
p_i	the (promised) payment to user i
t_i	the required sensing time of user i
u_i	the utility function of user i
u_p	the utility function of the platform
n_i	number of the assigned tasks to user i
m	number of partitions
d	partition length
$f(x, y_0)$	the slice function of $f(x, y)$ by fixing $y = y_0$
B	budget, upper bound of the payment sum $\sum p_i$
R	the number of remaining sensing tasks to do

A. Crowdsensing Game

We introduce the general game model of crowdsensing system, which is used in the majority of existing literatures [9], [11], [22], [23], [24], [25]. The crowdsensing system consists of n participating users, who are all the users *willing* to contribute in the crowdsensing work (e.g. smartphones, mobile sensors) and one crowdsensing platform (e.g. BOINC [26]). The n users voluntarily participate in crowdsensing and contribute their sensing results. The platform aggregates the sensing results from the n users and maintains the crowdsensing system. To compensate the users' sensing costs and stimulate their participation, the platform distributes some payment to each user as the reward. The payment is determined by each user's strategy (i.e., its claimed sensing cost). Denote by c_i the sensing cost of user i, and let p_i be the promised payment from the platform to user i. If user i can complete the sensing task as promised, then the utility of user i is

$$u_i = p_i - c_i. \tag{1}$$

Due to individual rationality, each user wants to maximize its own utility.

For the platform, we may consider the budget feasibility for many practical scenarios [27], [28], [12]. That is, the total sum of the payments p_i must be no more than some budget B. In this paper, we will show that the payment sum of our proposed mechanisms can be upper bounded.

B. Collusion Attack and Group Strategyproof Equilibrium

The major objective of designing incentive mechanisms is to achieve an equilibrium, the state when each user has maximized its own utility and no user can increase its utility by unilaterally taking a different strategy. If one expects to eliminate the possibility that some users take actions deceitfully, a *strategyproof* equilibrium is required. Unfortunately, a strategyproof equilibrium is not strong enough to eliminate collusion among the users, since it is possible that some users *collude* to alter their strategies simultaneously, in order to let some of them gain more utilities without decreasing any utility of theirs [17].

To eradicate collusion, we have the standard game theoretic solution concept of Group Strategyproof Equilibrium as follows.

Definition 1. ***(Group Strategyproof Equilibrium)*** *[20] [14]. Let U be the set of all the players. A strategy profile s^* is Group Strategyproof Equilibrium if for any nonempty subset $S \subseteq U$, for any strategy profile s, either for any $i \in S$, we have*

$$u_i(s_S^*, s_{\overline{S}}) = u_i(s_S, s_{\overline{S}}), \tag{2}$$

or there exists a user $i \in S$ s.t.

$$u_i(s_S^*, s_{\overline{S}}) > u_i(s_S, s_{\overline{S}}). \tag{3}$$

Here $s^* = \kappa$, the real cost profile.

When an equilibrium is group strategyproof, if a subset of users change their strategies simultaneously, either none of them can gain more utility, or some of them have their utilities decreased. In fact, group strategyproof equilibrium can make any collusion without profit trading among the users fail [17].

C. Profit Trading

We also investigate the profit trading collusions in the crowdsensing game. Since utility can always be represented in some monetary form, the users involved in profit trading collusion can share their own utilities with other colluded users. For example, by taking another strategy profile, the utility of user A increases by 3 units, and the utility of user B decreases by 1 unit. Then user A can send 2 units to B to let each of them earn 1 unit. Since group strategyproof equilibrium does not consider utility transfer, it is not strong enough to eliminate the possibility of profit trading. In order to defend against profit trading, we need to achieve the equilibrium in which for any subset of colluded users, their total utilities are maximized. *t-Truthfulness* [29] describes the extent of profit-trading collusion resistance of the incentive mechanism.

Definition 2. ***(t-Truthfulness)*** *Let U be the set of all the players. An incentive mechanism can achieve t-Truthfulness, if for any nonempty subset $S \subseteq U$ such that $|S| \leq t$, for any strategy profile s, we have*

$$\sum_{i \in S} u_i(s_S^*, s_{\overline{S}}) \geq \sum_{i \in S} u_i(s_S, s_{\overline{S}}). \tag{4}$$

Here $s^* = \kappa$, the real cost profile.

Here t is the upper bound of collusion size that the mechanism can resist. It is obvious that if $t = |U|$, a mechanism that can achieve t-Truthfulness can always resist any profit trading collusion.

III. System Models and Problem Formulation

In this section, we formalize a popular type of crowdsensing scenarios, and then describe a typical strategyproof incentive mechanism. We also present an example to illustrate how collusion attack can make this incentive mechanism fail.

A. A Typical Strategyproof Crowdsensing Game

We have discussed the basics of crowdsensing games in Section II-A. However, we have not given any closed-form of each user's sensing cost c_i. According to [11], the sensing cost should increase with the user's participation level. For real-world applications, usually sensing time can be a good metric of each user's participation. For example, more time the user spends on sensing, more sensed data the user can contribute, and more sensing cost may be incurred. We denote by t_i the sensing time (or participation level in more general sense) of user i, and let κ_i be the cost unit of user i. Then we have the cost $c_i = \kappa_i t_i$. Given (1), we have

$$u_i = p_i - \kappa_i t_i. \tag{5}$$

Here the cost unit κ_i is *private information* owned only by user i, since κ_i is closely associated with individual attributes (i.e., battery capacity, smartphone types, etc.). Since each user i has no knowledge of others' real-time cost units κ_{-i}, it is impossible to directly leverage the platform-centric incentive mechanism in [9]. However, we can let each user declare its own cost unit as a strategy (like [17] did). If we can propose an incentive mechanism that can achieve strategyproof equilibrium, the platform can know the cost units globally based on each user's truthful strategy.

We now present a general incentive mechanism that can achieve strategyproof equilibrium. Since the platform has no preknowledge about any user's cost unit, the users need to upload *claimed* cost units to the platform, and then the platform determines the payment and required sensing time of each user. The detailed procedures are as follows:

1) Each user i uploads its claimed cost unit s_i (can be different from κ_i) to the platform;
2) The platform calculates the payment p_i and required sensing time t_i of each user i based on all the claimed cost units $s_1, s_2, \cdots, s_n$;
3) The platform lets each user i know its payment p_i and required sensing time t_i;
4) Each user must finish its required sensing time to earn the payment. Each user can also quit the crowdsensing to avoid deficit;
5) The users, who have finished the sensing time, upload the sensed data to the platform;
6) The platform verifies the uploaded data and confirms the payment.

To be strategyproof, our incentive mechanism requires all users to be honest on their claimed cost units s_i. That is, we desire $s_i = \kappa_i$ for each user i. To achieve this goal, we need delicate closed-forms of payment p_i and required sensing time t_i. It turns out that the concrete mechanism design is open-ended and there exist many different solutions. Since there are extensive literatures [11], [22], [23], [24] on the solutions to achieve strategyproofness in similar scenarios, we leave the discussions in Appendix A.

B. An Example of Collusion on Strategyproof crowdsensing Game

Unfortunately, a strategyproof incentive mechanism is usually not strong enough to resist collusion rigorously. For example, suppose a game has two users i, j s.t. $\kappa_i = \kappa_j = 1$. Given the user strategies s_i and s_j, for user i let its sensing time be $t_i(s_i, s_j) = s_j \mathrm{e}^{-s_i}$ and let its payment be

$$p_i(s_i, s_j) = s_i t_i(s_i, s_j) + \int_{s_i}^{+\infty} t_i(s, s_j)\mathrm{d}s = (s_i + 1)s_j \mathrm{e}^{-s_i}.$$

The case of user j is analogous. Thus the user utilities are

$$\begin{aligned} u_i(s_i, s_j) =& p_i(s_i, s_j) - \kappa_i t_i(s_i, s_j) \\ =& (s_i + 1)s_j \mathrm{e}^{-s_i} - 1 \times s_j \mathrm{e}^{-s_i} \\ =& s_j s_i \mathrm{e}^{-s_i}, \end{aligned} \tag{6}$$

and similarly $u_j(s_i, s_j) = s_i s_j \mathrm{e}^{-s_j}$. It can be shown this game is strategyproof. If both i, j act truthfully, we have $u_i(1, 1) = u_j(1, 1) = 1/e$, and each utility is maximized by fixing the other strategy to be 1. However, if both i, j declare their cost units as 1.1, we have $u_i(1.1, 1.1) = u_j(1.1, 1.1) > 0.4 > 1/e$ and thus i, j can together conduct a valid collusion to achieve higher utilities. It is worth researching the criteria that any strategyproof game must satisfy in order to be collusion resistant.

IV. Defense Against Collusion Attacks By Group Strategyproofness

In Section III and Appendix A, we have given the closed-forms of sensing time and payment of each user if we want the crowdsensing incentive mechanism to be strategyproof. In this section, we research the criteria that a strategyproof mechanism can also achieve group strategyproof equilibrium. We leverage a graph model to analyze the strategyproof crowdsensing game. We give the theoretic criteria to judge whether a strategyproof mechanism can be group strategyproof.

According to (13) and (14), we can uniquely determine the payment to each user i, if and only if we know the function form of the required sensing time $t_i(s_i)$. $t_i(s_i)$ must be a decreasing a.e. differentiable function such that the integral in (13) exists. For example, $t_i(s_i)$ may be e^{-s_i}, which is parametric dependent (see Definition 4) on none of any other user's strategy. $t_i(s_i)$ can also be parametric dependent on other user's strategy, such as $t_i(s_i) = s_j \mathrm{e}^{-s_i}, (j \neq i)$. We leverage a directed graph to represent the parametric dependencies among the user strategies. That is Parametric Dependency Graph.

Definition 3. ***(Parametric Dependency Graph)*** *For any strategyproof crowdsensing game, denote by $G = (U, E)$ its parametric dependency graph, in which U is the set of the participating users, and for any edge $(i, j) \in E$, user i's utility function u_i is parametric dependent on s_j.*

See Figure 1 for an example of parametric dependency graph, which has a circle of users 1, 2, 3. In fact, the strategyproof equilibrium of Figure 1 cannot be group strategyproof, since users 1, 2, 3 can form a collusion group to change their strategies simultaneously, in order to increase their utilities without decreasing any utility of theirs. Lemma 1 reveals that such valid collusion always exists by slightly perturbing the truthful strategies.

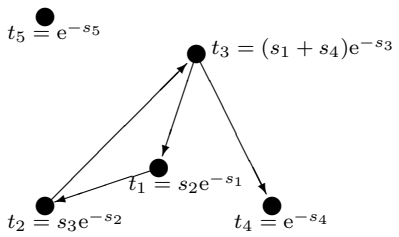

Fig. 1. An example of parametric dependency graph. Here user 1 points to user 2, indicating user 1's utility u_1 is parametric dependent on user 2's strategy s_2, i.e. u_1 takes s_2 as input.

Lemma 1. For any strategyproof crowdsensing game with user utility equation (5), if its parametric dependency graph is cyclic, this game is not group strategyproof.

Proof: If the PDG (parametric dependency graph) $G = (U, E)$ has circles, there always exists a smallest circle $C \subseteq U$. That is, there is no smaller circle among the users in C. Let $C = \{i_1, i_2, \cdots, i_k\}$ and the edges in the circle are $i_1 \to i_2 \to \cdots \to i_k \to i_1$. For any edge $i \to j$ in circle C, we have $\frac{\partial u_i}{\partial s_i}(\kappa_i, \kappa_j) = 0$ and $\frac{\partial u_i}{\partial s_j}(\kappa_i, \kappa_j) \neq 0$ hold for all the possible cost units κ_i of user i. Thus, according to Lemma 4 in the Appendix, there exists $\delta > 0$ such that for any point (s_i, s_j) that satisfies $0 < |s_i - \kappa_i| < \delta$ and exactly one of $s_j - \kappa_j = |s_i - \kappa_i|$ and $\kappa_j - s_j = |s_i - \kappa_i|$[2], we have $u_i(s_i, s_j) > u_i(\kappa_i, \kappa_j)$. We enumerate all the δs of the users $i_1, i_2, \cdots, i_k$ in C as $\delta_1, \delta_2, \cdots, \delta_k$ and let $\underline{\delta} = \frac{1}{2}\min\{\delta_1, \delta_2, \cdots, \delta_k\}$. For $i_1 \to i_2$, if $\frac{\partial u_{i_1}}{\partial s_{i_2}}(\kappa_{i_1}, \kappa_{i_2}) > 0$, $s_{i_1} = \kappa_{i_1} \pm \underline{\delta}$ and $s_{i_2} = \kappa_{i_2} + \underline{\delta}$ suffice $u_{i_1}(s_{i_1}, s_{i_2}) > u_{i_1}(\kappa_{i_1}, \kappa_{i_2})$; if $\frac{\partial u_{i_1}}{\partial s_{i_2}}(\kappa_{i_1}, \kappa_{i_2}) < 0$, $s_{i_1} = \kappa_{i_1} \pm \underline{\delta}$ and $s_{i_2} = \kappa_{i_2} - \underline{\delta}$ suffice $u_{i_1}(s_{i_1}, s_{i_2}) > u_{i_1}(\kappa_{i_1}, \kappa_{i_2})$. Similarly, we can determine the strategies (s_{i_2}, s_{i_3}) to make $u_{i_2}(s_{i_2}, s_{i_3}) > u_{i_2}(\kappa_{i_2}, \kappa_{i_3})$, and (s_{i_3}, s_{i_4}) to make $u_{i_3}(s_{i_3}, s_{i_4}) > u_{i_3}(\kappa_{i_3}, \kappa_{i_4})$ and so on. For $i_k \to i_1$, the similar method can determine whether $s_{i_1} = \kappa_{i_1} + \underline{\delta}$ or $s_{i_1} = \kappa_{i_1} - \underline{\delta}$ to guarantee $u_{i_k}(s_{i_k}, s_{i_1}) > u_{i_k}(\kappa_{i_k}, \kappa_{i_1})$. Thus, if the strategies of the users in C change from the truthful ones to $s_{i_1}, s_{i_2}, \cdots, s_{i_k}$, no utility of any user in C will decrease, and at least one utility among them will strictly increase. Thus, this strategyproof game is not group strategyproof, and the collusion group can be the users in C. ■

On the other hand, if the PDG is acyclic, we can guarantee that the strategyproof game must be group strategyproof, and thus it is impossible to conduct any valid collusion for the game. That is Lemma 2.

Lemma 2. For any strategyproof crowdsensing game with

[2]Here the choice depends on the sign of $\frac{\partial u_i}{\partial s_j}(\kappa_i, \kappa_j)$.

user utility equation (5), if its parametric dependency graph is acyclic, the equilibrium of this game is group strategyproof.

Proof: Let the acyclic PDG be $G = (U, E)$. For any subset $C \subseteq U$, there must exist one user $i \in C$ who has no outgoing edge (Section 3.3 in [30]). Since the utility of user i is not parametric dependent on any other user, there is no strategy other than κ_i that can achieve more utility than $u_i(\kappa_i)$, whatever other users strategize. Our requirement, $\frac{\partial t_i}{\partial s_i}(s_i) < 0$ a.e. for $s_i \in \mathbb{R}^+$, suffices that if $s_i \neq \kappa_i$, we have $u_i(s_i) < u_i(\kappa_i)$. Thus the utility u_i will decrease if user i colludes. Since such user exists for any subset of U, there is no subset of users that can conduct any valid collusion. Thus this game is group strategyproof. ■

Lemma 1 and Lemma 2 together give our main result, Theorem 1.

Theorem 1. For any strategyproof crowdsensing game with user utility equation (5), its equilibrium is group strategyproof, if and only if its parametric dependency graph is acyclic.

Based on Theorem 1, we need to define some partial order relations among the users in order to defend against collusion. The platform determines the partial orders solely and there is no need to let each user know all the orders. If the PDG is cyclic, even though the game cannot be group strategyproof, it is still possible to defend against collusion to some extent. For example, if the colluded users have no circles in the PDG, it is impossible to collude successfully. Let C be the users in the smallest circle in the PDG. It follows that such strategyproof game can resist any collusion that has less than $|C|$ colluded users.

V. Defense Against Profit Trading By n-Truthfulness

In this section, we talk about how to defend against profit trading collusion in the strategyproof crowdsensing game. We have demonstrated that group strategyproof equilibrium can not guarantee profit trading free in Section II-C. We next research the criteria in which the strategyproof mechanism can resist any profit trading collusion. The result is Theorem 2.

Theorem 2. For any strategyproof crowdsensing incentive mechanism with user utility equation (5), the mechanism can achieve n-Truthfulness, where n denotes the number of all the participating users: $n = |U|$, if and only if its parametric dependency graph $G = (U, E)$ satisfies $E = \emptyset$.

Proof: We first show the necessity by proving the contrapositive. That is, there may exist a couple of users that may collude to trade profit, if there is at least one edge in the PDG of the game. We may assume there exists one edge from user i to j in the PDG, indicating u_i is parametric dependent on s_j. Note that it is impossible to have a path from j to i. Otherwise, the PDG is cyclic and thus there must exist a subset of users that may succeed to collude based on Lemma 4. In fact, the collusion here is valid without profit trading, since cyclic PDG means there is no means to achieve group strategyproof equilibrium in order to resist collusion. Then we have

$$\begin{aligned} &\frac{\partial u_i}{\partial s_i}(\kappa_i, \kappa_j) = 0, \quad \frac{\partial u_i}{\partial s_j}(\kappa_i, \kappa_j) \neq 0; \\ &\frac{\partial u_j}{\partial s_i}(\kappa_i, \kappa_j) = 0, \quad \frac{\partial u_j}{\partial s_j}(\kappa_i, \kappa_j) = 0. \end{aligned} \tag{7}$$

Let u_{ij} be the total utility of users i, j: $u_{ij} = u_i + u_j$. Based on (7), we have

$$\frac{\partial u_{ij}}{\partial s_i}(\kappa_i, \kappa_j) = 0, \quad \frac{\partial u_{ij}}{\partial s_j}(\kappa_i, \kappa_j) \neq 0. \tag{8}$$

According to Lemma 4, there always exist uncountably infinite points $(s_i, s_j) \neq (\kappa_i, \kappa_j)$ such that $u_{ij}(s_j, s_j) > u_{ij}(\kappa_i, \kappa_j)$. Thus, users i, j may collude to trade profit.

We next show the sufficiency. Since there is no edge in the PDG, any user utility is not parametric dependent on any other user's strategy. That is, each user can maximize its utility, as long as it strategizes truthfully. Then for any s, for any subset $S \subseteq U$, we have

$$\sum_{i \in S} u_i(s_S, s_{\overline{S}}) \leq \sum_{i \in S} u_i(\kappa_i, s_{-i}) = \sum_{i \in S} u_i(\kappa_S, s_{\overline{S}}), \tag{9}$$

which shows the mechanism must achieve n-Truthfulness. ■

The result of Theorem 2 is slightly pessimistic, since it requires the PDG must have no edges to achieve n-Truthfulness. But it is not always necessary. The condition of n-Truthfulness (Definition 2) is so strong that it assumes each colluded user may offer full trust to any other conspirator. However, in real-world crowdsensing applications, it is not always that all users share the unconditional trust when colluding. If there is no such complete trust, it is fine to have some edges (no cycles) in the PDG.

VI. Collusion Resistant Discrete Crowdsensing Incentive Mechanism

In this section, we will give our discrete crowdsensing incentive mechanism that can resist collusion attacks, even including profit trading. We will also rigorously show that our mechanism can achieve group strategyproof and n-truthful equilibrium.

A. Discrete Crowdsensing Game

In the general crowdsensing game described in Section III, the participation level of each user may be measured by a positive sensing time. There are also many other popular sensing scenarios in which each user is assigned with discrete sensing tasks. For example in location based sensing [31], [32], if a crowdsourced smartphone expects to sense the interesting locations within its neighborhood, then each interesting location can be regarded as one sensing task. Thus its participation level is measured by discrete multiples.

In discrete crowdsensing model, the whole sensing work consists of a number of tasks (chunk data, aggregated nodes, etc.). Each user i suffers a cost unit κ_i by finishing one sensing task. User i reports its cost unit to the platform as a strategy s_i (may be different from κ_i), and as a response the platform lets user i know its assigned number of tasks n_i and the promised

Algorithm 1 *Computing (n, p) on the Platform*

Input: $[\underline{\kappa}_i, \overline{\kappa}_i]$: the publicly known range of user i's cost unit; P_m: maximum payment to each user; s_i: strategy (claimed cost unit) from user i; R: number of remaining tasks.

Output: $\overline{n}$: number of assigned tasks that each user should finish; $\overline{p}$: payment to each user.

$d \leftarrow \frac{1}{2} \min_{i \in U} \overline{\kappa}_i - \underline{\kappa}_i$, $r \leftarrow R$;
Choose the maximum integer m s.t. $\frac{1}{2} m(m+1)d \leq P_m$;
if $md > \max_{i \in U} \overline{\kappa}_i$ **then**
  $d \leftarrow \max_{i \in U} \overline{\kappa}_i / m$;
end if
for each user i in the FIFO queue **do**
  $n_i \leftarrow \max\left\{m - \lfloor \frac{s_i}{d} \rfloor, 0\right\}$;
  if $n_i < r$ **then**
    $p_i \leftarrow \sum_{j=m+1-n_i}^{m} jd = \frac{1}{2} n_i (2m + 1 - n_i) d$;
  else
    $n_i \leftarrow 0$, $p_i \leftarrow 0$;
  end if
  $r \leftarrow r - n_i$;
end for
Let $\overline{n} := \bigsqcup_{i \in U} \{n_i\}$, $\overline{p} := \bigsqcup_{i \in U} \{p_i\}$;
return $(\overline{n}, \overline{p})$;

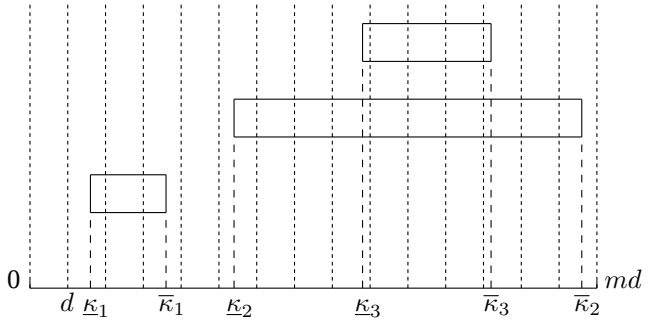

Fig. 2. The idea how to determine the length d. Each rectangular covers the possible range of each user's cost unit. In this example, there are 3 users in total. We choose d as the half of the $\min_{i \in U} \overline{\kappa}_i - \underline{\kappa}_i$ (here $\overline{\kappa}_1 - \underline{\kappa}_1$ is the least) so that no range interval can be completely contained in any partition with length d.

payment p_i once user i finishes the n_i assigned tasks. Hence the utility of user i is $u_i = p_i - \kappa_i n_i$.

We assume for each user i, its cost unit κ_i must fall in some publicly known interval $[\underline{\kappa}_i, \overline{\kappa}_i]$. The lower and upper bounds can be found by the historical distribution of claimed cost units from user i [11].

Note that in many practical applications, the total number of discrete sensing tasks is upper bounded [31], [32], [33]. We also assume the payment sum is upper bounded by the budget B. To upper bound each n_i, we introduce P_m, the maximum payment to any user. Let n be the number of participating users. Then we require $B \geq nP_m$. On the other hand, for each user, the assigned number of sensing tasks cannot exceed the number of remaining tasks to do. Otherwise, the platform will not allow the user to do any task. This is reflected by the threshold R, the number of remaining tasks in Algorithm 1.

B. Our Mechanism

The platform assigns sensing tasks and computes the payment to each user based on the users' strategies. Each user will get the promised payment provided the assigned tasks are finished. Algorithm 1 gives the procedures for the platform to compute the number of assigned tasks and the promised payment for each user. We give our algorithm idea as follows:

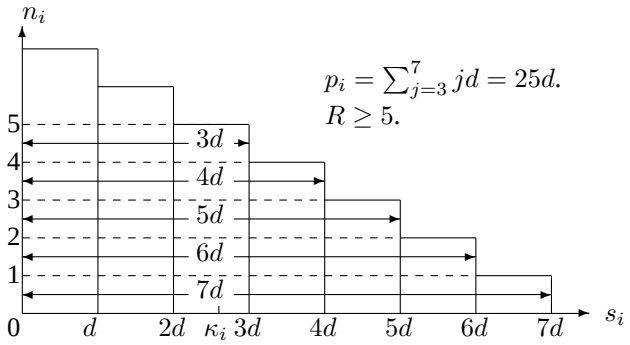

Fig. 3. The computation of the payment p_i. If user i truthfully strategizes $s_i = \kappa_i$, its payment will be $25d$. Here we assume $R \geq 5$. Otherwise, $n_i = p_i = 0$.

(1) Firstly we *partition* the cost unit interval $[0, +\infty)$. Our algorithm chooses the partition length d as half of the minimum interval length of $[\underline{\kappa}_i, \overline{\kappa}_i]$, and chooses the number of partitions allowed by the maximum possible payment P_m, which can be given by the budget B, i.e., $B \geq P_m \cdot n$. The platform will reduce the value of d if md is larger than the maximum possible cost unit. Then our algorithm divides $[0, +\infty)$ into $m + 1$ partitions, with m of them are of length d and the rightmost one is $[md, +\infty)$. By doing this, none of the cost unit range intervals $K_i = [\underline{\kappa}_i, \overline{\kappa}_i]$ can completely reside within any partition here. Figure 2 presents the resulted scenario by our choice of d. Small partition intervals may be beneficial to our truthful mechanism, as we will see later.

(2) Then our algorithm computes n_i and p_i for each user i. We use an FIFO queue to store the applications from the users. That is, user i who submits her application *earlier* that user j can get the feedback (n_i, p_i) *earlier* than j. On the other hand, for each application, the *higher* strategy s_i the user i claims, the *fewer* tasks will be assigned to the user i. Overall, the payment is determined by when user i submits her application, and which interval the strategy s_i falls in. Figure 3 gives the overview of our computation idea, i.e. the payment is the total area of the horizontal slices.

We will give more discussions on the underlying reasons of our algorithm in the following sections.

C. Proofs of Collusion and Profit Trading Resistance

We now show that Algorithm 1 can not only resist collusion attacks (by achieving group strategyproofness), but also eradicate the possibility of any profit trading (by achieving n-truthfulness). Parametric Dependency Graph (PDG) will be used to judge whether the strategyproof equilibrium can be group strategyproof or n-truthful.

Before we introduce our main results, we need Lemma 3, which gives the strategyproof equilibrium for the discrete crowdsensing game.

Lemma 3. The nonnegative integer n_i that maximizes the user utility $u_i = p_i - \kappa_i n_i$ is $\max_{i \in U}\{m - \lfloor \kappa_i / d \rfloor, 0\}$. That

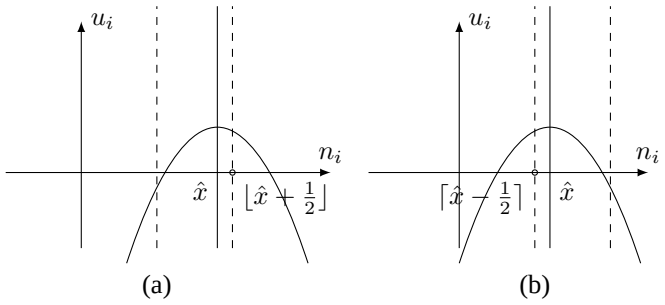

Fig. 4. Two cases of the parabola symmetric axis. The dashed vertical lines denote $x = k \in \mathbb{Z}$, and each solid vertical line is the symmetric axis of the parabola. Each circled point is the integer closest to the axis.

is, if any $n_i = \max_{i \in U}\{m - \lfloor \kappa_i/d \rfloor, 0\}$, then the discrete crowdsensing game achieves strategyproof equilibrium.

Proof: In fact the utility may be real-extended into a concave quadratic function as follows:

$$u_i(n_i) = p_i - \kappa_i n_i = \frac{1}{2} n_i(2m + 1 - n_i)d - \kappa_i n_i.$$

Its function graph is a parabola with the axis of symmetry $\hat{x} = m - \frac{\kappa_i}{d} + \frac{1}{2}$. It is easy to verify that the integer closest to the axis of symmetry can obtain the maximum utility. Figure 4 gives the two possible cases of the axis position. For both cases $m - \lfloor \kappa_i/d \rfloor$ is the integer closest to the symmetric axis. It is possible that the axis satisfies both the cases. Then it has two nearest integers and we can choose either of them. If the closest integer is nonnegative, we get the desired n_i immediately. If the closest integer is negative, then the axis of symmetry must reside to the left of the y axis. Hence the utility function monotonically decreases over $[0, +\infty)$. As a result the nonnegative integer n_i that maximizes u_i must be 0. Both cases give the lemma. ■

Now we can show our main results, Theorem 3.

Theorem 3. For any discrete crowdsensing game with user utility equation $u_i = p_i - \kappa_i n_i$, the incentive mechanism given by Algorithm 1 can achieve group strategyproof and n-truthful equilibrium.

Proof: Clearly the utility of each user i is not parametric dependent on any other user strategy. We may view n_i as restricting the t_i in (5) onto integers. Theorem 1 guarantees that to complete this proof, it suffices to show that the incentive mechanism given by Algorithm 1 can achieve strategyproof equilibrium. For any user i, to maximize its utility $u_i = p_i - \kappa_i n_i$, n_i must be $\max_{i \in U}\{m - \lfloor \kappa_i/d \rfloor, 0\}$ by Lemma 3. In other words, there will be no gain if user i makes its strategy s_i different from the true cost unit κ_i. This fact implies the strategyproof equilibrium. ■

Note that there will be no gain or loss if user i selects any other points within the $(m + 1 - n_i)$-th interval. Thus a small partition interval is beneficial to the truthful mechanism, since each user has less freedom to perturb the truthful strategy. Also our mechanism is budget feasible, since the payment sum is upper bounded by the budget B, i.e., $\sum_{p_i} < P_m \cdot n \leq B$.

VII. Simulations

This section will show our simulations that support the theoretic results we have obtained in the previous sections. Our crowdsensing incentive mechanism can achieve group strategyproof and n-truthful equilibrium simultaneously. To verify all the collusion resistance results, we have designed five simulations. To verify the group strategyproof and n-truthful facts, we consider two variables: the size of the collusion (How many colluded users are there in the network?), and the perturbation size of colluded users' strategies (In collusion, how large the range of each colluded user's strategy can be?). For group strategyproofness, we focus on the average ratio between the number of the colluded users that are willing to quit collusion since the collusion can decrease their own utilities, and the number of all the colluded users. For n-truthfulness, we calculate the average utility loss of each colluded user, since the larger the colluded user's utility loss is, the less possible profit trading can happen. Under both cases above we also pay attention to the variations of the platform utility.

A. Default Parameters and Settings

Before we give the detailed simulation results, let us see some default settings of our simulations. There are 100 participating users. Each user reports a claimed cost unit as its strategy, and the platform returns the required number of sensing tasks and the promised payments based on all the user strategies. We assume the maximum cost unit κ_{max} is 10.0. Any coefficient λ_i for user i is no more than 20.0. To generate the cost units randomly, we always assume they are uniformly distributed within the interval $[0.01, \kappa_{max}]$. In our simulation we assume 0.01 is the least cost unit one user can have. We assume each colluded user randomly strategizes within the perturbation interval, whose radius is the user's perturbation size. Formally, for κ_i, if the perturbation size is θ, then the perturbation interval is $[\max\{\kappa_i - \theta, 0.01\}, \kappa_i + \theta]$. Given the perturbation interval, in order to earn more payment, each colluded user prefers to make its strategy as low as possible, since lower strategy usually means higher participation level and thus more payment from the platform. For example, a colluded user i's cost unit is 1.0, and the perturbation size is 2.0. Then this user i has probability of 25% to claim $s_i = 0.01$, and 75% to claim a strategy larger than 0.01. Defaultly we assume half of the users are colluding. The perturbation size is 5.0. The maximum payment to each user is 100. Each measurement takes 1000 test samples. One may see [34] for the source code of our simulations.

B. Evaluation Results

Based on Figure 5, most colluded users (around 60%) may complain the loss of their user utilities when in collusion. Figure 6 shows the losing ratio increases given more perturbation size. For any perturbation size larger than 2.0, more than half of the colluded users will lose. By a similar reasoning, Figure 7 and Figure 8 show that any collusion may have negative impacts on the user utilities of the colluded users. For different collusion sizes, each colluded user may suffer average

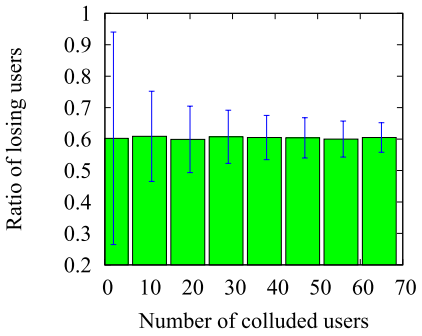

Fig. 5. Ratios of losing colluded users for different collusion sizes.

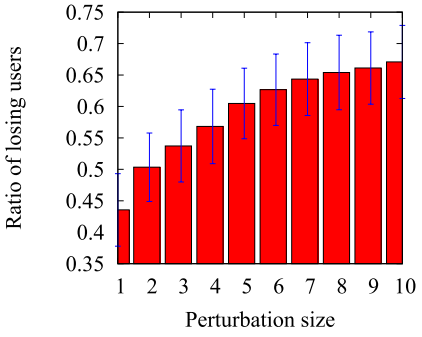

Fig. 6. Ratios of losing colluded users for different strategy perturbation sizes.

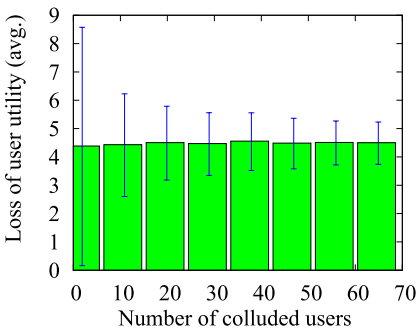

Fig. 7. Average loss on user utility for different collusion sizes.

utility loss of more than 4. The perturbation size increasing by 1.0 will give utility loss of 1.0 to each colluded user averagely. Hence our simulation has verified that our incentive mechanism is group strategyproof and n-truthful.

We also consider the platform utility. Collusion has negative impacts on the platform utility. Figure 9 shows that the platform utility suffers more loss given more colluded users. Figure 10 reveals that the platform utility also suffers larger loss given more deviant perturbation on strategies. The underling reason is that each colluded user prefers to earn more payment by reporting low cost units. For example, if the collusion allows the user to report 0.01 or 2.00 as strategy, this user will prefer to report 0.01 rather than 2.00. As a result, when more users collude, or the perturbation size grows, the payment to each user can have more probability to increase rather than decrease. This will reduce the platform utility. On the other hand, collusion may give more variations on the platform utility. Thus by considering the utility, the platform expects to eliminate collusion among the users.

VIII. Related Works

Incentive mechanism design in crowdsensing applications has attracted a lot of attention in research [1], [28], [2], [9], [35], [11], [10], [12], [13], [36]. Among them the question how to make the participating users strategize truthfully has a lot of interests [28], [2]. For example, Yang et al. [9] proposed a truthful user-centric incentive mechanism for mobile phone crowdsensing. They used the truthful auction rule (Theorem 2.1 in [37]) to implement the computationally efficient mechanism in which each user had no incentive by lying about its cost. But the authors did not consider the scenario that the platform budget could be limited. To make the mechanism budget feasible, Singla et al. [12] proposed another truthful incentive mechanism by making a few assumptions on each user's cost distribution. Similarly, Koutsopoulos [11] gave the closed form of truthful incentive payments for some continuous crowdsensing games by assuming each user's cost value is bounded within an interval. On the other hand, one may consider many other factors when designing incentive mechanisms for crowdsensing applications. Such factors include reputation mechanisms [10], users' locations [13], privacy concerns [36], etc. Unfortunately, many of them could not achieve truthful mechanisms [10], [13].

Collusion attack on collaborative systems is also an important issue for research [16], [15], [17], [38]. Basically researchers give truthful incentive mechanisms by achieving strategyproof equilibria [16]. However, only strategyproofness cannot eliminate all the possible collusions among the users, since it is possible that a group of users know and trust each other completely, and thus they may safely change their strategies simultaneously to make their utilities higher than those given by equilibrium [29], [17]. To give a truthful incentive mechanism which is also resistant to any collusion, one may achieve group strategyproofness [20], [39] and t-truthful [29], [40]. Unfortunately, it is often impossible to find group strategyproofness or t-truthfulness for many cases. For instance, Zhong et al [17] showed that group strategyproofness is impossible for many incentive-compatible routing games on ad hoc networks. Instead they achieved strong Nash equilibrium [41], [39], another standard game theoretic solution, to defend against any collusion. The authors also leveraged some public key cryptographic techniques called restricted verifier signatures to eliminate profit trading. However, strong Nash equilibrium cannot guarantee truthfulness. On the other hand, since crowdsensing games are usually different from routing games in ad hoc networks[4], one cannot directly use routing incentive solutions to solve crowdsensing incentive problems. There are very few literatures on collusion resistance for crowdsensing applications. To the best of our knowledge, we are the first to investigate the possibility of group strategyproofness for crowdsensing incentive mechanisms.

IX. Conclusions

This paper has systematically researched collusion attacks, including profit trading, in crowdsensing incentive mechanism. We have investigated the criteria that a truthful crowdsensing incentive mechanism can achieve group strategyproofness (eliminating collusion) or even n-truthfulness (eradicating profit trading). We have also proposed our incentive mechanism for discrete crowdsensing. Considering all the possible

[4]Usually one has to consider the graph model of network to reflect each participating user's task in routing games. However, we do not have to consider this graph model for most crowdsensing games.

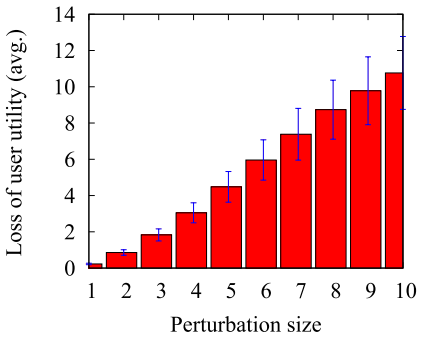

Fig. 8. Average loss on user utility for different strategy perturbation sizes.

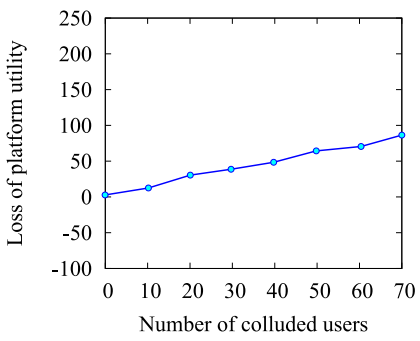

Fig. 9. Loss of platform utility for different collusion sizes.

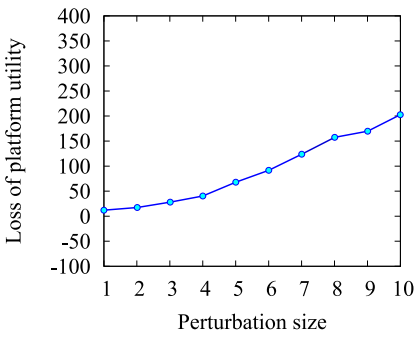

Fig. 10. Loss of platform utility for different strategy perturbation sizes.

combinations of collusion attack, even including profit trading, we have rigorously shown that our discrete mechanism can resist collusion attacks in any form. To verify our results, we have rigorously done the proofs that our solution can achieve the goals as stated above, and we have also given extensive simulations and presented all the supporting experimental results.

REFERENCES

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *Communications Magazine, IEEE*, vol. 49, no. 11, pp. 32–39, 2011.

[2] M. Vukovic, S. Kumara, and O. Greenshpan, "Ubiquitous crowdsourcing," in *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing-Adjunct*. ACM, 2010, pp. 523–526.

[3] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2. IEEE, 2002, pp. 1327–1332.

[4] S. Nawaz, C. Efstratiou, and C. Mascolo, "Parksense: a smartphone based sensing system for on-street parking," in *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 2013, pp. 75–86.

[5] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 293–304.

[6] T. Yan, B. Hoh, D. Ganesan, K. Tracton, T. Iwuchukwu, and J.-S. Lee, "Crowdpark: A crowdsourcing-based parking reservation system for mobile phones," *University of Massachusetts at Amherst Tech. Report*, 2011.

[7] R. Kazman and H.-M. Chen, "The metropolis model a new logic for development of crowdsourced systems," *Communications of the ACM*, vol. 52, no. 7, pp. 76–84, 2009.

[8] J. Howe, "The rise of crowdsourcing," *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006.

[9] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 173–184.

[10] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2140–2148.

[11] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 1402–1410.

[12] A. Singla and A. Krause, "Truthful incentives in crowdsourcing tasks using regret minimization mechanisms," in *Proceedings of the 22nd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2013, pp. 1167–1178.

[13] L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador, "A location-based incentive mechanism for participatory sensing systems with budget constraints," in *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*. IEEE, 2012, pp. 103–108.

[14] H. Moulin and S. Shenker, "Strategyproof sharing of submodular costs: budget balance versus efficiency," *Economic Theory*, vol. 18, no. 3, pp. 511–533, 2001.

[15] C. Chen and Y. Wang, "Sparc: Strategy-proof double auction for mobile participatory sensing," in *Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference on*. IEEE, 2013, pp. 133–140.

[16] Z. Feng, Y. Zhu, and L. M. Ni, "imac: strategy-proof incentive mechanism for mobile crowdsourcing," in *Wireless Algorithms, Systems, and Applications*. Springer, 2013, pp. 337–350.

[17] S. Zhong and F. Wu, "On designing collusion-resistant routing schemes for non-cooperative wireless ad hoc networks," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. ACM, 2007, pp. 278–289.

[18] W. Wang and M. Li, "Low-cost routing in selfish and rational wireless ad hoc networks," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 5, pp. 596–607, 2006.

[19] R. Jurca and B. Faltings, "Collusion-resistant, incentive-compatible feedback payments," in *Proceedings of the 8th ACM conference on Electronic commerce*. ACM, 2007, pp. 200–209.

[20] K. Jain and V. V. Vazirani, "Group strategyproofness and no subsidy via lp-duality," in *Proc. of ASPLOS*. Citeseer, 1999.

[21] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their applications," in *Advances in Cryptology EUROCRYPT96*. Springer, 1996, pp. 143–154.

[22] T. Luo, H.-P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 127–135.

[23] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li, "Free market of crowdsourcing: Incentive mechanism design for mobile sensing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 12, pp. 3190–3200, 2014.

[24] T. Liu and Y. Zhu, "Social welfare maximization in participatory smartphone sensing," *Computer Networks*, vol. 73, pp. 195–209, 2014.

[25] Y. Weiy, Y. Zhuy, H. Zhuy, Q. Zhangz, and G. Xue, "Truthful online double auctions for dynamic mobile crowdsourcing," in *accepted by INFOCOM 2015*. IEEE, 2015.

[26] D. P. Anderson, "Boinc: A system for public-resource computing and storage," in *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*. IEEE, 2004, pp. 4–10.

[27] D. Zhao, X.-Y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 1213–1221.

[28] Y. Singer and M. Mittal, "Pricing mechanisms for crowdsourcing markets," in *Proceedings of the 22nd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2013, pp. 1157–1166.

[29] A. V. Goldberg and J. D. Hartline, "Collusion-resistant mechanisms for single-parameter agents," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2005, pp. 620–629.

[30] S. Dasgupta, C. H. Papadimitriou, and U. Vazirani, *Algorithms*. McGraw-Hill, Inc., 2006.
[31] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, "Location-based crowdsourcing: extending crowdsourcing to the real world," in *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, 2010, pp. 13–22.
[32] X. Liu, K. Liu, L. Guo, X. Li, and Y. Fang, "A game-theoretic approach for achieving k-anonymity in location based services," *Proceedings - IEEE INFOCOM*, vol. 12, no. 11, pp. 2985–2993, 2013.
[33] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 729–737.
[34] *Simulation source code*, 2015. [Online]. Available: https://github.com/anonymous10101/collusion_resist_codes
[35] Y. Zhao and Q. Zhu, "Evaluation on crowdsourcing research: Current status and future direction," *Information Systems Frontiers*, pp. 1–18, 2012.
[36] A. Singla and A. Krause, "Incentives for privacy tradeoff in community sensing," in *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.
[37] Y. Singer, "Budget feasible mechanisms," in *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*. IEEE, 2010, pp. 765–774.
[38] Z. Zheng, F. Wu, and G. Chen, "A strategy-proof combinatorial heterogeneous channel auction framework in noncooperative wireless networks," *IEEE Transactions on Mobile Computing*, 2014.
[39] A. Mas-Collel, M. D. Whinston, and J. Green, "Microeconomic theory," 1995.
[40] A. V. Goldberg and J. D. Hartline, "Competitiveness via consensus," in *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2003, pp. 215–222.
[41] R. J. Aumann, "Acceptable points in general cooperative n-person games," *Contributions to the Theory of Games*, vol. 4, pp. 287–324, 1959.
[42] J. C. Oxtoby, *Measure and Category, Graduate Texts in Mathematics: A Survey of the Analogie Between Topological and Measure Spaces*. Springer, 1971.

APPENDIX

A. Discussions on the Strategyproof Incentive Mechanism

In Section III we have given the general incentive mechanism in which each user's private cost unit is unknown initially. To make each user report its own cost unit honestly, we need to find the forms of payment p_i and sensing time t_i that can achieve strategyproofness. In this section we discuss the closed-forms of payment and required sensing time for a strategyproof incentive mechanism.

We first analyze the required sensing time of each user. The major objective of our incentive mechanism is to make each user strategize truthfully ($s_i = \kappa_i$ always holds) if each user expects to maximize its own utility. This requirement restricts each user's required sensing time $t_i(s_i)$[5] to be non-increasing as the user strategy s_i grows unilaterally. This restriction can be shown by the sum of the following inequalities:

$$\begin{aligned} p_i(\kappa_i) - \kappa_i t_i(\kappa_i) &\geq p_i(\kappa_i') - \kappa_i t_i(\kappa_i') \\ p_i(\kappa_i') - \kappa_i' t_i(\kappa_i') &\geq p_i(\kappa_i) - \kappa_i' t_i(\kappa_i) \end{aligned} \tag{10}$$

where $\kappa_i' > \kappa_i$. The first equation in (10) describes the case in which the cost unit of user i is κ_i, and the second equation is when the cost unit is κ_i'. By summing both the equations in (10), one may verify the restriction $t_i(\kappa_i) \geq t_i(\kappa_i')$. We assume the required sensing time $t_i(s_i)$ is a non-increasing function from $\mathbb{R}^+$ to $\mathbb{R}^+$. Based on Theorem 7.8 in [42], $t_i(s_i)$ is differentiable a.e. (almost everywhere). Thus, we have $\frac{\partial t_i}{\partial s_i} \leq 0$ a.e. holds.

[5]Here $t_i(s_i)$ denotes the required sensing time of user i if its strategy is s_i, while other users' strategies are fixed. This paper always follows this notational convention.

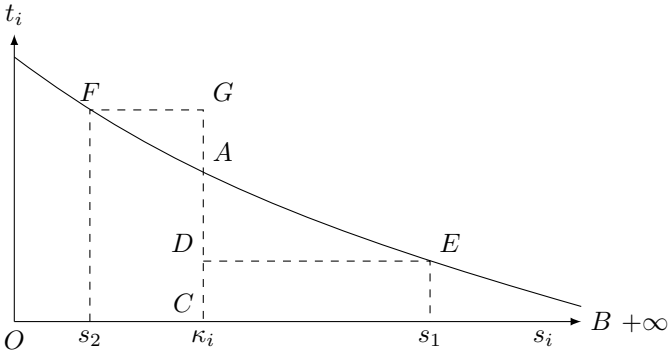

Fig. 11. The relation between user strategy s_i and user utility u_i in the strategyproof crowdsensing incentive mechanism. Here $s_1 > \kappa_i > s_2$. For s_1, we have $u_i(s_1)$ is equal to the area of $DEBC$. For s_2, we have $u_i(s_2)$ is equal to the area of ABC minus the area of AFG. Obviously the area of user utility maximizes at the value $s_i = \kappa_i$.

We next analyze the payment to each user. For the truthfulness, the utility u_i of each user i maximizes at its real cost unit κ_i whatever strategies other users take. Based on (5), we have

$$\frac{\partial u_i}{\partial s_i}(\kappa_i) = \frac{\partial p_i}{\partial s_i}(\kappa_i) - \kappa_i \frac{\partial t_i}{\partial s_i}(\kappa_i) = 0 \tag{11}$$

holds for any $\kappa_i \in \mathbb{R}^+$. Usually, for each user, the more cost unit it will suffer, the more reluctant it will be to participate in crowdsensing. If the user would like to be out of the crowdsensing game, it just declares a very large cost unit. As a result, its utility should be 0 in nature. Thus, it is reasonable to assume both the payment p_i and the required sensing time t_i will finally converge to 0: $\lim_{s_i \to +\infty} p_i(s_i) = 0$, $\lim_{s_i \to +\infty} t_i(s_i) = 0$. Based on this boundary condition and (11), we have[6]

$$\int_{s_i}^{+\infty} \frac{\mathrm{d}u_i}{\mathrm{d}s}(s)\mathrm{d}s = -p_i(s_i) + s_i t_i(s_i) + \int_{s_i}^{+\infty} t_i(s)\mathrm{d}s = 0, \tag{12}$$

which gives the closed-form of $p_i(s_i)$[7]

$$p_i(s_i) = s_i t_i(s_i) + \int_{s_i}^{+\infty} t_i(s)\mathrm{d}s. \tag{13}$$

Thus, the utility of user i is

$$u_i(s_i) = (s_i - \kappa_i) t_i(s_i) + \int_{s_i}^{+\infty} t_i(s)\mathrm{d}s. \tag{14}$$

We discuss the payment p_i and the user utility u_i here in an intuitive way. Figure 11 illustrates the geometric relation between payment and user utility. Clearly user utility maximizes at the value $s_i = \kappa_i$ exactly, and the maximum user utility is the area of ABC.

[6]Here we replace s_i by s to avoid notational confusion.

[7]Here we choose proper $t_i(s)$ to make the integral $\int_{s_i}^{+\infty} t_i(s)\mathrm{d}s$ converge.

We next analyze the user utility obtained by the above truthful mechanism. For each user i's own strategy s_i, we have

$$\frac{\partial u_i}{\partial s_i} = (s_i - \kappa_i)\frac{\partial t_i}{\partial s_i}. \tag{15}$$

It is easy to verify $\frac{\partial u_i}{\partial s_i}(\kappa_i) = 0$. To be truthful, we have $\frac{\partial t_i}{\partial s_i}(s_i) \leq 0$ a.e. holds. Thus $\frac{\partial u_i}{\partial s_i} \leq 0$ a.e. for $s_i > \kappa_i$, and $\frac{\partial u_i}{\partial s_i} \geq 0$ a.e. for $s_i < \kappa_i$. Since u_i can be established by the following Lebesgue integration:

$$u_i(s_i) = u_i(\kappa_i) + \int_{\kappa_i}^{s_i} \frac{\partial u_i}{\partial s_i}(s)\mathrm{d}s. \tag{16}$$

Clearly $u_i(\kappa_i) \geq u_i(s_i)$ for any $s_i \in \mathbb{R}^+$. But the uniqueness of maximum points of u_i needs further discussion. If κ_i is a condensation point[8] of some open interval I s.t. for any $s_i \in I$, $\frac{\partial t_i}{\partial s_i}(s_i) = 0$, then (16) shows that every point in I maximizes u_i. Our discrete crowdsensing model is one instance of this scenario. Otherwise, $s_i = \kappa_i$ is the *unique* solution to maximize u_i, since κ_i is within some open interval E, in which $\frac{\partial t_i}{\partial s_i}(s_i) < 0$ a.e. holds. Suppose for any $s_i > \kappa_i$, we have

$$\begin{aligned} u_i(s_i) &= u_i(\kappa_i) + \left\{\int_{(\kappa_i, s_i)\cap E} + \int_{(\kappa_i, s_i)\setminus E}\right\}\frac{\partial u_i}{\partial s_i}(s)\mathrm{d}s \\ &\leq u_i(\kappa_i) + \int_{(\kappa_i, s_i)\cap E}\frac{\partial u_i}{\partial s_i}(s)\mathrm{d}s < u_i(\kappa_i). \end{aligned} \tag{17}$$

The case for any $s_i < \kappa_i$ is analogous.

For any other user strategy s_j, we have

$$\frac{\partial u_i}{\partial s_j}(\kappa_i, s_j) = \int_{\kappa_i}^{+\infty}\frac{\partial t_i}{\partial s_j}(s, s_j)\mathrm{d}s. \tag{18}$$

We are not sure whether $\frac{\partial u_i}{\partial s_j}(\kappa_i, s_j) = 0$ based on (18). Since we have no knowledge of κ_i when we design the closed-form of $t_i(s_i)$, in order to assume $\frac{\partial u_i}{\partial s_j}(\kappa_i, s_j) \neq 0$, we must guarantee $\int_{s_i}^{+\infty}\frac{\partial t_i}{\partial s_j}(s, s_j)\mathrm{d}s \neq 0$ holds for any $s_i \in \mathbb{R}^+$.[9] Insightfully, this reveals the *dependency* between the function value $u_i(s_i, s_j)$ and the parameter s_j. That is, the strategy s_j contributes significantly to the utility of user i. Thus, we introduce the concept of *parametric dependency*.

Definition 4. ***(Parametric Dependency)*** *The utility function $u_i(s_i, s_j) = p_i(s_i, s_j) - \kappa_i t_i(s_i, s_j)$ is parametric dependent on s_j if and only if $\int_{s_i}^{+\infty}\frac{\partial t_i}{\partial s_j}(s, s_j)\mathrm{d}s \neq 0$ holds for any $s_i \in \mathbb{R}^+$.*

For example, let $t_i(s_i, s_j) = s_j \mathrm{e}^{-s_i}$, and thus $\frac{\partial t_i}{\partial s_j}(s_i, s_j) = \mathrm{e}^{-s_i} > 0$. Then we have $\int_{s_i}^{+\infty}\frac{\partial t_i}{\partial s_j}(s, s_j)\mathrm{d}s > 0$ and thus $u_i(s_i, s_j)$ is parametric dependent on s_j. In fact, when $s_i = \kappa_i$, $\frac{\partial u}{\partial s_j}(\kappa_i, s_j) = \int_{\kappa_i}^{+\infty}\frac{\partial t_i}{\partial s_j}(s, s_j)\mathrm{d}s > 0$. That is, the utility of user i will increase as user j's strategy increases around the maximum points. Similarly, it can derive that $u_i(s_i, s_j)$ is not parametric dependent on any s_k such that $k \neq i, j$. If u_i is not parametric dependent on s_j, s_j has no impact on u_i, and thus user i can maximize its utility by choosing truthful strategy, whatever s_j is. This is because for any different strategies s_j and s_j', we have

$$u_i(s_j) = u_i(s_j') + \int_{s_j'}^{s_j}\frac{\partial u_i}{\partial s_j}(s)\mathrm{d}s = u_i(s_j'), \tag{19}$$

in which $\frac{\partial u_i}{\partial s_j} = 0$ always holds.

B. Local Properties of Bivariate Functions

This paper leverages the local properties of the bivariate functions (i.e., the user utility functions) at the maximum points of the slices. A differentiable bivariate function $u(x, y)$ maps $\mathbb{R}^2$ to $\mathbb{R}$, and we can obtain a slice function $u(x, y_0)$ by fixing $y = y_0$. We assume there exists a point (x_0, y_0) such that $\frac{\partial u}{\partial x}(x_0, y_0) = 0$ and $\frac{\partial^2 u}{\partial x^2}(x_0, y_0) < 0$. That is, the slice function $u(x, y_0)$ is *concave* over the neighborhood of (x_0, y_0). However, for the other variable y we assume $\frac{\partial u}{\partial y}(x_0, y_0) \neq 0$. That is, the slice function $u(x_0, y)$ is either increasing or decreasing over the neighborhood of (x_0, y_0). Then there is a boundary distinguishing the areas $\{(x, y)|u(x, y) < u(x_0, y_0)\}$ and $\{(x, y)|u(x, y) > u(x_0, y_0)\}$, and each point (x, y) on the boundary satisfies $u(x, y) = u(x_0, y_0)$ (see Figure 12). Over the neighborhood of (x_0, y_0), the boundary curve $\{(x, y)|u(x, y) = u(x_0, y_0)\}$ satisfies

$$\frac{\partial u}{\partial x}(x_0, y_0) + \frac{\partial u}{\partial y}(x_0, y_0)\frac{\mathrm{d}y}{\mathrm{d}x}\bigg|_{y=y_0} = 0. \tag{20}$$

Since $\frac{\partial u}{\partial x}(x_0, y_0) = 0$ and $\frac{\partial u}{\partial y}(x_0, y_0) \neq 0$, we have $\frac{\mathrm{d}y}{\mathrm{d}x} = 0$ at (x_0, y_0). That is, for any positive number $\varepsilon > 0$, we have $\delta > 0$ such that for any $0 < |x - x_0| < \delta$, $|\frac{y - y_0}{x - x_0}| < \varepsilon$ holds. We may assume $\varepsilon = 1$, and then there exists $\delta > 0$ such that $|y - y_0| < |x - x_0|$ for any $0 < |x - x_0| < \delta$. Thus for any point (x, y) satisfies $|y - y_0| = |x - x_0|$ and $0 < |x - x_0| < \delta$, we have $u(x, y) \neq u(x_0, y_0)$. If $\frac{\partial u}{\partial y}(x_0, y_0) > 0$, for any point (x, y) satisfies $y - y_0 = |x - x_0|, 0 < |x - x_0| < \delta$, we have $u(x, y) > u(x_0, y_0)$. Similarly, if $\frac{\partial u}{\partial y}(x_0, y_0) < 0$, for any point (x, y) satisfies $y_0 - y = |x - x_0|, 0 < |x - x_0| < \delta$, we have $u(x, y) > u(x_0, y_0)$. That is, we can always find uncountably infinite points in the neighborhood of (x_0, y_0) to increase the function value $u(x, y)$ and make it larger than $u(x_0, y_0)$. We summarize the above discussions as Lemma 4.

Lemma 4. For any bivariate function $u(x, y)$ such that $\frac{\partial u}{\partial x}(x_0, y_0) = 0$ at point (x_0, y_0),

1) if $\frac{\partial u}{\partial y}(x_0, y_0) > 0$, there exists $\delta > 0$ such that for any point (x, y) satisfies $y - y_0 = |x - x_0|, 0 < |x - x_0| < \delta$, we have $u(x, y) > u(x_0, y_0)$;
2) if $\frac{\partial u}{\partial y}(x_0, y_0) < 0$, there exists $\delta > 0$ such that for any point (x, y) satisfies $y_0 - y = |x - x_0|, 0 < |x - x_0| < \delta$, we have $u(x, y) > u(x_0, y_0)$.

[8]Every open interval (neighborhood) including the condensation point must have *uncountably infinite* points in the given set. Assuming Continuum Hypothesis, the intersection of every open neighborhood and the given set must contain some open interval.

[9]Similarly, we must guarantee $\int_{s_i}^{+\infty}\frac{\partial t_i}{\partial s_j}(s, s_j)\mathrm{d}s = 0$ holds for any $s_i \in \mathbb{R}^+$ to ensure $\frac{\partial u_i}{\partial s_j}(\kappa_i, s_j) = 0$, which is the opposite case of Definition 4. That is, we avoid the case $\int_{s_i}^{+\infty}\frac{\partial t_i}{\partial s_j}(s, s_j)\mathrm{d}s = 0$ holds partially over $\mathbb{R}^+$ when designing the function t_i for this paper.

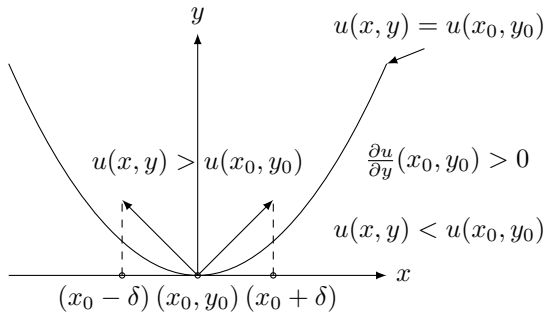

Fig. 12. Local scenario of bivariate function at maximum point of its slice.