Designing Secure and Dependable Mobile Sensing Mechanisms With Revenue Guarantees

Yuan Zhang, He Zhang, Siyuan Tang, and Sheng Zhong

***Abstract*— In many existing incentive-based mobile sensing applications, the sensing job owner runs an auction with the mobile phone users to maximize its purchased sensing resource. We notice that both the mobile phone users and the job owner could behave dishonestly to pursue their own interests. This motivates us to design secure and dependable auction mechanisms that generate the correct, promising output even when both of them could cheat. In particular, in this paper, we consider a general auction in which a buyer, who acts as the auctioneer, purchases the resource under a limited budget from a group of sellers who act as the bidders. Considering bidders' privacy and their limited computing capacity, we construct our mechanisms by integrating the innovative game theoretical techniques, logic deductions, and efficient cryptographic operations. Our mechanisms are not only proved to be strategy-proof against dishonest bidders in the sense that they are incentivized to bid their private types truthfully, but also enable all the bidders to efficiently verify the correctness of the auction's outcome, that is computed by the auctioneer, without revealing their private types to each other. Meanwhile, our mechanisms are proved to have the theoretical guarantee that the auctioneer/buyer's expected revenue (i.e. the amount of service it acquires after the auction) is no less than a certain portion of the optimal revenue that the auctioneer can acquire when it knows all the bidders' types at no cost. Our extensive evaluations show that our mechanisms achieve good performance in terms of the revenue maximization and their efficiency.**

***Index Terms*— Cheating behaviors, privacy-preserving verification, mobile sensing.**

I. Introduction

AUCTIONS have been widely accepted as an efficient way of allocating resource, thus are widely used in nowadays incentive-based applications or systems such as mobile sensing or crowdsourcing [1]–[3], and spectrum selling or renting [4]–[7].

In this paper, we consider a general incentive-based resource allocation system that consists of one buyer who wants to buy resource from a group of sellers who are interested in trading their resource for monetary payments from the buyer. We consider the buyer aims to maximize the total amount of resource that are bought by performing an auction (as the auctioneer) with the sellers (as the bidders), and design practical auction mechanisms to achieve buyer's goal. To make our problem meaningful, we assume the buyer has a limited budget.

As most existing auction mechanisms [1], [2], [8]–[11], we design our mechanism to be strategy-proof against possible cheating strategies of the bidders. In the meantime, we also hope the auction mechanism allows the buyer to get as much resource as possible with its limited budget, thus we require our mechanism to guarantee that the total amount of services purchased by the buyer (a.k.a the buyer's *revenue*) is no less than a certain portion of the optimal amount of service that a buyer could acquire in an omniscient auction (i.e. an auction in which bidders' private types are known to the auctioneer). Although there have been a few works that design auction mechanisms that are strategy-proof and meanwhile optimize buyer's revenues, or utility functions, or social welfare etc., most of them assume the buyer possess the knowledge of sellers' real types or their types' probability distribution. Since we do not make such assumptions, our problem is much more difficult.

Furthermore, we also consider the case that the auctioneer or the buyer could cheat by tampering the outcome of the auction, which has been seldom considered in existing revenue maximizing auction mechanisms, and design our mechanism to be strategy-proof against the auctioneer. Note that this issue can be directly solved by utilizing standard cryptographic techniques such as secure multi-party computation and/or zero-knowledge proof. However, in order to detect a tampered outcome, all users need to jointly solve a complicated optimization problem without revealing their own inputs to each other (since our auction is sealed-bid and these inputs are sellers' private type information). If we directly apply the cryptographic techniques to make such computation secure, the computational cost and the communication cost would be so heavy that most resource-limited applications (e.g. mobile sensing or mobile crowdsourcing applications) could not afford.

Despite the difficulties above, we manage to propose a randomized mechanism that achieves our goals by novelly applying a "bipartition-and-estimating" idea on the fixed-price auctions, and construct a highly efficient verification mechanisms by utilizing behavior analyses, logic deductions, as well as a semantically secure, additively homomorphic

Manuscript received April 25, 2015; revised July 20, 2015 and September 2, 2015; accepted September 2, 2015. Date of publication September 14, 2015; date of current version October 30, 2015. This work was supported by the National Natural Science Foundation of China under Grant 61321491, Grant 61425024, Grant 61300235, and Grant 61402223. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Wanlei Zhou. *(Corresponding author: Sheng Zhong.)*

The authors are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China, and also with the Computer Science and Technology Department, Nanjing University, Nanjing 210023, China (e-mail: zhangyuan05@gmail.com; mattzhang9@gmail.com; siyuantang2013@gmail.com; zhongsheng@nju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2015.2478739

cryptosystem. Specifically, our major contributions can be summarized as follows:

- We study the incentive mechanism design problem for a general resource allocation system, and design an efficient **S**trategy-proof **Auc**tion mechanism with **R**evenue **G**uarantee: *SRG-Auc*.
- We rigorously prove that *SRG-Auc* is strategy-proof against untruthful bidders. This helps the buyer to avoid possible price manipulations by sellers.
- We prove *SRG-Auc*'s *competitive ratio*, i.e. the ratio of the buyer's expected revenue in the auction to the optimal revenue that it could acquire when all bidders' private types are given to the auctioneer at no cost, has a theoretical lower bound under a mild assumption. This allows *SRG-Auc* to have performance guarantees in helping the buyer to get as much resource as possible at most circumstances.
- We propose highly efficient enhancements *SRG-Verify1* and *SRG-Verify2* to *SRG-Auc* to enable sellers to verify the correctness of the auction's outcome, and rigorously prove their correctness. These two mechanisms do not reveal the sellers' private types to each other, and meanwhile help these sellers to thwart the buyer from tampering the auction's outcome without being noticed.
- We perform extensive simulation experiments to evaluate our mechanisms. Experimental results verify the efficiency of our mechanisms and show the auctioneer could acquire a good portion of the optimal revenue.

The rest of this paper is organized as follows. In Section II, we introduce the preliminaries of our work. In Section III, we consider the buyer is truthful and the sellers could cheat, and present *SRG-Auc* and its theoretical analysis. In Section IV, we consider the buyer may also cheat, and present our two verification mechanisms *SRG-Verify1* and *SRG-Verify2*. In Section V, we conduct experiments to evaluate our mechanisms' performance. In Section VI, we discuss three practical attacks or issues, and provide possible solutions to deal with them. Finally, after introducing the related work in Section VII, we conclude our work in Section VIII.

II. Preliminaries

A. System Model and Assumptions

To ease the understanding, we use a general mobile sensing application as a concrete example, and explain our system model and assumptions accordlingly. In this setting, the sensing job owner is the buyer/auctioneer, and a group of mobile phone users who are interested in selling their sensing resource (e.g. sensing time, bandwidth, data quota, etc.) are the sellers/bidders.

Consider a mobile sensing job crowdsourcing system which mainly consists of a group of registered mobile phone users, a group of registered mobile sensing job owners, and a supporting infrastructure. Mobile phone users connect to the supporting infrastructure via their network connections. We assume, with the help of the supporting infrastructure, the job owner and all mobile phone users could communicate in an authenticated manner.

Specifically, we assume a job owner could broadcast its sensing job by "writing" descriptions of its job on a public bulletin board which is maintained by the supporting infrastructure. In addition, we assume a mobile phone user who competes for a job could post a message on a bulletin board that can be seen by all other mobile users who also compete for the same job. We note such supporting infrastructures are provided by some existing crowdsourcing system such as Amazon's Mechanical Turk [12] and CrowdFlower [13].

B. Auction Model and Solution Concepts

Consider a sensing job owner $\mathcal{O}$ who purchases a certain kind of sensing resource from a group of mobile device users. For the ease of presentation, we assume the resource here is users' available time for sensing in the rest of the paper. Denote by $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$ n interested sellers/mobile device users. Each seller U_i $(1 \leq i \leq n)$ has a private *type* $I_i = (c_i, l_i)$ that consists of c_i, which equals the cost per unit of its available sensing time, and l_i which equals the maximum sensing time it is able to provide or sell.

To make our problem meaningful, we assume that buyer $\mathcal{O}$ has a limited budget $R \in \mathbb{R}^+$ to pay for the sellers' sensing time. We aim to design auction mechanisms that allow $\mathcal{O}$ to purchase as much sensing time as possible.

After the auction starts, each seller U_i, as the bidder, submits a two-parameter bid $J_i = (d_i, m_i)$ to the buyer, where $d_i \in \mathbb{R}^+$ equals its claimed unit cost of its sensing time and $m_i \in \mathbb{R}^+$ equals its claimed maximum sensing time.

As the auctioneer, the buyer $\mathcal{O}$ runs a predefined auction mechanism which takes all sellers' bids $\{J_i\}_i$ as inputs and outputs $\{(r_i, t_i)\}_i$, where $r_i \in [0, R]$ equals the amount of payment $\mathcal{O}$ makes to seller U_i and $t_i \in [0, m_i]$ equals the amount of time that U_i is required to sense for $\mathcal{O}$.

As most game theoretical work, each seller U_i is considered to be *rational* [14], and always aims to maximize its own *utility*:

$$u_i = \begin{cases} r_i - c_i t_i & \text{if } t_i \leq l_i, \\ -\infty & \text{otherwise.} \end{cases} \tag{1}$$

Here we assume every seller, no matter honest or not, has a *hard* constraint on its maximum total sensing time. By hardness, we mean a seller would face a very large utility deficit if it is required to sense a period of time that is longer than its real capacity. To make this true, we assume no seller could submit fake data without being noticed (this can be possibly achieved by adopting sophisticated data authenticity verification schemes or requiring each seller to use trusted sensors [15]), and charge a large amount of fine when a user fails to deliver the required amount of authentic sensing data.

The notations used in this paper are summarized in Table I.

To make our auction mechanism practical, we require it possesses the following fundamental characteristics:

- *Computationally Efficient*: The auction has a polynomial time complexity.
- *Individual Rational*: All bidders' utilities are non-negative when they participate in the auction and submit their types truthfully as their bids.

TABLE I
NOTATION

Symbol	Description
O	the buyer
U_i	the ith seller
c_i	the cost per unit of U_i's sensing time
l_i	the maximum sensing time of U_i
d_i	the claimed unit cost of U_i's sensing time
m_i	the claimed maximum sensing time of U_i
J_i	the bid U_i submits, $J_i = (d_i, m_i)$
R	buyer $\mathcal{O}$'s limited budget
r_i	the payment to U_i after auction
t_i	the sensing time that is required from U_i after auction
u_i	the utility U_i gets from the auction
n	the count of sellers participating in the auctions
S_i	the ith group of sellers
$p^\star$	the clearing price for all sellers in S_1
k	the count of sellers in S_0
α	the parameter controlling the estimation

- *Budget Balanced*: The total amount of payment to all sellers should be no greater than the buyer's budget.

In addition, we are interested in making the auction mechanism

- *Strategy-proof against the bidders*: No bidder (seller) could improve its utility by submitting a bid that deviates from its true type, no matter how other bidders bid.

Furthermore, we consider the buyer could tamper the outcome of the auction, thus we require that the auction mechanism is

- *Strategy-proof against the auctioneer*: The auctioneer (buyer) could not tamper the true outcome of the auction without being noticed by the bidders.

Finally, we aim to guarantee that the job owner could acquire as much sensing time as possible in our auction mechanism. We analyze an auction mechanism's performance regarding this goal by applying the competitive analysis, and require our auction

- ϵ-*competitive*: The expected total sensing time acquired by the buyer in the auction is no less than ϵT^*, where $\epsilon \in (0, 1]$ is known as the *competitive ratio* and T^* equals the maximum amount of sensing time that the buyer could purchase in an omniscient, single-price auction.[1]

C. The Semantically Secure, Additively-Homomorphic Cryptosystem

In this paper, we will use a cryptosystem that is *additively-homomorphic* [17], and *semantically secure* [18] to protect sellers' bids and to perform verifications on the correctness of the auction's outcome. When a cryptosystem is additively-homomorphic, it means there is an efficient algorithm that takes the ciphertexts of a group of plaintexts as inputs, outputs the ciphertext of the plaintexts' sum without performing any decryption. When a cryptosystem is semantically secure, it basically means no one could succeed in differentiating a ciphertext of a known plaintext x and a uniformly random number of the same length with a non-negligible probability. There are a few existing cryptosystems that are both semantically secure and additively homomorphic, e.g. the Paillier cryptosystem [17] and the BGN cryptosystem [19].

Formally, taking the BGN cryptosystem as an example, the additively-homomorphic, semantically secure cryptosystem can be defined as $\mathcal{E} = \{KeyGen, Enc, Dec, CipAdd\}$ that consists of the following four efficient algorithms:

- $KeyGen(\cdot)$: a randomized function that takes a security parameter $L \in \mathbb{N}^+$ and returns a key pair $(priv, pub)$.
- $Enc_{pub}(\cdot, \cdot)$: a function that maps the public key pub, a plaintext $x \in \mathcal{X}$ and a seed $r \in \mathcal{R}$ to a ciphertext $\overline{x}$, where $\mathcal{X}$ and $\mathcal{R}$ are the domains of the plaintext and the random seeds that are usually determined by the security parameter.
- $Dec_{priv}(\cdot)$: a function that maps the private key $priv$, and the ciphertext $\overline{x} = Enc_{pub}(x, r)$ to its corresponding plaintext x.
- $CipAdd_{pub}()$: a function that maps pub and a group of ciphertexts $Enc_{pub}(x_1, r_1), \ldots, Enc_{pub}(x_n, r_n)$ to $Enc_{pub}(x_1 + \ldots + x_n, r_1 + \ldots + r_n)$.

III. *SRG-Auc*: A Strategy-Proof Auction Mechanism With Revenue Guarantees

In this section, we present *SRG-Auc*, an efficient auction mechanism that is strategy-proof against the bidders, and also has theoretical revenue guarantees.

A. Designing Rationale

As introduced in the Introduction section, the main idea of our mechanism is to adopt a bipartitioning-and-estimating procedure.

In more detail, *SRG-Auc* randomly partitions all sellers into two groups: a large group in which a fixed-price auction is conducted, and one small group based on which the estimation is performed. The fixed-price auction adopts a clearing price that is determined only by bids of the sellers from the small group. With this setting, the sellers who participate in the fixed-price auction cannot manipulate the clearing price, thus *SRG-Auc* achieves strategy-proofness. Meanwhile, to achieve a good revenue or competitive ratio, the proper clearing price is estimated from the sellers in the small group under the intuition that distributions of sellers' types in the large group and in the smaller one should be similar since they are randomly sampled from the entire seller space.

B. Solving the Optimal Omniscient Auction

Before we introduce our auction mechanism, we first show how to solve the optimal omniscient auction. We will use the results we get in this section when we construct *SRG-Auc*, and when we evaluate it.

In particular, we are interested in the maximum revenue that the buyer can acquire in an omniscient, single-price auction, given all sellers' claimed unit-costs $d = (d_1, d_2, \ldots, d_n)$,

[1] In single-price auctions, we have $pt_i = r_i$ for every seller U_i. It has been proved in [16] that the maximum amount of sensing time that the buyer can acquire in an omniscient, multi-price auction is no greater than twice of it in an omniscient, single-price auction. Therefore, it suffices to analyze the buyer's revenue in our auction by comparing it to the revenue in the omniscient, single-price auctions.

claimed maximum sensing time constraints $m = (m_1, m_2, \ldots, m_n)$, and an arbitrary budget of the buyer R ($R \in \mathbb{R}^+$). Note that in the omniscient auction, $d = c = (c_1, \ldots, c_n)$ and $m = l = (l_1, \ldots, l_n)$.

Define by $T(d, m, R)$ the function that returns the corresponding maximum revenue, i.e.,

$$T(d, m, R) = \max\ t \quad \text{s. t.} \sum_{d_i \le R/t} m_i \ge t. \tag{2}$$

For the ease of presentation, we assume $d_1, \ldots, d_n$ has been sorted in a non-decreasing order. It is not difficult to verify that $T(d, m, R)$ can be computed as follows.

Algorithm 1 $T(d, m, R)$

1: Compute $i^* = \arg\max_i (d_i \sum_{j=1}^{i-1} m_j \le R)$.
2: Output $T(d, m, R) = \min(R/d_{i^*}, \sum_{j=1}^{i^*} m_j)$.

In addition, we define the "optimal" price at which the maximum total sensing time is achieved by $F(d, m, R)$. It is easy to see

$$F(d, m, R) = R/T(d, m, R). \tag{3}$$

C. Dealing With Untruthful Bidders

In this section, we assume the job owner or the auctioneer is honest, i.e. always runs the predefined auction mechanism without any deviation, and construct *SRG-Auc* that is able to thwart bidders' cheating behaviors and has theoretical revenue guarantees.

1) Bipartitioning the Sellers: SRG-Auc separates all sellers into two groups randomly, and determines the auction's outcome regarding sellers in one group based on the bids of sellers in the other group.

Note that the randomness introduced in this step makes the partitioning independent from all sellers' bids. Therefore, no seller can adjust his bid to increase the chance that it is assigned to one particular group. In addition, this also makes the distributions of sellers' types in two groups are similar to each other, therefore estimating promising clearing price for one group from the types or bids in the other group is possible.

In particular, *SRG-Auc* first partitions all sellers randomly into two sets: $\mathcal{U}_{S_0}$ with k sellers ($k \in \{1, 2, \ldots, \lfloor n/2 \rfloor\}$) whose index set is S_0, and $\mathcal{U}_{S_1}$ with $n - k$ sellers whose index set is S_1. Here, k is one of the two parameters in *SRG-Auc* that can be adjusted to achieve different revenue guarantees.

For an arbitrary index set S ($S \subseteq \{1, 2, \ldots, N\}$), denote by d^S and m^S the claimed unit costs, and claimed maximum sensing time of sellers whose index are in S respectively. Given S and an arbitrary budget R, define

$$f(S, R) = F(d^S, m^S, R). \tag{4}$$

2) Estimating a Promising Clearing Price: Next, *SRG-Auc* computes a clearing price for all sellers in S_1 based on the bids of sellers in S_0 as:

$$p^* = f(S_0, R\alpha k/n). \tag{5}$$

Here $\alpha \ge 1$ is introduced to limit the chance of *SRG-Auc* being cancelled or failed. A greater α would let *SRG-Auc* use a greater budget to estimate, thus gets a greater clearing price. With the clearing price increased, the chance that *SRG-Auc* uses up the assigned budget would increase. Accordingly, the chance that the *SRG-Auc* is cancelled or failed would decrease.

3) Performing the Auction: Finally, *SRG-Auc* performs a sub-auction $FPA_{(p,r)}$ on all sellers in S_1 with $p = p^*$ and $r = R(n - k)/n$. Here $FPA_{(p,r)}$, specified by Algorithm 2, is the fix-price-auction that uses a budget r to purchase sensing time from sellers at a predefined price p.

Note that the fixed-price auction makes purchases from all winning sellers (i.e. ones whose claimed unit-cost is no greater than the clearing price) one by one in a random order. When a seller's position is near the tail of the sorted sequence, it might win nothing since the budget may have been used up on sellers in front of it. To avoid sellers from manipulating this order by adjusting their bids, we choose to sort sellers randomly.

Algorithm 2 $FPA_{(p,r)}$(S)

1: Let $\mathcal{U}_S = \{U_i\}_{i \in S}$.
2: Sort all sellers in $\mathcal{U}_S$ randomly.
3: **for** each sorted seller U_i **do**
4: **if** U_i's claimed unit-cost $d_i \le p$ **and** any budget remains i.e. $r > 0$ **then**
5: Owner O pays U_i with r_i that equals $p \times m_i$ or all remaining budget r, whichever is smaller.
6: U_i senses for O at the rate p: $t_i = r_i/p$.
7: O updates r with $r - r_i$
8: **end if**
9: **end for**
10: **if** any budget still remains i.e. $r > 0$ **then**
11: the auction is cancelled and all sellers in U lose.
12: **end if**

In more detail, we summarize *SRG-Auc* with its two controlling parameters k ($k \in \{1, 2, \ldots, \lfloor n/2 \rfloor\}$) and α ($\alpha \ge 1$) in Algorithm 3.

Algorithm 3 $SRG\text{-}Auc_{(k,\alpha)}$

1: Buyer O partitions all sellers randomly into two sets: $\mathcal{U}_{S_0}$ of k sellers whose index set is S_0 and $\mathcal{U}_{S_1}$ of $n - k$ sellers whose index set is S_1.
2: O computes a cutting price based on the bids of sellers in $\mathcal{U}_{S_0}$: $p^* = f(S_0, R\alpha k/n)$.
3: O lets all sellers in $\mathcal{U}_{S_0}$ lose the auction.
4: O runs a fix-price auction $FPA_{(p^*, R(n-k)/n)}$ on all sellers in $\mathcal{U}_{S_1}$.

D. Theoretical Analysis

First, we prove *SRG-Auc* is strategy-proof against the bidders.

Theorem 1: $SRG\text{-}Auc_{(k,\alpha)}$ is strategy-proof against the bidders.

Proof: According to Algorithms 3 and 2, a seller's utility is determined by the following three steps: 1) It is

assigned to one of the two sets; 2) Depending on the set it is assigned to, the seller's payment is calculated accordingly; 3) Depending on the final remaining budget, its payment is finalized or cancelled. Here we prove a seller cannot fake its bid to make the outcomes of these steps in favor of itself.

Since all sellers are partitioned randomly, it is straightforward to see changing its bid would not change the outcome of step 1).

Now we focus on step 2). We show when a seller fakes its type, its utility would either not change at all, or become negative after winning the auction, or decrease its utility.

Specifically, consider an arbitrary seller i.

When it is assigned to set S_0, its payment is always 0 no matter how it changes its bid.

When it is assigned to set S_1, its payment is calculated using equation 1, thus is determined by the clearing price p^*, its bidding price/cost d_i, its bidding maximum sensing time m_i, its real price/cost c_i, and the remaining budget when this user's turn arrives $R_{remaining}$ as below:

$$u_i = \begin{cases} (p^* - c_i) \cdot t_i & t_i \le l_i \text{ and } d_i \le p^* \\ -\infty & t_i > l_i \text{ and } d_i \le p^* \\ 0 & d_i > p^* \end{cases} \tag{6}$$

where $t_i = \min\{R_{remaining}/p^*, m_i\}$.

- In the case that seller i's cost is greater than the clearing price, it loses the auction and gets a zero utility when it truthfully bids. However, if it fakes its type, either it wins the auction by bidding $d_i \le p^*$ but decreases its utility (it gets a negative utility since $p^* < c_i$.), or it still loses, thus its utility does not change.
- In the case that seller i's cost is no greater than the clearing price, depending on the remaining budget $R_{remaining}$, by truthfully bidding, it would either get a non-negative payment if $R_{remaining} > 0$, or a zero utility otherwise.

  In the later occasion, changing its bid would make it lose the auction (by submitting $d_i > p^*$), or does not change anything (when its changes m_i), both resulting in a zero utility.

  In the first occasion, changing its bid would make it lose the auction (similarly by submitting $d_i > p^*$), or sells a non-increased amount of sensing time by submitting $m_i < l_i$ and gets a non-increased amount of payment, or sells an amount of sensing time that is beyond its capability and gets a negative utility.

Finally, consider step 3). If the auction is cancelled when a seller truthfully bids, this seller can change this outcome only if it wins in step 2) and it bids a greater $m_i > l_i$ to use up the budget. However, this cause user i to get a negative utility according to (6). □

Next, we study *SRG-Auc*$_{(k,\alpha)}$'s theoretical revenue guarantees. Since we have proved that our mechanism is strategy-proof against all bidders, we can directly analyze its revenue based on bidders' true types.

Lemma 2: The probability that SRG-Auc$_{(k,\alpha)}$ *fails has an upper bound of* $P(n,k,\alpha)$ *where,*

$$P(n,k,\alpha) = \max_{l} \sum_{q=\lfloor k\epsilon/(n-k+\alpha k)l \rfloor + 1}^{k} \binom{l}{q}\binom{n-l}{k-q} \Big/ \binom{n}{k}. \tag{7}$$

Proof: According to the algorithm, *SRG-Auc*$_{(k,\alpha)}$ fails (i.e. all sellers lose in the auction) if and only if $FPA_{(p^*,R(n-k)/n)}$, that runs on all sellers in $\mathcal{U}_{S_1}$ fails with the cutting price determined based on all sellers in $\mathcal{U}_{S_0}$. This is equivalent to some budget remains in the end of the fixed-price auction run on $\mathcal{U}_{S_1}$.

$$\begin{aligned} &Pr[\textit{SRG-Auc}_{(k,\alpha)} \text{ fails on } \mathcal{U}] && (8)\\ &= Pr[f(S_0, R\alpha k/n) = p^* \text{ and } FPA_{(p^*,R(n-k)/n)} \\ &\qquad \text{fails on } \mathcal{U}_{S_1}] && (9)\\ &= Pr[f(S_0, R\alpha k/n) = p^* \text{ and} \\ &\qquad p^* \sum_{i\in S_1; d_i \le p^*} m_i < R(n-k)/n] && (10) \end{aligned}$$

According to (4), we know $f(S_0, R\alpha k/n) = p^*$ implies

$$p^* \sum_{i\in S_0; d_i \le p^*} m_i \ge R\alpha k/n \tag{11}$$

Thus, we have

$$\begin{aligned} &Pr[\textit{SRG-Auc}_{(k,\alpha)} \text{ fails on } \mathcal{U}] && (12)\\ &\le Pr[p^* \sum_{i\in S_0; d_i \le p^*} m_i \ge R\alpha k/n \text{ and} \\ &\qquad p^* \sum_{i\in S_1; d_i \le p^*} m_i < R(n-k)/n] && (13)\\ &\le Pr[p^* \sum_{i\in S_0; d_i \le p^*} m_i \ge R\alpha k/n && (14)\\ &\qquad > \alpha k/(n-k) p^* \sum_{i\in S_1; d_i \le p^*} m_i] && (15)\\ &\le Pr[\sum_{i\in S_0; d_i \le p^*} m_i > \alpha k/(n-k) \sum_{i\in S_1; d_i \le p^*} m_i] && (16)\\ &\le Pr[(1+\alpha k/(n-k)) \sum_{i\in S_0; d_i \le p^*} m_i && (17)\\ &\qquad > \alpha k/(n-k) \sum_{i\in S; d_i \le p^*} m_i] && (18)\\ &\le Pr[\sum_{i\in S_0; d_i \le p^*} m_i && (19)\\ &\qquad > \alpha k/(n-k+\alpha k) \sum_{i\in S; d_i \le p^*} m_i]. && (20) \end{aligned}$$

Denote by $cnt(p,S)$ a counting function that returns the number of sellers in $\mathcal{U}_S$ whose claimed unit-cost is no greater than p. It is easy to see

$$\sum_{i\in S_0; d_i \le p^*} m_i \le cnt(p^*, S_0) m_{max} \tag{21}$$

and

$$\sum_{i\in S; d_i \le p^*} m_i \ge cnt(p^*, S) m_{min}. \tag{22}$$

Therefore, we know

$$\begin{aligned} &Pr[\textit{SRG-Auc}_{(k,\alpha)} \text{ fails on } \mathcal{U}] && (23)\\ &\le Pr[cnt(p^*, S_0)m_{max} && (24)\\ &\qquad > k/(n-k+\alpha k)cnt(p^*, S)m_{min}] && (25)\\ &\le Pr[cnt(p^*, S_0) > k\epsilon/(n-k+\alpha k)cnt(p^*, S)]. && (26)\end{aligned}$$

Denote by P the range of p^*, by $x(p)$ the value of $Pr[cnt(p, S_0) > k\epsilon/(n-k+\alpha k)cnt(p, S)$. We have

$$\begin{aligned} &Pr[cnt(p^*, S_0) > k\epsilon/(n-k+\alpha k)cnt(p^*, S)] && (27)\\ &\quad = \sum_{p\in P} Pr[p^* = p]x(p) && (28)\\ &\quad \le \max_{p\in P} x(p). && (29)\end{aligned}$$

When $p \in [d_l, d_{(l+1)})$ for $l = 1, ...n$ (let $d_{n+1} = +\infty$), it is easy to know

$$cnt(p^*, S) = l, \tag{30}$$

thus we have

$$\begin{aligned} &x(p) && (31)\\ &\quad = Pr[cnt(p, S_0) > k\epsilon/(n-k+\alpha k)l] && (32)\\ &\quad = \sum_{q=\lfloor k\epsilon/(n-k+\alpha k)l\rfloor+1}^{k} Pr[cnt(p, S_0) = q] && (33)\\ &\quad = \sum_{q=\lfloor k\epsilon/(n-k+\alpha k)l\rfloor+1}^{k} \binom{l}{q}\binom{n-l}{k-q}/\binom{n}{k}. && (34)\end{aligned}$$

According to (29) and (34), we know

$$\begin{aligned} &Pr[cnt(p^*, S_0) > k\epsilon/(n-k+\alpha k)cnt(p^*, S)] && (35)\\ &\quad \le \max_{l=1,...,n} \sum_{q=\lfloor k\epsilon/(n-k+\alpha k)l\rfloor+1}^{k} \binom{l}{q}\binom{n-l}{k-q}/\binom{n}{k} && (36)\\ &\quad = P(n, k, \alpha). && (37)\end{aligned}$$

□

Theorem 3: Assuming $T(d^{S_0}, m^{S_0}, R\alpha k/n) \ge k/nT(d, m, R)$, *we know SRG-Auc*$_{(k,\alpha)}$*'s competitive ratio is no less than* $\frac{(1-P(n,k,\alpha))(n-k)}{n\alpha}$ *where* $P(n, k, \alpha) = \max_l \sum_{q=\lfloor k\epsilon/(n-k+\alpha k)l\rfloor+1}^{k} \binom{l}{q}\binom{n-l}{k-q}/\binom{n}{k}$.

Proof: Denote by T and T^* the total amount of sensing time that the buyer acquires in *SRG-Auc*$_{(k,\alpha)}$, and in the optimal omniscient auction respectively.

According to Section III-C, we have

$$T = \begin{cases} R(n-k)/(np^*) & \text{when } \textit{SRG-Auc}_{(k,\alpha)} \text{ on } S_0,\\ 0 & \text{otherwise,}\end{cases} \tag{38}$$

where $p^* = R\alpha k/(nT(d^{S_0}, m^{S_0}, R\alpha k/n))$. Therefore, we know

$$E(T) = (\sum_{S_0:\textit{SRG-Auc}_{(k,\alpha)} \text{ succeeds on } S_0} \frac{R(n-k)}{(np^*)})/|\{S_0\}|. \tag{39}$$

Assuming $T(d^{S_0}, m^{S_0}, R\alpha k/n) \ge k/nT(d, m, R)$, we immediately get

$$\begin{aligned} p^* &= (R\alpha k/n)/T(d^{S_0}, m^{S_0}, R\alpha k/n) && (40)\\ &\le (R\alpha)/T(d, m, R), && (41)\end{aligned}$$

and

$$\begin{aligned} R(n-k)/(np^*) &\ge \frac{R(n-k)}{n(R\alpha)/T(d, m, R)} && (42)\\ &= (n-k)T(d, m, R)/(n\alpha). && (43)\end{aligned}$$

Therefore,

$$\begin{aligned} &E(T) && \\ &\ge \frac{|\{S_0 : \textit{SRG-Auc}_{(k,\alpha)} \text{ succeeds on } S_0\}|(n-k)T(d, m, R)}{|\{S_0\}|n\alpha} && (44)\\ &= (1 - Pr[\textit{SRG-Auc}_{(k,\alpha)} \text{ fails}])\frac{(n-k)T(d, m, R)}{n\alpha} && (45)\\ &\ge (1 - P(n, k, \alpha))\frac{(n-k)T(d, m, R)}{n\alpha}. && (46)\end{aligned}$$

Note that

$$T^* = T(d, m, r), \tag{47}$$

thus the competitive ratio of *SRG-Auc*$_{(k,\alpha)}$

$$E(T)/T^* \ge \frac{(1 - P(n, k, \alpha))(n-k)}{n\alpha}. \tag{48}$$

□

IV. *SRG-Ver*: An Enhancement for Dealing With Dishonest Auctioneers

In this section, we propose *SRG-Ver*, an efficient enhancement to our auction mechanism *SRG-Auc* that deals with dishonest auctioneers.

A. To Enable Sellers to Verify the Outcome Without Revealing Their Bids to Each Other

Note that the buyer and the sellers are running a sealed-bid auction in *SRG-Auc*. Accordingly, only the buyer, who is the auctioneer, knows all sellers' bids and the correct outcome of the auction. If the buyer tampers the output of the auction to increase its own benefit, sellers cannot discover the buyer's cheating action. To deal with this issue, we aim to design an efficient protocol that allows sellers to verify the correctness of the outcome. In the meantime, considering sellers' bids contain sellers' private type information, we hope the protocol does not reveal any seller's bid to any other seller.

It is known that our problem here can be solved by letting the buyer generate zero-knowledge proofs [20] on the correctness of the outcome, or by letting all sellers jointly compute the outcome using a general-purpose secure multi-party computation protocol [21], [22]. However, both solutions incur heavy computation cosst and/or communication costs, which makes them hardly applicable in mobile sensing systems.

To propose efficient solution to our problem, we first analyze all theoretically possible cheating behaviors of the buyer, filter out those cheating behaviors that would never be adopted

by a rational buyer, and focus on dealing with the rest ones. Although the analysis helps to reduce the complexity of our problem, we find it is still difficult to detect the rest cheating behaviors since, in order to verify the outcome is correct, sellers need to jointly solve an optimization problem which involves computing a complicated function without revealing the function's inputs. To overcome this difficulty, we perform logic reductions to transform our original verification problem into two equivalent verification problems. These two verification problems only require the sellers to jointly compute a conditional sum and verify a inequality of the (conditional) sum. Compared with the complicated optimization function, the conditional sum function is much easier for sellers to jointly compute. Finally, to make sure the computation do not reveal sellers' private bids, we utilize an additively-homomorphic, semantically secure cryptosystem to secure the computation process. Utilizing all the techniques above, we manage to construct a highly efficient verification mechanism that allows the sellers to verify the correctness of the outcome provided by the buyer without revealing their private bids to each other.

B. The Buyer's Three Kinds of Cheating Actions

According to *SRG-Auc*, the buyer could perform the following cheating actions:

- a1 To partition all sellers according to their bids instead of following a uniform distribution.
- a2 To perform the FPA auction with an incorrect input p', instead of p^*.
- a3 To stop executing the auction procedure before the auction is completed.

We assume the buyer would not sabotage auction's strategy-proofness against bidders under all circumstances. Otherwise, sellers could bid arbitrarily high unit-costs which causes the buyer to waste most of its revenue. It is easy to see action a1 would allow sellers to adjust their bids to avoid being partitioned into the smaller group, thus breaks the incentive-proofness of the *SRG-Auc* auction (Recall all sellers in the smaller group lose the auction). Therefore, we can neglect action a1.

In addition, cheating behavior a3 can be easily noticed by all sellers if they do not receive the outcome of the auction from the buyer within a certain amount of time. Therefore, we can also neglect action a3.

Therefore, we only need to focus on dealing with cheating action a2. According to equations (2), (3), (4) and (5), the correct clearing price p^* should be computed as:

$$p^* = R\alpha k/(nT(d^{S_0}, m^{S_0}, R\alpha k/n)). \tag{49}$$

Denote by p' the clearing price that is announced public by the buyer who claims that the price is computed following the *SRG-Auc* auction without any deviation. To prevent the buyer from performing action a2 covertly, the sellers need to be able to verify

$$p' = p^* \tag{50}$$

is true or not.

C. Using Logic Reductions to Simplify Detections of Cheating Action a2

Although p^* can be easily computed by the buyer who possesses all sellers' bids, it is difficult to be computed by the sellers since $T(d, m, r)$ is a rather complicated optimization function and its inputs consist of a group of sellers' bids that are supposed to be kept private. Therefore, it is difficult for the sellers to jointly compute p^*, and verify (50)'s correctness directly to thwart the cheating action a2.

To deal with this problem, we apply logic deductions to transform the verification of (50) to two verifications that only require the sellers to jointly compute a few conditional sum functions. Compared with the optimization function, it is much easier for sellers to jointly compute these conditional sum functions. (We will demonstrate how to compute them shortly in the next subsection.)

Same as most Logic literatures, we denote by "$\neg$", "$\Rightarrow$", "$\Leftrightarrow$", "$\wedge$" and "$\vee$" the logic relations of "negation", "implication", "equivalence", "logic conjunction" and "logic disjunction". And we add angle brackets "$\langle$" and "$\rangle$" to the beginning and the end of a statement to represent a predicate. It is easy to see that the following proposition holds.

Proposition 4: $\langle p' = p^* \rangle \Leftrightarrow \langle p' \geq p^* \rangle \wedge \langle p' \leq p^* \rangle$.

Lemma 5: To verify if $p' \geq p^*$ is true, it is equivalent to verify if

$$\sum_{d_i \leq p'} m_i \geq R\alpha k/(np') \tag{51}$$

is true.

Proof: According to (49) and (2), the definitions of p^* and $T(d, m, r)$, we know:

$$\begin{aligned}
&\langle p' \geq p^* \rangle && (52)\\
&\quad \Leftrightarrow \neg\langle R\alpha k/(np') > T(d^{S_0}, m^{S_0}, R\alpha k/n) \rangle && (53)\\
&\quad \Leftrightarrow \neg\langle \sum_{d_i \leq p'} m_i < R\alpha k/(np') \rangle && (54)\\
&\quad \Leftrightarrow \langle \sum_{d_i \leq p'} m_i \geq R\alpha k/(np') \rangle. && (55)
\end{aligned}$$

Remark: (53) and (54) are equivalent is due to the following reasons. Since $T = R\alpha k/(np^*)$ is the maximum t that satisfies the constrained sum inequation in (2), it immediately follows that, the constrained sum inequation dose not hold for $t = R\alpha k/(np')$ which corresponds to (54), when (53) holds, and vice versa. □

Lemma 6: To verify if $p' \leq p^*$ is true, it is equivalent to verify if

$$\sum_{d_i \leq R\alpha k/(nT')} m_i < T' \tag{56}$$

holds for every $\delta > 0$, where $T' = R\alpha k/(np') + \delta$.

Proof:

$$\begin{aligned}
&\langle p' \leq p^* \rangle && (57)\\
&\quad \Leftrightarrow \neg\langle R\alpha k/(np') < T(d^{S_0}, m^{S_0}, R\alpha k/n) \rangle && (58)\\
&\quad \Leftrightarrow \neg\langle \exists \delta > 0, R\alpha k/(np') + \delta \leq T(d^{S_0}, m^{S_0}, R\alpha k/n) \rangle && (59)\\
&\quad \Leftrightarrow \langle \forall \delta > 0, R\alpha k/(np') + \delta > T(d^{S_0}, m^{S_0}, R\alpha k/n) \rangle && (60)\\
&\quad \Leftrightarrow \langle \forall \delta > 0, T' = R\alpha k/(np') + \delta, \sum_{d_i \leq R\alpha k/(nT')} m_i < T' \rangle. && (61)
\end{aligned}$$

Remark: (60) and (61) are equivalent is due to the following reasons. Since $T = R\alpha k/(np^*)$ is the maximum t that satisfies the constrained sum inequation in (2), it immediately follows that, the constrained sum inequation dose not hold for $T' = R\alpha k/(np') + \delta$ which corresponds to (61), when (60) holds, and vice versa. □

Instead of testing every $\delta > 0$ (which is actually impossible due to the fact that there are infinite such δs), we show that it suffices to verify if the inequality holds for one particular value in our problem. Specifically, we have.

Proposition 7: To verify if $p' \leq p^*$, it is equivalent to verify if

$$\sum_{d_i \leq R\alpha k/(nT')} m_i < T' \tag{62}$$

holds for $\delta^* = 1/(AnPp'D!)$, where $T' = R\alpha k/(np') + \delta^*$, $A, P \in \mathbb{N}^+$ are the smallest integers that make $A\alpha$ and $Pp\prime$ integers respectively, and $D! = D \times (D-1) \times \ldots \times 1$.

Proof: Let $\Delta = AnPp'D!$. According to Proposition 6, we know

$$\langle p' \leq p^* \rangle \tag{63}$$
$$\Leftrightarrow \langle \forall \delta > 0, R\alpha k/(np') + \delta > T(d^{S_0}, m^{S_0}, R\alpha k/n) \rangle \tag{64}$$
$$\Leftrightarrow \langle \forall \delta > 0, \Delta \times (R\alpha k/(np') + \delta) > \Delta \times T(d^{S_0}, m^{S_0}, R\alpha k/n) \rangle \tag{65}$$
$$\Leftrightarrow \langle \forall \delta > 0, \Delta \times R\alpha k/(np') + \Delta \times \delta > \Delta \times T(d^{S_0}, m^{S_0}, R\alpha k/n) \rangle \tag{66}$$

Notice that Δ is an integer. Also it is easy to verify

$$\Delta \times R\alpha k/(np') = R(A\alpha)PD! \tag{67}$$

is also an integer given the definitions of A, P, and D. Furthermore, according to the definition of function $T(d, m, r)$ in Algorithm 1, we know

$$T(d^{S_0}, m^{S_0}, R\alpha k/n) = R\alpha k/(nd_i) \tag{68}$$

for some $i \in S_0$ or

$$T(d^{S_0}, m^{S_0}, R\alpha k/n) = \sum_{j \in S^0; j < i} m_j. \tag{69}$$

In either case, it is easy to verify $\Delta \times T(d^{S_0}, m^{S_0}, R\alpha k/n)$ is an integer.

Now, given both $\Delta \times R\alpha k/(np')$ and $\Delta \times T(d^{S_0}, m^{S_0}, R\alpha k/n)$ are integers, it is easy to verify the following is true.

$$\langle \forall \delta > 0, \Delta \times R\alpha k/(np') + \Delta \times \delta > \Delta \times T(d^{S_0}, m^{S_0}, R\alpha k/n) \rangle \tag{70}$$
$$\Leftrightarrow \langle \Delta \times R\alpha k/(np') + 1 > \Delta \times T(d^{S_0}, m^{S_0}, R\alpha k/n) \rangle \tag{71}$$
$$\Leftrightarrow \langle R\alpha k/(np') + \delta^* > T(d^{S_0}, m^{S_0}, R\alpha k/n) \rangle \tag{72}$$
$$\Leftrightarrow \langle T' = R\alpha k/(np') + \delta^*, \sum_{d_i \leq R\alpha k/(nT')} m_i < T' \rangle \tag{73}$$

□

Based on results above, we immediately have the following theorem regarding the verification of (50).

Theorem 8: To verify if $p' = p^*$ is true, it is equivalent to verify if

$$\sum_{d_i \leq p'} m_i \geq R\alpha k/(np'), \tag{74}$$

and

$$\sum_{d_i \leq R\alpha k/(nT')} m_i < T' \tag{75}$$

both hold, where $T' = R\alpha k/(np') + \delta^*$, $\delta^* = 1/(AnPp'D!)$, and $A, P \in \mathbb{N}^+$ are the smallest integers that make $A\alpha$ and $Pp\prime$ integers respectively.

D. Efficiently Performing Verifications Without Revealing Sellers' Bids

Notice that the verifications of (74) and (75) require sellers to compute conditional sum functions. In addition, the computation should be done without revealing the elements within the sum. Based on the specifics of our problem and particular scenario, we propose efficient solutions based on homomorphic encryptions. For the ease of presentation, we assume all sellers' claimed unit costs and claimed maximum sensing time are both integers in this section.[2] Assume $D \in \mathbb{Z}^+$ is the upper bound of all sellers' claimed unit cost, i.e., $d_i \in \{1, 2, \ldots, D\}$. In addition, we assume the buyer and all sellers have agreed on a semantical secure, additively-homomorphic cryptosystem $\mathcal{E} = \{KeyGen, Enc, Dec, CipAdd\}$.

Before the verifications, the buyer O generates a random key pair, keeps the private key private, and publishes the public key to all sellers:

$$O : (priv, pub, R) \leftarrow KeyGen(L); \tag{76}$$
$$O : pub \dashrightarrow \mathcal{U}_S. \tag{77}$$

1) Verifying the First Inequality: To verify (74), each seller in S_0 generates its input to conditional sum function based on the condition specified in (74), encrypts it and sends the ciphertext and the random seed used in the encryption to the buyer:

$$\forall i \in S_0, \quad U_i : r_i \xleftarrow{\$} R, \overline{x_i} \leftarrow Enc_{puk}(x_i, r_i); \tag{78}$$
$$\forall i \in S_0, \quad U_i : (\overline{x_i}, r_i) \dashrightarrow O, \tag{79}$$

where

$$x_i = \begin{cases} m_i & \text{if } d_i \leq p'; \\ 0 & otherwise. \end{cases} \tag{80}$$

Then buyer computes the conditional sum by performing ciphertext-additions and one decryption, then publishes all ciphertexts it received, the conditional sum, and the sum of the seeds that are received with the ciphertexts:

$$O : \overline{x} \leftarrow CipAdd(\{\overline{x_i}\}_{i \in S_0}), \quad x \leftarrow Dec_{priv}(\overline{x}), \; r \leftarrow \sum_{i \in S_0} r_i; \tag{81}$$
$$O : \{\overline{x_i}\}_{i \in S_0}, \quad x, \; r \dashrightarrow \mathcal{U}_S. \tag{82}$$

[2]In case the assumption does not hold, all sellers could multiply a same large integer to their bids and/or perform truncations to make them integers.

Secondly, each seller in $\mathcal{U}_{S_0}$ verifies if (74) is correct by performing two kinds of verifications. First, all sellers verify if the conditional sum computed by the buyer makes (74) holds as:

$$x \geq R\alpha k/(np'). \tag{83}$$

In addition, each seller in U_{S_0} verifies if the conditional sum is correctly computed by verifying: 1) if the published ciphertext is not tampered; 2) if the encryption of the published conditional sum using the published sum of rand seeds equals the result of ciphertext additions on all published ciphertexts:

$$Enc_{pub}(x, r) = CipAdd(\{\overline{x_i}\}_{i\in S_0}). \tag{84}$$

It is straightforward to see (74) is true if and only if all verifications above pass. In more detail, we present the entire verification process above in Algorithm 4.

Algorithm 4 *SRG-Verify1*(S_0, p')

Input: S_0, p' and pub that are published by the buyer O.
Output: Whether p' is no greater than $p^* = R\alpha k/(nT(d^{S_0}, m^{S_0}, R\alpha k/n))$ is TRUE or FALSE.

1: For every seller U_i in $\mathcal{U}_{S_0}$, U_i computes $r_i \xleftarrow{\$} R, \overline{x_i} \leftarrow Enc_{puk}(x_i, r_i)$, and sends $(r_i, \overline{x_i})$ to the buyer O, where x_i equals m_i if $d_i \leq p'$, and equals 0 otherwise.
2: O computes $\overline{x} \leftarrow CipAdd(\{\overline{x_i}\}_{i\in S_0})$, $x \leftarrow Dec_{priv}(\overline{x})$, and $r \leftarrow \sum_{i\in S_0} r_i$.
3: O publishes $\{\overline{x_i}\}_{i\in S_0}, x, r \dashrightarrow \mathcal{U}_S$.
4: Every seller verifies if $x \geq R\alpha k/(np')$ is true. If not, the seller aborts the protocol and reports FALSE.
5: Every seller U_i in $\mathcal{U}_{S_0}$ verifies if its ciphertext $\overline{x_i}$ is correctly published. If any inconsistency is found, U_i aborts the protocol and reports FALSE.
6: Every seller computes $CipAdd(\{\overline{x_i}\}_{i\in S_0})$ and $Enc_{pub}(x, r)$. If these two are not equal, the seller aborts the protocol and reports FALSE.
7: If no seller reports FALSE, output TRUE; otherwise output FALSE.

2) Verifying the Second Inequality: To verify (75), all sellers could similarly perform the above process given the verification of (75) also requires all sellers to compute a conditional sum and then verify an inequality of it.

One major differences is, in the verification here, each seller selects its private input using a different criteria:

$$x_i = \begin{cases} m_i & \text{if } d_i \leq R\alpha k/(nT'); \\ 0 & otherwise, \end{cases} \tag{85}$$

where $T' = R\alpha k/(np') + \delta^*$.

In more detail, we present *SRG-Verify2* in Algorithm 5.

E. Security Analysis

Besides the correctly verifying if the buyer cheats, we also want to make sure our verification mechanisms reveal little about sellers' bids which are their private types. Here, we analyze our verification algorithms' security performance by analyzing the information that our verification mechanisms reveal.

Algorithm 5 *SRG-Verify2*(S_0, p')

Input: S_0, p' and pub that are published by the buyer O.
Output: Whether p' is no less than $p^* = R\alpha k/(nT(d^{S_0}, m^{S_0}, R\alpha k/n))$ is TRUE or FALSE.

1: For every seller U_i in $\mathcal{U}_{S_0}$, U_i computes $T' = R\alpha k/(np') + 1/(AnPp'D!)$.
2: For every seller U_i in $\mathcal{U}_{S_0}$, U_i computes $r_i \xleftarrow{\$} R, \overline{x_i} \leftarrow Enc_{puk}(x_i, r_i)$, and sends $(r_i, \overline{x_i})$ to the buyer O, where x_i equals m_i if $d_i \leq R\alpha k/(nT')$, and equals 0 otherwise.
3: O computes $\overline{x} \leftarrow CipAdd(\{\overline{x_i}\}_{i\in S_0})$, $x \leftarrow Dec_{priv}(\overline{x})$, and $r \leftarrow \sum_{i\in S_0} r_i$.
4: O publishes $\{\overline{x_i}\}_{i\in S_0}, x, r \dashrightarrow \mathcal{U}_S$.
5: Every seller verifies if $x < T'$ is true. If not, the seller aborts the protocol and reports FALSE.
6: Every seller U_i in $\mathcal{U}_{S_0}$ verifies if its ciphertext $\overline{x_i}$ is correctly published. If any inconsistency is found, U_i aborts the protocol and reports FALSE.
7: Every seller computes $CipAdd(\{\overline{x_i}\}_{i\in S_0})$ and $Enc_{pub}(x, r)$. If these two are not equal, the seller aborts the protocol and reports FALSE.
8: If no seller reports FALSE, output TRUE; otherwise output FALSE.

Theorem 9: The verification mechanisms SRG-Verify1 and SRG-Verify2 do not reveal any knowledge to all sellers other than the conditional sums in (74) and (75) that are computed in them respectively.

Proof: According to Algorithm 4 and Algorithm 5, both mechanisms reveal only the ciphertexts of all sellers' private inputs to the conditional sum, the conditional sum itself, and the sum of random seeds selected by each seller. Since the ciphertexts are generated using a semantically secure cryptosystem and the sellers do not know the private key, these ciphertexts are no different from a random number to the sellers. In addition, given the fact that each seller selects its seed uniformly at random, the sum of these seeds is also uniformly at random. Thus, the sum of these seeds is also no different from a random number to the sellers. □

Note that although our verification mechanisms reveal the conditional sum that equals the sum of a group of sellers' private claimed maximum sensing time, it is difficult to infer the private claimed maximum sensing time of a particular seller from this sum especially when no seller knows which sellers are in the group. In our mechanisms, every seller in U_{S_0} submits a ciphertext of its claimed maximum sensing time or zero to the buyer. Due to the semantical security of the cryptosystem that guarantees no seller can differentiate a random encryption of zero with a random encryption of a non-zero number, no seller is able to tell which sellers are in U_{S_0}, and submit the encryptions of their claimed maximum sensing time instead of encryptions of zero. This means no seller knows which sellers are in the group.

V. Performance Evaluations

A. Simulation Setup

In this section, we conduct two sets of experiments to evaluate the performance of our mechanisms. We let the

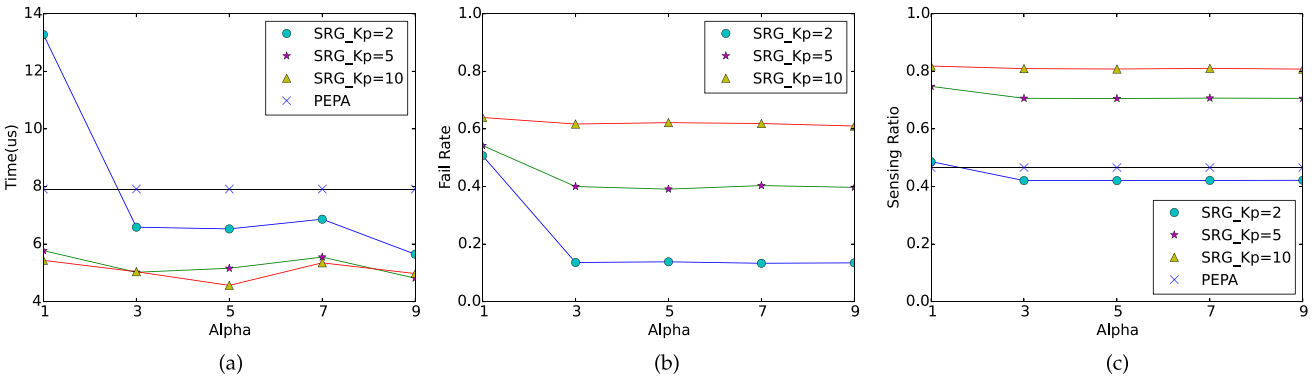


Fig. 1. *SRG-Auc*$_{(k,\alpha)}$'s performance when cnt(sellers) = 20. (a) Efficiency. (b) Failing rate. (c) Competitive ratio.

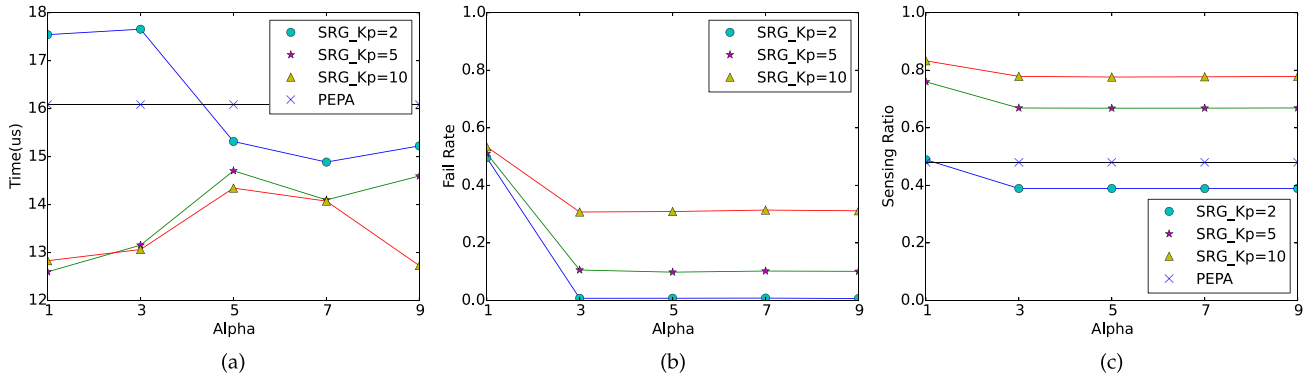


Fig. 2. *SRG-Auc*$_{(k,\alpha)}$'s performance when cnt(sellers) = 50. (a) Efficiency. (b) Failing rate. (c) Competitive ratio.

buyer's budget R equals 1000 and randomly generate sellers' unit costs and their maximum sensing time by sampling the nearest integer of a random variable follows the norm distribution $N(W,(W/3)^2)$, where $W = \sqrt{R/cnt(sellers)}$, and the $cnt(sellers)$ denotes the total number of sellers in our tests. In both sets of experiments, we test three different $cnt(sellers)$s which equal 20, 50, and 100.

In one set of experiments, we evaluate our auction mechanism *SRG-Auc*$_{(k,\alpha)}$ performance in three aspects. First, we test the $SRG-Auc(k,\alpha)$'s efficiency by testing the time that is needed for the buyer to compute the outcome of the auction. In addition, since *SRG-Auc*$_{(k,\alpha)}$ is a randomized algorithm, we test the failing rate of *SRG-Auc*$_{(k,\alpha)}$ which equals ratio between the total number of successful runs and the total number of runs that are tested in our experiment. Finally, we test the competitive ratio of *SRG-Auc*$_{(k,\alpha)}$ which equals the ratio between the total sensing time purchased in *SRG-Auc*$_{(k,\alpha)}$ and the total sensing time purchased in the optimal, omniscient, single-price auction. Let $k_p = cnt(sellers)/k$. In our experiments, we test three different k_ps which equal 2, 5, and 10. In addition, we let α vary from 1 to 9 in steps of 2. Each test is repeated for 10000 times and the final result is the average.

In the other set of experiments, we evaluate our verification mechanisms' efficiency. Specifically, we test the total running time that is used by the buyer and the sellers to complete our auction mechanism *SRG-Auc* with two verification mechanisms *SRG-Verify1* and *SRG-Verify2*. The cryptosystem we used is the BGN cryptosystem [19] with 512-bit keys. Also, we test three different k_ps which equal 2, 5, and 10. Since the parameter α does not affect the running time of both verification mechanisms, we simply choose $\alpha = 3$ in our test. Each test is repeated for 100 times, and the final result is the average.

To provide a better understanding of our auction mechanism' performance, we also implement the Profit Extract Partition Auction (PEPA) mechanism [16], which is currently the best strategy-proof mechanism that achieves a bounded competitive ratio without relying on bayesian assumptions, and compare the efficiency and competitive ratio of our mechanism and the Profit Extract Partition Auction mechanism.

All programs for the experiments are implemented using the C++ programming language, and compiled using gcc-4.7 with the O2 optimization. All experiments were run on a laptop running Mac OS X 10.10 operating system with Intel Core i7 4750HQ CPU, 8GB 1600MHz DDR3 RAM.

B. The Performance of SRG-Auc and Its Verification Mechanisms

Fig. 1, Fig. 2 and Fig. 3 show the testing results about the performance of our mechanism and the PEPA mechanism in the first set of experiments. Fig. 4 demonstrates the efficiency of *SRG-Auc*'s verification mechanisms.

SRG-Auc's Efficiency: From Fig. 1(a), Fig. 2(a) and Fig. 3(a), we can see *SRG-Auc* has a very good efficiency, and outperforms the PEPA mechanism. Even in the case of 100 sellers, the maximum running time that is needed

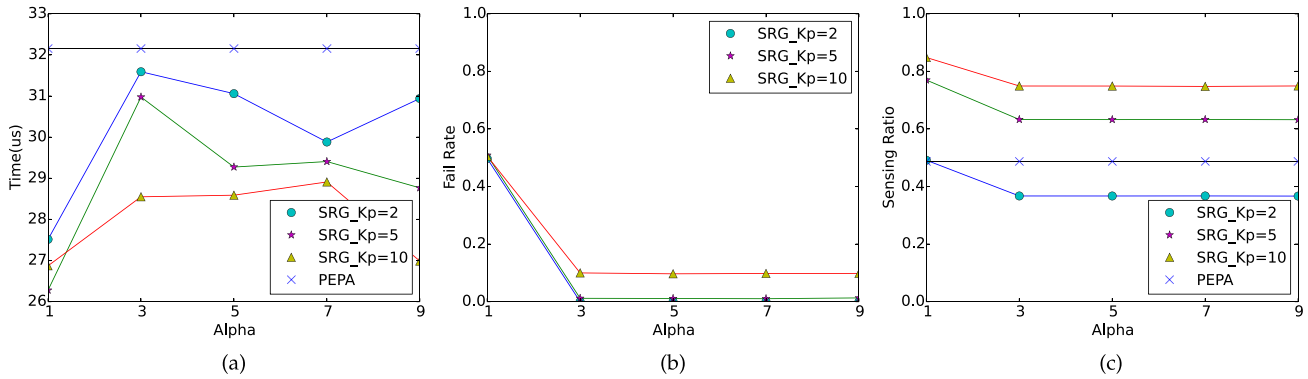


Fig. 3. *SRG-Auc*$_{(k,\alpha)}$'s performance when cnt(sellers) $=$ 100. (a) Efficiency. (b) Failing rate. (c) Competitive ratio.

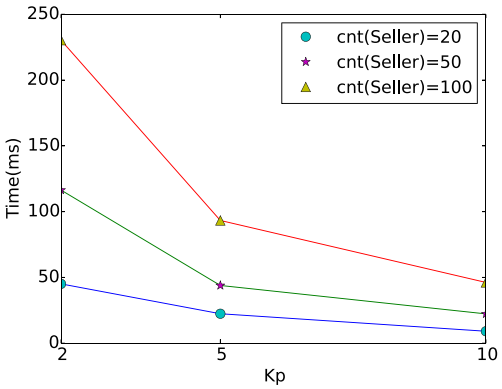


Fig. 4. The efficiency of *SRG-Auc* with *SRG-Verify1* and *SRG-Verify2*.

for the buyer to compute the auction's outcome is only 3.5×10^{-5} seconds.

SRG-Auc's Failing Rate: From Fig. 1(b), Fig. 2(b) and Fig. 3(b), we can see that when α, or the total number of sellers, or k gets greater, *SRG-Auc*'s failing rate becomes lower. In all tests, when $\alpha > 3$, *SRG-Auc*'s the failing rate becomes stable. And the maximum failing rate we can notice in our experiments is around 60% when happens in an quite extreme case in which 2 sellers among 20 sellers are sampled and the final clearing price is merely determined by these two sellers' bids. Since the PEPA mechanism never fails, we do not perform this test on PEPA.

SRG-Auc's Competitive Ratio: From Fig. 1(c), Fig. 2.(c) and Fig. 3.(c), we can see our mechanism outperforms the PEPA mechanism in this test. PEPA achieves stable average competitive ratios that are around 0.5, while our mechanism could achieve greater competitive ratios that are around 0.7 to 0.8. In addition, we can see that the average competitive ratio grows higher when k becomes smaller. This is mainly because when k becomes smaller, more sellers are distributed to $\mathcal{U}_{S_1}$, and the buyer has a greater portion of the entire budget that can be used on these sellers. The lowest average competitive ratio in our tests is 30% which is quite a promising result.

Efficiency of SRG-Auc With SRG-Verify1 and SRG-Verify2: From Fig.4 we can see the entire mechanism which consists of the basic auction mechanism and the verification mechanisms has very good efficiency. Among all tests, the maximum total running time that is needed to complete all computations in *SRG-Verify1* and *SRG-Verify2* is only 0.4 second in average. Since the both verification mechanisms require every seller in $\mathcal{U}_{S_0}$ to perform one encryption, and require the buyer to perform ciphertext additions on all these ciphertexts. Therefore, we can see the total running time decreases as parameter k_p increases. (Recall the size of $\mathcal{U}_{S_0}$ is $k = cnt(sellers)/k_p$.)

VI. Discussions

In this section, we discuss possible issues that could happen when our mechanisms are used in real life.

A. Dealing With Collision Attacks

When the sellers are able to collude, they can manipulate the market, and force the buyers to accept a higher price. We note that designing a strategy-proof auction mechanism with collusion in our problem is highly challenging, and existing game theoretical techniques can hardly produce perfect solutions.

One possible solution to mitigate this issue is as follows. First, the buyer performs the test-run of our auction mechanism for several times. In each test-run, a clearing price and its corresponding auction outcome is computed and recorded. In the end, the buyer chooses the smallest clearing price, and set the corresponding auction outcome as the final outcome of the auction. Since our mechanism chooses S_0 randomly, it is likely in one of these test-runs the clearing price is not manipulated. In addition, the buyer could preset a clearing price threshold based on its knowledge of the market. If the clearing price exceeds the threshold, the buyer could abort the auction to prevent loss caused by collision attacks.

B. Dealing With Sybil Attacks

Another possible attack is the Sybil attack in which a dishonest seller submits multiple bids to increase its chances of winning the bid or manipulating the price.

In fact, if a dishonest seller succeeds to submit multiple bids in the auction for sellers in S_1, we can show that it cannot increase its expected utility from doing this. (However, we note that there is a chance that a dishonest seller may manipulate the clearing price by submitting multiple bids as sellers in S_0.) For the ease of presentation, we refer to these

multiple bids as "sub-bids". In more detail, if the sum of claimed maximum sensing time in all sub-bids exceeds this seller's real maximum sensing time, it is easy to see there is a chance that all sub-bids are approved and then it would receive a very large utility deficit since it is required to sense longer than it is capable to. This large utility deficit would have a dominating impact on the expected utility, and make it negative. If the sum of claimed maximum sensing time in all sub-bids is no greater than its real maximum sensing time, we can prove the expected utility of this user, when it submits multiple bids, is no greater than its expected utility, when it truthful bids once. The main reasoning is as follows. Consider all possible sorted sequences in which the first appearance of this seller's sub-bid is on the i-th position ($i = 1, \ldots, n$). It is easy to see, given the sorted order is any of these sequences, this seller's utility is no greater than the utility that it receives given it directly submits its bid and the position of this users' bid in the sorted sequence is also i.

In order to defense the Sybil attack, we need to restrict the number of bids that a seller could submit. One possible solution is to require each bidder to authenticate itself with its unique identification, e.g. the cellphone number or the mobile identification number (MIN), before submitting its bid to the buyer. In this way, we can guarantee that one mobile phone users with one cellphone could only submit one bid to the buyer.

C. Dealing With Uncooperative Sellers in the Verification

Notice that our verification mechanism requires the sellers in S_0 to honestly participate in the verification process. However, in practice, these sellers do not have the incentive to cooperate. In fact, they might even intentionally submit fake data to make the auction on sellers in S_1 fail, causing the job owner cannot buy enough sensing time and rerun the auction. In this way, they may get a second chance to win the auction.

One possible solution to this issue is to provide incentives to the users in S_0, and meanwhile to prohibit a buyer from posting jobs in the system if any user reports that it cheats. Specifically, recall we only use $(n-k)R/n$ to purchase sensing time from users in S_1, this allows us to offer rewards with the remaining budget to users in S_0 for their participation in the verification process. It is easy to see now users have incentives to help the verification, while have no incentives to sabotage a truthful auction since it cannot gain any benefit when the auction is cancelled.

VII. Related Work

There are many works (e.g. [23]–[27]) that develop practical mobile sensing applications or systems to perform different kinds of sensing jobs. However, incentives are not discussed in most of these works.

In general, existing incentive mobile sensing mechanisms can be divided into two categories based on the model they use.

One category contains the works that adopt the *user/seller-centric model* in which sellers have private types and buyers determine the outcome of the mechanism based on bids of the sellers. Our work also belongs in this category. Works in this category often aim to design mechanisms that are able to optimize the buyer's interests or the social welfare, and meanwhile guarantee promising economic properties such as individual rationality, strategy-proofness against the sellers. For example, Koutsopoulos [11] considers the problem of minimizing the total payments the buyer make for providing a required level of service quality. Sun [1] considers an online auction and proposes mechanisms that maximizes the social welfare based on heterogenous belief values about sellers' types. In [2], Sun and Ma study an online all-pay auction and design mechanisms that aim to maximize the buyer's revenue with a limited budget. In [3], Luo et al also consider an all-pay auction and aim to design auction mechanisms that maximize the buyer's profit. All works above construct their solutions in a bayesian framework and assume the probability distribution of sellers' types are known. There are a few works that do not make such assumptions. For example, In [8], Zhao et al. consider an online auction and propose strategy-proof mechanisms that maximize the buyer's valuation with a limited budget. In [28], Zhang et al. also consider an online auction and design auction mechanisms that maximize the buyer's utility with an unlimited budget. Both works above assume the optimization function is submodular, and design strategy-proof mechanism based on this property. One major difference between all works above and our work is that we do not make any of the two assumptions.

Besides, there are a number of works (e.g. [29], [30]) that consider the incentive mechanisms in the mobile sensing market setting and design double auctions or two-sided auctions for it. These works can be considered as a special case of works based on seller-centric model. Note that these works generally adopt a demand-and-supply model which is totally different from ours.

Finally, there are a number of works that design auction mechanism for specific purpose or based on specific considerations of the problem. For example, In [31], Danezis et. al. propose to use second price auction to persuade sellers to reveal their true valuation for privacy; In [32], Lee and Hoh propose reverse auction with dynamic pricing to retain participants; truthfulness is only want; In [33], Krontiris and Albers propose auction mechanisms for the special case in which sellers' commodities have multiple attributes; In [34], Feng et. al. take the sellers' locations into considerations. Due to the specificities in their problems or goals, their solutions do not apply in our problem.

The other one category includes the works that adopt the *platform/auctioneer/buyer-centric model* in which the buyer may provides a fixed payment and the sellers jointly determine the outcome of the mechanisms. Works in this category [9], [35] generally aim to study sellers' strategic behaviors and analyze the resulting equilibriums. For example, in [9], Yang et al. consider a strategic game that is played by all sellers and study the Nash Equilibrium of sellers' strategies.

We also notice that there are a few works that study similar revenue maximization problems in the spectrum auction area in recent years. Among these works, many (e.g. [4]–[6]) consider the problem in a bayesian model which assumes

the distribution of player's types is known before the auction. There are also a few works (e.g. [7], [36]) which do not make such assumptions. However, their solutions mainly focus on how to avoid spectrum interferences and explore spectrum sharing properties, which do not apply in our problem. Furthermore, these works do not consider the impact of bidders' limited resource which is one of our major concerns.

Since one goal of our auction mechanisms is to maximize the buyer's revenue, theoretically this part of our work can be categorized into the general revenue-maximizing auction design problem. Most existing works (e.g. [37]–[39]) on this topic construct their solutions in the Bayesian setting which assumes that the auctioneer knows the probability distribution of the bidders' types. A recent work [16] on the revenue maximization in multi-unit budget-constraint auctions does not rely on the above assumption. Despite a similar bipartition idea is adopted in the mechanism proposed in [16] and our auction mechanism, the partition algorithm, the outcome generation and thus the final performance guarantee of the two mechanisms are totally different.

VIII. Conclusion

In this paper, we consider a general incentive-based resource allocation scenario in which a buyer aims to purchase as much resource as possible with a limited budget from a group of rational sellers, and study how to design secure and dependable mechanism for it. We first propose an auction mechanism that is proven to be strategy-proof against all bidders, and meanwhile, has a theoretical guarantee on the competitive ratio regarding the amount of resource that is purchased by the buyer. In addition, we further consider that the buyer/auctioneer could temper the auction's outcome, and extend our auction mechanism to make it strategy-proof against the buyer and meanwhile preserver sellers' privacy. Experiments show our auction mechanisms enjoy strategy-proofness, as well as a good competitive ratio, and meanwhile are highly efficient. Due to these advantages, our auction mechanisms can be easily applied in practical resource allocation applications in areas such as mobile sensing, crowdsourcing, spectrum auctions, etc.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable time and effort in reviewing this paper. Their insightful suggestions helped in improving the quality of this paper significantly.

References

[1] J. Sun, "An incentive scheme based on heterogeneous belief values for crowd sensing in mobile social networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 1717–1722.

[2] J. Sun and H. Ma, "A behavior-based incentive mechanism for crowd sensing with budget constraints," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 1314–1319.

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

[4] A. P. Subramanian, M. Al-Ayyoub, H. Gupta, S. R. Das, and M. M. Buddhikot, "Near-optimal dynamic spectrum allocation in cellular networks," in *Proc. 3rd IEEE Symp. New Frontiers Dyn. Spectrum Access Netw. (DySPAN)*, Oct. 2008, pp. 1–11.

[5] J. Jia, Q. Zhang, Q. Zhang, and M. Liu, "Revenue generation for truthful spectrum auction in dynamic spectrum access," in *Proc. 10th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2009, pp. 3–12.

[6] M. Al-Ayyoub and H. Gupta, "Truthful spectrum auctions with approximate revenue," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 2813–2821.

[7] A. Gopinathan and Z. Li, "A prior-free revenue maximizing auction for secondary spectrum access," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 86–90.

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

[9] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. 18th Annu. Conf. MOBICOM*, 2012, pp. 173–184.

[10] J.-S. Lee and B. Hoh, "Dynamic pricing incentive for participatory sensing," *Pervasive Mobile Comput.*, vol. 6, no. 6, pp. 693–708, 2010.

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

[12] *Amazon Mechanical Turk*. [Online]. Available: https://www.mturk.com/mturk/

[13] *CrowdFlower*. [Online]. Available: http://www.crowdflower.com/

[14] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. New York, NY, USA: MIT Press, 1994.

[15] S. Saroiu and A. Wolman, "I am a sensor, and i approve this message," in *Proc. 11th Workshop Mobile Comput. Syst. Appl.*, 2010, pp. 37–42.

[16] Z. Abrams, "Revenue maximization when bidders have budgets," in *Proc. Annu. ACM-SIAM SODA*, 2006, pp. 1074–1082.

[17] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 1999, pp. 223–238.

[18] O. Goldreich, *Foundations of Cryptography: Basic Applications*, vol. 2. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[19] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Theory of Cryptography*. Berlin, Germany: Springer-Verlag, 2005, pp. 325–341.

[20] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems," *J. ACM*, vol. 38, no. 3, pp. 691–729, 1991.

[21] A. C. Yao, "Protocols for secure computations (extended abstract)," in *Proc. 23rd Annu. Symp. Found. Comput. Sci.*, Chicago, IL, USA, Nov. 1982, pp. 160–164.

[22] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," in *Proc. 19th Annu. ACM. Symp. Theory Comput.*, New York, NY, USA, 1987, pp. 218–229.

[23] H. J. Lee *et al.*, "Ubiquitous healthcare service using Zigbee and mobile phone for elderly patients," *Int. J. Med. Informat.*, vol. 78, no. 3, pp. 193–198, 2009.

[24] A. Thiagarajan *et al.*, "VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proc. 7th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2009, pp. 85–98.

[25] R. Lubke, D. Schuster, and A. Schill, "MobilisGroups: Location-based group formation in mobile social networks," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM Workshops)*, Mar. 2011, pp. 502–507.

[26] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment framework for participatory sensing data collections," in *Proc. 8th Int. Conf. Pervasive Comput. (Pervasive)*, 2010, pp. 138–155.

[27] Y. Zhao, J. Wu, F. Li, and S. Lu, "On maximizing the lifetime of wireless sensor networks using virtual backbone scheduling," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 8, pp. 1528–1535, Aug. 2012.

[28] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li, "Free market of crowdsourcing: Incentive mechanism design for mobile sensing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3190–3200, Dec. 2014.

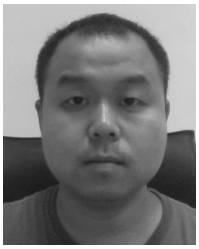
[29] C. Chen and Y. Wang, "SPARC: Strategy-proof double auction for mobile participatory sensing," in *Proc. Int. Conf. Cloud Comput. Big Data (CloudCom-Asia)*, 2013, pp. 133–140.

[30] W. Xu, H. Huang, Y.-E. Sun, F. Li, Y. Zhu, and S. Zhang, "DATA: A double auction based task assignment mechanism in crowdsourcing systems," in *Proc. 8th Int. ICST Conf. Commun. Netw. China (CHINACOM)*, 2013, pp. 172–177.

[31] G. Danezis, S. Lewis, and R. J. Anderson, "How much is location privacy worth?" in *Proc. WEIS*, 2005, pp. 1–13.

[32] J.-S. Lee and B. Hoh, "Sell your experiences: A market mechanism based incentive for participatory sensing," in *Proc. IEEE Int. Conf. PerCom*, Mar./Apr. 2010, pp. 60–68.

[33] I. Krontiris and A. Albers, "Monetary incentives in participatory sensing using multi-attributive auctions," *Int. J. Parallel, Emerg. Distrib. Syst.*, vol. 27, no. 4, pp. 317–336, 2012.
[34] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 1231–1239.
[35] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1701–1709.
[36] R. Zhu, Z. Li, F. Wu, K. Shin, and G. Chen, "Differentially private spectrum auction with approximate revenue maximization," in *Proc. 15th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2014, pp. 185–194.
[37] Y.-K. Che and I. Gale, "Expected revenue of all-pay auctions and first-price sealed-bid auctions with budget constraints," *Econ. Lett.*, vol. 50, no. 3, pp. 373–379, 1996.
[38] J.-J. Laffont and J. Robert, "Optimal auction with financially constrained buyers," *Econ. Lett.*, vol. 52, no. 2, pp. 181–186, 1996.
[39] E. S. Maskin, "Auctions, development, and privatization: Efficient auctions with liquidity-constrained buyers," *Eur. Econ. Rev.*, vol. 44, no. 4, pp. 667–681, 2000.



Yuan Zhang received the B.S. degree in automation and the M.S. degree in software engineering from Tsinghua University, in 2005 and 2009, respectively, and the Ph.D. degree in computer science from the State University of New York at Buffalo, in 2013. He is interested in security, privacy, and economic incentives.

He Zhang is currently pursuing the bachelor's degree in computer science with the Computer Science and Technology Department, Nanjing University. He is interested in security, privacy, and economic incentives.

Siyuan Tang received the B.S. degree in computer science from Nanjing University, in 2014, where he is currently pursuing the master's degree in computer science with the Computer Science and Technology Department. He is interested in economic incentives.

Sheng Zhong received the B.S. and M.S. degrees from Nanjing University, in 1996 and 1999, respectively, and the Ph.D. degree from Yale University, in 2004, all in computer science. He is interested in security, privacy, and economic incentives.