Sybil-Proof Online Incentive Mechanisms for Crowdsensing

Jian Lin, Ming Li, Dejun Yang, Guoliang Xue

Abstract—Crowdsensing leverages the rapid growth of sensor-embedded smartphones and human mobility for pervasive information collection. To incentivize smartphone users to participate in crowdsensing, many auction-based incentive mechanisms have been proposed for both offline and online scenarios. It has been demonstrated that the Sybil attack may undermine these mechanisms. In a Sybil attack, a user illicitly pretends multiple identities to gain benefits. Sybil-proof incentive mechanisms have been proposed for the offline scenario. However, the problem of designing Sybil-proof online incentive mechanisms for crowdsensing is still open. Compared to the offline scenario, the online scenario provides users one more dimension of flexibility, i.e., active time, to conduct Sybil attacks, which makes this problem more challenging. In this paper, we design Sybil-proof online incentive mechanisms to deter the Sybil attack for crowdsensing. Depending on users’ flexibility on performing their tasks, we investigate both single-minded and multi-minded cases and propose SOS and SOM, respectively. SOS achieves computational efficiency, individual rationality, truthfulness, and Sybil-proofness. SOM achieves individual rationality, truthfulness, and Sybil-proofness. Through extensive simulations, we evaluate the performance of SOS and SOM.

I. INTRODUCTION

In recent years, the rapid proliferation of smartphones with rich embedded sensors has attracted great attentions from both academy and industry [7]. The effectiveness of crowdsensing to collect data enabled numerous crowdsensing applications in a wide variety of domains, such as transportation [28], marketing [6], environmental monitoring [22], cellular coverage map [19], and etc.

The number of participating users is a critical factor for the success of a crowdsensing application. Most of the early systems [18, 22] assume that the smartphone users contribute to the platform voluntarily. In practice, however, smartphone users taking part in crowdsensing cause extra cost while performing the sensing tasks, e.g., sensing time, battery expenditure, transmission expense and potential privacy threats from the exposure of their locations. Therefore, it is necessary to design incentive mechanisms to stimulate users to participate in crowdsensing.

At present, a number of auction-based incentive mechanisms have been proposed for crowdsensing. These incentive mechanisms model the interaction between the crowdsensing platform and the smartphone users as a reverse auction. The buyer is the platform, and the sellers are smartphone users who bid to perform sensing tasks. In these mechanisms, the platform and the smartphone users as a reverse auction. The platform selects users according to their submitted bids. Most of existing incentive mechanisms (e.g., [10, 15, 31, 32, 36]) focus on offline scenario in which smartphone users are required to submit their bids at the beginning of the auction, and the platform selects a subset of users according to some criteria for different objectives, e.g., maximize social welfare and platform utility. Some works (e.g., [4, 5, 8, 9, 27, 37, 39, 40, 40]) consider a more practical yet dynamic online scenario in which smartphone users participate in the system in a random order, as shown in Fig. 1. Once a user arrives, the platform has to make irrevocable decisions on whether to select it and how much it should be paid without knowing future information. However, none of above mechanisms take into consideration the Sybil attack [2], also known as false-name attack [34]. In recent years, the potential threat from Sybil attack has been investigated in various areas, such as cloud resource allocation [26], social networks [23, 24], and crowdsourced mobile apps [25]. These works focus on designing detection methods to eliminate Sybil attackers. The impact of Sybil attack in auctions has been analyzed in [33, 34].

In both offline and online crowdsensing systems, a user may submit multiple bids under different fictitious identities in the hope to increase its utility. This attack is easy to conduct (e.g., creating multiple accounts) but difficult to detect. The vulnerability to Sybil attack may make a mechanism fail to achieve its desired properties, and the fairness of the system will be jeopardized, since the increase of an attacker’s utility may decrease other users’ utility.

Lin et al. [16] are the first to investigate the Sybil attack in crowdsensing. They proposed two Sybil-proof incentive mechanisms in the offline scenario. However, their mechanisms cannot be directly applied to the online scenario, since...
the online scenario provides users one more dimension of flexibility to conduct Sybil attacks, i.e., active time window (to be elaborated latter). Existing online incentive mechanisms for crowdsensing [4, 5, 9, 27, 37–39] are all vulnerable to Sybil attack. Among them, the VCG-based incentive mechanism [5] is not Sybil-proof, since the VCG auction has been proved not Sybil-proof in [34]. The mechanism proposed in [4] is not Sybil-proof, since a user can increase its critical value and thus increase its payment by submitting multiple bids. For mechanisms in [9, 30, 37, 38], a user can increase its utility by changing from a loser to a winner via Sybil attack. We will use examples to demonstrate the vulnerability of existing online mechanisms to Sybil attack in Section IV. Therefore, the problem of designing Sybil-proof online incentive mechanisms for crowdsensing remains open.

In this paper, we focus on designing Sybil-proof online incentive mechanisms for crowdsensing. A mechanism is Sybil-proof if, participating in crowdsensing using a single identity is a dominant strategy of each user. The main contributions of this paper are as follows:

- To the best of our knowledge, we are the first to investigate Sybil attack in online incentive mechanisms for crowdsensing.
- We analyze existing online incentive mechanisms and demonstrate that they are all vulnerable to Sybil attack.
- Depending on users’ flexibility on performing their tasks, we consider both the single-minded and multi-minded cases. We design SOS and SOM for these two cases, respectively. In order to design SOS, we provide a sufficient condition for an online mechanism to be Sybil-proof. We prove that SOS achieves computational efficiency, individual rationality, truthfulness, and Sybil-proofness, and that SOM achieves individual rationality, truthfulness, and Sybil-proofness.

The remainder of this paper is organized as follows. In Section II, we review the related work. In Section III, we introduce the system model and the objectives. In Section IV, we analyze the vulnerability of existing online mechanisms to Sybil attack. In Section V and Section VI, we present two online mechanisms for single-minded case and multi-minded case and prove their desired properties, respectively. Performance evaluations are presented in Section VII. We conclude this paper in Section VIII.

II. RELATED WORK

In recent years, a number of auction-based incentive mechanisms have been proposed for crowdsensing. Most of them are offline mechanisms with different objectives e.g., maximizing the utility of the platform under a budget constraint [35], minimizing the social cost [3, 30], and preserving users’ privacy [12, 14, 17]. The quality of sensing data has been considered in [11, 13, 20, 29]. Lin et al. [16] formalized the Sybil attack in crowdsensing and demonstrated that previous offline mechanisms are all vulnerable to Sybil attack. Two Sybil-proof offline incentive mechanisms are prosed in this paper, i.e., SPIM-S and SPIM-M. However, these two mechanisms cannot be directly applied to the online scenario.

Several works considered the online scenario where smartphone users come to the system in a random order. Among them, some online pricing mechanisms have been analyzed in [8, 40]. In addition, a number of auction-based online incentive mechanisms have been proposed for crowdsensing. Most of them aim to design online mechanisms, which have comparable performance to offline mechanisms. To get the information about upcoming users, two-sage based mechanisms have been proposed in [9, 37]. However, these mechanisms cannot guarantee the consumer sovereignty, since the first batch of users are rejected no matter how they bid. Zhao et al. [38, 39] proposed two multi-stage mechanisms, which satisfy consumer sovereignty. Furthermore, Gao et al. [5] proposed a VCG-based mechanism to incentivize users to participate in the system for a long-term. Feng et al. [4] considered a system with dynamic users and dynamic tasks. The users’ privacy has been considered in [27]. However, none of existing online mechanisms take into consideration the Sybil attack.

Recently, the impact of Sybil attack has been widely analyzed in areas including virtual machine instance allocation [26], social networks [23] and crowdsourced mobile apps [25]. As pioneers, Yokoo et al. [34] analyzed the effects of Sybil attack on combinatorial auctions. They proved that VCG auction is not Sybil-proof in this paper. In addition, the price-oriented rationing-free protocols, which characterize the Sybil-proof protocols for combinatorial auction have been proposed in [33].

The problem of designing Sybil-proof online incentive mechanisms for crowdsensing is still open. All existing online incentive mechanisms are vulnerable to Sybil attack as explained in Section I, and we will show their vulnerabilities to Sybil attack in Section IV.

III. MODEL AND PROBLEM FORMULATION

A. System Model

In this paper, we consider a crowdsensing system consisting of a platform and a crowd of smartphone users $U = \{1, 2, \ldots, n\}$, where $n$ is unknown. The platform first publicizes a set $T = \{\tau_1, \tau_2, \ldots, \tau_m\}$ of $m$ sensing tasks, aiming at finding some users to complete these tasks before a specified deadline $T$, which is divided into slots of equal size. Each task $\tau_i \in T$ has a value $v_i$ to the platform. We use bundle to refer to any subset of $T$. There is a function $V(B)$ to calculate the value of bundle $B$ to the platform, i.e., $V(B) = \sum_{T \subseteq B} v_i$, $B \subseteq T$. Each user $i$ has a active time window within which it promises to complete the tasks if it is assigned, and a task set $\Gamma_i \subseteq T$, which $i$ can complete within its active time window. Let $\tilde{a}_i \in \{1, \ldots, T\}$ denote the begin of active time window and $\tilde{d}_i \in \{1, \ldots, T\}$, $\tilde{d}_i \geq \tilde{a}_i$ denote the end of active time window. Note that the platform has to make decision to each user $i$ by $\tilde{d}_i$. As with [16], we assume that each user $i$ has a cost function $c_i(B)$, which determines the cost for $i$ to perform all tasks in bundle $B$. The cost function $\hat{c}_i(\cdot)$ satisfies the following properties:
\[ c_i(\emptyset) = 0; \]
\[ c_i(\{t_j\}) = \infty, \forall t_j \in T \setminus \hat{\Gamma}_i; \]
\[ c_i(B') \leq c_i(B'), \forall B' \subseteq B'' \subseteq T; \]
\[ c_i(B) \leq c_i(B') + c_i(B'') \quad \forall B', B'' \subseteq T \text{ and } B = B' \cup B''. \]

The first two properties depict user's capability. The third property implies that a user may incur more cost by performing more tasks. The last property means that a user's cost of performing a set of tasks is not greater than that of performing these tasks separately. These four properties together closely characterize a user's cost when participating in crowdsensing.

Depending on users' flexibility on performing their task sets, we consider two cases in this paper: single-minded (SM) and multi-minded (MM). For the SM case, each user \( i \in U \) is willing to perform only \( \Gamma_i \) and behaves in a "win all or nothing" manner. For the MM case, each user \( i \in U \) is willing to perform any subset of \( \Gamma_i \) and behaves in a flexible manner.

We use the sealed-bid reverse auction to model the interaction between the platform and the smartphone users. In our model, the buyer is the platform buying sensing services, and the sellers are smartphone users bidding for performing tasks. User \( i \) is a winner if it is assigned tasks, and a loser otherwise. Let \( \beta_i = (a_i, d_i, \Gamma_i, b_i) \) denote the bid of user \( i \). Similar to most online crowdsensing systems \([4, 39]\), we assume that a user cannot announce an earlier arrival or a later departure, i.e., \( \hat{a}_i \leq a_i \leq \hat{d}_i \leq d_i \). A user is active at time slot \( t \) if \( a_i \leq t \leq d_i \). Note that, \( b_i \) is a value in the SM case, while \( b_i \) is a cost function in the MM case. We call user \( i \)'s bid \( \beta_i \) is true if \( \beta_i = (\hat{a}_i, \hat{d}_i, \hat{\Gamma}_i, \hat{c}(\hat{\Gamma}_i)) \) in the SM case; \( \beta_i = (a_i, d_i, \Gamma_i, c(\cdot)) \) in the MM case. At each time slot \( t \), each newly arriving user \( i \), i.e., \( a_i = t \) submits its bid to the platform, which is not necessarily to be true. Given the bids of all active users at any time slot \( t \), the platform will assign each active user \( i \) a bundle \( A_i \subseteq \hat{\Gamma}_i \) to complete. Note that \( A_i = \emptyset \) means user \( i \) is not assigned any task to perform at time \( t \). In addition, the platform calculates the payment \( p_i^t \) to user \( i \) for time slot \( t \). Let \( A^t \) and \( p^t \) denote the assignment profile and the payment profile of all active users at time slot \( t \), respectively. Besides, let \( A_i = \bigcup_{i \in [a_i, d_i]} A_i^t \) and \( p_i = \sum_{t=a_i}^{d_i} p_i^t \) denote the overall assignment and overall payment to user \( i \), respectively. At last, let \( A = (A^1, \ldots, A^T) \) denote the overall assignment profile, and \( p = (p^1, \ldots, p^T) \) denote the overall payment profile. The platform pays users once it receives the results of assigned tasks. Note that \( p_i = 0 \), if \( A_i = \emptyset \) or user \( i \) fails to perform assigned tasks. The utility of \( i \) in SM case is
\[ \hat{u}_i = \begin{cases} 
 p_i - \hat{c}_i(A_i), & \text{if } A_i = \hat{\Gamma}_i; \\
 0, & \text{otherwise.} 
\end{cases} \quad (1) \]

The utility of \( i \) in MM case is
\[ \hat{u}_i = \begin{cases} 
 p_i - \hat{c}_i(A_i), & \text{if } A_i \subseteq \hat{\Gamma}_i; \\
 0, & \text{otherwise.} 
\end{cases} \quad (2) \]

The utility of the platform is
\[ u_0 = V(\bigcup_{i \in U} A_i) - \sum_{i \in U} p_i. \quad (3) \]

### B. Attack Model

In this paper, we assume that all users are selfish but rational. Hence it is possible that user \( i \) submits a false bid to maximize its utility. Specifically, a false bid may have a false active time window, i.e., \( a_i \neq \hat{a}_i \) or \( d_i \neq \hat{d}_i \). In addition, \( b_i \) may not be true, i.e., \( b_i \neq \hat{c}_i(\hat{\Gamma}_i) \) in the SM case; \( b_i \neq \hat{c}_i(\cdot) \) in the MM case. Furthermore, user \( i \) could also misreport its task set i.e., \( \Gamma_i \neq \hat{\Gamma}_i \).

We also assume that an attacker could conduct Sybil attack at any time slot in its active time by submitting multiple bids under fictitious identities. As a simple case, attacker \( i \) could submit two bids under two identities \( i' \) and \( i'' \), respectively. Note that \( i \) could submit these two bids simultaneously or at different time slots in its active time window. This case is sufficient to represent the general Sybil attack. We extend the definition of attacker's utility in the following two cases.

#### Single-Minded Case: Each attacker \( i \) is only willing to perform \( \hat{\Gamma}_i \). Attacker \( i \) submits \( \beta_i = (a_i, d_i, \Gamma_i, b_i) \) and \( \beta_{i'} = (a_i, d_i, \Gamma_i, b_i) \) within \([\hat{a}_i, \hat{d}_i] \) using identities \( i' \) and \( i'' \), respectively, where \( \Gamma_i \cup \Gamma_{i'} = \hat{\Gamma}_i \). Attacker \( i \)'s utility is
\[ u_i = \begin{cases} 
 p_i - p_{i'} + \hat{c}_i(\hat{\Gamma}_i), & \text{if } A_i \cup A_{i'} = \hat{\Gamma}_i; \\
 0, & \text{otherwise.} 
\end{cases} \quad (4) \]

#### Multi-Minded Case: Each attacker \( i \) is willing to perform any subset of its task set \( \Gamma_i \). Attacker \( i \) submits \( \beta_i = (a_i, d_i, \Gamma_i, b_i) \) and \( \beta_{i'} = (a_i, d_i, \Gamma_i, b_i) \) within \([\hat{a}_i, \hat{d}_i] \) using identities \( i' \) and \( i'' \), respectively, where \( \Gamma_i \subseteq \hat{\Gamma}_i \) and \( \Gamma_{i'} \subseteq \hat{\Gamma}_i \). Attacker \( i \)'s utility is
\[ u_i = \begin{cases} 
 p_i - p_{i'} + \hat{c}_i(A_i \cup A_{i'}), & \text{if } A_i \cup A_{i'} \subseteq \hat{\Gamma}_i; \\
 0, & \text{otherwise.} 
\end{cases} \quad (5) \]

It is obvious that attacker \( i \) has an incentive to conduct Sybil attack if \( u_i > \hat{u}_i \) in either case. We will use examples to show the vulnerability of existing online incentive mechanisms to Sybil attack in Section IV. Note that any user can be an attacker, we use attacker and user interchangeably in the rest of this paper.

### C. Desired Properties and Objective

In this paper, we consider the following properties:

- **Computational Efficiency:** A mechanism is computationally efficient if it terminates in polynomial time.
- **Individual Rationality:** A mechanism is individually rational if each user has a non-negative utility when bidding its true bid.
- **Truthfulness:** A mechanism is truthful if any user’s utility is maximized when bidding its true bid including both true active time window and true cost.
- **Sybil-Proofness:** A mechanism is Sybil-proof if any user’s utility is maximized when bidding its bid using a single identity.

The objective of this paper is to design Sybil-proof online incentive mechanisms satisfying above properties. The main notations are summarized in Table I.
IV. VULNERABILITY OF EXISTING ONLINE MECHANISMS TO SYBIL ATTACK

In this section, we use examples to show existing online incentive mechanisms are vulnerable to Sybil attack.

A. Mechanism Classification

We classify existing online incentive mechanisms into three categories according to their vulnerabilities to Sybil attack. The first category is the VCG-based mechanism [5]. The second category is the critical value-based mechanism in [4] where each winner is paid its critical value. The third category consists of threshold-based mechanisms [9, 27, 37–39]. The mechanism in the first category is not Sybil-proof since VCG auction is proved not Sybil-proof in [34]. Next, we analyze the vulnerabilities to Sybil attack for the last two categories.

B. Vulnerabilities of Critical Value-based Mechanisms

The mechanism in [4] executes a reverse auction round by round. In one round of auction, the mechanism sorts all active users in a non-decreasing order by their bids. The first \( m_t \) users will be selected as winners, where \( m_t \) is the number of tasks at time slot \( t \). At last, the payment to each winner \( i \) is set to its critical value. Let \( t_i \) denote the time slot \( i \) wins. Winner \( i \)'s critical value is the largest of all the \( m_t \)-th users' bids at each time slot \( t \in [t_i, d_i] \). It is obvious that at any time slot \( t \in [t_i, d_i] \) in which the \( m_t \)-th user does not exist, \( i \) can submit a higher bid using a fictitious identity to take the \( m_t \)-th place. Therefore, \( i \) can increase its critical value and thus increase its payment via Sybil attack.

C. Vulnerabilities of Threshold-based Mechanisms

Within this category, we further divide the mechanisms into two groups. The first group comprises mechanisms in [9, 37]. These two mechanisms are two-stage mechanisms in which the first arrived \( \frac{n}{2} \) users are rejected and their bids are used as the sample for the next stage, where \( n \) is the number of participating users and \( e \) is the base of the natural logarithm. In the first stage, the largest \( v_i/b_i \) value will be used as a threshold for the user selection in the next stage, where \( v_i \) is user \( i \)'s marginal value and \( b_i \) is its bid. In the second stage, the first user \( i \) whose \( v_i/b_i \) value is no less than the threshold will be selected as a winner. We use the example in TABLE II. to show that these mechanisms are not Sybil-proof. In this example, the value of \( \tau_1 \) is 1 and the value of \( \tau_2 \) is 3. The mechanism will reject the first user (User1) since \( n = 5 \) and \( \left\lceil \frac{5}{2} \right\rceil = 1 \). Next, assume user 1 conducts Sybil attack by submitting two bids \( \beta_1 = \left(1, \{\tau_1\}, 2\right) \) and \( \beta_2 = \left(2, \{\tau_2\}, 2\right) \) under fictitious identities \( 1' \) and \( 1'' \), respectively. In this case, the first two users (User 1' and User 2) are rejected since \( n = 6 \) and \( \left\lceil \frac{6}{2} \right\rceil = 2 \), and the threshold is 0.5. User \( i'' \) is the third arrived user whose \( v_i''/b_i'' = 1.5 > 0.5 \), and thus it is selected as a winner. Therefore, these mechanisms are not Sybil-proof, since a user can increase its utility by changing from a loser to a winner via Sybil attack.

The second group comprises of an improved two-stage mechanism [27] and a multi-stage mechanism called OMG [38, 39]. Given the deadline \( T \) and budget \( B \), these two mechanisms will set the cutoff time of the first stage by \( T = \frac{B}{2^{3/2}} \) and \( B = \frac{B}{2^{3/2}} \) respectively, and allocate the first stage a stage-budget \( \frac{B}{2^{3/2}} \), respectively. We take the OMG as an example and show that it is still not Sybil-proof using the example in TABLE II. In the first stage, OMG selects users iteratively according to users' marginal density, \( v_i/b_i \) where \( v_i \) is user \( i \)'s marginal value, and \( b_i \) is its bid. In each iteration, the user with the largest marginal value will be selected, and if its marginal density is not less than a preset density threshold \( \rho \) and its bid does not exceed the stage-budget, it will be selected as a winner. In this example, density threshold \( \rho = 0.5 \), \( T = 8 \) and \( B = 16 \), and thus the first stage ends at time slot 1 and the stage-budget in the first stage is 2. The value of \( \tau_1 \) is 1 and the value of \( \tau_2 \) is 2. In this case, user 1 will not be selected since its bid is greater than the stage-budget. Next, assume user 1 conducts Sybil attack by submitting two bids \( \beta_1 = \left(1, \{\tau_1\}, 2\right) \) and \( \beta_2 = \left(2, \{\tau_2\}, 2\right) \) under fictitious identities \( 1' \) and \( 1'' \), respectively. In this case, user 1' will win with non-negative utility since OMG is individually rational. Therefore, OMG is not Sybil-proof, since a user can increase its utility by changing from a loser to a winner via Sybil attack.

<table>
<thead>
<tr>
<th>User</th>
<th>Bid ((a_i, d_i, \Gamma_i, b_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((1,2,{\tau_1, \tau_2}, 4))</td>
</tr>
<tr>
<td>2</td>
<td>((2,2,{\tau_1}, 2))</td>
</tr>
<tr>
<td>3</td>
<td>((3,4,{\tau_1}, 3))</td>
</tr>
<tr>
<td>4</td>
<td>((3,4,{\tau_1, \tau_2}, 4))</td>
</tr>
<tr>
<td>5</td>
<td>((3,4,{\tau_1, \tau_2}, 4))</td>
</tr>
</tbody>
</table>

V. SOS: SYBIL-PROOF ONLINE INCENTIVE MECHANISM FOR SINGLE-MINDED CASE

In this section, we design and analyze SOS, a Sybil-proof online incentive mechanism for SM case.
A. Design Rationale

In SM case, a user could maximize its utility by submitting multiple subsets of its task set using multiple identities with different active time window in the hope that all the identities will be selected as winners. In order to design Sybil-proof mechanisms, we provide a sufficient condition for an online mechanism to be Sybil-proof in the following theorem.

**Theorem 1:** An online mechanism is Sybil-proof if it satisfies the following two conditions: If any user i pretends two identities i' and i", and both i' and i" are selected as winners with assignment $A_i'$ and $A_i''$, respectively within $[\tilde{a}_i, \tilde{d}_i]$, then

1) i should be selected as a winner with assignment $A_i = A_i' \cup A_i''$ within $[\tilde{a}_i, \tilde{d}_i]$ while using only one identity;
2) $p_i \geq p_{i'} + p_{i''}$.

*Proof:* Assume user i pretends two identities i' and i", and both i' and i" are winners within its true active time window with assignment $A_i' \cup A_i''$. According to (1), user i would have been a winner within its true active time window with assignment $A_i = A_i' \cup A_i''$. According to (4), user i's utility via Sybil attack is $u_i = p_i - c_i(A_i)$. Accordingly, user i cannot increase its utility via Sybil attack.

The following theorem will be used to guarantee the truthfulness of SOS.

**Theorem 2:** [4] An online mechanism is truthful iff:

- The winner selection rule is monotone: If user i wins the auction by bidding $\beta_i = (a_i, d_i, \Gamma_i, b_i)$, it also wins by bidding $\beta_i' = (a_i, d_i', \Gamma_i, b_i')$, where $a_i' \leq a_i, d_i' \geq d_i, \Gamma_i \subseteq \Gamma_i', b_i' \leq b_i$.
- Each winner is paid the critical value, which is the smallest value such that user i would lose the auction if it bids higher than this value.

In order to guarantee Sybil-proofness and truthfulness, SOS should satisfy both Theorem 1 and Theorem 2.

B. Design of Mechanism

In this section, we describe the details of SOS, which is comprised of two subroutines: winner selection with assignment and payment determination.

The winner selection with assignment is illustrated in Algorithm 1. It selects winners iteratively at each time slot until all tasks are assigned or deadline T is reached. Let $R'^t$ denote the set of currently unassigned tasks and $v'^t_i = V(R'^t \cap \Gamma_i)$ denote the marginal value of user i to the platform at time slot t. At each time slot t, Algorithm 1 selects the user with the largest criterion value, $v'^t_i - b_i$, from all active users in $U'^t$. If its criterion value is non-negative, this user will be put into winner set $W'$ and assigned the tasks it submits. Otherwise, it will not be assigned tasks. All active users’ task assignments constitute the assignment profile $A'^t$ of time slot t. The assignment profile of every time slot constitutes the overall assignment profile $A$. The outcome of Algorithm 1 are $A$ and $W$.

The payment determination is illustrated in Algorithm 2. The input are user i's ID, the time slot $t[i]$, and the set $R'^{t[i]}$ of unassigned tasks at the beginning of time slot $t[i]$. For each time slot in $[t[i], d_i]$, Algorithm 2 calculates the highest price i can bid in order to be a winner. At last, the payment to user i is set to the highest price among these prices. Note that

$$p_i = \arg \max_{t \in [t[i], d_i]} \{v'^t_i - c^t\},$$

where $c^t = \max\{0, v'^t_i - b_i\}$, and $i_j$ is the user with the largest criterion value at time slot t when i is not in $U'^t$.

The main algorithm of SOS is illustrated in Algorithm 3. Note that, there is at most one winner at each time slot according to Algorithm 1. Therefore, SOS iterates all time slots and calculates the payment for the winner at each time slot using Algorithm 2.

**Algorithm 1:** SOS-WSA($T, T$)

1. $W \leftarrow \emptyset$, $A' \leftarrow \emptyset$, $\forall t \in [1, T]$, $t \leftarrow 1$, $R' \leftarrow T$;
2. while $R' \neq \emptyset$ and $t \leq T$ do
3. $U'^t \leftarrow$ the set of active users at time slot t;
4. $A'_t \leftarrow \emptyset$, $\forall j \in U'^t$;
5. $i_j \leftarrow \arg \max_{i \in U'^t}(v'^t_i - b_i)$;
6. if $b_i \leq v'^t_i$ then
7. $W \leftarrow W \cup \{i_j\}$, $A'_t \leftarrow \Gamma_i$, $R'^{t+1} \leftarrow R'^t \setminus \Gamma_i$;
8. $t \leftarrow t + 1$;
9. end
10. end
11. $A \leftarrow A_1, \ldots, A^T$;
12. return $(A, W)$.

**Algorithm 2:** SOS-PD(i, $t[i]$, $R'^{t[i]}$)

1. $t \leftarrow t[i]$, $p'^{t[i]}_i = 0$;
2. while $t \leq d_i$ and $R'^t \neq \emptyset$ do
3. $U'^t \leftarrow$ the set of active users at time slot t;
4. $U'^t \leftarrow U'^t \setminus \{i\}$;
5. $i_j \leftarrow \arg \max_{i \in U'^t}(v'^t_i - b_i)$;
6. if $b_i \leq v'^t_i$ then
7. $p'^{t[i]}_i \leftarrow \max\{p'^{t[i]}_i, v'^t_i - (v'^t_i - b_i)\}$;
8. $R'^{t+1} \leftarrow R'^t \setminus \Gamma_i$;
9. else
10. $p'^{t[i]}_i \leftarrow \max\{p'^{t[i]}_i, v'^t_i\}$;
11. end
12. $t \leftarrow t + 1$;
13. end
14. return $p'^{t[i]}_i$.

**Algorithm 3:** SOS($T, T$)

1. $t \leftarrow 1$, $R' \leftarrow T$;
2. $(A, W) \leftarrow$ SOS-WSA($T, T$);
3. while $R' \neq \emptyset$ and $t \leq T$ do
4. $i \leftarrow j \in W \text{ s.t. } A'_i \neq \emptyset$;
5. $p'^t_i \leftarrow$ SOS-PD(i, t, $R'^t$);
6. $R'^{t+1} \leftarrow R'^t \setminus A'_i$;
7. $t \leftarrow t + 1$;
8. end
9. return $(A, p)$.

C. Analysis of SOS

In this section, we prove the properties of SOS in the following theorem.
Theorem 3: SOS is computationally efficient, individually rational, truthful and Sybil-proof in SM case.

We prove this theorem with the following lemmas.

Lemma 1: SOS is computationally efficient.

Due to space limit, we omit the proof for this lemma.

Lemma 2: SOS is individually rational.

Proof: For any winner $i$, assume it is selected at time slot $t[i]$ with its true bid, i.e., $b_i = c_i$. If there exists a winner $j$ at time slot $t[j]$ when $i$ is not in $U[t[i]]$, we have $v_i[t[i]] - b_i \geq v_j[t[j]] - b_j \geq 0$ since $i$ was the winner at time slot $t[j]$. According to Line 7 in Algorithm 2, we have $p_i[t] \geq v_i[t[i]] - (v_f[t][j] - b_j) \geq b_i$. If there is no winner at time slot $t[i]$ when $i$ is not in $U[t[i]]$, we have $p_i[t[i]] \geq v_i[t[i]] \geq b_i$ according to Line 10. Therefore, $u_i = p_i - c_i = p_i - b_i \geq 0$, and SOS is individually rational.

Lemma 3: SOS is truthful.

Proof: We first prove that user $i$ cannot increase its utility by submitting a false task set. We then prove that user $i$ cannot increase its utility by submitting a false active time window or a false cost. If user $i$ submits a false task set $A_i \subset \Gamma_i$, the utility of $i$ is 0 according to (1). On the contrary, if $\Gamma_i \setminus A_i \neq \emptyset$, user $i$ will not be paid since it cannot finish all the tasks in $\Gamma_i$. Thus, there is no incentive for $i$ to submit a false task set.

To prove that user $i$ cannot increase its utility by submitting a false active time window or a false cost, it suffices to prove that the selection rule of SOS is monotone and the payment to each winner is its critical value according to Theorem 2. Obviously, the criterion value of a user will increase with the decrease of user’s cost. Meanwhile, due to the submodularity of user’s marginal value, the criterion value of a user at each time slot will not decrease if it bids a wider active time window. Therefore, the selection rule of SOS is monotone. Next, we prove that the payment $p_i$ to winner $i$ is its critical value. Assume $i$ was selected at time slot $t[i]$ with $b_i$, and thus $p_i = p_i[t[i]] \geq b_i$. If user $i$ bids $b_i > p_i$ it is obvious that $i$ still loses at any time slot $t \in [t[i], t[i])$ since its criterion value is less than that when $i$ bids $b_i$ but loses in $[t[i], t[i)]$. At any time slot $t \in [t[i], d_i]$ $i$ still loses since there always exists a user $j$ such that $v_i[t[i] - t_i] < b_j$ according to (6). If user $i$ bids $b_i < p_i$, $i$ wins at least within $[t[i], d_i]$ according to (6), if not earlier. Therefore, $p_i$ is the critical value for user $i$.

Lemma 4: SOS is Sybil-proof.

Proof: We prove SOS is Sybil-proof by proving it satisfies the sufficient conditions in Theorem 1. Assume user $i$ submits $(a_i, d_i, \Gamma_i, b_i)$ and $(a_i, d_i, \Gamma_i, b_i)$ using two fictitious identities $i'$ and $i''$, respectively, where $\Gamma_i \cup \Gamma_i' = \Gamma_i$.

We first prove that SOS satisfies the first condition in Theorem 1. We assume that both $i'$ and $i''$ are selected as winner at time $t[i']$ and $t[i'']$ (w.l.o.g. $t[i'] < t[i'']$), respectively. It implies that $\Gamma_i' \subset \Gamma_i$ and $\Gamma_i'' \subset \Gamma_i$, since one will make the other lose otherwise. In addition, we have $v_i[t[i']][t[i']] - b_i \geq v_i[t[i'']][t[i'']] - b_i$ and $v_i[t[i'']][t[i'']] \geq b_i$, since both $i'$ and $i''$ are winners. If user $i$ use a single identity and submits its true bid $(a_i, d_i, \Gamma_i, b_i)$. At time slot $t[i]$, we have $v_i[t[i]] - b_i \geq v_i[t[i]][t[i']] - b_i \geq v_i[t[i]][t[i]] + v_i[t[i]][t[i']] - b_i$. The first inequation lies in the fact that $\Gamma_i' \cup \Gamma_i'' = \Gamma_i$. The second inequation is based on the fourth property of the cost function. Therefore, we have $v_i[t[i]] - b_i \geq v_i[t[i]][t[i']] - b_i \geq v_i[t[i]][t[i']] - b_i$. Since $v_i[t[i]][t[i']] - b_i \geq 0$. This implies that $i$ wins at $t[i']$ at the latest while using a single identity. Therefore, the first condition in Theorem 1 is satisfied.

We next prove that SOS satisfies the second condition in Theorem 1. We know that user $i$ wins at $t[i] \leq t[i']$. Let $t_c$, $t_e$ and $t_f$ denote the time that determine the payment of $i'$ and $i''$ according to (6), respectively. We know that $t_c \in [t[i], d_i]$, $t_e \in [t[i'], d_i]$, and $t_f \in [t[i''], d_i]$ according to Algorithm 2. We then prove by cases. In Case 1, $t_c \in [t[i'], d_i]$ and $t_e < t_f$. According to (6), we have $p_i \geq v_i[t[i]] - c_i \geq v_i[t[i]][t[i']] - c_i \geq v_i[t[i]][t[i']] + v_i[t[i]][t[i']] - c_i \geq p_i + p_i$. The first inequation results from the fact $t_c \in [t[i], d_i]$. The second inequation is based on the fact $\Gamma_i' \cup \Gamma_i'' = \Gamma_i$. The third inequation is based on the fact $p_i \geq v_i[t[i]][t[i']] - c_i$ and $p_i \leq v_i[t[i]][t[i']]$. Hence, the second condition in Theorem 1 is satisfied.

Therefore, SOS is Sybil-proof according to Theorem 1. We can use a similar proof for the case where a user pretends more than two identities.

VI. SOM: SYBIL-PROOF ONLINE INCENTIVE MECHANISM FOR MULTI-MINDED CASE

In this section, we design and analyze SOM, a Sybil-proof online incentive mechanism for MM case.

A. Design Rationale

In MM case, a user is willing to perform any subset of its task set and tries to maximize its utility by submitting multiple bids under fictitious identities. To guarantee that each user submits its true cost function, SOM gives the payment to each user, which is independent of its own cost function. The time-truthfulness is based on the monotonic task assignment rule and the submodularity of users’ marginal value. To achieve Sybil-proofness, we extend the characterization of Sybil-proof mechanisms in [33] to the online scenario.

B. Design of Mechanism

The main algorithm of SOM is shown in Algorithm 4. It selects users iteratively at each time slot until all tasks are assigned or deadline $T$ is reached. Given the sensing tasks $T$ and deadline $T$, $\text{SOM}$ outputs the overall assignment profile $A$ and the overall payment profile $p$.

Let $T_i \subseteq \Gamma_i$ denote a set of unassigned tasks that $i$ can perform at time slot $t$. Let $B_i(t) = \{B \mid B \subseteq \Gamma_i, B_i(t) \cap B = \emptyset\}$. At each time slot $t$, SOM first calculates the payment $p_i^B$ to each active user $i$ for any bundle $B \subseteq T_i$. Note that the payment to user $i$ for any bundle $B \subseteq T_i$ is independent of its cost function $c_i(\cdot)$, i.e., $p_i^B = V(B) - \max_{j \neq i, B \in B_i(t)} \{V(B') - c_j(B')\}$.

(7)
Algorithm 4: SOM(T, T)

1. \( F \leftarrow 0, A^t_i \leftarrow 0, \forall t \in [1, T], t \leftarrow 1; \)
2. while \( F \neq T \) and \( t \leq T \) do
3. \( U^t \leftarrow \) the set of active users at time slot \( t; \)
4. \( \Gamma^t_i \leftarrow \Gamma_i \setminus F, \forall i \in U^t; \)
5. foreach \( i \in U^t \) do
6. Calculate the payment to \( i \) for any bundle \( B \subseteq \Gamma^t_i \)
7. \( p^t_i = \max_{B \subseteq \Gamma^t_i} (p^t_i(B) - c_i(B)); \)
8. \( F \leftarrow F \cup A^t_i; \)
9. end
10. end

At last, SOM will assign each active user \( i \) a set of tasks \( A^t_i \), which is a bundle \( B \subseteq \Gamma^t_i \) maximizing its utility based on the calculated payment, i.e.,
\[
A^t_i = \arg \max_{B \subseteq \Gamma^t_i} (p^t_i(B) - c_i(B)).
\]

The payment \( p^t_i \) to each user \( i \) at time slot \( t \) for assignment \( A^t_i \) is \( p^t_i(A^t_i) \). Note that \( p^t_i = p^t_i(A^t_i) = 0 \), if \( A^t_i = \emptyset \).

C. Analysis of SOM

In this section, we prove the properties of SOM in the following theorem.

Theorem 4: SOM is individually rational, truthful and Sybil-proof in MM case.

We prove this theorem with the following lemmas.

Lemma 5: SOM is individually rational.

Proof: The utility of any active user \( i \) at any time slot \( t \) is 0 when the assignment \( A^t_i = \emptyset \) according to (2). According to (8), at any time slot \( t \), SOM assigns any active user \( i \) a bundle \( A^t_i \) maximizing its utility. It implies that \( A^t_i \neq \emptyset \) only if \( u_i = p^t_i(A^t_i) > 0 \), and thus the utility of any user \( i \) is non-negative. Therefore, SOM is individually rational.

Lemma 6: SOM is truthful.

Proof: We first prove that user \( i \) cannot increase its utility by submitting a false task set. Then, we prove that user \( i \) cannot increase its utility by submitting a false active time window or a false cost function. Assume user \( i \) submits a false bid \( \beta_i = (a_i, d_i, \Gamma_i, c_i(\cdot)) \). By (7), at any time slot \( t \), the payment to \( i \) for any bundle \( B \subseteq \Gamma^t_i \) is calculated independently of \( i \)'s own cost function. If \( \Gamma_i \subseteq \Gamma_i \), the payment to \( i \) for any subset of \( \Gamma_i \) is the same as that when \( i \) submits \( \Gamma_i \). In addition, at each time slot \( t \), SOM assigns \( i \) a bundle maximizing its utility by (8). Therefore, user \( i \) cannot increase its utility by submitting a false task set. On the contrary, if \( \Gamma_i \cap \Gamma_i \neq \emptyset \), it will not be assigned tasks in \( \Gamma_i \cap \Gamma_i \neq \emptyset \), since its utility is negative according to the second property of the cost function. Therefore, user \( i \) has no incentive to submit a false task set.

Next, we prove that user \( i \) has no incentive to submit a false active time window, i.e., \( a_i > \bar{a}_i \) or \( d_i < \bar{d}_i \). By (7), SOM only considers \( \Gamma^t_i \) for each active user \( i \) at any time slot \( t \). Besides, the size of \( \Gamma^t_i \) is non-increasing with time. Thus, a narrow active time window will not increase a user’s chance to be a winner. Therefore, a user has no incentive to submit a false time window.

At last, a false cost function \( c_i(\cdot) \neq c_i(\cdot) \) can only affect the result of (8) according to SOM. Let \( A^t_i \) and \( \hat{A}^t_i \) denote the assignments to \( i \) when \( i \) submits \( c_i(\cdot) \) and the true cost function \( c_i(\cdot) \), respectively.

Lemma 7: SOM is Sybil-proof.

Proof: We assume that user \( i \) pretends two identities \( i' \) and \( i'' \) who are assigned \( A^t_i \) and \( \hat{A}^t_i \), respectively. Let \( u_i(A_i) \) denote the utility of user \( i \) when assigned \( A_i \). For any time slot \( t \in \{a_i, d_i\} \), we have \( u_i(A_i) \geq u_i(A_i) \cup A_i \) since \( A_i \cup A_i \subseteq \Gamma_i \) and SOM assigns \( i \) the bundle that maximizes its utility. Next, we prove that \( u_i(A_i) \cup A_i \) is the utility of any user \( i \). Assume \( A_i \neq \emptyset \), and thus the utility of any user \( i \) is non-negative. Therefore, SOM is individually rational.
\( A'_i \times A'_j = \emptyset \). We also have \( V(A'_i) = V(A'_i \cup A'_j) = V(A'_i) + V(A'_j) \), since \( A'_i = A'_i \cup A'_j \). By (7), the payment to \( i \) is 
\[
p_i(\mathcal{A}) = V(A'_i) - \max\{0, m_i\}
\]
\[
\geq V(A'_i) - (m_{i'} + m_{i''})
\]
\[
= V(A'_i \cup A'_{i''}) - (m_{i'} + m_{i''})
\]
\[
= V(A'_i) - m_{i'} + V(A'_{i''}) - m_{i''}
\]
\[
\geq p_i(\mathcal{A}) + p_{i''}(\mathcal{A})
\]

In addition, we have \( c_i(A'_i) \leq c_i(A'_j) \) because of the fourth property of the cost function. By (2) and (5), the utility of \( i \) when using two identities is not greater than that obtained by using a single identity at any time slot \( t \). User \( i \)’s utility via Sybil attack is not greater than that obtained by using a single identity.

Therefore, SOM is Sybil-proof.

Remark: We can use a proof similar to that in Lemma 7 to prove that \( A'_i \cap A'_j = \emptyset \) for any two users \( i \) and \( j \). In addition, SOM does not satisfy computational efficiency, since at each time slot \( t \) it calculates the payments to each active user \( i \) for every subset of \( \Gamma_i \), and the time complexity is exponential to the largest \( |\Gamma_i| \) for all \( i \in S' \). In reality, however, the number of tasks each user can perform is very small because of various constraints, e.g., travel budget [21], and thus the execution time of SOM is still practical.

VII. PERFORMANCE EVALUATION

In this section, we compare the performances of SOS and SOM with three benchmarks. The first benchmark is an online mechanism adapted from [4] for SM case, denoted by Greedy. Note that, this mechanism is not Sybil-proof. The second benchmark is SPIM-S [16], which is a Sybil-proof mechanism for SM case. The third benchmark is SPIM-M [16], which is a Sybil-proof mechanism for MM case. The performance metrics include total payment, platform utility and Sybil-proofness.

A. Evaluation Setup

For a fair comparison with SPIM-S and SPIM-M, we use the same dataset, which is a real-world dataset consisting of the traces of taxi drivers [1]. As in [16], we consider a crowdsensing system in which the tasks are measuring the Wi-Fi signal strength at specific locations. In this system, tasks are represented by GPS locations of the taxi drivers in the dataset, and users are all the taxi drivers.

In our evaluation, we randomly select locations on taxi drivers’ traces as the sensing tasks. The value of each task is uniformly distributed over \([1,5]\) and users’ cost for each task is uniformly distributed over \([1,5]\). We set the deadline \( (T) \) to 60 (min), each user \( i \)’s \( \hat{a}_i \) is uniformly distributed over \([0,60]\), and the active time window \( (\hat{d}_i - \hat{a}_i) \) of each user is uniformly distributed over \([0,5]\). To evaluate the impact of the number of sensing tasks \( (m) \) on the performance metrics, we fix the number of users \( (n) \) at 200 and vary \( m \) from 20 to 60 with a step of 10. To evaluate the impact of the number of users on the performance metrics, we fix \( m \) at 150 and vary \( n \) from 100 to 300 with a step of 50. All the results are averaged over 1000 independent runs.

B. Evaluation of Total Payment

The impacts of \( m \) and \( n \) on the total payment to users are shown in Fig. 2. In Fig. 2 (a), we see that the total payment of both offline and online mechanisms increase with the increase of \( m \). This is because the platform may recruit more users when \( m \) increase, and thus has a higher payment. In addition, we see that the total payment of the online mechanisms (Greedy, SOS, SOM) are higher than that of the offline mechanisms (SPIM-S, SPIM-M). This is because the online mechanisms may recruit more users when users arrive in different time slots, and thus has a higher payment. In Fig. 2 (b), we observe that the total payment of SPIM-M decrease with the increase of \( n \). This is because, with more users, SPIM-M may find more low-cost users to perform the tasks. The total payment of the other four mechanisms increase slightly with the increase of \( n \). This is because, with more users, these mechanisms may assign more tasks incurring a higher payment. In addition, we see that the total payment of the offline mechanisms are higher than that of the offline mechanisms as explained before. Note that, SOS has a similar performance as Greedy, which is not Sybil-proof.

C. Evaluation of Platform Utility

The impacts of \( m \) and \( n \) on the platform utility are shown in Fig. 3. In both Fig. 3 (a) and Fig. 3 (b), we see that the platform utilities achieved by the offline mechanisms (SPIM-S, SPIM-M) are larger than that achieved by online mechanisms (Greedy, SOS, SOM). This is because offline mechanisms know all users’ bids before making decision, while online mechanisms have no information of future users. In addition, we see that SOM outperforms SOS in term of the platform utility. This is because SOM assigns each task to at most one user, and thus avoids paying users to perform duplicated tasks.

D. Evaluation of Sybil-proofness

Fig. 4 shows the utility of Sybil attacker and the other users without Sybil attack, denoted by Attacker and Other, and that
via Sybil attack, denoted by Attacker-Sybil and Other-Sybil. In both Fig. 4 (a) and Fig. 4 (b), we see that the attacker increase its utility while decreasing the utility of others’ in Greedy. This is because Greedy is vulnerable to Sybil attack. However, the attacker cannot increase its utility in SPIM-S, SOS and SOM, since they are Sybil-proof.

VIII. Conclusion

In this paper, we demonstrated that existing online incentive mechanisms for crowdsensing are all vulnerable to Sybil attack. We proposed two Sybil-proof online incentive mechanisms SOS and SOM for single-minded case and multi-minded case, respectively. Specifically, SOS achieves computational efficiency, individual rationality, truthfulness and Sybil-proveness. SOM achieves individual rationality, truthfulness and Sybil-proveness. We rigorously proved the desired properties of the mechanisms and evaluated their performances through extensive simulations.

REFERENCES