Robust Incentive Tree Design for Mobile Crowdsensing

Xiang Zhang, Guoliang Xue, Ruozhou Yu, Dejun Yang, Jian Tang

***Abstract*—With the proliferation of smart mobile devices (smart phone, tablet, and wearable), mobile crowdsensing becomes a powerful sensing and computation paradigm. It has been put into application in many fields, such as spectrum sensing, environmental monitoring, healthcare, and so on. Driven by promising incentives, the power of the crowd grants crowdsensing an advantage in mobilizing users who perform sensing tasks with the embedded sensors on the smart devices. Auction is one of the commonly adopted crowdsensing incentive mechanisms to incentivize users for participation. However, it does not consider the incentive for user solicitation, where in crowdsensing, such incentive would ease the tension when there is a lack of crowdsensing users. To deal with this issue, we aim to design an auction-based incentive tree to offer rewards to users for both participation and solicitation. Meanwhile, we want the incentive mechanism to be robust against dishonest behavior such as untruthful bidding and sybil attacks, to eliminate malicious price manipulations. We design RIT as a Robust Incentive Tree mechanism for mobile crowdsensing which combines the advantages of auctions and incentive trees. We prove that RIT is truthful and sybil-proof with probability at least H, for any given $H \in (0, 1)$. We also prove that RIT satisfies individual rationality, computational efficiency, and solicitation incentive. Simulation results of RIT further confirm our analysis.**

Key Words: Crowdsensing; Mobile Networks; Wireless Networks; Incentive Mechanism; Sybil-proofness; Truthfulness.

1. INTRODUCTION

With the proliferation of smart mobile devices (e.g., smart phone, tablet, and wearable), the richness of embedded sensors (e.g., accelerometer, compass, gyroscope, GPS), and the increasingly powerful processors, mobile users can perform basic sensing tasks single-handedly. Crowdsensing [10], which falls into the category of mobile crowdsourcing [15], has blossomed into a fertile area for commercial business, where mobile device users are recruited to collectively share their sensing data. Unlike traditional labor markets where jobs are assigned to designated workers, crowdsensing out-sources the jobs to the crowd who can be recruited online or through social networks. Driven by promising incentives, the power of the crowd grants crowdsensing advantages in mobilizing *crowdsensing users* to provide services in various areas, such as spectrum sensing [1], environmental monitoring [19], traffic prediction [42], healthcare [14, 28, 30], and so on.

Zhang, Xue, and Yu are with Arizona State University, Tempe, AZ 85287. Email: {xzhan229, xue, ruozhouy}@asu.edu. Yang is with Colorado School of Mines, Golden, CO 80401. Email: djyang@mines.edu. Tang is with Syracuse University, Syracuse, NY 13244. Email: jtang02@syr.edu.
This research was supported in part by NSF grants 1421685, 1420881, 1444059, 1461886, 1525920, 1717197, and 1717315. The information reported here does not reflect the position or policy of the federal government.

Behind the success of these applications, one of the forces that motivate the users into participation is the incentives paid to the crowdsensing users. When performing crowdsensing jobs, it incurs costs to crowdsensing users, in terms of power consumption, private information leakage, time spent, etc. These costs might prevent the users from further participation. Thus, monetary rewards would be offered to the crowdsensing users as compensations such that despite the costs, users would still devote their efforts to perform the crowdsensing jobs.

A good incentive mechanism, which allocates tasks and computes proper payments, is crucial to the success of a crowdsensing application. Auction is one of the commonly adopted incentive mechanisms for crowdsensing [9, 17, 20, 23, 35, 40]. There are several desired properties for an auction: truthfulness, individual rationality, and computational efficiency. These properties aim to make the mechanism robust, fair, and executable. We will present formal definitions of these properties in Section 3-C. Among these properties, truthfulness is the crowning jewel which guarantees the robustness of the auction, such that each user is better off being honest to reveal its true cost. If truthfulness is not guaranteed for an auction, dishonest users may misreport their sensing costs and receive payments higher than they deserve, and honest users may leave the crowdsensing platform in fear of price manipulation.

Though being widely-adopted, auction has its limitation. In a crowdsensing auction, it is assumed that all users are known to be present and are aware of the existence of the auction. However, this is not always true for crowdsensing. In many scenarios, finishing a crowdsensing job requires a large amount of users' efforts, and the efforts from existing users are far from enough. Thus, auction alone cannot satisfy the need when there are not enough users to perform the sensing tasks. One possible solution is to provide extra incentives to users who could recruit more users for the crowdsensing job.

On the other hand, incentive tree mechanisms are designed [3, 6–8, 24] to incentivize individuals for solicitation. An incentive tree is a tree-structured incentive mechanism which offers rewards to each individual for solicitation. Incentive tree mechanisms can be applied to provide incentives for existing users to spread the word to other potential users hidden in their social networks. It can depict the solicitation process of the crowdsensing users (if user P_1 refers user P_2, then in the incentive tree P_1 is the parent of P_2), and computes the payment for each user based on its own contribution and the contribution from its descendants.

A perfect poster child of the incentive tree mechanisms is the Pyramid Scheme [27], which provides promising rewards for solicitation. The DARPA Network Challenge [4] is another example where the incentive tree demonstrated its efficiency in crowdsourcing. In 2009, DARPA initiated the challenge where contestants were asked to locate ten balloons deployed across the United States with a $\$40,000$ reward for the winner. An MIT team [26] won the challenge by applying an incentive tree mechanism which recruited nearly 4,400 participants within nine hours to find all ten balloons. Their strategy is as follows. Each balloon finder is rewarded with $\$2,000$. The inviter of the finder is rewarded with $\$1,000$, and the one who refers the inviter is rewarded with $\$500$, and so on. This mechanism not only mobilizes participants to locate the balloons, but to further refer others into action. Using similar strategies, the incentive tree has shown its efficiency in many contests such as the TAG Challenge [29] and the DARPA Shredder Challenge [5].

There is a drawback in these incentive tree mechanisms despite their efficiencies – they are not robust against sybil attacks. A sybil attack is that a user may generate multiple fake identities, and gain a higher utility without devoting extra efforts. E.g., in the MIT strategy of the DARPA Network Challenge, Bob is the balloon finder and Alice is the inviter of Bob. Then Bob receives $\$2,000$ and Alice receives $\$1,000$. Now suppose that Bob launches a sybil attack and splits himself into Bob1 and Bob2, where Bob1 is the balloon finder, Bob2 is the inviter of Bob1, and Alice is the inviter of Bob2. According to the payment rule, Bob1 receives $\$2,000$, Bob2 receives $\$1,000$, and Alice receives $\$500$. Then Bob will receive $\$2,000 + \$1,000 = \$3,000$ comparing to what he deserves ($\$2,000$). On the other hand, Alice will receive $\$500$ comparing to what she deserves ($\$1,000$). From this example, we know that without robustness against sybil attacks, a dishonest user may increase its utility with no extra efforts devoted, and an honest user may not receive the utility that she deserves. Thus, an incentive tree mechanism should not only offer rewards for solicitation, but be robust against sybil attacks (and this property is named as sybil-proofness).

To design an incentive mechanism that incentivizes both participation and solicitation with truthfulness and sybil-proofness, we cannot simply combine an existing sybil-proof incentive tree with an existing truthful auction, as these two properties may bring new issues to each other. For instance, there may be coalitions from the identities generated by one dishonest user. This coalition may violate the truthfulness of the mechanism as most crowdsensing auctions are not designed for robustness against coalitions. Furthermore, it is proved that no deterministic auction can be robust against coalitions [12]. Thus, we turn to randomized algorithms for a satisfactory probability of truthfulness and sybil-proofness. Therefore, it is non-trivial to design a crowdsensing incentive mechanism which is truthful and meanwhile sybil-proof.

We propose **RIT** (**R**obust **I**ncentive **T**ree Mechanism for Crowdsensing) to incentivize users for both participation and solicitation. We prove that **RIT** is truthful and sybil-proof with probability at least H, for any given $H \in (0, 1)$. We also prove that **RIT** satisfies individual rationality, computational efficiency, and solicitation incentive. We implemented **RIT** and the extensive evaluations further confirm our analysis.

The main contributions of this paper are:

- To the best of our knowledge, we initiate the problem of designing crowdsensing incentive mechanisms which incentivize users for both participation and solicitation, while guaranteeing truthfulness and sybil-proofness.
- We design **RIT** as the first robust auction-based incentive tree mechanism for crowdsensing. We prove that **RIT** is truthful and sybil-proof with probability at least H, for any given $H \in (0, 1)$. We also prove that **RIT** satisfies individual rationality, computational efficiency, and solicitation incentive.

The remainder of this paper is organized as follows. In Section 2, we review the state-of-art works on truthful crowdsensing auctions and sybil-proof incentive tree mechanisms. In Section 3, we describe the crowdsensing model, formulate the problem studied in this paper, and present the design goal. In Section 4, we illustrate the challenges when designing truthful and sybil-proof incentive mechanisms. We present **RIT** and its analysis in Section 5 and Section 6, respectively. In Section 7, we present and analyze the extensive performance evaluation results. We draw our conclusions in Section 8.

2. RELATED WORK

There is an abundance of research efforts on truthful auction design for crowdsensing. Two mobile crowdsensing models were proposed by Yang *et al.* [35], where a unique Stackelberg Equilibrium was computed in the first and a truthful auction was designed for the second. Jin *et al.* [17] proposed a truthful and quality-aware crowdsensing auction which approximates the social welfare. Wen *et al.* [34] designed a quality-driven auction for crowdsensing which guarantees truthfulness and maximizes the social welfare. A truthful auction for crowdsensing was proposed by Koutsopoulos [20] to determine the participation levels and the payments. Luo *et al.* [23] designed an all-pay auction which approximates the utility of the organizer in crowdsensing. Feng *et al.* studied auctions for location-aware crowdsensing [9]. Wang *et al.* [32] designed a quality-aware truthful auction for crowdsensing, which minimizes the expected expenditure. There are also existing works on collusion-resistant auctions. Goldberg and Hartline [12] studied truthful auctions using a consensus rounding technique to achieve truthfulness against coalitions with a high probability. Following this line, Wang *et al.* [33] applied multiple rounds of an algorithm AEM from [12] to allocate resources in cloud computing. Zhou and Zheng [43] combined collusion-resistance with spatial reusability for cognitive radio networks. Zhang *et al.* [38] designed a truthful and sybil-proof auction for crowdsourcing. However, none of these auctions considers providing incentives for solicitation.

As for the sybil-proof incentive trees, Douceur and Moscibroda [6] proposed Pachira Lottree for distributed service installations, which is sybil-proof. Emek *et al.* [8] proposed a sybil-proof diffusion mechanism within a social network. Two families of sybil-proof incentive trees were proposed by Lv and Moscibroda [24], where each satisfies a maximal subset of the desired properties. Following this line, Zhang *et al.* [37] proposed a time-sensitive and sybil-proof incentive tree for crowdsourcing. Drucker and Fleischer [7] proposed a family of sybil-proof mechanisms by modifying incentive mechanisms with natural properties. Chen *et al.* [3] proposed a direct referral tree under the query incentive networks. None of these incentive tree mechanisms considers using truthfulness to guide the behavior of each participant.

3. Crowdsensing Model and Problem Formulation

In this section, we present the crowdsensing model, describe the dishonest behaviors, and state the design goal.

A. Crowdsensing Model

In the crowdsensing model, there is a *crowdsensing platform* who has a *sensing job* $\mathcal{J}$ to be finished. $\mathcal{J}$ consists of many sensing *tasks* that can be completed by a crowdsensing user single-handedly. Each sensing task is indivisible. We categorize the tasks into m *types*: $\tau_1, \tau_2, ...,$ and τ_m, according to their locations. For instance, in mobile spectrum sensing, users are required to sense the spectrum usage over a large region. We divide the whole region into many different and non-overlapping areas, where each area contains several points of interest (*POI*s) to be sensed by the users. We regard the sensing in each area as one task type, and regard the sensing at one POI as one task. We use Γ to denote the set $\{\tau_1, \tau_2, ...\tau_m\}$.

There is a set of n *crowdsensing users* $\mathcal{P} = \{P_1, P_2, ..., P_n\}$ who can provide services to *complete* tasks. Each crowdsensing user P_j chooses one area (task type) $t_j \in \Gamma$ to perform the sensing tasks. In mobile spectrum sensing, this relates to the geographical location of the user, where each user has difficulties to sense the spectrum usage in two different areas in the same time window. Each user P_j can complete at most $K_j > 0$ tasks in type $t_j \in \Gamma$. E.g., if P_1 can complete at most three tasks in type τ_2, then $t_1 = \tau_2$ and $K_1 = 3$. Meanwhile, P_j has a private *unit cost* $c_j > 0$ to complete one task in type t_j. We define the largest K_j as K_{max}, i.e., $K_{max} = \max_{P_j \in \mathcal{P}}\{K_j\}$.

The job $\mathcal{J}$ is described as a multi-subset of Γ and it is *finished* if and only if all tasks in $\mathcal{J}$ are completed. We use m_i to denote the number of tasks in type τ_i requested by $\mathcal{J}$. For instance, if there is a crowdsensing job $\mathcal{J} = \{\tau_1, \tau_2, \tau_3, \tau_3\}$, we have $m = 3$, $m_1 = 1$, $m_2 = 1$, and $m_3 = 2$. $\mathcal{J}$ is finished if and only if one task in type τ_1, one task in type τ_2, and two tasks in type τ_3 are completed.

The crowdsensing model works as follows. The platform posts the sensing job $\mathcal{J}$. After $\mathcal{J}$ is posted, several users would join and perform the sensing tasks. However, these users may not be able to finish the job (perform all sensing tasks) by themselves. Thus, the platform offers extra incentives such that the existing users would further refer other users to join them. We use an *incentive tree* T to depict this solicitation process, where each user is represented by a node in the tree. There is an edge from P_i to P_j *iff* P_j joins by the solicitation of P_i. To make the structure of T a tree instead of a forest, we set the crowdsensing platform as the *root* and the users who join at the very beginning are the children of the root. When the number of crowdsensing users reaches a threshold value N, T stops growing and the solicitation comes to an end. We will discuss how to choose the threshold value of N in Remark 6.1. Upon joining the incentive tree, each user will notify the platform from whom it is solicited. Thus, the structure of T is known to the platform at the end of solicitation. We use T'_j to denote the set of nodes who are descendants of P_j, and r_j to denote the distance from P_j to the root.

Upon joining the incentive tree, each user P_j submits an *ask* (t_j, k_j, a_j) to the platform, where $k_j > 0$ is the maximum number of tasks that P_j claims to be able to complete, and *ask value* $a_j > 0$ is the minimum amount of reward that P_j requires to complete one task in type t_j. This submission is seal-bid, which means that by the time of submission, no user is aware of the ask from any other user. Note that k_j is not necessarily equal to K_j, but we assume that $k_j \leq K_j$ since P_j is not able to complete more than K_j tasks. Furthermore, a_j is not necessarily equal to c_j, since P_j may gain a higher utility by not revealing its cost. We use $\mathbb{A}$ to denote the *ask vector* $((t_1, k_1, a_1); (t_2, k_2, a_2); ...; (t_N, k_N, a_N))$.

After collecting the asks, the crowdsensing platform first computes an *auction payment* p_j^A for each user P_j. The platform also computes the *indicator* x_j for P_j which indicates the number of tasks in type t_j that is allocated to P_j. Combining the auction payments and the incentive tree T, the platform computes the *final payment* p_j for each user P_j, which is the actual amount that the platform pays to P_j. *Note that the auction payment is not the payment that each user receives. We only use it to compute the final payment* p_j. We use $\mathbf{p^A}$, $\mathbf{p}$, and $\mathbf{x}$ to denote vectors $(p_1^A, p_2^A, ..., p_N^A)$, $(p_1, p_2, ..., p_N)$, and $(x_1, x_2, ..., x_N)$, respectively.

The *utility* of user P_j with ask (t_j, k_j, a_j) is defined as $U_j(t_j, k_j, a_j) = p_j - x_j c_j$, which is P_j's payment minus its incurred cost.

B. Dishonest Behaviors

We consider two situations that a user P_j may gain a higher utility from being dishonest. The first one is that P_j may submit an ask value a_j deviating from its cost c_j. The second one is that P_j may generate fake identities to manipulate the auction payments or to increase the rewards from the incentive tree. We use truthfulness to incentivize P_j to reveal its cost c_j. To prevent P_j from generating fake identities, we first give a formal description of how a sybil attack is launched before introducing the concept of sybil-proofness.

A *sybil attack* is that a user P_j may generate $\delta(j)$ fake identities: $P_{j_1}, P_{j_2}, ..., P_{j_{\delta(j)}}$, where $\delta(j) > 1$. An identity of P_j resides in the incentive tree either as a child of P_j's parent or as a child of another identity of P_j. It does not attach to other users because the other users didn't reach out for P_j during the solicitation. For each child of P_j in the original incentive tree T, it is attached to one of P_j's identities after the sybil attack, while the other parts of the incentive tree remain unchanged.

Remark 3.1: When defining sybil attacks, we attach an identity of P_j to P_j's parent or P_j's other identities as a technical convention of sybil attacks [6–8, 24]. □

Each identity of P_j, denoted as P_{j_l}, acts as a user who submits an ask $(t_{j_l}, k_{j_l}, a_{j_l})$, where t_{j_l}, k_{j_l}, and a_{j_l} have similar definitions to those of t_j, k_j and a_j, respectively. We assume that P_j's identities' asks do not exceed P_j's capability, i.e., $t_{j_l} = t_j$ and $\sum_{l=1}^{\delta(j)} k_{j_l} \leq K_j$. Thus, the ask $(t_{j_l}, k_{j_l}, a_{j_l})$ can be written as (t_j, k_{j_l}, a_{j_l}). The unit cost of P_{j_l} is $c_{j_l} = c_j$.

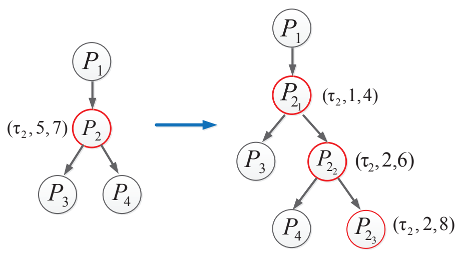


Fig. 1. A sybil attack from P_2

Fig. 1 presents an example of a sybil attack from P_2 with ask $(\tau_2, 5, 7)$. P_2 generates three identities P_{2_1}, P_{2_2}, and P_{2_3} with asks $(\tau_2, 1, 4)$, $(\tau_2, 2, 6)$, and $(\tau_2, 2, 8)$, respectively.

Similar to the definition of P_j's utility in Equation, P_{j_l}'s utility is defined as $U_{j_l}(t_j, k_{j_l}, a_{j_l}) = p_{j_l} - x_{j_l} c_j$, where p_{j_l} and x_{j_l} have similar definitions to those of p_j and x_j, respectively. The utility of P_j from a sybil attack is $\sum_{l=1}^{\delta(j)} p_{j_l} - \sum_{l=1}^{\delta(j)} x_{j_l} c_j = \sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, a_{j_l})$.

We claim that for each P_j, it could not generate more than K_{max} fake identities. This is because for each fake identity P_{j_l}, $K_j \geq k_{j_l} \geq 1$. Thus, there will be no more than K_j fake identities of P_j. This claim is closely related to the design goal introduced in Section 3-C.

C. Desired Properties and Design Goal

The concepts of the desired properties are presented as follows.

- **Truthfulness:** No user could increase its utility by reporting an ask value other than its cost, i.e., $U_j(t_j, k_j, c_j) \geq U_j(t_j, k_j, a_j)$;
- **Sybil-Proofness**: No user could benefit from generating multiple fake identities, i.e., $U_j(t_j, k_j, c_j) \geq \sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, a_{j_l})$, where $k_j \geq \sum_{l=1}^{\delta(j)} k_{j_l}$, for any k_j and $\delta(j)$;
- **Individual Rationality:** No user has a negative utility by revealing its cost, i.e., $U_j(t_j, k_j, c_j) \geq 0$;
- **Computational Efficiency:** The mechanism can be executed within polynomial time;
- **Solicitation Incentive:** If P_l is about to join T, then P_j's utility when P_l joins as P_j's child is no less than P_j's utility when P_l joins as another user's child.

There are many other properties that receive interests from research communities when designing crowdsensing incentive mechanisms, such as data quality guarantee [11, 36] and privacy protection [16, 18, 22]. These properties are out of the scope of this paper and are subject to future research.

In this paper, we want to design an incentive mechanism to guarantee both truthfulness and sybil-proofness, i.e., $U_j(t_j, K_j, c_j) \geq \sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, a_{j_l})$. However, we cannot simply combine a sybil-proof incentive tree with an existing truthful crowdsensing auction to achieve this. The reasons are two-folded. On one hand, by lying about the asks, an identity of a user may increase the payment of another identity in the incentive tree. On the other hand, multiple identities from the same user may form coalitions to violate truthfulness of the mechanism, since most truthful crowdsensing auctions do not consider such coalitions.

To overcome such difficulty, we take the inspiration from a consensus rounding technique [12], and develop new schemes to guarantee truthfulness against coalitions with a high probability. By defining high probability, it regards to some parameter of the input, e.g., the number of asks and the number winners in the auction. We introduce the following concepts.

- d**-truthfulness:** For any coalition of size at most d, the total utility of the coalition could not be increased if some of them reveal their ask values other than their costs;
- (d, η)**-truthfulness:** A mechanism is (d, η)-truthful if it is d-truthful with probability at least η.

Since P_j could not generate more than K_{max} fake identities, the incentive mechanism needs to be K_{max}-truthful for each task type. It has been proved in [12] that no deterministic auction can achieve K_{max}-truthfulness when $K_{max} \geq 2$. Thus, our goal is to design a (K_{max}, H)-truthful mechanism.

The design goal of this paper: under the crowdsensing model presented in Section 3-A, given a probability $H \in (0, 1)$, design an incentive mechanism such that for each crowdsensing user P_j, $U_j(t_j, K_j, c_j) \geq \sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, a_{j_l})$ for any K_j and $\delta(j)$ with probability at least H. This encourages P_j to reveal K_j and c_j, and not launch sybil attacks. The mechanism should meanwhile guarantee individual rationality, computational efficiency, and solicitation incentive.

4. Design Challenges

In Section 3-C, we have briefly introduced the difficulty to achieve both truthfulness and sybil-proofness. In this section, we illustrate the challenges in detail when designing the incentive mechanism. We focus on the impact of truthfulness and sybil-proofness on each other. *Through these discussions, we show that we cannot simply combine an existing truthful auction and an existing sybil-proof incentive tree mechanism. Thus, it is non-trivial to design a mechanism that achieves both truthfulness and sybil-proofness.*

A. Impact of Auctions on Sybil-proofness

Most sybil-proof incentive mechanisms are contribution-based [2, 6, 7, 24], where payments are computed based on the contribution from the users. In this paper, we use the auction payment to quantify the contribution of each user, since payment itself is a measurement of the contribution.

For existing sybil-proof mechanisms, they are robust against multi-identity attacks, such that if a user launches an attack, it will not have an increment in utility. However, if we combine the auctions with a sybil-proof incentive tree mechanism, the sybil-proofness might be violated.

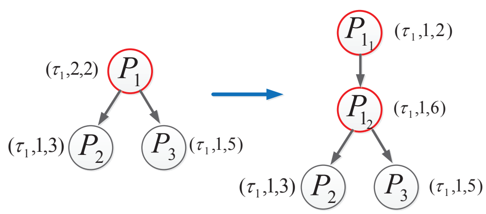


Fig. 2. An illustration of the impact from the auctions on sybil-proofness

We use the Fig. 2 to provide an example, where there are three users P_1, P_2, and P_3 with truthful asks $(\tau_1, 2, 2), (\tau_1, 1, 3)$, and $(\tau_1, 1, 5)$, respectively. The job $\mathcal{J}$ requires two tasks of type τ_1.

For ease of understanding, we use the k-th lowest price auction instead of the complicated truthful crowdsensing auctions for this example. In the k-th lowest price auction, there are several bidders, each of whom sells an item (or service). Each bidder has a private cost and submits an ask. The winners are the ones who submit the $k-1$ lowest asks, and their payments are the k-th lowest ask. Each bidder's utility is the payment received minus its cost. It has been proven that the k-th lowest price auction is a truthful auction [31]. The sybil-proof incentive tree mechanism that we use to compute the final payment is from [24], where the reward for P_j is $p_j = 2p_j^A + \ln(1 - \frac{p_j^A}{1+\sum_{P_j \in T_j} p_l^A})$. (We use the auction payment as the contribution of each user.)

We focus on P_1. If each user submits its truthful ask, P_1 is assigned to complete two tasks, and the auction payment is $2 \times 3 = 6$. Thus, $p_1 = 5.85$ and P_1's utility is $5.85 - 2 \times 2 = 1.85$. Suppose that P_j launches a multi-identity attack as shown in Fig. 2. Then P_{1_1} is assigned with one task and its auction payment is 5. P_2 is assigned with the other task and its auction payment is 5. Thus, $p_{1_1} = 4.39$, $p_{1_2} = 0$, and P_1's utility is $4.39 - 2 = 2.39 > 1.85$. From this example, we know that combining auctions with sybil-proof incentive trees may violate sybil-proofness.

B. Impact of Incentive Trees on Truthfulness

There is a large body of efforts on truthful incentive mechanisms for crowdsensing [9, 17, 20, 23, 35, 36, 39–41]. However, if we combine these mechanisms with incentive trees, where users have incentives for solicitation, the truthfulness of these mechanisms may be violated since the payment is also related to the solicitation incentive.

We use the instance in Fig. 3 to illustrate such impact.

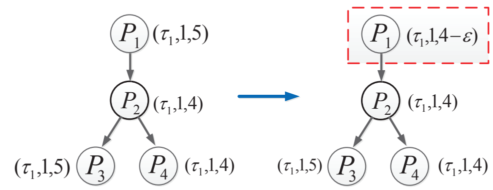


Fig. 3. An illustration of the impact from the incentive trees on truthfulness

In Fig. 3, there are four bidders P_1, P_2, P_3, and P_4 who sell services for one task type τ_1, with cost values $5, 4, 5$, and 4, respectively. The job $\mathcal{J}$ requires two tasks of type τ_1. We use the third price auction and the sybil-proof incentive tree in [24] to compute the final payment, where the reward for P_j is $p_j = 2p_j^A + \ln(1 - \frac{p_j^A}{1+\sum_{P_j \in T_j} p_l^A})$.

We focus on user P_1. If each user submits its truthful ask, $p_1^A = 0$, $p_1 = 0$, and P_1's utility is 0. If P_1 bids $4 - \epsilon$ instead, then $p_1^A = 4$, $p_1 = 7.41$, and P_1's utility is $7.41 - 5 = 2.41 > 0$. Thus, P_1 has the incentive to bid untruthfully, which indicates that incentive trees may violate the truthfulness of an auction. Furthermore, when applying incentive trees on truthful auctions, the auction payment may also be influenced if there is a multi-identity attack. This is because the auction payment of one identity depends on the asks from other identities of the same user.

5. Design of RIT

In this section, we present the incentive mechanism **RIT** (**R**obust **I**ncentive **T**ree Mechanism for Crowdsensing).

A. Design Rationale

The mechanism **RIT** consists of two phases: the *auction phase* and the *payment determination phase*. The auction phase allocates tasks to each user j and computes the auction payment p_j^A. In the auction phase, we propose two algorithms: **CRA** (**C**ollusion **R**essistant **A**uction) in Algorithm 1 as a basic auction, which allocates at most m_i tasks to users and guarantees k-truthfulness with a high probability, and $Extract$ in Algorithm 2, which converts the asks from the users into the format that **CRA** requires. To make sure that exactly m_i tasks are assigned to the users for each τ_i, we run multiple rounds of **CRA**s to allocate the tasks and meanwhile guarantee (K_{max}, H)-truthfulness. In the payment determination phase, based on the auction payment p_j^A and the solicitation of user j in the incentive tree, we propose an incentive tree mechanism to compute p_j.

B. Design Details

We first present **CRA** in Algorithm 1 and $Extract$ in Algorithm 2 which are used by **RIT** in Algorithm 3.

We design **CRA** in Algorithm 1 to allocate at most q tasks in type τ_i. Let $\alpha = (\alpha_1, \alpha_2, ...)$ be a vector of asks where each α_ω bids for one task in type τ_i. Let x'_ω be the indicator such that if α_ω is allocated to one task in **CRA**, $x'_\omega = 1$; $x'_\omega = 0$ otherwise. Let p'_ω be the auction payment for α_ω. **CRA** takes the ask vector α for type τ_i, the number of unallocated tasks (in τ_i) q, and m_i as input, and outputs the indicator vector $\mathbf{x}' = (x'_1, x'_2, ...)$ and payment vector $\mathbf{p}' = (p'_1, p'_2, ...)$. From Line 1 to Line 16 of Algorithm 1, we select at most $q + m_i$ asks as potential winning asks. If there are more than q asks selected, we randomly choose q asks among them with equal probability as the winning asks and allocate one task to each winning ask. The reason why we first select no more than $q+m_i$ potential winning asks and then choose q winning asks is to guarantee that the auction phase is K_{max}-truthful with high probability. We will provide more detailed explanations in Lemma 6.2 and Remark 6.1.

In Algorithm 1, we first sample a subset $\mathcal{S}$ of asks from α randomly with equal probability $\frac{1}{q+m_i}$ in Line 2. We define s as the smallest sampled ask value in Line 3. Next we calculate n_s based on its definition in Line 5. If $n_s \le q+m_i$, we temporarily choose the smallest n_s asks in α in Line 7. Otherwise, among the smallest n_s asks, we choose each one with equal probability $\frac{q+m_i}{2n_s}$ independently in Line 10. However, the number of selected asks may still exceed $q+m_i$. Therefore, we apply a $q + m_i + 1$-st auction in Line 14 and Line 15, where we choose the smallest $q + m_i$ asks, and set the payment s as the $q + m_i + 1$-st smallest ask value. Till here, the algorithm selects no more than $q + m_i$ asks. If there are more than q asks chosen, there will not be enough tasks to be allocated. To make sure that **CRA** allocates no more than q tasks, we select q asks randomly among these chosen asks with equal probability as final winning asks in Line 18 and all the others are losing asks. The corresponding payment is s for each winning ask in Line 22.

In **CRA**, each ask bids for one task, whereas in **RIT** the ask from each user is in the format of (t_j, k_j, a_j). Thus, we use $Extract$ in Algorithm 2 to transform the ask vector $\mathbb{A}$ into the vector α for each task type τ_i, such that α contains all asks that bid for tasks in type τ_i. $Extract$ takes the task type τ_i and the ask vector $\mathbb{A}$ as input, and outputs the ask vector $\alpha = (\alpha_1, \alpha_2, ...)$ for task type τ_i and a function $\lambda(\cdot)$, such that $\lambda(\omega) = j$ indicates that α_ω comes from user P_j. For the ask (t_j, k_j, a_j) from each user P_j, if t_j equals to τ_i, $Extract$ expands its ask into k_j asks of value a_j in α and updates the function $\lambda(\cdot)$ in Line 6 of Algorithm 2. As an illustration, if $\mathbb{A} = ((\tau_1, 2, 3); (\tau_2, 3, 4); (\tau_1, 4, 2))$, after $Extract(\tau_1, \mathbb{A})$, the ask vector $\alpha = (3, 3, 2, 2, 2, 2)$, and the λ function is computed as $\lambda(1) = 1$, $\lambda(2) = 1$, $\lambda(3) = 3$, $\lambda(4) = 3$, $\lambda(5) = 3$, and $\lambda(6) = 3$.

Algorithm 1: CRA(α, q, m_i)

1 $\mathbf{x}' \leftarrow \mathbf{0}$, $\mathbf{p}' \leftarrow \mathbf{0}$;
2 Let $\mathcal{S}$ be a sample of α where each ask is sampled with probability $\frac{1}{q+m_i}$ independently;
3 $s \leftarrow \min_{\alpha_\omega \in \mathcal{S}}\{\alpha_\omega\}$;
4 Let y be a uniform random value over $[0, 1]$;
5 $n_s \leftarrow \lfloor \sharp_s(\alpha) \rfloor_{\{2^{z+y}:z\in\mathbb{Z}\}}$, where $\sharp_s(\alpha)$ is the number of asks at most s in α, and $\lfloor \sharp_s(\alpha) \rfloor_{\{2^{z+y}:z\in\mathbb{Z}\}}$ is the nearest round-down value to $\sharp_s(\alpha)$ in $\{2^{z+y} : z \in \mathbb{Z}\}$;
6 **if** $n_s \le q + m_i$ **then**
7   Choose the smallest n_s asks;
8 **else**
9   **for** *each ask value* α_ω *among the* n_s *asks* **do**
10   Choose α_ω with probability $\frac{q+m_i}{2n_s}$;
11   **end**
12 **end**
13 **if** *there are more than* $q + m_i$ *asks chosen* **then**
14   Choose the smallest $q + m_i$ asks as winning asks;
15   Let s be the $q + m_i + 1$-st smallest ask value among the winning asks;
16 **end**
17 **if** *there are more than* q *asks chosen* **then**
18   Randomly select q asks as winners with equal probability and set the others as losers;
19 **end**
20 **for** *each* α_ω *in* α **do**
21   **if** α_ω *is a winning ask* **then**
22         $x'_\omega \leftarrow 1$, $p'_\omega \leftarrow s$;
23   **end**
24 **end**
25 **return** $(\mathbf{x}', \mathbf{p}')$.

Algorithm 2: $Extract(\tau_i, \mathbb{A})$

1 $\omega \leftarrow 0$;
2 **for** $j = 1, 2, ..., N$ **do**
3   **if** $t_j = \tau_i$ **then**
4   **for** $f = 1, 2, ..., k_j$ **do**
5             $\omega \leftarrow \omega + 1$;
6             $\alpha_\omega \leftarrow a_j$, $\lambda(\omega) \leftarrow j$;
7   **end**
8   **end**
9 **end**
10 **return** $(\alpha, \lambda(\cdot))$.

In Algorithm 3, **RIT** takes the job $\mathcal{J}$, users' asks $\mathbb{A}$, the incentive tree T, and the probability H as input, and outputs the indicator $\mathbf{x}$ and the final payment vector $\mathbf{p}$. **RIT** consists of two phases, the auction phase and the payment

Algorithm 3: RIT$(\mathcal{J}, \mathbb{A}, T, H)$

/* *Auction Phase* */

1 $\mathbf{x} \leftarrow 0$, $\mathbf{p} \leftarrow 0$;
2 $\mathbf{p}^A \leftarrow 0$, $\eta \leftarrow H^{\frac{1}{m}}$;
3 **for** *each task type* τ_i **do**
4 Define $\mathbb{A}'$ as $((t'_1, k'_1, a'_1); (t'_2, k'_2, a'_2); ...)$;
5 $\mathbb{A}' \leftarrow \mathbb{A}$;
6 $q \leftarrow m_i$, $rounds \leftarrow 0$;
7 $max \leftarrow \lfloor \frac{\lg \eta}{\lg((1-\frac{1}{m_i})^{K_{max}} + \log(1-\frac{2K_{max}}{m_i}) - e^{-\frac{m_i}{8}})} \rfloor$;
8 **while** $rounds < max$ *and* $q > 0$ **do**
9 $(\alpha, \lambda(\cdot)) \leftarrow Extract(\tau_i, \mathbb{A}')$;
10 $(\mathbf{x}', \mathbf{p}') \leftarrow$**CRA**$(\alpha, q, m_i)$;
11 **for** *each ask* α_ω *in* α **do**
12 **if** $x'_\omega = 1$ **then**
13 $x_{\lambda(\omega)} \leftarrow x_{\lambda(\omega)} + 1$;
14 $p^A_{\lambda(\omega)} \leftarrow p^A_{\lambda(\omega)} + p'_\omega$;
15 ${k'}_{\lambda(\omega)} \leftarrow {k'}_{\lambda(\omega)} - 1$;
16 $q \leftarrow q - 1$;
17 **end**
18 **end**
19 $rounds \leftarrow rounds + 1$;
20 **end**
21 **end**

/* *Payment Determination Phase* */

22 **if** *all tasks in* $\mathcal{J}$ *are assigned* **then**
23 **for** *each user* j *in the incentive tree* **do**
24 $p_j = p^A_j + \sum_{P_l \in \Gamma_j, t_l \neq t_j} (\frac{1}{2})^{r_l} p^A_l$;
25 **end**
26 **else**
27 $\mathbf{x} \leftarrow 0$, $\mathbf{p} \leftarrow 0$;
28 **end**
29 **return** $(\mathbf{x}, \mathbf{p})$.

determination phase. The auction phase (Line 2 to Line 21) applies multiple rounds of **CRA**s to allocate tasks and compute auction payments for each τ_i. We first calculate the probability η in Line 2 such that **RIT** needs to be (K_{max}, η)-truthful for each type τ_i. From Line 3 to Line 21, we allocate m_i tasks and compute the corresponding auction payments. We set q as the number of unallocated tasks and $rounds$ as the number of finished rounds of **CRA**s. We calculate max for each τ_i as the maximum number of rounds to apply **CRA** in Line 7. We prove it in Lemma 6.3 that if we run **CRA** for no more than max rounds for each τ_i, **RIT** is (K_{max}, H)-truthful. Before applying **CRA** in Line 10, we first use $Extract$ to construct the ask vector α from the unallocated asks $\mathbb{A}'$ in Line 9. After **CRA**, we update the indicator $x_{\lambda(\omega)}$, the auction payment $p^A_{\lambda(\omega)}$, and the remaining capability $k'_{\lambda(\omega)}$ for the user who submits a winning ask α_ω in Line 13, Line 14, and Line 15, respectively. We also update q and $rounds$ in Line 16 and Line 19, respectively. The loop from Line 8 to Line 21 terminates when $rounds$ exceeds max or all tasks in type τ_i have been allocated.

The payment determination phase (Line 22 to Line 28) computes the final payment for each user. If all tasks required by $\mathcal{J}$ are allocated, we apply the function in Line 24 to compute p_j for each user P_j. Otherwise, $\mathcal{J}$ cannot be finished while satisfying the desired properties required by the design goal, and we set the payment for each user as 0 with no allocated tasks (Line 27). In Line 24, for each P_j, the payment p_j sums up the auction payment p^A_j and the weighted auction payments from its descendants whose task types are different from t_j. For each P_l whose auction payment contributes to p_j, the corresponding weight is $(\frac{1}{2})^{r_l}$.

6. Desired Properties of RIT

In this section, we analyze **RIT** and prove that **RIT** satisfies the desired properties required by the design goal of this paper.

Lemma 6.1: The auction phase of **RIT** is individually rational, i.e., if $a_j = c_j$, $p^A_j \geq x_j c_j$. □

Proof: Let $j = \lambda(\omega)$. In **CRA**, if an ask $\alpha_\omega = c_j$ loses, P_j's utility from this ask is 0 since the payment would be 0 and there's no cost incurred. If an ask $\alpha_\omega = c_j$ wins the auction, then the payment would be determined by Line 3 or Line 15 in Algorithm 1. If the payment is determined by Line 3, then all of the n_s winning asks have ask values lower than the payment s according to the definition of n_s in Line 5, which means that $s \geq c_j$. If the payment is determined by Line 15, since s is the $q+m_i+1$-st smallest ask value and α_ω is among the $q+m_i$ smallest ask values, we have $s \geq c_j$. Thus, for each P_j and τ_i, the auction payment p^A_j equals to the sum of the payment s in each round of **CRA**, and this value is no less than the total cost $x_j c_j$. ■

Theorem 1: RIT is individually rational. □

Proof: $U_j(t_j, k_j, c_j) = p_j - x_j c_j = (p^A_j - x_j c_j) + \sum_{P_l \in \Gamma_j, t_l \neq t_j} (\frac{1}{2})^{r_l} p^A_l$. By Lemma 6.1, we have $p^A_j \geq x_j c_j$. Since $p^A_l \geq 0$, we have $U_j(t_j, k_j, c_j) \geq 0$. ■

Lemma 6.2: CRA is k-truthful with probability no less than $(1 - \frac{1}{q+m_i})^k + \log(1 - \frac{2k}{q+m_i}) - e^{-\frac{q+m_i}{8}}$. □

Proof: Suppose there is a coalition of size k. Let $\mathcal{E}_s$ be the event that an ask from the coalition is in the sample $\mathcal{S}$ in Line 2 of Algorithm 1, let $\mathcal{E}_c$ be the event that n_s is not a k-consensus [12] and $n_s \geq q+m_i$, and let $\mathcal{E}_o$ be the event that there are more than $q+m_i$ winners in Line 13 of Algorithm 1. By Lemma 15 of [13], if none of these events occurs, **CRA** is k-truthful. Now we analyze the probabilities of these events:

- $Pr[\mathcal{E}_s] = 1 - (1 - \frac{1}{q+m_i})^k$, since each ask is sampled independently with probability $\frac{1}{q+m_i}$.
- If $n_s \leq q + m_i$, **CRA** is k-truthful when $\mathcal{E}_s$ does not occur; if $n_s > q + m_i$, by Lemma 9 and Lemma 15 of [13], this probability is less than $-\log(1 - \frac{2k}{q+m_i})$. Thus, $Pr[\mathcal{E}_c \cup \mathcal{E}_s] \leq 1 - (1 - \frac{1}{q+m_i})^k - \log(1 - \frac{2k}{q+m_i})$.
- Since the expected number of winning asks in Line 13 is $n_s \times \frac{q+m_i}{2n_s} = \frac{q+m_i}{2}$, by Chernoff bound [25] and

Lemma 15 of [13], $Pr[\mathcal{E}_o] \leq e^{-\frac{q+m_i}{8}}$.

Thus, **CRA** is k-truthful with probability $1 - Pr[\mathcal{E}_c \cup \mathcal{E}_s \cup \mathcal{E}_o] \geq (1 - \frac{1}{q+m_i})^k + \log(1 - \frac{2k}{q+m_i}) - e^{-\frac{q+m_i}{8}}$. ■

Remark 6.1: For the lower bound of the probability in Lemma 6.2, it is easy to verify that it decreases with the decrement of q. Thus, when $q = 0$, the lower bound is $(1 - \frac{1}{m_i})^k + \log(1 - \frac{2k}{m_i}) - e^{-\frac{m_i}{8}}$. When the coalition size is relatively small compared to the job size, i.e., $\frac{k}{m_i} \ll 1$, the lower bound is close to one. Since each user can only have at most K_{max} fake identities to form a coalition in **CRA** and $K_{max} \ll m_i$, **CRA** is K_{max}-truthful with a high probability. E.g., when $K_{max} = 10$ and $m_i = 1,000$, the lower bound is 0.98, which means that **CRA** is $(10, 0.98)$-truthful.

However, we cannot directly apply the consensus technique from [12]. If we use the same technique, the new lower bound becomes $(1 - \frac{1}{q})^k + \log(1 - \frac{2k}{q}) - e^{-\frac{q}{8}}$ by a similar analysis of Lemma 6.2. After several rounds of **CRA**s in Algorithm 3, it is possible that there is a small value of q such that $\frac{k}{q}$ is not close to zero, which drags down the lower bound of the probability. E.g., if $k = 10$ and $q = 50$, the new lower bound is 0.59, which is too low to be a satisfactory probability. To select $q + m_i$ winners in **CRA**, we need at least $2m_i$ asks to bid for tasks in τ_i in Algorithm 2, as the number of unallocated tasks q can be as most m_i for each τ_i. Thus, the incentive tree should propagate to N users such that for each τ_i, the users are able to complete at least $2m_i$ tasks. □

Lemma 6.3: For all $P_j \in \mathcal{P}$, $\sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, c_j) \geq \sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, a_{j_l})$ holds with probability at least H. □

Proof: For each task type τ_i, we apply at most $max = \lfloor \frac{\lg \eta}{\lg((1-\frac{1}{m_i})^{K_{max}} + \log(1-\frac{2K_{max}}{m_i}) - e^{-\frac{m_i}{8}})} \rfloor$ rounds of **CRA**s. Each round of **CRA** is K_{max}-truthful with probability at least $(1 - \frac{1}{m_i})^{K_{max}} + \log(1 - \frac{2K_{max}}{m_i}) - e^{-\frac{m_i}{8}}$. To make all rounds of **CRA**s be K_{max}-truthful for τ_i, it requires that each round of **CRA** is K_{max}-truthful, and this probability is no less than $((1 - \frac{1}{m_i})^{K_{max}} + \log(1 - \frac{2K_{max}}{m_i}) - e^{-\frac{m_i}{8}})^{max} \geq \eta$. Thus, the probability that the auction phase is truthful for all users is at least $\eta^m = H$.

In the payment determination phase, according to Line 24, $p_j - x_j c_j = p_j^A - x_j c_j + \sum_{P_l \in \Gamma_j, t_l \neq t_j} (\frac{1}{2})^{r_l} p_l^A$. P_j has no influence on $\sum_{P_l \in \Gamma_j, t_l \neq t_j} (\frac{1}{2})^{r_l} p_l^A$ by changing its own ask value. Therefore, $p_j - x_j c_j$ is maximized if and only if $p_j^A - x_j c_j$ is maximized. Since the probability that $p_j^A - x_j c_j$ is maximized when revealing c_j is at least H for all $P_j \in \mathcal{P}$, the probability that the inequality holds is at least H. ■

Lemma 6.4: RIT is sybil-proof when the ask values of all P_j's identities are the same, i.e., $U_j(t_j, k_j, a_j) \geq \sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, a_j)$, where $k_j = \sum_{l=1}^{\delta(j)} k_{j_l}$. □

Proof: From Algorithm 3, we know that for each task type τ_i, **CRA** is applied after $Extract$, which constructs the unit ask vector α from the ask vector $\mathbb{A}$. Thus, the number of fake identities does not influence the payment of the auction phase when $k_j = \sum_{l=1}^{\delta(j)} k_{j_l}$ and $a_{j_l} = a_j$ for each P_{j_l}, which indicates that $\sum_{l=1}^{\delta(j)} x_{j_l} = x_j$ and the auction payment for each ask does not change. Since $U_j(t_j, k_j, a_j) = p_j - x_j c_j$, $U_{j_l}(t_j, k_{j_l}, a_j) = p_{j_l} - x_{j_l} c_j$, and $x_j = \sum_{l=1}^{\delta(j)} x_{j_l}$, we need to prove that $p_j \geq \sum_{l=1}^{\delta(j)} p_{j_l}$ in order to prove this lemma.

Since an identity of P_j is attached to either P_j's parent or another identity of P_j according to Section 3-B, for any sybil attack, we can imitate the attack by a series of "simpler" attacks, where in each of these "simpler" attacks, an identity of P_j splits itself into two new identities. Therefore, if we can prove that none of these attacks could bring P_j a higher payment, we can prove $p_j - x_j c_j \geq \sum_{l=1}^{\delta(j)} p_j^l - \sum_{l=1}^{\delta(j)} x_{j_l} c_j$.

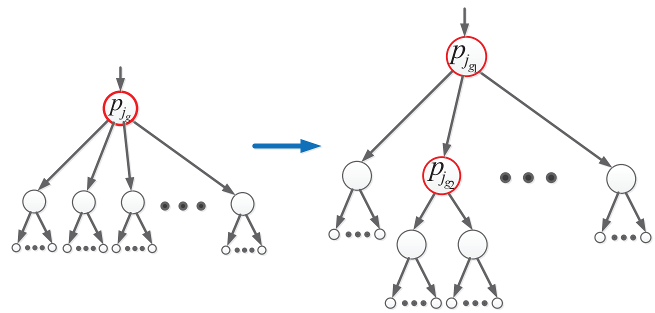


Fig. 4. An illustration of the attack when $P_{j_{g_1}}$ is the parent of $P_{j_{g_2}}$

There are two kinds of sybil attacks if P_j splits one of its identities P_{j_g} into two new identities, $P_{j_{g_1}}$ and $P_{j_{g_2}}$. The first is that $P_{j_{g_1}}$ is the parent of $P_{j_{g_2}}$, as illustrated in Fig. 4. Let z_l be the number of P_j's identities where P_l is a descendant of these identities. Let r_l' be the distance from a user P_l to the root after the attack. Let $T_{j_{g_2}}'$ denote the set of users who are descendants of $P_{j_{g_2}}$. The auction payment for each user does not change after the attack. If P_l is not a descendant of $P_{j_{g_2}}$, $r_l' = r_l$; otherwise, $r_l' = r_l + 1$. Thus, the change of P_j's payment from this sybil attack is $\sum_{P_l \in T_{j_{g_2}}', t_l \neq t_j} [(z_l + 1)(\frac{1}{2})^{r_l'} p_l^A - z_l (\frac{1}{2})^{r_l} p_l^A] = \sum_{P_l \in T_{j_{g_2}}', t_l \neq t_j} \frac{1-z_l}{2} (\frac{1}{2})^{r_l} p_l^A$. Since $z_l \geq 1$, this change of payment is non-positive, which indicates that the first kind of attack could not increase P_j's utility.

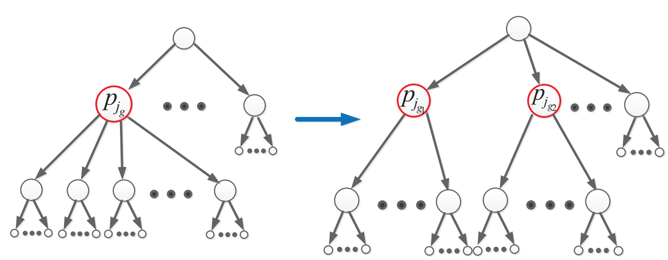


Fig. 5. An illustration of the attack when $P_{j_{g_1}}$ and $P_{j_{g_2}}$ are siblings

The second kind of attack is that $P_{j_{g_1}}$ and $P_{j_{g_2}}$ are siblings and their parents are the parent of P_{j_g}, as illustrated in Fig. 5. The total auction payment of all identities of P_j does not change. The auction payments for all other users do not change. $r_l' = r_l$ for each user P_l. Thus, the utility of P_j does not change after the sybil attack.

Therefore, the sybil attack with the same ask value could not bring P_j a higher utility. ■

Theorem 2: **RIT** is truthful and sybil-proof with probability at least H, i.e., $U_j(t_j, K_j, c_j) \geq \sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, a_{j_l})$ with probability at least H, where $\sum_{l=1}^{\delta(j)} k_{j_l} \leq K_j$. □

Proof: From Lemma 6.3, we know that $\sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, c_j) \geq \sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, a_{j_l})$ for any user P_j with probability at least H. From Lemma 6.4, we have $\sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, c_j) \leq U_j(t_j, \sum_{l=1}^{\delta(j)} k_{j_l}, c_j)$. Suppose that there is a user $P_{j'}$ whose parent is an identity of P_j with $t_{j'} = t_j$, $K_{j'} = K_j - \sum_{l=1}^{\delta(j)} k_{j_l}$, and $c_{j'} = c_j$. Since **RIT** is individually rational, we have $U_{j'}(t_j, K_{j'}, c_j) \geq 0$. Thus, $\sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, c_j) \leq U_j(t_j, \sum_{l=1}^{\delta(j)} k_{j_l}, c_j) + U_{j'}(t_j, K_{j'}, c_j)$. If we take $P_{j'}$ as another fake identity of P_j, by Lemma 6.4 we have $U_j(t_j, \sum_{l=1}^{\delta(j)} k_{j_l}, c_j) + U_{j'}(t_j, K_{j'}, c_j) \leq U_j(t_j, \sum_{l=1}^{\delta(j)} k_{j_l} + K_{j'}, c_j) = U_j(t_j, K_j, c_j)$. Thus, we have $U_j(t_j, K_j, c_j) \geq \sum_{l=1}^{\delta(j)} U_{j_l}(t_j, k_{j_l}, a_{j_l})$ with probability at least H for any $P_j \in \mathcal{P}$. ■

Theorem 3: **RIT** is computationally efficient. □

Proof: First we analyze the time complexity of the auction phase. For each task type τ_i, we run at most $\lfloor \frac{\lg \eta}{\lg((1-\frac{1}{m_i})^{K_{max}} + \log(1-\frac{2K_{max}}{m_i}) - e^{-\frac{m_i}{8}})} \rfloor$ rounds of **CRA**s. Since $(1-\frac{1}{m_i})^{K_{max}} + \log(1-\frac{2K_{max}}{m_i}) \simeq e^{-\frac{K_{max}}{m_i}} - \frac{2K_{max}}{m_i}$ when $\frac{K_{max}}{m_i} \ll 1$, we have $(1-\frac{1}{m_i})^{K_{max}} + \log(1-\frac{2K_{max}}{m_i}) - e^{-\frac{m_i}{8}} \leq e^{-\frac{K_{max}}{m_i}}$. Thus, the number of rounds of **CRA**s is $O(\sum_{i=1}^{m} \frac{-\lg \eta \times m_i}{K_{max}}) \in O(\frac{|\mathcal{J}|}{K_{max}})$. For each **CRA**, the running time is $O(\sum_{P_j \in \mathcal{P}, t_j = \tau_i} k_j) = O(NK_{max})$. Each $Extract$ is also bounded by $O(NK_{max})$. Thus, the time complexity for the auction phase is $O(\frac{|\mathcal{J}|}{K_{max}} \times NK_{max}) = O(N|\mathcal{J}|)$.

For the payment determination phase, to calculate the payment for each user P_j, we sum up the weighted auction payments from the users in T'_j. By computing this payment recursively from leaf to root, the time complexity of this phase is $O(N)$. Thus, the time complexity of **RIT** is $O(N|\mathcal{J}|)$, which indicates that **RIT** is computationally efficient. ■

Theorem 4: **RIT** satisfies solicitation incentive. □

Proof: Consider that there is a user P_l joining the incentive tree. Let $\hat{P}_j^A$ be the new auction payment of P_j. If $t_l = t_j$, no matter where P_l joins in the incentive tree, the change of P_j's utility is $\hat{P}_j^A - P_j^A$. If $t_l \neq t_j$, when P_l joins as a descendant of P_j, the change of P_j's utility is $(\frac{1}{2})^{r_l} P_l^A \geq 0$. To make this increment of utility as large as possible, P_j would like P_l to join as its child. When P_l joins as a non-descendant of P_j, P_j's utility does not change. Thus, comparing with P_l joining as a child of another user, P_j prefers P_l to joining as its own child for a higher utility, which indicates that **RIT** satisfies solicitation incentive. ■

7. Performance Evaluation

In this section, we implemented **RIT** and present the extensive performance evaluation results.

A. Simulation Setup

We present the setup of the simulation as follows. The job $\mathcal{J}$ consists of tasks in $m = 10$ task types. Each user P_j's task type t_j is randomly distributed among the 10 task types, and k_j is uniformly distributed over $(0, 20]$. The ask value a_j is uniformly distributed over $(0, 10]$. We set the threshold probability $H = 0.8$. All evaluations were tested on a Linux system with 3.4 GHz Core i-7 CPU and 8GB memory. All results are averaged over 1000 times.

To build the incentive tree, we used the data from [21], a twitter social network with over $80,000$ users. A twitter user P_l follows user P_j indicates that P_j has an influence on P_l. Thus, P_l may join the incentive tree as a child of P_j. We generate a spanning forest of the social network where each user refers all of its un-joined neighbors into the incentive tree. We set the platform as the root of the incentive tree and attach all roots of the spanning forest as the children of the root. If multiple invitations arrive at a user at the same time, we break the ties by choosing the one with the smallest index among the inviters as the parent. E.g., if P_6 receives invitation from P_1 and P_5 at the same time, we add P_6 as P_1's child.

B. Performance Metrics

During the simulation, we study *average user utility*, *total payment*, *running time*, and *dishonest user utility* as the metrics to evaluate **RIT**. We study the impact of the payment determination phase on the performance metrics introduced above, by comparing the results of the auction phase with the final results of **RIT**.

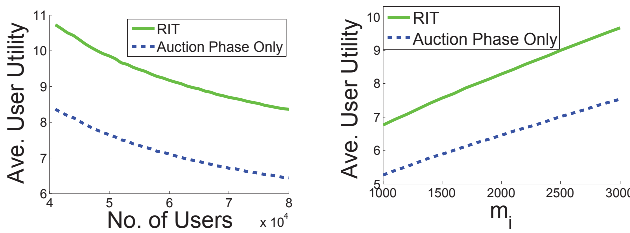
To evaluate the impact of the number of users on average user utility, total payment, and running time, we set $m_i = 5000$ for each τ_i and varied the number of users from 40000 to 80000 with an increment of 1000. The results are presented in Fig. 6(a), Fig. 7(a), and Fig. 8(a), respectively.

To evaluate the impact of the number of tasks on average user utility, total payment, and running time, we fixed $n = 30000$, and varied m_i from 1000 to 3000 with an increment of 100 for each task type τ_i. The results are presented in Fig. 6(b), Fig. 7(b), and Fig. 8(b), respectively.

To evaluate the properties of sybil-proofness and truthfulness, we fixed $n = 10000$ and let m_i be uniformly distributed over $(100, 500]$ for each task type τ_i. We randomly picked a user P_{29} whose auction payment is non-zero when asking its true cost $c_{29} = 5.5$. We also have $K_{29} = 17$, which means that P_{29} cannot generate more than 17 identities. We varied the number of identities of P_{29} from 2 to 17, and let P_{29} randomly generate the identities. We monitored the total utility of P_{29} from its dishonest behavior. Further, we picked three ask values for P_{29}'s identities and monitored the corresponding utilities. The three ask values are $a_{29} = c_{29} = 5.5$, $a_{29} = 6.5$, and $a_{29} = 6.25$. We present the results in Fig. 9.

C. Simulation Results

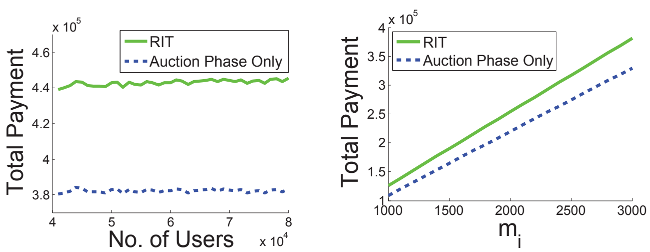
In this subsection, we present our simulation results on **RIT** and provide some analysis on these results.



(a) $m = 5000$ (b) $n = 30000$

Fig. 6. Average user utility

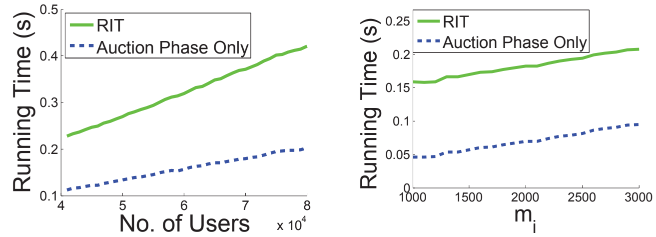
From Fig. 6(a) we observe that with more users, the average user utility decreases. This is because with more users, the competition among the users becomes more fierce and as a result, the average auction payment decreases, which leads to a decrement to the average user utility. In Fig. 6(b), when the number of users are fixed, the average user utility increases with the increment of the job size. This is because with more tasks to be allocated, the average auction payment increases, which leads to the increment of average user utility. Furthermore, by adding the payment determination phase to the auction phase, the average user utility increases since each user gets rewards for its solicitation, which confirms our analysis of solicitation incentive and demonstrates that **RIT** incentivizes users for solicitation.



(a) $m = 5000$ (b) $n = 30000$

Fig. 7. Total payment

From Fig. 7(a) it is observed that the total payment of the platform does not increase remarkably with the increment of the number of users. The reason is that despite the increment of the number of users, the number of allocated tasks is fixed. Besides, with the increment of users, the average auction payment decreases. However, with more users, the payment for solicitation of each user may increase. Combining these factors, the total payment of the platform does not increase remarkably. In Fig. 7(b), the total payment of the platform increases with the job size. This is because with more tasks, the number of winning users increases, and the payments from **RIT** increase as well, which leads to the increment of the total payment. Furthermore, we observe that by applying the payment determination phase to the auction phase, the total payment increases. The increment is no more than the total auction payment, i.e., $\sum_{P_j \in \mathcal{P}}(p_j - p_j^A) \leq \sum_{P_j \in \mathcal{P}} r_j (\frac{1}{2})^{r_j} p_j^A \leq \sum_{P_j \in \mathcal{P}} p_j^A$, where $r_j \geq 1$ for each user P_j. This indicates that by applying the incentive tree based payment determination phase, the platform pays no more than the total auction payment for solicitation.



(a) $m = 5000$ (b) $n = 30000$

Fig. 8. Running time

In Fig. 8(a) and Fig. 8(b), we observe that the running time is increasing in an approximately linear speed with the size of the users and the job size, respectively, which backs up the analysis of computational efficiency in Lemma 3. Furthermore, by applying the payment determination phase, the time complexity of the algorithm still grows linearly.

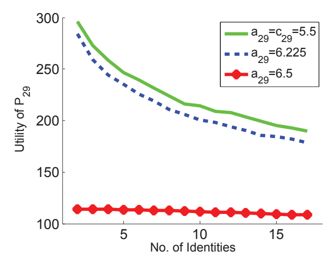


Fig. 9. Utility of P_{29} in sybil attacks

In Fig. 9, we observe that with the increment of the number of identities, the utility of P_{29} decreases, which is a demonstration of the sybil-proofness of **RIT**. Apart from revealing its cost $c_j = 5.5$, we also evaluate the utility of P_{29} when it asks values deviating from its true cost by having $a_{29} = 6.5$ and $a_{29} = 6.225$. We observe that when $a_{29} = c_{29}$, P_{29}'s utility is larger than those of the rest of the ask values, which further confirms the truthfulness of **RIT**.

8. Conclusions

In this paper, we designed a robust crowdsensing incentive mechanism **RIT**, which is an auction-based incentive tree mechanism, to motivate users for participation and solicitation. We proved that **RIT** is truthful and sybil-proof with probability at least H, for any given $H \in (0, 1)$. We also proved that **RIT** satisfies individual rationality, computational efficiency, and solicitation incentive. We implemented **RIT** and the performance evaluation results confirm our analysis.

REFERENCES

[1] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Phys. Commun.*, vol. 4, no. 1, pp. 40–62, 2011.

[2] M. Babaioff, S. Dobzinski, S. Oren, and A. Zohar, "On bitcoin and red balloons," in *ACM EC'12*, pp. 56–73.

[3] W. Chen, Y. Wang, D. Yu, and L. Zhang, "Sybil-proof mechanisms in query incentive networks," in *ACM EC'13*, pp. 197–214.

[4] DARPA Network Challenge, http://archive.darpa.mil/networkchallenge/.

[5] DARPA Shredder Challenge, http://archive.darpa.mil/shredderchallenge/.

[6] J. R. Douceur and T. Moscibroda, "Lottery trees: Motivational deployment of networked systems," in *ACM SIGCOMM'07*, pp. 121–132.

[7] F. A. Drucker and L. K. Fleischer, "Simpler sybil-proof mechanisms for multi-level marketing," in *ACM EC'12*, pp. 441–458.

[8] Y. Emek, R. Karidi, M. Tennenholtz, and A. Zohar, "Mechanisms for multi-level marketing," in *ACM EC'11*, pp. 209–218.

[9] Z. Feng, Y. Zhu, Q. Zhang, L. Ni, and A. Vasilakos, "TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *IEEE INFOCOM'14*, pp. 1231 – 1239.

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

[11] L. Gao, F. Hou, and J. Huang, "Providing long-term participation incentive in participatory sensing," in *IEEE INFOCOM'15*, pp. 2803–2811.

[12] A. V. Goldberg and J. D. Hartline, "Collusion-resistant mechanisms for single-parameter agents," in *ACM-SIAM SODA '05*, pp. 620–629.

[13] ——, "Collusion-resistant mechanisms for single-parameter agents," Microsoft Research - Silicon Valley, Tech. Rep., 2004.

[14] healthmap, http://www.healthmap.org/en/.

[15] J. Howe, *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Crown Publishing Group, 2008.

[16] Q. Huang, Y. Gui, F. Wu, G. Chen, and Q. Zhang, "A general privacy-preserving auction mechanism for secondary spectrum markets," *IEEE/ACM TON'15*, vol. PP, no. 99.

[17] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *ACM MobiHoc'15*, pp. 167–176.

[18] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Inception: Incentivizing privacy-preserving data aggregation for mobile crowd sensing systems," in *ACM MobiHoc'16*, pp. 341–350.

[19] KickStarter, https://www.kickstarter.com/.

[20] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *IEEE INFOCOM'13*, pp. 1402–1410.

[21] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *NIPS'2012*, pp. 548–556.

[22] Q. Li and G. Cao, "Providing efficient privacy-aware incentives for mobile sensing," in *IEEE ICDCS'14*, pp. 208–217.

[23] T. Luo, H. P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in *IEEE INFOCOM'14*, pp. 127–135.

[24] Y. Lv and T. Moscibroda, "Fair and resilient incentive tree mechanisms," in *ACM PODC'13*, pp. 230–239.

[25] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*.

[26] G. Pickard, W. Pan, I. Rahwan, M. Cebrian, R. Crane, A. Madan, and A. Pentland, "Time-critical social mobilization," *Science*, vol. 334, pp. 509–512, 2011.

[27] Pyramid Scheme, http://www.fbi.gov/scams-safety/fraud.

[28] Sickweather, http://www.sickweather.com/.

[29] Tag Challenge, https://www.tag-challenge.com.

[30] G. F. Trend, https://www.google.org/flutrends/about/.

[31] H. R. Varian, "Position auctions," *International Journal of Industrial Organization*, vol. 25, no. 6, pp. 1163–1178, 2007.

[32] J. Wang, J. Tang, D. Yang, E. Wang, and G. Xue, "Quality-aware and fine-grained incentive mechanisms for mobile crowdsensing," in *IEEE ICDCS'16*, pp. 354–363.

[33] Q. Wang, K. Ren, and X. Meng, "When cloud meets ebay: Towards effective pricing for cloud computing," in *IEEE INFOCOM'12*, pp. 936–944.

[34] Y. Wen, J. Shi, Q. Zhang, X. Tian, Z. Huang, H. Yu, Y. Cheng, and X. Shen, "Quality-driven auction-based incentive mechanism for mobile crowd sensing," *IEEE TVT'15*, vol. 64, no. 9, pp. 4203–4214.

[35] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *ACM MOBICOM'12*, pp. 173–184.

[36] Q. Zhang, Y. Wen, X. Tian, X. Gan, and X. Wang, "Incentivize crowd labeling under budget constraint," in *IEEE INFOCOM'15*.

[37] X. Zhang, G. Xue, D. Yang, and R. Yu, "A sybil-proof and time-sensitive incentive tree mechanism for crowdsourcing," in *IEEE GLOBECOM'15*, pp. 1–6.

[38] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Countermeasures against false-name attacks on truthful incentive mechanisms for crowdsourcing," *IEEE JSAC'17*, vol. PP, no. 99, pp. 1–1.

[39] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li, "Free market of crowdsourcing: Incentive mechanism design for mobile sensing," *IEEE TPDS'14*, vol. 25, no. 12, pp. 3190–3200.

[40] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Truthful incentive mechanisms for crowdsourcing," in *IEEE INFOCOM'15*, pp. 2830–2838.

[41] D. Zhao, X.-Y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *IEEE INFOCOM'14*, pp. 1213–1221.

[42] P. Zhou, Y. Zheng, and M. Li, "How long to wait? predicting bus arrival time with mobile phone based participatory sensing," *IEEE TMC14*, vol. 13, no. 6, pp. 1228–1241.

[43] X. Zhou and H. Zheng, "Breaking bidder collusion in large-scale spectrum auctions," in *ACM MobiHoc '10*, pp. 121–130.