SCALABLE PRIVACY-PRESERVING PARTICIPANT SELECTION WITH APPROPRIATE INCENTIVE IN MOBILE CROWDSENSING SYSTEMS

by

Ting Li

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computer Science

Charlotte

2019

Approved by:

_____
Dr. Yu Wang

_____
Dr. Weichao Wang

_____
Dr. Dazhao Cheng

_____
Dr. Tao Han

_____
Dr. Hsin-Hao Su

ABSTRACT

TING LI. Scalable Privacy-Preserving Participant Selection with Appropriate Incentive in Mobile Crowdsensing Systems. (Under the direction of DR. YU WANG)

Mobile crowdsensing (MCS) has been emerging as a new sensing paradigm where vast numbers of mobile devices are used for sensing and collecting data in various applications. Unlike traditional sensor networks (or the static sensing paradigm), which use pre-deployed sensors to collect specific information at fixed locations, MCS leverages a large number of participants (smart mobile device users) to jointly perform sensing and other crowd sourcing tasks. The MCS solution brings several advantages, including low infrastructure cost, real-time and wide coverage, and potential integration with human intelligence. However, it also faces several research challenges. These include participant selection, incentive mechanisms, and privacy protection and so on. In this work, we mainly focus on designing scalable privacy-preserving participant selection with appropriate incentive for mobile crowdsensing system.

Auction based participant selection has been widely used for current MCS systems to achieve user incentive and task assignment optimization. However, participant selection problems solved with auction-based approaches usually involve participants' privacy concerns because a participant's bids may contain her private information, and disclosure of participants' bids may disclose their private information as well. Following the classical VCG auction, we carefully design a scalable grouping based privacy-preserving participant selection scheme, which uses Lagrange polynomial interpolation (LPI) to perturb participants? bids within groups. The proposed solution, which built on the current MCS platform, can protect such bid privacy in a temporally and spatially dynamic MCS system. Later, we analyze the bidding game of our proposed solution with three implications to prove the security.

To address the participant grouping problem with the constraint of communication

cost during participant bidding process, we propose two algorithms: sorting and dynamic programming (DP). We prove that sorting algorithm could efficiently achieve a feasible solution with a certain approximation ratio for different problems, while dynamic programming algorithm is proved to provide the optimal solution.

However, the selection scheme with cloud-based MCS platform suffers from high overheads, poor scalability and more important. To address this issue and to enhance the protection of user privacy, we further propose a set of novel privacy-preserving grouping methods, which place participants into small groups over hierarchical edge clouds. Our design goal is to group participants in a way that minimizes the communication cost during secure sharing/bidding, while satisfying each participant's requirement for privacy preservation. For different scenarios and optimization functions, we propose a set of grouping schemes to fulfill this goal.

For all of above work, extensive simulations over both synthetic and real-life datasets are conducted to verify the efficiency and security, and confirm the effectiveness of proposed mechanisms.

ACKNOWLEDGEMENTS

I would like to thank a lot of people since I wouldn't have come this far without their help.

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Yu Wang, without whose consistent help I couldn't have finished all of my work. I am very fortunate to have joined WiNS lab under his supervision, with persistent support, valuable instructions and excellent guidance. He is the best mentor I ever had. Words cannot describe how much I learned from him and how thankful I am.

I am so grateful to all the members of my dissertation committee, for their indispensable and extensive professional guidance, constructive criticism, and valuable feedback in my dissertation. In addition, I would like to thank UNC Charlotte for providing me a spectacular academic environment with professional faculty guidance, heart-warming staff support, and excellent peer cooperation.

Also, special thanks to my husband, Joe and my parents, for their care, affection and encouragement. Finally, I would like to thank all my friends for their support over the last few years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Over the past decades, there has been a proliferation of smart mobile devices (including smart phones, smart watches, tablets, smart shared bikes, and connected vehicles), which are capable of sensing, computing, and communicating information. This has lead to the development of the *Mobile Crowd Sensing* (MCS) [1, 2] paradigm for data collection. Unlike traditional sensor networks (or the static sensing paradigm), which use pre-deployed sensors to collect specific information at fixed locations, MCS leverages a large number of participants (smart mobile device users) to jointly perform sensing and other crowd sourcing tasks. Large-scale MCS systems have been used for many applications, including traffic monitoring [3], noise pollution assessment [4], trajectory recovery [5], environment monitoring [6], and on-street parking space tracking [7, 8]. For more MCS applications, please refer to [1, 2].

The MCS solution brings several advantages, including low infrastructure cost, real-time and wide coverage, and potential integration with human intelligence. However, it also faces several research challenges. These include participant selection [9–14], incentive mechanisms [15–20], and privacy protection [21–24].

Participant selection aims to find the optimal set of participants for a given sensing task with the goal of maximizing the coverage or minimizing the cost. While the coverage determines the degree of task completion, the cost often affects the platform's utility. participant selection can be formulated as an optimization problem in several ways. One approach is to maximize the coverage, while ensuring that the cost satisfies a budget constraint (e.g. the number of participants). Another approach is to maximize the overall utility, while ensuring that the coverage is larger than a certain threshold. Recently, several studies addressed this important and challenging aspect

of MCS [9–11, 11–14, 25–28].

Another issue is motivating participants to perform the required sensing tasks, as it may consume resources from the participants' devices [25, 28, 29] or compromise their privacy [19, 21, 30]. Though voluntary participation played an important role, most of current MCS systems are incentive/payment based. To ensure adequate participation, different incentive mechanisms have be used, including reward-based incentives, quality-based incentives, and privacy-based incentives.

Finally, privacy protection is essential for MCS systems [31], as MCS platforms need to access private user information such as location [21, 32–39], payment information [32–34], sensing data [36–42], and mobility traces [5]. A number of privacy preserving mechanisms have been developed, which includes using a trusted third party [32, 36, 38], a blind signature, differential privacy [23, 43–45], the blockchain technical [46], or cloaked locations [21].

In summary, participant selection is one of the key challenges in MCS systems, and it is tightly coupled with incentive mechanisms and privacy protection.

Auction based participant selection is a common solution, where participants submit their bids (reflecting their sensing costs) over different sensing tasks to the MCS system and the platform selects the winners (usually with the lowest bids) among all the bidders to perform the tasks. Here we assume that various sensing tasks may request sensing data at different locations and time. Further, different mobile participants may have their own mobility patterns, as a result, they perform these sensing tasks at various costs. The optimal goal is to pick the appropriate participants who can perform the tasks with the minimum cost. In addition, to guarantee the truthfulness of participants on their bids, a Vickrey-Clarke-Groves(VCG)-based auction [79] can be applied. Notice that the platform in an auction-based MCS system (Fig. 3.1) has the bid information and can easily conjecture the bid patterns of each participant through a long time learning process because the bids are temporally and spatially

correlated in MCS auctions. Furthermore, the bidding value (i.e., the private sensing cost value) may also reflect certain level of privacy information, such as the likeness of visiting that place or the distance to the task location. Therefore, exposing the bid information to the platform brings privacy concerns of users and may hurt the users' enthusiasm to participate. Further, this may result in insufficient participants for the completeness of sensing tasks. Therefore, it is a critical issue for the MCS system.

In first work, we focus on a new solution to protect the bid privacy of participants while still guarantee the truthfulness property of auction and the efficient operation of the MCS system. For potential participants, bid privacy is preserved unless they win the competition and perform the assigned sensing tasks. Our contribution is made by taking care of those challenges. First, we focus on a temporally and spatially dynamic MCS system, where both sensing tasks and mobile participants have dynamic characteristics in both spatial and temporal domains. Second, the users in the bidding system are dynamic. Mobile devices can join and leave, thus the bidder pool keeps changing over different tasks. Third, we aim to achieve accurate sensing results. Fourth, we target at protecting bid privacy without a trusted third party (TTP) participating in every round of auctions, considering that such a party capable of coordinating every single auction hardly exists. Finally, we hope that the proposed solution is built on the existing MCS system and does not affect the operation of current platform. By leveraging Lagrange polynomial interpolation (LPI), minimization approximation and several additional semi-honest parties, bid information is protected during the group bidding and the final platform bidding. Notably, our theoretical analysis shows that no statistical information about bids is disclosed from the ciphers generated by our solution (semantic security), implying that even the temporally and spatially correlated bids can be protected by our approach.

In second work, we mainly deal with how to group participant together for the secret sharing [72] and grouping bidding [73], which have been proposed for participant

selection recently. To protect the participants privacy during participation selection, both of them adopt the idea from *k-anonymity*, which the information of each individuals is contained in the release cannot be distinguished from at least $k-1$ individuals whose information also exist in the release, to perform our *participant grouping*. We first model participant grouping into multiple optimization problems, *Min-Max* of group size (or its square) and *Min-Sum* of group size (or its square) under the circumstance of considering the communication cost for group members exchanging their ID or not. Then we propose *Sorting* and prove that *Sorting* is $2-approximation$ comparing to optimal solution with complexity $O(N \log N)$ for *Min-Max* problem. We also apply *Dynamic Programming (DP)* to solve both *Min-Max* and *Min-Sum* problem with optimal solution and complexity $O(N^2)$.

Existing MCS systems are mainly cloud-based, and participant selection is performed by the platform, which is located on a remote cloud sever. Thus, participants must upload their bids and sensing quality to the cloud in order to take part in the selection procedure. However, this not only leads to privacy concerns, but also causes long communication delays, which are not tolerable for time-sensitive tasks. To mitigate these issues, we propose to use edge computing, which places small-scale servers at the edge of the network and performs data processing closer to the users. This way, in addition to shorter communication latency, sensitive information will only be submitted to nearby edge servers instead of to the remote platform.

In the last piece of work, we carefully study the *privacy-preserving participant grouping* problem for MCS, where participant groups must satisfy the privacy requirements of each user and its challenges are shown as follows. First, there is an important trade-off in finding an optimal location on the edge cloud. On the one hand, to minimize communication latency, it is preferable for a mobile user's group to be located on a server close to the user, but on the other hand, to ensure greater privacy, one may need to pool users from several locations and thus host their in-

formation further away. Second, not only the location, but the size of the group matters. This also requires a trade-off since larger groups provide more privacy, but they require larger computation costs when performing the secret sharing/bidding procedure. Third, the overall costs and loads among edge servers and groups need to be optimized or balanced.

To the best of our knowledge, this is the first work tackling the privacy preserving participant grouping problem for MCS over hierarchical edge clouds. Our contributions can be summarized as follows: We model participant grouping into multiple optimization problems.Considering the communication cost between edge servers, we propose two heuristic algorithms *Top-Down* and *Bottom-Up* with $O(N \log N)$ to solve the grouping problem.

Last but not least, we conduct the simulation with two real-life datasets to confirm that the method is efficient compared with other existing methods for participant selection, participant grouping and grouping over hierarchical edge clouds, respectively. Experiments results validate the proposed approaches are efficient for the scalable privacy preserving participant selection with grouping over hierarchical edge clouds.

CHAPTER 2: RESEARCH BACKGROUND AND RELATED WORK

2.1 Architecture of MCS Mobile Crowdsensing System

MCS is a system for solving various sensing tasks, where individuals with smart mobile devices collect and contribute data to complete a given task [1]. This can be implemented in several ways: a cloud-based system or a distributed system.

A *cloud-based MCS system* is comprised of three main components: the task owners, the MCS platform, and the participants. These are illustrated in Figure 2.1. The task owners start the process by initiating tasks, and they end it by collecting sensed data from and making payments to either the selected participants or the MCS platform. The MCS platform is in charge of participant modeling, participant selection, incentive mechanisms, privacy protection, data retrieval, and truth discovery. The participants are the mobile device users who can perform various sensing tasks. Participation in a particular task can be either voluntary or motivated by incentives. In MCS systems, these three main components (task owners, MCS platform, and mobile participants) are inter-connected with sensing tasks, sensing data, payments, and rewards as illustrated in Figure 2.1.

A *distributed MCS system* has the same component as the cloud-based MCS one but with multiple platforms. Here, the task owners can send their tasks to multiple platforms, and a mobile participant may fulfill tasks for more than one platform. In addition, each individual device could be a task owner (having its own sensing tasks which need to be performed by others), an agent of distributed platform (selecting participants and sensing out sensing tasks, later collecting sensed data from assigned participants), and a mobile participant (accepting sensing tasks from others and performing them). Without a centralized controller, the platforms may not share

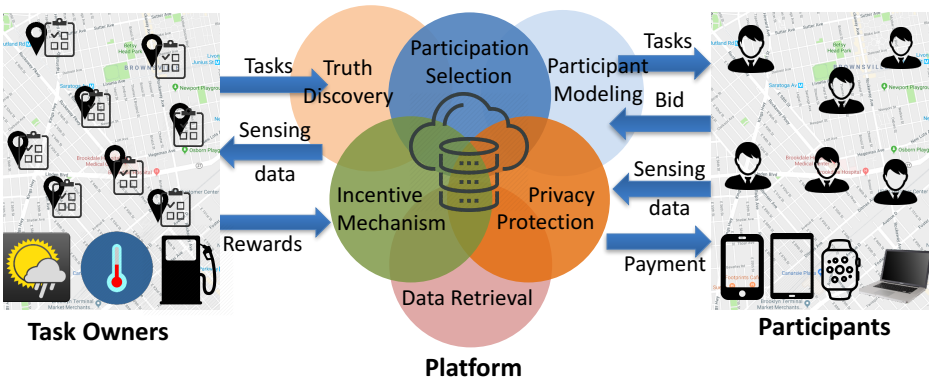


Figure 2.1: The framework of a cloud-based MCS system.

information with each other. In such a distributed system, due to the lack of global information and the limited resources at local devices, the overall performance of MCS may be less optimal. However, the interactions among multiple agents (platforms and users) make participant selection more interesting and challenging.

2.2 Challenges in Mobile Crowdsensing

To perform an efficient participant selection and to optimize performance, an MCS platform/system must deal with a number of issues. These include *participant modeling*, *participant selection*, *data retrieval*, *incentive mechanism*, *privacy protection*, and *truth discovery*, see Figure 2.1. These issues are often tangled with each other and cannot be solved separately. We now briefly discuss these.

2.2.1 Participant Modeling

To select the optimal set of participants to perform the given sensing tasks, the MCS platform must acquire participant information, such as mobility patterns and sensing qualities. However, due to privacy concerns, this information may not be available. In this case, the MCS platform constructs a model to predict the unknown information of participants. When a participant's capability (either mobility patterns or sensing qualities) is unknown or difficult to retrieve due to privacy concerns [21, 22, 36], learning techniques [47–50] can be used.

2.2.1.1 Modeling Sensing Quality

A number of methods have been used to model a user's sensing quality. [47] used an online learning approach to acquire statistical information about the sensing values from participants throughout the selection process with the assumption that the quality of sensing data is random. [48] modeled the situation as an online labeling problem *e.g.* labeling task with 0 or 1, where the true label is unknown. They proposed an online algorithm, which used the majority voting rule to differentiate high and low quality participants over time and proved that their method has a bounded regret under mild assumptions on the collective quality of the crowd. [49] also considered expertise-aware task allocation and truth analysis in MCS, where user expertise is estimated via a general online learning framework. [50] modeled a cumulative participant selection problem as a combinational multi-armed bandit problem and presented an online selection algorithm, which leverages the historical performance records of participants to learn the different capabilities (both sensing probability and uploading time delay) of participants.

2.2.1.2 Mobility Model

Many MCS tasks aim to collect sensing data at a particular location and time. For such problems, knowledge of a participant's mobility patterns allows the platform to select either better or fewer participants [51]. For this reason, the modeling of participant mobility is important for MCS. In some cases, a participant's mobility patterns are known by using Global Positioning Systems (GPS) or other localization schemes as in [52]. In other cases, the participant's mobility patterns are unknown and need to be predicted, as in [26, 28, 53, 54]. [54] suggested that user mobility can be predicted by combining *Bayesian* inference with *Markov* models. Similarly, [53] applied what they call a semi-Markov process model to calculate the probability that a user will be at a certain location during a predefined time slots. Beside, the mobility

trace could be obtained based on information retrieval from the route scheduling or the navigator of vehicles [55] to achieve high quality sensing with a limited budget. Another type of mobility is for task-oriented participants, when either the participant routes are unknown, including their historical data, to the platform, the participants' travel plan is determined by their assigned tasks. Since, in this case, the mobility patterns are unknown to the system, participant selection/task assignment is mostly used to help each participant to make their own travel plan as in.

2.2.2 Participant Selection

Participant selection aims to effectively select appropriate participants from a huge participants pool to perform various sensing tasks while satisfying certain constraints or achieving certain goals. The goals or the constraints are often related to three aspects, which include the sensing cost (*e.g.* the energy consumption or the rewards to selected participants), the task coverage, and the quality of the sensed data. The ultimate goal of participant selection can be minimizing the overall cost while achieving the required degree of coverage or quality level of the sensed data; or it can be maximizing the coverage or quality of the fulfilled tasks within a fixed cost budget.

Participant selection is the key function of an MCS system. Based on MCS system's architecture, centralized and distributed solutions can be applied. We now briefly illustrate possible solutions over the centralized and distributed architectures.

2.2.2.1 Participant Selection over Centralized Architecture

The centralized method, generally uses a cloud-based platform to hold all the candidate information and selects the participants for every task, see, *e.g.*, [25,28]. Using knowledge of each candidate's information (*e.g.* their mobility patterns and bid information), the platform aims to select the participants to optimize performance. Many MCS systems formulate the selection problem as either a coverage problem (covering the sensing tasks with minimal cost) or an auction problem (where the platform is

the auctioneer and the sensing tasks are auction items). We will see many examples in next section. In order to select participants with the lowest cost, [56] used the insights of group buying to aggregate the tasks first and then recruit participants by auction. Since the platform has all the information, the optimization of participant selection can usually be solved more effectively and efficiently than over the distributed architecture. Nevertheless, the centralized architecture suffers from high computation, communication overheads, and privacy concerns. In addition to offline solutions (where the sensing tasks are known), some works [57, 58] have proposed online/real-time mechanisms with budget constraints or using cache [12] to store the sensing results in order to select fewer participants.

2.2.2.2 Participant Selection over Distributed Architecture

In the distributed architecture, several task requesters/platforms [20, 59–61] can receive information from the participants as long as they are within the participants' range, make the selection decision locally, and then may synchronize the outcome of each requester. For example, [60] applied the distributed auction on each crowd sourcer (*e.g.* platform) and selected the user with the lowest cost. Similarly, in [20], multiple distributed platforms published heterogeneous tasks within their communication range and each participant could submit only one bid for all of the tasks. The platform could choose only one participant for each task. In addition to being chosen passively, the participants could also select and schedule the tasks individually. For instance, [62] proposed a distributed algorithm, where each participant receives a set of tasks and she or he then selects an individual or a subset of the tasks.

2.2.3 Data Retrieval

After performing sensing tasks, the selected participants need to send the collected data to the task owners (directly or via the platform). There are several data retrieval methods in MCS. In most systems, the participants hold on the collected data and

upload it until they have access to the platform. If they have cellular data service, they can upload the data at any time with certain cost. One possible improvement is *piggyback-style* retrieval [26, 28], where the selected participants upload the collected data while they make a phone call or access cellular data. This saves energy since the on/off cellular transmission process can consume a significant amount of energy. Another method is *gateway-aided* retrieval [9, 63], where the selected participants upload the data through WiFi or Bluetooth gateways. Such solutions are cheaper in term of energy cost, but the delay in retrieval is longer than cellular solution due to potential long waiting time to encounter a gateway. Further, the participants in [64] can either move to where the gateways are located at or pass the data to other participants who may move to the gateways in future. Such *device-to-device* solutions work well for delay tolerant sensing tasks or distributed MCS systems.

2.2.4 Truth Discovery

Note that each participant, who performs the same assigned sensing task, may have different observations because of various sensor qualities, environment noises, or lack of sensor calibration. Therefore, in many MCS systems, more than one participant [59] will be selected to perform the same task and hence there will be multiple observations for the same task. Several mechanisms have been proposed for aggregating and analyzing the observations to discover the truth (which is unknown to the platform or owners). These methods include maximum likelihood estimation [49, 65], majority voting and block coordinate descent [66, 67]. Besides, the method based on block coordinate descend in [67] was also used to make up the missing data because of the sparsity of sensing data.

2.2.5 Incentive Mechanism

One of the assumptions for participants selection in MCS is the sufficient large number of participants, which guarantees the complement or quality of sensing tasks.

However, it is not easy to attract large number of participants to the MCS platform. There are usually two categories of participants, *voluntary participants* and *non-voluntary participants*. Voluntary participants are usually altruistic and willing to contribute their efforts for the common good (such as environment protection or society improvement) of MCS. Several MCS systems [28, 51, 54] assume that their tasks could be finished with enough voluntary participants who are unselfish and willing to obey assignments. However, in reality, individual mobile users may be selfish and lack of motivation to participant sensing tasks since performing those tasks conserves their resources and significantly impact their own performances. Even a rational user might attempt to only maximize his own utility and conserve his resource without considering system-wide criteria. Therefore, it is crucial for the MCS system to develop appropriate incentive mechanisms to stimulate individual mobile nodes to participant MCS. Designing incentive mechanisms is tightly coupling with participant selection and can be jointly addressed by various pricing, selection, and reward methods [68, 69].

2.2.6 Privacy Protection

Privacy protection is not only for motivating people participation, which is needed for providing incentives to engage as many participants as possible, but for the personal requirement recently. With the development of communication system, the fields of privacy is multifaceted and comprises several other dimensions [70]. Because of the voluminous of mobile devices acting as the major part, their consensus on sensitive sensing data is the mainly concern in MCS. As a recently developing and brand new application, there's no clear definition on the privacy in MCS. However, the privacy in participatory sensing [71] has been declared as the guarantee that participants maintain control over the release of their sensitive information which includes the protection of information that can be inferred from both the sensor readings themselves as well as from the interaction of the users with the participatory sensing system.

Having the similar components and analogue framework as the participatory sensing, the privacy in MCS could refer the definition from participatory sensing.

When participating in a sensing task, the participants are concerned about compromising their confidentiality by sharing their personal information. This information may include temporal-spatial attributes [21,37–39,44,72], the sensed data [38–40], and the participants' bids or payments [24,32,33,73]. A well-developed privacy mechanism encourages participants to contribute without privacy concerns. Therefore, privacy protection is a critical component, which can impact the performance of MCS. Different privacy protection mechanisms have been developed in literature. Some proposed frameworks adopt external components, such as a trusted third party [32, 33] or mobile security agents [35]. Privacy information can also be protected through data hiding, see, *e.g.*, [21, 31, 39], adding extra noise, see [41], or utilizing differential privacy [43, 44].

2.3 Background and Related Work

In this work, we aim to construct a scalable privacy-preserving participant selection system with the help of hierarchical edge cloud. Hence, we only focus on the related work of participant selection, privacy protection, auction in incentive mechanism and grouping.

2.3.1 Participant Selection in Mobile Crowdsensing

Due to the large number of participants and the diverse sensing tasks in mobile crowdsensing, the selection of participants for different tasks (i.e. task assignment) becomes a challenging task. On one hand, assigning more participants for certain task can lead to better quality of the sensed data. On the other hand, MCS have to pay more rewards to the participants to cover their sensing cost. Recently, there are several studies on participant selection in MCS with various optimization goals such as coverage maximization [10, 11, 74], energy efficiency [25, 28, 29], user incentive and

truthfulness [18, 52]. In this work, we also consider the participant selection problem, but focus on a different aspect: bidding privacy. We only consider a simple bidding scenario where the MCS platform aims to minimize the payment by choosing the lowest bid among all bids.

2.3.2 Privacy Protection in Mobile Crowdsensing

To protect the participants' privacy in mobile crowd sensing or participatory sensing, several privacy preserving schema have been proposed using different techniques, such as data transform language [37], data aggregation [40] , location obfuscation [39], cloaking [21], k-anonymity [36], pseudonym [32] and adding noise [41]. These solutions usually introduce additional entities (registration authority / trusted third part) [32, 36, 37] or an aggregation server [39, 40] to achieve the protection of the sensing data privacy, participants' anonymity or their location privacy. Note that TTP-free method in [32] uses pseudonym and bling signature to protect user privacy, but its encryption operations may bring a burden of cost.

In contrast to the data privacy or location privacy solutions above, there are also recent efforts [72, 73, 75] on protecting bid privacy or sensing quality privacy during the procedure of participant selection. Jin *et al.* [75] consider bid privacy in an aggregated MCS system. They define the bid privacy with differential privacy over the aggregated sensing data (labels) and assume that all sensing tasks are binary classification (labelling) tasks. Li *et al.* [73] consider a more general and direct model, where sensing tasks have sensing requests on both temporal and spacial domains and privacy is defined on the bids from participants. They introduce one or multiple semi-honest third parties to perform grouping of participants. By leveraging Lagrange polynomial interpolation (LPI), their solution perturbs the participants' bids within groups so that bid information is protected during the group bidding and the final bidding at the platform. The communication cost of the LPI-based secure biding within a group with k members is $O(k)$. Xiao *et al.* [72] also address privacy-preserving

participant selection by using secret sharing schemes. They consider quality-aware participant selection while trying to protect the inputs (sensing qualities) of each user from being revealed to the platform or to other users. The basic idea is to leverage the secure multi-party multiplication and secure multi-party comparison protocols. Their solution's communication cost is $O(k^2)$, where k is the number of participants. Note that secure sharing/bidding schemes has already been studied [72, 73] and will be not discussed here. In this work, we focus on privacy-preserving user grouping, which can be combined with these secure sharing/bidding schemes to achieve overall privacy-preserving participant selection.

2.3.3 Secure VCG Auction

Auction theory is a branch of game theory which deals with how participants act in auction markets and studies the properties of auction markets. There are two major kinds of members in the auction theory: the *sellers* who want to sell the items and the *buyer* who compete to buy the items. Usually, the auction theory satisfies two conditions: the auction process is universal and the outcome of the auction has no relationship with the identity of bidders, which means the auctions are anonymous. There are two types of the auction model: the regular auction and reverse auction. In regular auction, the sellers sell the item with highest bidder, which is pushed up by multiple buyers who bid against each others. While in the reverse auction model, the buyer buy the item with lowest bidder which generated from the competition of sellers' bidding process and the sellers push down the price for item. The auction theory has been applied in MCS system to model the interaction between participants and platform during the participants selection process. In MCS, most of the work exploit the reverse auction model [15, 19, 52, 76–78], in which the platform acts as the buyer who wants to get the sensing data with lowest payment and the participants are the seller who wants to earn the rewards for their contribution on sensing data.

In our work, Vickrey-Clarke-Groves(VCG) auction [79] is leveraged as a building

block, and we propose a novel privacy-preserving design of VCG auction in order to protect users' bid privacy. Related research exists in the literature, who targets at protecting bid privacy in the VCG auction as this work does. However, existing works have common limitations due to the building blocks they employed to realize secure VCG auctions, and this made them less attractive than our approaches when implemented and deployed in the real-life applications.

Naor *et al.* [80] proposed how to design general auctions with mechanism design without revealing private bid information by leveraging the secure multi-party computation (MPC) with garbled circuits [81]. However, it is shown that auction mechanisms based on secure multi-party computation is inefficient because the complexity inherently increases exponentially with the number of goods to be auctioned and the bit-length of the bid, and the actual overhead is large as well as due to the large constant factors [82]. Besides, an auction issuer, who is a party that is assumed not to collude with the auctioneer, needs to engage every time an auction is run.

Huang *et al.* [83] and Lipmaa *et al.* [84] proposed approaches that are both based on homomorphic encryption , but they require a third party at every auctioning as well. Larson *et al.* [85] proposed to use the homomorphism in homomorphic encryption to enable secure VCG auction without revealing individual bids. However, they introduce a group key among the group of users in extra, and this limits the application in real world where users may come and go because group key sharing must occur for every new group, and this will not be practical in many cases as users cannot communicate with each other during the auction.

A series of works have been proposed by different researchers to realize privacy-preserving VCG auctions [86–91]. All of these works are based on the homomorphic encryption, and they do not require a third party engagement as in our approach. However, they achieved this by generating one cipher per possible bid value. That is, if the bid length is b bits and the size of bid space is 2^b, the computa-

tion/communication/storage complexities are inherently exponential to the input size.

Unlike all aforementioned existing works, our solution does not require third-party[1] engagement in every auction running, and our solution scales well with the number of goods, number of bidders, and the bit-length of the bid values.

2.3.4 Grouping/Clustering for Privacy

Grouping/clustering has been studied in some recent works [92–94] for privacy preserving. Given a set of n points in general metric space and a value r, the r-gather problem is defined as clustering the points into groups at least r points each such that the largest diameter of clusters are minimized [94]. Aggarwal *et al.* [94] prove that there's a polynomial time algorithm that give a 2-approximation to the problem and show that it's a NP-completeness. Armon [93] extends the result of Aggarwal *et al.* and shows that it's NP-hard to approximate with a ratio better than 3 for $r > 2$ for general metric space. Zeng *et al.* [92] describes a distributed algorithm with an approximation factor of 4 for r-gather problem. All of those existing works cluster the points with only one parameter r, however, in our model, the privacy criteria r is different from each user and the optimization problem is formulated differently. Also, the existing algorithms (for example sweep algorithm) could not applied here in the hierarchical edge architecture since the location of users' group could be only on its ancestor node but anywhere else. In addition, none of algorithm are proposed for problem *Min-Sum*.

2.4 Summary

In this chapter, we first introduce the architecture of mobile crowdsensing system with the perspectives of centralized structure and decentralized structure in section 2.1. Then we review the challenges in mobile crowdsensing system so far in section 2.2. At last, the solution with related work are summarized in section 2.3.

[1]Note that the third party (TP) we defined in next section is the auctioneer in the group auction. It is called third party since it is a new entity added between the platform and the participants.

CHAPTER 3: PRIVACY-PRESERVING PARTICIPANT SELECTION

3.1 Introduction

The proliferation of mobile devices equipped with built-in sensors enables a new sensing paradigm, mobile crowd sensing (MCS), which has been widely used in numerous applications [2]. Compared with traditional static sensing, MCS leverages existing sensing and mobile communication infrastructures to provide unprecedented spatiotemporal coverage. Meanwhile, it brings many new challenges in the system design. Participant selection is one of them, where appropriate participants are selected to perform certain sensing tasks [10, 11, 13, 14, 18, 25, 28, 29, 52, 74].

Auction based participant selection is a common solution, where participants submit their bids (reflecting their sensing costs) over different sensing tasks to the MCS system and the platform selects the winners (usually with the lowest bids) among all the bidders to perform the tasks. Fig. 3.1 illustrates the architecture of such a MCS system. Here we assume that various sensing tasks may request sensing data at different locations and time. Further, different mobile participants may have their own mobility patterns, as a result, they perform these sensing tasks at various costs. The optimal goal is to pick the appropriate participants who can perform the tasks with the minimum cost. In addition, to guarantee the truthfulness of participants on their bids, a Vickrey-Clarke-Groves(VCG)-based auction [79] or other game theoretical approaches [18, 52] can be applied.

Existing auction-based solutions solved the participant selection and incentive issues, but we observed that there exists user privacy concerns on the other hand. In most cases, bids are related to participants' contexts (*e.g.,* location), and such information may leads to privacy breach (*e.g.,* a participant with a higher bid in

certain MCS problems indicate closer proximity of his/her location to the place where crowdsensing is performed). Recently, various privacy-preserving schemes [21, 32, 36, 37, 39–41, 95, 96] have been proposed for the protection of the participants' privacy, however none of them consider the privacy leakage from the bid values. In this work, we would like to complement existing works by protecting the bid values in order to achieve better anonymity and privacy protection.

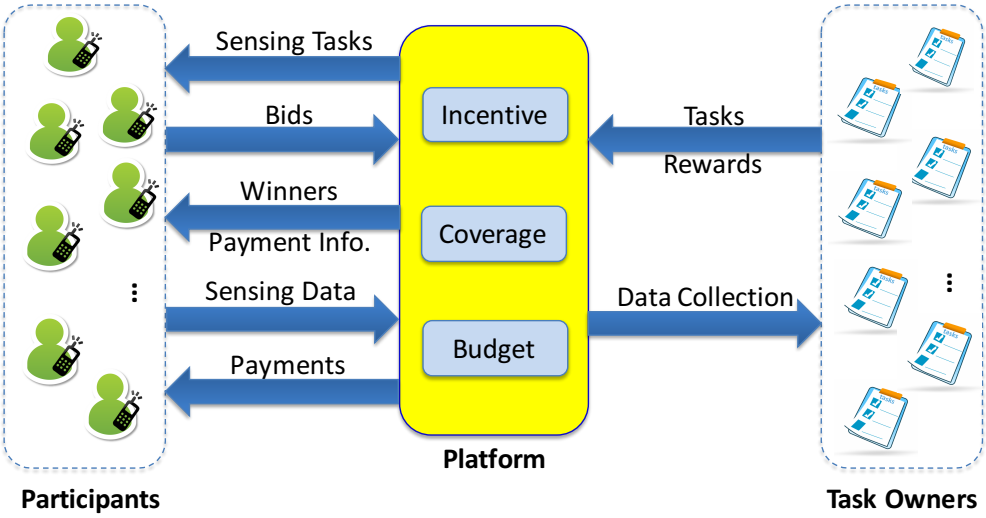


Figure 3.1: **MCS System:** the platform distributes sensing tasks to participants, collects their bids, decides the winning bids (i.e., selecting participants for each task), collects sensing data, and makes payment to the participants.

Notice that the platform in an auction-based MCS system (Fig. 3.1) has the bid information and can easily conjecture the bid patterns of each participant through a long time learning process because the bids are temporally and spatially correlated in MCS auctions. For a more concrete example, the platform may know particular participant route if that participant often bids on some particular location and time. Furthermore, the bidding value (i.e., the private sensing cost value) may also reflect certain level of privacy information, such as the likeness of visiting that place or the distance to the task location. Therefore, exposing the bid information to the platform brings privacy concerns of users and may hurt the users' enthusiasm to participate.

Further, this may result in insufficient participants for the completeness of sensing tasks. Therefore, it is a critical issue for the MCS system. In this work, we focus on a new solution to protect the bid privacy of participants while still guarantee the truthfulness property of auction and the efficient operation of the MCS system. For potential participants, bid privacy is preserved unless they win the competition and perform the assigned sensing tasks.

Achieving the bid privacy in MCS problems involves multiple challenges. First, we focus on a temporally and spatially dynamic MCS system, where both sensing tasks and mobile participants have dynamic characteristics in both spatial and temporal domains. This makes bids in the auctions temporally and spatially correlated, making it hard to protect end-to-end bid privacy over the long time. Second, the users in the bidding system are dynamic. Mobile devices can join and leave, thus the bidder pool keeps changing over different tasks. This makes most of the existing privacy-preserving data aggregation schemes ([97–99]) unfitting since one suite of keys need to be distributed to one specific group of users. Third, we aim to achieve accurate sensing results. This is a critical issue as noisy bid information may lead to unnecessary overpayment and/or even the failure in completing the sensing task. As a result, traditional perturbation-based approaches such as Laplacian mechanism with differential privacy [100] is hardly applicable. Fourth, we target at protecting bid privacy without a trusted third party (TTP) participating in every round of auctions, considering that such a party capable of coordinating every single auction hardly exists. Finally, we hope that the proposed solution is built on the existing MCS system and does not affect the operation of current platform.

We assumes that both the participants and the platform are semi-honest (i.e., they follow the protocol but try to infer sensitive information). Further, we introduce one or multiple semi-honest third parties (TPs) to perform grouping of participants (Section 5.2). By leveraging Lagrange polynomial interpolation (LPI) and key values

generated by a key generator (KG), bid information is protected during the group bidding and the final platform bidding (Section 3.3). Here, KG is only in charge of key generation and does not participate in the auction and crowd sensing process, and its participation is minimized. Notably, our theoretical analysis (Section 3.4) shows that no statistical information about bids is disclosed from the ciphers generated by our solution (semantic security), implying that even the temporally and spatially correlated bids can be protected by our approach. Finally, we conclude in Section 3.6. Experiments with two real-life datasets (Chapter 6) also confirm that the method is efficient compared with other existing methods.

3.2 Problem Definition and Security Models

3.2.1 Participation Selection Problem and VCG Mechanism

In general, a MCS system includes three main components, as shown in Fig. 3.1: a large number of *mobile participants* who can perform sensing tasks and contribute sensing data, a set of *task owners* who generate various sensing tasks and are willing to pay for sensing data (acting as data consumers), and the *platform* who plays a vital role in the MCS system and acts as the MCS marketplace to connect the mobile participants with the task owners. The participation selection aims to select a set of participants who could complete the sensing tasks but with the minimum payment. This could be a challenging task because of large number of participants and various requirements of the tasks.

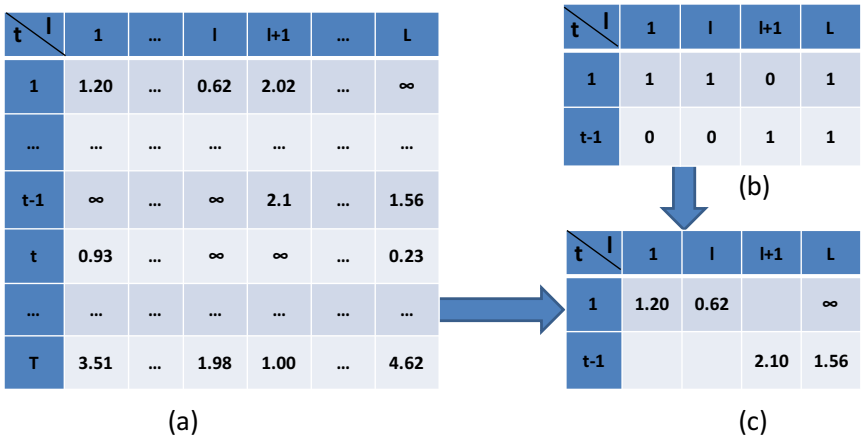


Figure 3.2: **Spatiotemporal Matrixes:** (a) private cost matrix C_i of participant u_i; (b) binary task matrix S; and (c) bidding matrix B_i generated from u_i, in which the bid value may not be equal to the real cost value.

In our model, there are n mobile participants $U = \{u_1, u_2, \cdots, u_n\}$. Each participant u_i keeps a dynamic spatiotemporal matrix about her real sensing cost $C_i = \{c_i(t,l)\}$ privately, as shown in Fig. 3.2(a), where $c_i(t,l)$ is the real sensing cost for u_i to obtain sensing value at location l at time t. The real sensing cost information is sensitive since it may reveal the visiting pattern of this participant. We assume that we have finite number of l and t, i.e., $t \in \{t_1, t_2, \cdots, t_T\}$ and $l \in \{l_1, l_2, \cdots, l_L\}$.

Suppose there are m sensing tasks $S = \{s_1, s_2, \cdots, s_m\}$ and each of them has a strict spatiotemporal coverage requirement, s.t., task s_j could be described as a binary spatiotemporal matrix $S_j = \{s_j(t,l)\}$, where $s_j(t,l) = 1$ represents that s_j request data at location l during time slot t and $s(t,l) = 0$ otherwise. Since we assume that each requested cell within the binary spatiotemporal matrix can be fulfilled by one selected participant within the same requested cell, we take a union of all sensing tasks into a single binary spatiotemporal matrix $S = \{s(t,l) = \oplus_{j=1}^{m} s_j(t,l)\}$, as shown in Fig. 3.2(b), where $s(t,l) = 1$ represents that there is at least one task requesting the data from l at t. Then, the task assignment can be treated as assigning a single

participant to each cell with $s(t,l) = 1$.

Each participant u_i, if interested, can submit a bidding matrix $B_i = \{b_i(t,l)\}$, as shown in Fig. 3.2(c), to the platform based on her real cost. Note that $b_i(t,l)$ may be different from $c_i(t,l)$. After receiving bids from all participants, the platform will make a decision about winning bids for tasks at $s(t,l) = 1$ based on certain strategy and pay the corresponding rewards $p(t,l)$ to the winners of these tasks. We assume that the mobile participants can finish the tasks assigned to them as long as they participate and win the bid competition. In other words, the completeness of tasks is guaranteed if enough bidding participants can cover the task spatiotemporal matrix.

Based on bidding matrices $\{B_i\}$ provided by participants U for task set S, the mission of the platform is to efficiently find the optimal set of participants for tasks such that the total payment (i.e., $P = \sum_{t,l} p(t,l)$) is minimum. At the same time, the platform wants the selection mechanism to be truthful, i.e., the participant bids at its real sensing cost for each cell. To achieve this goal, we adopt the classical Vickrey-Clarke-Groves (VCG) auction [101–103] in our participation selection problem. Each participant has no knowledge about others' bids during the auction since the bid matrix is private. The lowest bidder wins but the payment is equal to the second lowest bid, which gives the participants an incentive to bid at their true cost value in this optimal strategy. The whole VCG auction process includes the winning bid decision and the critical payment calculation.

Definition 1. ***Winning Bid and Critical Payment.*** *The winning bid is the lowest bid among all bids submitted by participants within U for each cell $s(t,l) = 1$, which could be defined as follows:*

$$b(t,l) = \min_{u_i \in U_{(t,l)}} b_i(t,l) \text{ and } w(t,l) = \arg\min_{u_i \in U_{(t,l)}} b_i(t,l) \tag{3.1}$$

where $w(t,l)$ is the single winner for this requested cell (if there is a tie, an arbitrary

one can be selected as the winner) and $U_{(t,l)}$ are the set of participants who submit their bids for task cell (t, l). The payment $p(t, l)$ for winner $w(t, l)$ is defined as the lowest bid among all the bids except the winner's bid. i.e.,

$$p(t,l) = \min_{u_i \neq w(t,l)} b_i(t,l) \tag{3.2}$$

By applying the VCG mechanism, it is easy to prove the bid truthfulness, i.e., the participant will maximize its utility when it bids truthfully at its real sensing cost (i.e., $b_i(t, l) = c_i(t, l)$). In addition, the VCG mechanism minimizes the total payment for the participant selection. Both the winning bid and critical payment can be decided very efficiently with a simple sorting. Notice that it is possible for a task cell, there are bid ties. This will not affect the effectiveness of VCG mechanism.

3.2.2 Adversary Model and Assumptions

Recall that we aim to design a MCS system with grouping and security techniques to protect the bid privacy of participants while still guarantee the truthfulness property of auction and the operation of MCS system. We assume that task owners, the platform, and the participants in the system may all become semi-honest adversaries. A semi-honest adversary follows the protocol specification, however she may try to infer sensitive information from the communication strings generated by the protocol. More specifically, the task owners as well as the platform may try to infer true bids of the participants, and the participants may try to infer other participants' true bids as well as owing to the bidding competition. Further, in order to bootstrap the mobile crowdsensing, we introduce a semi-honest third party (TP) and the only single trusted party in our system – key generator (KG). TP is used for grouping bids, while KG is in charge of key generation only and it does not participate in the auction and crowd sensing process.

Notably, we assume that the adversaries may have certain background knowledge

about the participants' true bids, and we also assume that they are capable of the cryptanalysis. Such adversaries are quite powerful in the attack, and therefore the protection scheme must be strong enough such that no side information is leaked from the communication strings.

3.2.3 Security Model

The security of our system is defined by following standard security game between the adversary and the challenger.

Secure Bidding Game:

- Setup: two disjoint time domains are chosen: T_1 for phase 1, T_c for challenge phase, and T_2 for phase 2.
- Init: The adversary declares that one role in the MCS system will be under his control (*i.e.,* the platform or a participant). The challenger controls the remaining entities in the MCS system. Subsequently, both of them engage themselves in the exchange of public/private parameters according to the protocol specification.
- Phase 1 in T_1: The adversary receives all the communication strings generated during multiple auctions in T_1. The only constraint is that the auctions occur in the time domain T_1.
- Challenge in T_c: The adversary declares any victim participant, and he declares two distinct challenge bids b_0, b_1. The challenger then flips a fair binary coin $\mu = \{0, 1\}$ and generates the disguised bid of b_μ.
- Phase 2 in T_2: Phase 1 is repeated adaptively, but the time window should be chosen from T_2.
- Guess: The adversary gives a guess μ' on μ.

The advantage of an adversary $\mathcal{A}$ in this game is defined as

$$\mathsf{adv}_{\mathcal{A}}^{MCS} = \left|\mathbf{Pr}[\mu' = \mu] - \frac{1}{2}\right| \tag{3.3}$$

Definition 2. *An MCS protocol is indistinguishable against chosen-plaintext attack (IND-CPA) if all polynomial time adversaries' advantages in the above game are of a negligible function w.r.t. of the security parameter λ when T_1, T_c, T_2 are all pair-wise disjoint.*

Intuitively, our security definition indicates that the followings hold in a MCS protocol with IND-CPA.

- Even if adversaries have some knowledge on the distribution of victims' bids, they are still not able to infer any information about the bids from the communication strings.
- Even if temporal or spatial correlation exists in victims' bids, adversaries are not able to link disguised bids whose true bids are correlated to each other.
- Adversaries are not able to learn any information about the victim's private key if they do not know the exact value of the victim's bid.

In other words, the bid disguising is semantically secure against polynomially bounded adversaries and therefore no statistical information about the bids is disclosed.

3.3 Privacy-Preserving Participation Selection

In this section, we present our design of privacy-preserving participation selection, which leverages combinatorial group strategy to find the minimum bid for each task cell in the group while the bid information of every participant is unknown by anyone else except for the participant himself. To preserve privacy, two additional parts, key generator (KG) and third party (TP), are added to the original MCS framework, as shown in Fig. 3.3. The Key generator randomly generates and distributes a series of polynomials outcomes and IDs for all enrolled participants. The third party is the data aggregator, and it calculates the minimum bid among all the participants without the knowledge of each individual bid value. The introducing of KG and TP does not affect the operation of MCS platform. In the view of the platform, TP and KG together are agents of virtual participants (groups).

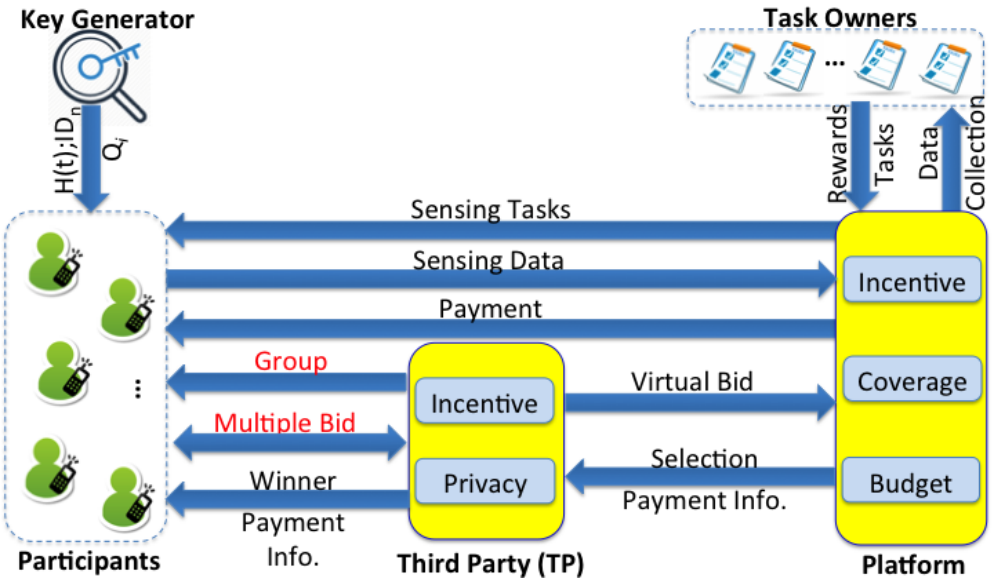


Figure 3.3: **Privacy-preserving participant selection:** each new participant receives her ID, public parameter $H(t)$ and retrieves a set of polynomial values for all requested cells in spatiotemporal matrix S with her ID; task owners give out the tasks and rewards to platform; TP is in charge of grouping and privacy-preserving auction; the platform selects final participants to complete the tasks and make the payments.

3.3.1 Preliminaries

We use the following theories to obtain the minimum bid and the critical payment (second lowest bid) without leaking the bid privacy of participants to any of the other parts, including TP. Further, the calculation could be verified by using fixed point representation.

- Minimum Approximation: For a large integer number R and the upper bound Υ, known by the whole system, the approximation of the minimum number among all the $x_i, i \in [1, I]$ could be obtained by:

$$\Upsilon - \sqrt[R]{\sum (\Upsilon - x_i)^R} \approx \min (x_1 \dots x_i \dots x_I). \tag{3.4}$$

- Lagrange Polynomial Interpolation (LPI): Given a polynomial $Q^j(x)$ with a highest degree j no more than $W - 1$ (i.e., $j \leq (W - 1)$) who passes through the W points $(x_1, q^j(x_1)), (x_2, q^j(x_2)), \dots (x_W, q^j(x_W))$, any other point $(x, q^j(x))$ can

be given by

$$q^j(x) = \sum_{w=1}^{W} \left(q^j(x_w) \prod_{\substack{v=1 \\ v \neq w}}^{W} \frac{x - x_v}{x_w - x_v} \right). \tag{3.5}$$

If $Q^j(x)$ is a polynomial where $q^j(0) = 0$, then the right part of equation is equal to 0, which is

$$\sum_{w=1}^{W} \left(q^j(x_w) \prod_{\substack{v=1 \\ v \neq w}}^{W} \frac{0 - x_v}{x_w - x_v} \right) = 0. \tag{3.6}$$

- Fixed Point Representation: We can transfer one type of fixed point data type with scaling factor A to another data type with scaling factor B by multiplying A and dividing B. We could use the fixed point representation to represent real numbers so that the key in our proposed strategy could be trivially verified as shown in Section 3.3.4.

3.3.2 Sketch of Basic Idea

Our basic idea is inspired by [104]. We let participants form groups first and then the privacy preserving auction is performed within each group. Then the winning bids within each group will be disguised as the virtual participants and submitted to the platform for the final participation selection. VCG auction is performed during both the group bid session and the final participation selection process. The challenges are: (1) how to perform privacy preserving auction within the groups to prevent from leaking the participants' bid information to any party, including TP, and (2) how to make these operations efficient without causing much overhead. Fig. 3.3 shows the structure of our design.

After participants receive the requirements of the sensing task, they will form several groups (with their own group size requirements, which reflects their privacy level). The groups can be formed by either the participants themselves or by TP.

Further, the groups may have different sizes, and a larger group size usually leads to better privacy. For simplicity, hereafter, we consider the group is formulated by TP and the size is a standard system parameter. Each participant in a group uses her group member ID and her related polynomial value to disguise the original bid information. With the feature of LPI pass through origin and the minimum approximation, TP could obtain the minimum bid within each group. The second lowest bid in the group could also be obtained in the same way after excluding the winner. These two bids in each group will be reported to the platform as virtual bids from regular participants. Then the platform uses the VCG method to select the winner and calculates her payment.

Except for the bid winner, all the bid information could be well-protected by our strategy with light overhead and efficient computation. Also, the bid truthfulness could be protected by VCG. Note that this method does not affect the operation of current VCG-based MCS platform. The virtual bids from groups can be treated as regular participants of the platform. Our method can support hybrid participants (both virtual and regular participants) at the platform and also multiple TPs (as different agents) for grouping. In the next section, we will describe the design details of the proposed system.

3.3.3 Detailed Design of Privacy-Preserving Group Bidding

For the simplicity, we omit all the modulo operations, however all numbers appearing in our mechanisms are within a finite field $\mathbb{Z}/p\mathbb{Z}$ where p is a safe prime number of bit length λ, and λ is also denoted as the *security parameter*. We also focus on the a single requested cell (t, l) where $s(t, l) = 1$.

Initialization: KG generates a set of polynomials $Q = \{Q(t, l)\}$, where $Q(t, l) = \{q^2_{(t,l)}(x) \dots q^{\kappa}_{(t,l)}(x) \dots q^{K-1}_{(t,l)}(x)\}$, for the requested cell (t, l) in the spatiotemporal matrix securely, in which all constants are equal to 0. Here K is the upper bound of the group size. The polynomials with same degree κ are distinct from each requested cell.

For the consideration of security, the polynomials need to be updated once they are used in a group, whose overhead will be analyzed in Section 3.4.2. Every participant u_i retrieves her polynomial values Q_i for the whole spatiotemporal matrix with each degree using her ID_i from KG. In requested cell (t, l), each participant u_i holds the polynomial set $Q_i(t,l) = \{q^2_{(t,l)}(\mathrm{ID}_i) \dots q^{\kappa}_{(t,l)}(\mathrm{ID}_i) \dots q^{K-1}_{(t,l)}(\mathrm{ID}_i)\}$. The $q^{\kappa-1}_{(t,l)}(\mathrm{ID}_i)$ for tasks $s(t,l) = 1$ should be used when the group size is κ. The participants only know the values of polynomials on their spatiotemporal matrix with their ID_i but not the polynomial themselves. Also, KG assigns a public parameter V to each participants, which will be used for breaking bid tie later. Note that KG does not participant in the actual auction processes.

Group and disguised bid formulation: Platform broadcasts the tasks S and the bid upper bound Υ. For these tasks, at current time τ, TP randomly allocate the κ participants $(u^j_1, u^j_2, \cdots, u^j_\kappa)$ into a group set U_j (with $U = \cap_{j=1,\cdots,\lfloor \frac{n}{\kappa} \rfloor} U_j$) and asks for the bid information from these participants. The true bid from participant u^j_i in a group U_j is denoted as $b^j_i(t,l)$. Participant u^j_i in group U_j can calculate her disguised bid using the group members' IDs and report her bid to TP:

$$\begin{aligned} &f(ID_i, b^j_i(t,l)) = \\ &q^{\kappa-1}_{(t,l)}(ID_i) \prod_{\substack{o=1 \\ o \neq i}}^{\kappa} \frac{0 - ID_o}{ID_i - ID_o} \cdot H(\tau) + (\Upsilon - b^j_i(t,l))^R, \end{aligned} \tag{3.7}$$

where $H()$, a hash function, is a public secret among the participants.

Winning bid decision within groups: First, TP aggregates all the participants' disguised bids together,

$$\sum_{i=1}^{\kappa} f(ID_i, b^j_i(t,l)) = \sum_{i=1}^{\kappa} (\Upsilon - b^j_i(t,l))^R \tag{3.8}$$

and then uses the Minimum Approximate to find the minimum bid in the current

group.

$$b^j(t,l) = \Upsilon - \sqrt[R]{\sum_{i=1}^{\kappa} (\Upsilon - b_i^j(t,l))^R} = \Upsilon - \sqrt[R]{\sum_{i=1}^{\kappa} f(ID_i, b_i^j(t,l))}. \tag{3.9}$$

Note that all κ, Υ, and R are the public system parameters. After calculating the minimum bid, TP broadcasts it among all the group members.

Next, TP needs to find the winner, find the second lowest bid and break bid ties in each group. To select a single winner and break bid ties, we use the following procedure (also illustrated in Fig. 3.4). The participants whose bids are larger than the winning bid report the pre-assigned value V to TP and assume that the number of values received is F. If $F < \kappa - 1$, TP knows that there is a tie and the second lowest bid in current group is the same as the winning bid. In this case, TP could randomly select one from the participants who did not provide the V value in this step as the group winner since ID is a public parameter. Further, TP can consider other parameters such as credit or worker ability [105] to select the winner when there is a tie. Otherwise, if $F = \kappa - 1$, TP will set the winner's bid to the bid upper bound and repeat the process again to get the second lowest bid in the current group. After this procedure, TP obtains the winner $w^j(t,l)$, its winning bid $b^j(t,l)$, and the second lowest bid $p^j(t,l)$ in this group U_j.

Winning bid decision at platform: For each group U_j, TP will presents two virtual participants (with virtual bids at the winning bid $b^j(t,l)$ and the second lowest bid $p^j(t,l)$) to the platform. The platform receives $2G$ virtual participants' bids for sensing tasks cell (t,l), where G is the number of groups. Then, the platform will make the virtual participants selection and obtain the lowest bid $b(t,l)$ and the critical payment $p(t,l)$ over all participants. The group selection is the same as optimizing participant selection in general case without the third party. The lowest bid (winner)

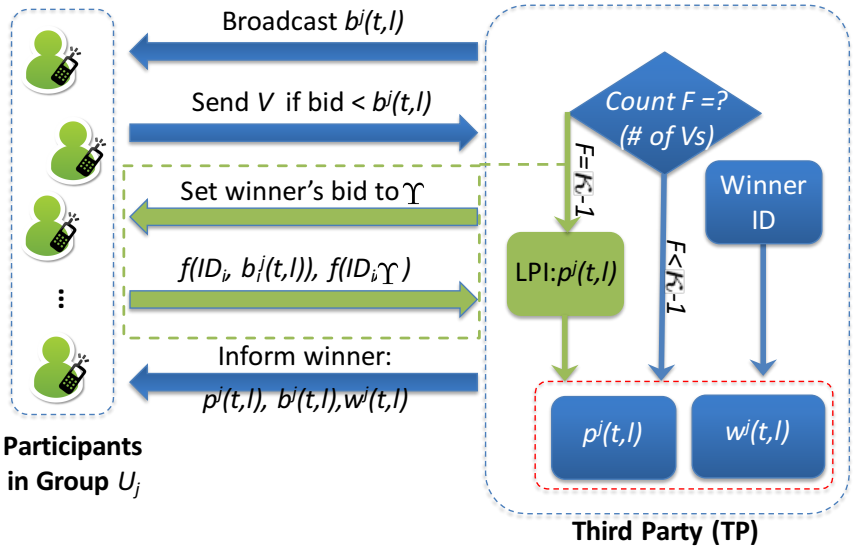


Figure 3.4: **Breaking ties within a group**: TP selects a single winner and obtains the second lowest bid after knowing $b^j(t,l)$. Note that the green exchanges are only needed when there is no tie (i.e. $F = \kappa - 1$).

and the critical payment calculated by platform is described as below:

$$\begin{aligned} b(t,l) &= \min_{j=1}^{G} b^j(t,l), w(t,l) = \arg\min_{j=1}^{G} b^j(t,l) \\ p(t,l) &= \min_{j\in[1,G], b^j(t,l)\neq b(t,l)} \{b^j(t,l), p^j(t,l)\} \end{aligned} \tag{3.10}$$

After the selection decision, platform broadcasts the winning bid (and the winning group) the payment information to the TP. Then TP notifies winner, who then perform the corresponding task. Note that our proposed solution do not affect the selection algorithm (VCG auction) at the platform. The platform can also accept bids from real participants.

3.3.4 Some Critical Issues

$H(\tau)$ and IDs. Note that $H(\tau)$ is the common secret which is known by all participants but not by the third party. $H(\tau)$ could let the whole system be securer since it changes each time when TP aggregates bids from each group. This requires the synchronization among all participants. The participants' IDs are also public in

our system so that they could be directly used in the calculation.

Verification for $q^X_{(t,l)}(ID_i)$. Each participant could only receive the value of polynomials with her own ID_i. Although KG is assumed to be a trusted party, it is possible that an erroneous value is delivered to the participants due to unknown errors. However, since the polynomial is the master secret which is kept hidden to anyone except KG, the participants are not able to verify the correctness the received values. To solve this problem, we extend the zero-knowledge proof (ZKP) [106] and introduce a simple verification protocol below. The protocol allows the participants to verify that the value is indeed calculated from the polynomial owned by KG, but the entire protocol keeps the polynomial itself hidden to the participants.

Key generator publishes the generator g of a multiplicative cyclic group $\mathbb{G}$ where the DDH assumption holds (*e.g.,* a Schnorr group). Then, a series of g^{c_x}'s, where each c_x is the coefficient for ID_i^x, are published. Each participant can calculate the following formula:

$$\prod (g^{c_x})^{ID_i^x} = g^{c_1 \cdot ID_i^1 + \cdots + c_x \cdot ID_i^x + \cdots + c_X \cdot ID_i^X} \tag{3.11}$$

If this value is equal to $g^{q^X_{(t,l)}(ID_i)}$, where $q^X_{(t,l)}(ID_i)$ is the received value from KG before, the participant verifies that the received value is correctly calculated. Because the DDH assumption holds in the group $\mathbb{G}$, no statistical information about c_x is leaked from g^{c_x}, therefore this verification does not tamper the IND-CPA guaranteed by our MCS protocol.

ID Updates. From the formula of the disguised bid, we know that the polynomial value is the only secret except the true bids. As the ranges of polynomial value is much larger than bids, the attackers could estimate the bid value in several rounds with the same polynomial value applied. As a result, we need to update the used ID_i and the related $q^\kappa_{(t,l)}(ID_i)$. In our current system, each participant has multiple

unduplicated IDs. For each participation selection round, the bidders need to mark the polynomial value they used. When the selection with same group size is performed in the same requested cell, the participants should request to renew the polynomial values. We will analyze the involvement of KG in Section 3.4.2.

3.4 Theoretic Analysis

3.4.1 Security Proof

Theorem 1. *Our bid disguising is semantically secure.*

Proof. A bidder's (with ID ID_i) bid $b_i(t,l)$ for the auction occurring at the (t,l) of the spatiotemporal matrix is disguised as the following format:

$$\begin{aligned} &f(ID_i, b_i(t,l)) = \\ &q_{(t,l)}^{\kappa-1}(ID_i) \prod_{\substack{o=1 \\ o\neq i}}^{\kappa} \frac{0 - ID_o}{ID_i - ID_o} \cdot H(\tau) + (\Upsilon - b_i(t,l))^R, \end{aligned} \tag{3.12}$$

when κ bidders participate in the auction. The IDs of the bidders, the current time slot τ as well as the hash function $H(\cdot)$ are public parameters. The only two unknown secrets are $q_{(t,l)}^{\kappa-1}(ID_i)$ and $(\Upsilon - b_i(t,l))^R$. Therefore, multiple disguised bids with distinct polynomial values cannot be used to infer the true bids because there are more unknown variables than the equations.

In reality, there are three cases. First, the disguised bids are received from different auctions occurring at different cells in the spatiotemporal matrix. Second, the auctions occur at the same cell in the spatiotemporal matrix and the number of bidders are different in the auctions. In these situations, different polynomials are used to disguise the bids, therefore the adversaries do not benefit. Third, multiple auctions with the same number of bidders occur at the same cell in the spatiotemporal matrix. Note that every time an auction occurs at a cell at which another auction with the same number of bidders has occurred before, our mechanism ensures that the par-

ticipants' IDs are refreshed, and all participants will receive new polynomial values corresponding to the new IDs. Therefore, even in this case, the disguised bids from multiple auctions are based on different polynomials. In summary, no matter how auctions are performed, combining multiple disguised bids does not help to infer the true bids.

In the sequel, we further prove that our mechanism guarantees semantic security by disguising the true bids . For any single disguised bid $f(ID_i, b_i(t,l))$, let us simplify the terms first. Let $f(ID_i, b_i(t,l))$ be simplified as

$$f(ID_i, b_i(t,l)) = q \cdot \Pi \cdot H + b \tag{3.13}$$

where $q, \Pi, H,$ and b represent $q_{(t,l)}^{\kappa-1}(ID_i)$, $\prod \frac{0-ID_o}{ID_i-ID_o}$, $H(\tau)$, and $(\Upsilon - b_i(t,l))^R$ respectively. Then, for any b and its disguised bid $f(ID_i, b_i(t,l))$, there must exist $b' \neq b, q' \neq q$ such that

$$q \cdot \Pi \cdot H + b = q' \cdot \Pi \cdot H + b' \tag{3.14}$$

Such b', q' exist because of the following reason. Recall that all operations are closed under the finite field $\mathbb{Z}/p\mathbb{Z}$ with a safe prime p. Then, ΠH and p must be coprime, and therefore the inverse $(\Pi H)^{-1} \mod p$ must exist, which implies $q' = (q\Pi H + b - b')(\Pi H)^{-1}$ will make the above equation hold. In other words, for any $b' \in \mathbb{Z}/p\mathbb{Z}$, the disguised bid created with $q' = (q\Pi H + b - b')(\Pi H)^{-1}$ will be exactly the same as b's disguised bid $f(ID_i, b_i(t,l))$.

Recall that the coefficients of the polynomials are chosen from $\mathbb{Z}/p\mathbb{Z}$ uniform randomly. Then, a given disguised bid can be the disguised bid of any valid bid with equal likelihood, which indicates that the disguised bid does not disclose any statistical information about the true bid. □

Theorem 2. *Our MCS protocol guarantees ciphertext indistinguishability against chosen-plaintext attack (IND-CPA).*

Proof. In the aforementioned Secure Bidding Game, although the adversary can adaptively query communication strings corresponding to any input bid he submits, the semantic security of our bid disguising guarantees that he does not gain any statistical information about the bid or the polynomial value. This implies that, even if the adversary submits two challenge bids and receive their disguised bids in Phase 1 or Phase 2, they are not able to statistically correlate them to the disguised bid of b_μ he receives in the Challenge phase. Therefore, his advantage will be a negligible function of the security parameter λ. □

3.4.2 Involvement of Key Generator

As we illustrated in 3.3.3, a participant may need to refresh her ID and polynomial value from the key generator for privacy protection. Note that KG only needs to refresh the parameters for the participant who wants to respond to a task request. The request occurs in the same cell (t, l) of the spatiotemporal matrix S as a previous task that she participated in, and the participant wants to require the same group size κ for both tasks. In the worst case, assume that each participant is willing to bid for tasks falling in each cell of S. Thus, when the participant encounters the same group size at the same cell, she has to contact the key generator to renew her parameters. Therefore, the involvement of KG is influenced by the task distribution over the spatiotemporal matrix ($T \times L$ choices) and participant's required group size κ ($K-2$ choices from 3 to K). We now analyze the average frequency of KG involvement using amortized analysis.

For each cell (t, l) in the $T \times L$ spatiotemporal matrix the possible group size κ varies from 3 to K. Each combination (t, l, κ) can be represented by a box. A task (which the participant want to bid) belongs to a box if it has the corresponding values of t, l, κ. Clearly, there are $D = TL(K - 2)$ boxes.

Now assume that we have m tasks randomly distributed to the D boxes. Let $N(t, l, \kappa)$ be the number of tasks in box (t, l, κ) and let $p(t, l, \kappa)$ be the probability

that a task is located in box (t, l, κ). Note that $N(t, l, \kappa) \sim B(m, p(t, l, \kappa))$, where $B(m, p(t, l, \kappa))$ represents the binomial distribution with parameters m and $p(t, l, \kappa)$. We have

$$\sum_{t,l,\kappa} p(t, l, \kappa) = 1 \text{ and } \sum_{t,l,\kappa} N(t, l, \kappa) = m \tag{3.15}$$

Assuming that each participant participates in each task, then the number of KG involvements corresponds to

$$\sum_{t,l,\kappa} (N(t, l, \kappa) - 1) 1_{[N(t,l,\kappa)\geq 2]}, \tag{3.16}$$

where $1_{[\dots]}$ is the indicator function. We have

$$\begin{aligned}
&\mathrm{E}(\textit{frequency of KG involvement}) \\
=&\mathrm{E}[\sum_{t,l,\kappa} (N(t, l, \kappa) - 1) 1_{[N(t,l,\kappa)\geq 2]}] \\
=&m - D - \mathrm{E}[\sum_{t,l,\kappa} (N(t, l, \kappa) - 1) 1_{[N(t,l,\kappa)\leq 1]}] \\
=&m - D - \sum_{t,l,\kappa} \mathrm{E}[(N(t, l, \kappa) - 1) 1_{[(N(t,l,\kappa)\leq 1]}] \\
=&m - D - \sum_{t,l,\kappa} (-1) P(N(t, l, \kappa) = 0) \\
=&m - D + \sum_{t,l,\kappa} (1 - p(t, l, \kappa))^m.
\end{aligned} \tag{3.17}$$

In the case of a uniform distribution, where $p(t, l, \kappa) = 1/D$ we have

$$\mathrm{E}(\textit{frequency of KG involvement}) = m - D + \frac{(D-1)^m}{D^{m-1}}. \tag{3.18}$$

Similarly, the probability that a task type at (t, l) with group size requested at κ

requests x KG involvement is

$$\begin{aligned} P(x \ KG \ involvements \ in \ box(t,l,\kappa)) &= P(N(t,l,\kappa)=x+1) \\ &= C_m^{x+1}\left(\frac{1}{D}\right)^{x+1}\left(1-\frac{1}{D}\right)^{m-x-1}, \end{aligned} \tag{3.19}$$

where C_m^y refers to y choose m.

3.5 Extension with Quality

Although we only consider the participant selection process with choosing the user with the minimum bid above, there may be lots of participants with low reputation or work ability but bid least. It won't bring the issue of paying rewards in vain since the participants could only earn the rewards once they fulfill the tasks. In such situation, there will be the task incompleteness risk because of users' low work ability and reputations. And further, it will be a waste of time if MCS system could complete the tasks after bidding for several round.

In order to deal with this problem, we could easily extend our framework with one more public sealed parameter, for example, the users' work ability, to guaranteeing the sensing quality and task completeness. We only consider the task completeness of an single task and it's similar to other scenarios with different requirement and multiple tasks. Denote Δ_i is the public working ability of user i(i.e. shown up probability), which will be updated based on whether user i has fulfilled the task or not after he was selected with bid b_i. For each task, there will be criteria θ for its completeness requirement. Then during each group winning bid decision process, the objective function is formed as:

$$\begin{aligned} &\min_i b_i \\ &\arg\min_i b_i \\ &s.t. \ \Delta_i \geq \theta \end{aligned} \tag{3.20}$$

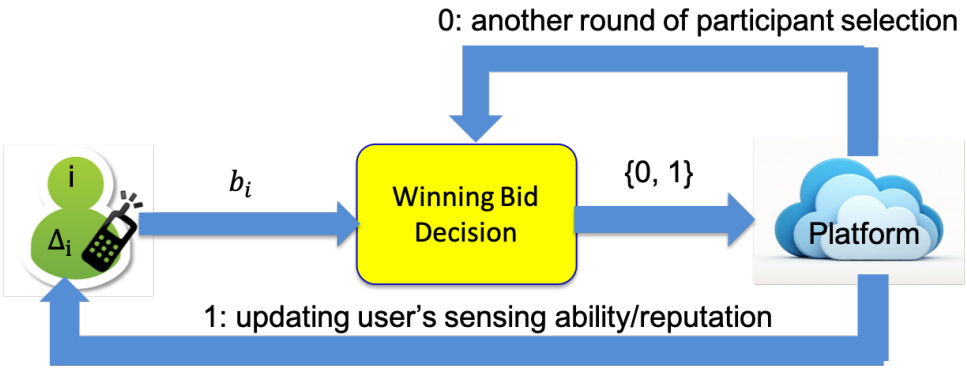


Figure 3.5: **Flowchart of participant selection with user's sensing ability/reputation**

It could be easily solved by making the binary decision so that we will not illustrated here. It will be more complex if the system need to select multiple users. We could use adjustive pace selection for decision making if we could chose one winner in each round. Otherwise, we could also use greedy algorithm to select multiple users in each round since it's a set cover based problem. The flowchart is shown as in figure 3.5. In each round, platform select participant based on bids and their work ability/reputation. If user's sensing ability/reputation is too low, platform will be launch another round of the winning bid selection to choose the user with higher ability and reputations. Otherwise, once the user win the bidding process and is selected to conduct the sensing task, the platform will update his work ability/reputation based on his performance with sensing quality(i.e. the resolution for a sensed photo). With such additional one more parameter, our system could not only minimize the cost for fulfilling the tasks but also guarantee the sensing quality of tasks.

3.6 Summary

In this work, we propose a new privacy-preserving participant selection mechanism for protecting bid privacy of participants in a dynamic auction-based MCS system. By grouping mobile participants into groups with semi-trusted TPs and carefully disguising their bids within the groups, we can achieve scalable selection and guarantee

the overall truthfulness and security while protect the individual bids from participants. The theoretical analysis confirms the efficiency and security of our proposed mechanism.

CHAPTER 4: PRIVACY-PRESERVING PARTICIPANT GROUPING

4.1 Introduction

Secret sharing [72] and grouping bidding [73] has been proposed for participant selection recently. To protect the participants privacy during participation selection, both of them adopt the idea from *k-anonymity*, which the information of each individuals is contained in the release cannot be distinguished from at least $k-1$ individuals whose information also exist in the release, to perform our *participant grouping*. Specifically, all the participants are divided into the small groups based on their requirement (i.e. group size γ_{u_i}). Then privacy-preserving participation selection will be performed within each group and at platform level to select the winners (which will be introduced in detail in the next section). Finally, the winners will perform the assigned sensing task and get the rewards. Note that the participant grouping performance has no influence on the winning bid decision and payment process.

Grouping/clustering has been studied in some recent works [92–94] for privacy preserving. Given a set of n points in general metric space and a value r, the r-gather problem is defined as clustering the points into groups at least r points each such that the largest diameter of clusters are minimized [94]. Aggarwal *et al.* [94] prove that there's a polynomial time algorithm that give a 2-approximation to the problem and show that it's a NP-completeness. Armon [93] extends the result of Aggarwal *et al.* and shows that it's NP-hard to approximate with a ratio better than 3 for $r > 2$ for general metric space. Zeng *et al.* [92] describes a distributed algorithm with an approximation factor of 4 for r-gather problem. All of those existing works cluster the points with only one parameter r, however, in our model, the privacy criteria r is different from each user and the optimization problem is formulated differently. Also,

the existing algorithms (for example sweep algorithm) could not applied here in the hierarchical edge architecture since the location of users' group could be only on its ancestor node but anywhere else. In addition, none of algorithm are proposed for problem *Min-Sum*.

Our contributions can be summarized as follows:

- We model participant grouping into multiple optimization problems, *Min-Max* of group size (or its square) and *Min-Sum* of group size (or its square) under the circumstance of considering the communication cost for group members exchanging their ID or not.

- We propose *Sorting* and prove that *Sorting* is $2 - approximation$ comparing to optimal solution with complexity $O(N \log N)$ for *Min-Max* problem. We also apply *Dynamic Programming (DP)* to solve both *Min-Max* and *Min-Sum* problem with optimal solution and complexity $O(N^2)$.

In this chapter, we introduce the participant grouping problem and solve it with two different grouping/clustering methods. We first define the grouping problem in section 4.2 and then propose two algorithms: one based simple sorting and the other one based on dynamic programming (DP) in section 4.3. Finally, we conclude in Section 4.5. Experiments with real-life datasets (Chapter 6) also confirm that the methods are efficient compared with optimal solution.

4.2 Mobile User Grouping Problem Definition

Assume that there are N mobile users $U = \{u_1, u_2, \cdots, u_N\}$ in our MCS system. For each user, $\gamma(u_i)$ is her group size requirement (privacy requirement) and G_i is the group index of u_i after grouping. After the grouping process, x groups are formed and in each group.

The goal of our privacy-preserving participant grouping is to divide the participants into groups, which satisfy the privacy requirements of all of the participants

while minimizing the total communication cost in the secure bidding process (i.e., participant selection process defined in Section 3.3). Assume that we wind up with x groups, where we call $G^1, G^2, \ldots, G^x$. The total communication cost depends on whether the identification of participants are public or not.Further, we consider two secure bidding/sharing approaches. In the first approach, *the number of messages exchanged in the bidding process* is $O(|G|)$ [73] while it is $O(|G|^2)$ [72] in the second. For simplicity, we use $|G|$ and $|G|^2$ as the communication costs for secure bidding/sharing. Besides, the objective for communication cost is also depend on whether the group is formed in parallel. If perform the secure bidding within groups in parallel, we will only care about the communication cost of the largest group. Hence, the participation grouping problem can be as follows:

$$\begin{aligned} f_{I_1} &= \min \max_{i=1}^{x} |G_i| \\ f_{I_2} &= \min \max_{i=1}^{x} |G_i|^2. \end{aligned} \tag{4.1}$$

When the groups are dealt with sequentially the goals are

$$\begin{aligned} f_{I_3} &= \min \sum_{i=1}^{x} |G_i| \\ f_{I_4} &= \min \sum_{i=1}^{x} |G_i|^2. \end{aligned} \tag{4.2}$$

For all above grouping problem. the constraint is

$$\textit{Subject to: } |G^i| \geq \max_{u_j \in G^i} \gamma(u_j) \tag{4.3}$$

The objective is to minimize the maximal group size, while the constraint is that the number of participants in each group must be larger than or equal to the largest group size requirement of any participant in that group. Note that f_{I_3} is meaningless since $\sum_{i=1}^{x} |G_i| = N$. Note further that it is equivalent to optimize over f_{I_1} and f_{I_2}.

Algorithm 1 Method based on Sorting

Input: each user's requirement $\gamma(u_i)$
Output: each user's group number $g(u_i)$

Sort all of the users into a list L such that their group size $\gamma(u_i)$ is in the descending order; $j = 1$
while $|L| > 0$ **do**
  if $\gamma(u_1) \leq |L|$ where u_1 is the first user in L **then**
    Create a group with the first $\gamma(u_1)$ users in L and remove them from L
    Set these members' group number $g(u_i)$ to j
    $j = j + 1$
  else
    Search in previous groups (from G^{j-1} to G^1) and insert this user u_1 to the first group with a size greater than or equal to $\gamma(u_1) - 1$
    Set the group number $g(u_1)$ to that group and remove this user from L
    Sort all of the groups such that their size is in the descending order
return $g(u_i)$ for all users

For these reasons, we only consider f_{I_1} and f_{I_4}.

4.3 Method Based on Simple Sorting

This algorithm is developed to optimize f, to minimize the size of the largest group. As described in Algorithm 1 (denoted as Sorting), we first sort all of the users based on their requirements. We then create a group to satisfy the requirements of a user with the highest requirement. For the remaining users, if there are enough to satisfy the requirements of the one with the largest requirements, we create a new group. Otherwise, we assign this user to a group with the smallest number of users that nevertheless satisfies her requirement. This process repeats until all users are assigned to groups. The time complexity of this algorithm is $O(n \log n)$ with n participants.

Although this Sorting algorithm can efficiently generate the final grouping, it cannot guarantee an optimal solution. Such an example is given in Figure 4.1 for both f_{I_1} and f_{I_4}. In the example there are 22 users. Algorithm 1 generates two groups with 11 users each, while the optimal solution includes three groups, one with 10 users and two with 6 users. For f_{I_1}, the optimal solution has a maximum group size of 10, while

Figure 4.1: The difference between grouping results of sorting and the optimal solution (also results from DP algorithm).

Sorting has a maximum group size of 11. For f_{I_4}, the cost of the optimal solution is 172 while for Sorting it is 242. Result for f_{I_2} has the same situation since it's the square of f_{I_1}. However, we can prove the following result for the Sorting algorithm's approximation ratios for f_{I_1} and f_{I_2}.

Theorem 3. *Algorithm 1 is a* 2 *approximation for problem* f_{I_1} *and a* 4 *approximation for problem* f_{I_2}.

Proof. We first prove the 2 approximation for f_{I_1}. Let max, opt, $sorting$ be the largest group requirement of all users, the largest group size of the optimal solution, and the largest group size of solution from Algorithm 1, respectively. $max \leq opt$, since the optimal solution needs to satisfy the maximum group requirement. Line 1 of Algorithm 1 first sorts the list L (in the order of decreasing group requirement), then after Line $3-8$ we have x remaining users with the largest group requirement as $\gamma(x)$. Obviously, $x < \gamma(x) \leq max$. In worst case, Algorithm 1 will put all the rest x users into the largest group (first group in L). Therefore, $sorting \leq (max + x) \leq 2max \leq 2opt$. This finishes the proof of 2-approximation for f_{I_1}. Similar proof of 4-approximation can be obtained for f_{I_2}. □

4.4 Method based on Dynamic Programming

Since Sorting does not guarantee an optimal solution, we also propose another algorithm based on dynamic programming (DP) can give the optimal solution. Al-

Algorithm 2 Method based on Dynamic Programming for f_{I_1}

Input: each user's requirement $\gamma(u_i)$
Output: each user's group number $g(u_i)$
1: Sort all users into a list L such that their group size γ is in the ascending order
2: **for** $i = 1$ to n **do**
3: **if** $i - \gamma(L[i]) \geq 0$ **then**
4: **for** $j = 1$ to $i - \gamma(L[i])$ **do**
5: $D[i] = \min\max(D[j], i - j)$
6: Store the group information which achieves $D[i]$
7: **return** $g(u_i)$ for all users based on the stored groups

Algorithm 3 Method based on Dynamic Programming for f_{I_2}

Input: each user's requirement $\gamma(u_i)$
Output: each user's group number $g(u_i)$
1: Sort all users into a list L such that their group size γ is in the ascending order
2: **for** $i = 1$ to n **do**
3: **if** $i - \gamma(L[i]) \geq 0$ **then**
4: **for** $j = 1$ to $i - \gamma(L[i])$ **do**
5: $D[m] = \min\max(D[j], (m - j)^2)$
6: Store the group information which achieves $D[i]$
7: **return** $g(u_i)$ for all users based on the stored groups

gorithm 2 provides the detailed algorithm. After sorting the users based on their requirements (let L be the ordered users with ascending requirements), we use a list D in length n to store the current optimal cost of grouping the list L. In other words, $D[i]$ is the optimal largest group size for grouping users from $L[1]$ to $L[i]$.The relationship for the dynamic program is as follows:

$$\mathsf{D}[i] = \min_{j=1}^{i-\gamma(L[i])} \max(\mathsf{D}[j], i - j) \tag{4.4}$$

The time complexity of DP algorithm is $O(n^2)$ with n participants. We can modify this algorithm to work for f_{I_2} in algorithm 3 and f_{I_4} in algorithm 4 . In this case, we only need to change the recursive function in Line 5 to $\mathsf{D}[m] = \min\max(\mathsf{D}[j], (m-j)^2)$ and $\mathsf{D}[m] = \min\max(\mathsf{D}[m], \mathsf{D}[m-j]+(m-j)^2)$. It is not difficult to show the following theorem holds.

Algorithm 4 Method based on Dynamic Programming for f_{I_4}

Input: each user's requirement $\gamma(u_i)$
Output: each user's group number $g(u_i)$

```
1: Sort all users into a list L such that their group size γ is in the ascending order
2: for i = 1 to n do
3:    if i − γ(L[i]) ≥ 0 then
4:       for j = 1 to i − γ(L[i]) do
5:          D[m] = min max(D[m], D[m − j] + (m − j)^2)
6:       Store the group information which achieves D[i]
7: return g(u_i) for all users based on the stored groups
```

Theorem 4. *Algorithm 2 (or the modified version for f_{I_2} or f_{I_4}) generates optimal groups for optimization problem f_{I_1} (or f_{I_2} or f_{I_4}).*

4.5 Summary

In this chapter, we propose two grouping algorithm, sorting and dynamic programming, for grouping or clustering the participants into small groups so that every group member could conduct the secure bidding process while the communication cost is minimized. We also prove that sorting algorithm could provide a certain approximation with different optimization scenarios and dynamic programming algorithm offers the optimal solution.

CHAPTER 5: GROUPING OVER HIERARCHICAL EDGE CLOUD

5.1 Introduction

With the rapidly increasing use of mobile devices equipped with built-in sensors, mobile crowdsensing (MCS) [1,2] has become a promising paradigm, which is already being used for many applications. It leverages a large number of mobile users to accomplish large-scale tasks. Compared to traditional static sensing, MCS provides better coverage at a lower cost. An MCS system consists of a large pool of mobile users who are willing to perform various sensing tasks and a platform, residing on the cloud, which recruits specific users for a given task. Perhaps the key challenge in the MCS system is participant selection (also called task allocation), i.e., how to select the appropriate users to perform sensing tasks under certain constraints. Many participant selection algorithms have been proposed, taking different issues into consideration (such as coverage [10, 11, 74], energy [28, 29], incentive [18, 52, 59], data collection [9, 13, 63] and truth discovery [66, 107, 108]). In this work, we focus on the important and less studied problem of privacy-preservation in participant selection.

In an MCS system, there are two main categories of sensitive information that a user needs to provide, both of which may lead to privacy breaches. The first is related to the actual sensing data, which has been collected by the participants [21,36,39,41,108]. The second is information related to participant selection. This typically includes bid value [73], which may indicate a user's context (e.g. location or route trace), and sensing quality [72], which reveals the users' ability to perform sensing tasks (e.g. mobile device quality). While there have been several studies on sensing data privacy, participant anonymity, and location privacy, *privacy-preserving participant selection* has rarely been discussed until quite recently [72, 73, 75].

These papers suggest a two-step approach to privacy protection during the participant selection process. The first is *participant grouping* [73] and the second is *secret sharing* [72, 73]. Participant grouping is based on the idea of k-anonymity. Specifically, mobile users are placed into small groups based on their privacy requirement (i.e. desired group size k). After this, a secret sharing (or bidding) protocol is performed within each group to selected the winners. Finally, the group winners will participate in a selection procedure at the platform level, while their identities are hidden by their groups. Such solutions not only guarantee k-anonymity, but also make the secret sharing/bidding more scalable (within each group instead of over all participants). Note that complex encryption methods may not be suitable for this scenario due to their high computation and communication overheads.

Although the idea of participant grouping for privacy-preserving participant selection was first proposed by [73], the previous work did not discuss how grouping is performed. In this work, we carefully study the *privacy-preserving participant grouping* problem for MCS, where participant groups must satisfy the privacy requirements of each user, i.e. the size of the group that a user is placed in should have k or more members, where k is the user's group size requirement. While larger groups have better privacy, they also have more communication overhead and higher computation costs. For this reason, we aim for groups that are as small as possible, while guaranteeing all users' privacy requirement. Furthermore, we consider this problem over hierarchical edge clouds (Figure 5.1).

Existing MCS systems are mainly cloud-based, and participant selection is performed by the platform, which is located on a remote cloud sever. Thus, participants must upload their bids and sensing quality to the cloud in order to take part in the selection procedure. However, this not only leads to privacy concerns, but also causes long communication delays, which are not tolerable for time-sensitive tasks. To mitigate these issues, we propose to use edge computing, which places small-scale servers

at the edge of the network and performs data processing closer to the users. This way, in addition to shorter communication latency, sensitive information will only be submitted to nearby edge servers instead of to the remote platform. This provides another layer of privacy protection for mobile users.

Privacy-preserving participant grouping over hierarchical edge clouds has its own challenges. First, there is an important trade-off in finding an optimal location on the edge cloud. On the one hand, to minimize communication latency, it is preferable for a mobile user's group to be located on a server close to the user, but on the other hand, to ensure greater privacy, one may need to pool users from several locations and thus host their information further away. Second, not only the location, but the size of the group matters. This also requires a trade-off since larger groups provide more privacy, but they require larger computation costs when performing the secret sharing/bidding procedure. Third, the overall costs and loads among edge servers and groups need to be optimized or balanced.

Our contributions can be summarized as follows:

- We model participant grouping into multiple optimization problems, *Min-Max* of group size (or its square) and *Min-Sum* of group size (or its square) under the circumstance of considering the communication cost between hierarchical edge servers and the communication cost for exchanging group members' ID or not. To the best of our knowledge, this is the first work tackling the privacy preserving participant grouping problem for MCS over hierarchical edge clouds.

- Considering the communication cost between edge servers, we propose two heuristic algorithms *Top-Down* and *Bottom-Up* with complexity $O(N \log N)$ to solve the *Min-Sum* problem and they could be easily extended to other problems.

- Experiments results in Chapter 6 validate the proposed approaches using both

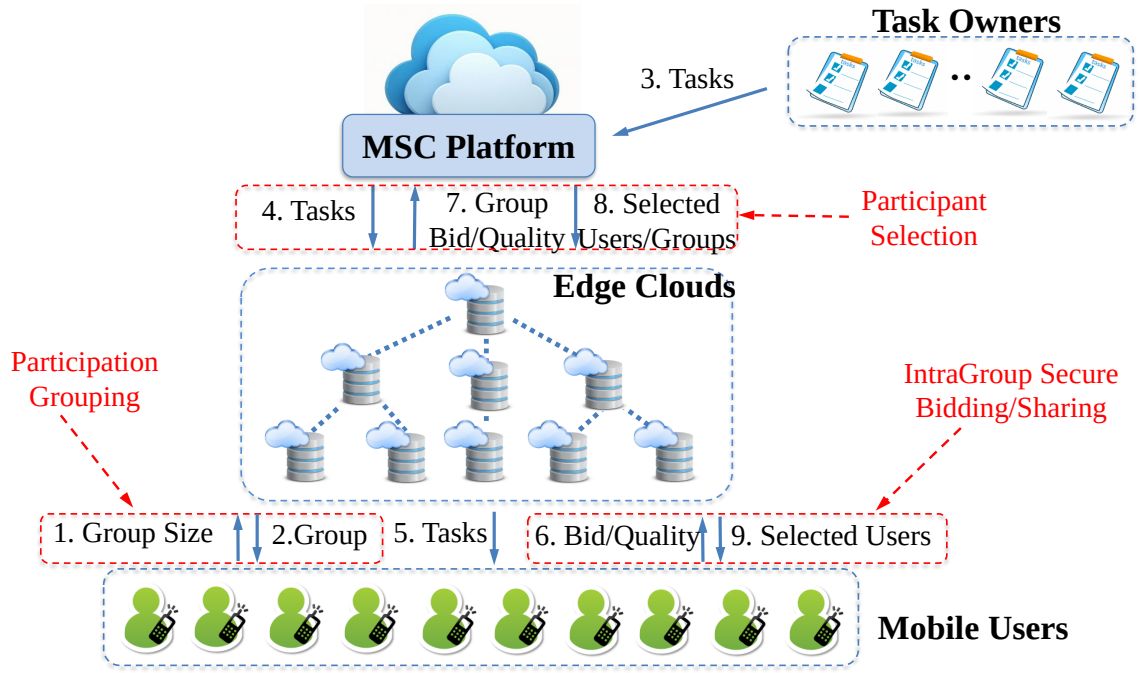


Figure 5.1: **MCS framework over edge clouds** includes three main parts: mobile users (participants), hierarchical edge clouds (acting as a third party to perform grouping and secure sharing/bidding), and the MCS platform on cloud with sensing tasks from requesters (task owners).

synthetic and real-life datasets. It confirms that our methods are efficient for the privacy preserving participant groups, while minimizing or balancing the overall costs over hierarchical edge clouds.

The remainder of this work is organized as follows. In Section 5.2, we give the problem definitions for four different situations. In Section 5.3, we propose and analyze our grouping algorithms. Finally, we conclude in Section 5.4.

5.2 Participant Grouping over Hierarchical Edge Clouds

5.2.1 MCS System over Hierarchical Edge Clouds

As shown in Figure 5.1, the proposed MCS system over the hierarchical edge clouds (MCS-HEC) has three major components: the mobile users, the hierarchical edge clouds, and the MCS platform on the cloud. Here, the hierarchical edge clouds act as a third party to perform grouping for all participants and then facilitate secure

sharing/bidding within each group. The final participant selection is performed by the MCS platform. We assume that both the platform and the hierarchical edge clouds are *semi-honest*, i.e., they follow the protocol but they are curious. In this work, we focus on the task of participant grouping. After grouping the participants, standard approaches for secure sharing/bidding such as those given in [72, 73] can be applied. Figure 5.1 demonstrates the overall flows in our MCS-HEC system.

5.2.1.1 Hierarchical Edge Clouds

We represent the hierarchical edge clouds by a tree with multiple tiers (also called levels), consisting of a set of M edge server nodes $V = \{v_1, v_2, \cdots, v_M\}$, as shown in Figure 5.2. Recall that the edge nodes are semi-honest and this only has influence on the secure bidding process but not grouping. The number of children for each node varies and is determined by the edge network architecture. For each node v, we denote its level by l_v, its ancestor node set and descendant node set by A_v and D_v respectively. Further, we define $A_{l,v}$ ($D_{l,v}$) as the ancestor (descendant) nodes of node v at tier l. We set $l = 0$ for the root node, and there are I levels in total. We assume that the participants are directly connected to the leaf servers in the hierarchical clouds. The higher the tier (with a smaller level) that a server is located at, the more resources it can provide for computation but with longer communication delay.

5.2.1.2 Mobile Users

Assume that there are N mobile users $U = \{u_1, u_2, \cdots, u_N\}$ in our MCS edge system. When user u_i joins the system, she chooses the nearest edge server v to join the MCS platform and sends in her registration data $(\gamma(u_i), s_0(u_i), s(u_i), g(u_i))$. Here $\gamma(u_i)$ is her group size requirement (privacy requirement), $s_0(u_i)$ is the edge node where her registration data is stored at the beginning, $s(u_i)$ is the edge node where her final group sits, and $g(u_i)$ is the group index of u_i after grouping. Initially, the latter two are empty.

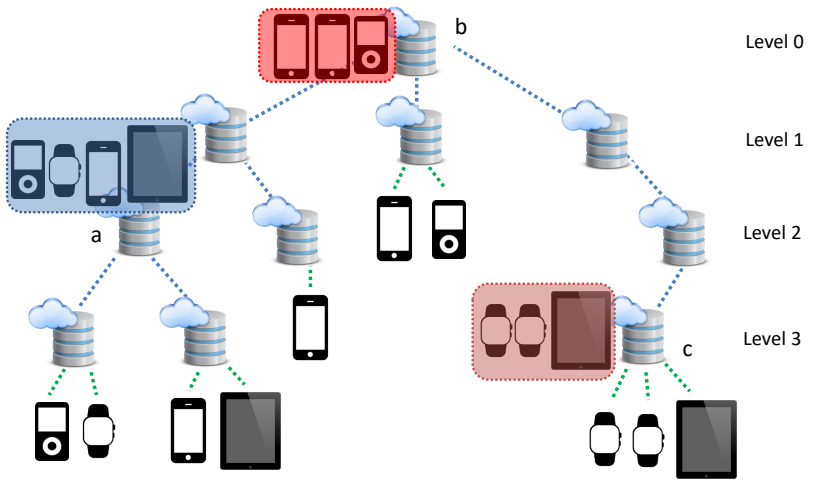


Figure 5.2: **Participant grouping over hierarchical edge clouds:** ten users are connected to the edge servers at different tiers, and they are divided into three groups at servers a, b & c.

During the grouping process, an edge server v can put its user u_i on a node at a higher tier if $\gamma(u_i)$ can not be satisfied locally or a node at a higher tier needs more participants. Due to the limitations of the tree structure, v can only put its users on one of its ancestor nodes ($A_v \subseteq V$). After the grouping process, each user u_i is placed into a group $G_{g(u_i)}$ which sits on edge server $s(u_i)$. Here, $s(u_i) \in \{s_0(u_i)\} \cup A_{s_0(u_i)}$ and $0 \leq l_{s(u_i)} \leq l_{s_0(u_i)}$. Assume that we wind up with x groups, which we call $G_1, G_2, \cdots, G_x$. Note that $\sum_{i=1}^{x} |G_i| = N$ and $|G_i| \geq \max_{u \in G_i} \gamma(u)$ (i.e., the number of participants in each group is larger than or equal to the largest group size requirement of any participant in that group).

5.2.2 Grouping Problems over Hierarchical Edge Clouds

The goal of our privacy-preserving participant grouping is to divide the participants into groups, which satisfy the privacy requirements of all of the participants while minimizing the total communication cost in the secure sharing/bidding process (i.e., participant selection process). In this chapter, we consider communication costs between edge nodes in to the total communication cos. Further, we still consider two secure bidding/sharing approaches and use $|G|$ and $|G|^2$ as the communication costs

for secure bidding/sharing.

Different with former chapter, we consider the delay among different levels of edge servers here. Although the servers at higher tiers can provide better privacy coverage, the communication between different tiers may cause long delays. Here, we use the level difference between $l_{s(u_i)}$ (where the final group sits) and $l_{s_0(u_i)}$ (where the mobile user originally sits) to represent the delay of a particular user u_i. In this case, the four types of optimization problems can be defined as following:

$$\begin{aligned} f_{II_1} &= \min \max_{i=1}^{x} \sum_{u_j \in G_i} (l_{s_0(u_j)} - l_{s(u_j)}) \\ f_{II_2} &= \min \max_{i=1}^{x} (|G_i|^2 \max_{u_j \in G_i} (l_{s_0(u_j)} - l_{s(u_j)})) \\ f_{II_3} &= \min \sum_{i=1}^{x} \sum_{u_j \in G_i} (l_{s_0(u_j)} - l_{s(u_j)}) \\ f_{II_4} &= \min \sum_{i=1}^{x} (|G_i|^2 \max_{u_j \in G_i} (l_{s_0(u_j)} - l_{s(u_j)})). \end{aligned} \tag{5.1}$$

Here, for simplicity, we assume that there is at most one group per edge server and the maximal level difference of a group is used to estimate the delay within the group.

Constraints for All Scenarios: The following constraints are applicable to all the objectives listed above.

$$\begin{aligned} &\forall G_i, \quad |G_i| \geq \max_{u \in G_i} \gamma(u) \\ &\forall u_i, \quad s(u_i) \in \{s_0(u_i)\} \cup A_{s_0(u_i)} \end{aligned} \tag{5.2}$$

The first constraint is that for each group, the number of its members should be equal to or larger than the largest group size requirement among its users. The second constraint is that each user's final group sits either on its original edge server or its server's ancestor nodes. Note that, for all scenarios, we assume that there exists at least one feasible solution, for example, if all of the users are grouped into a single

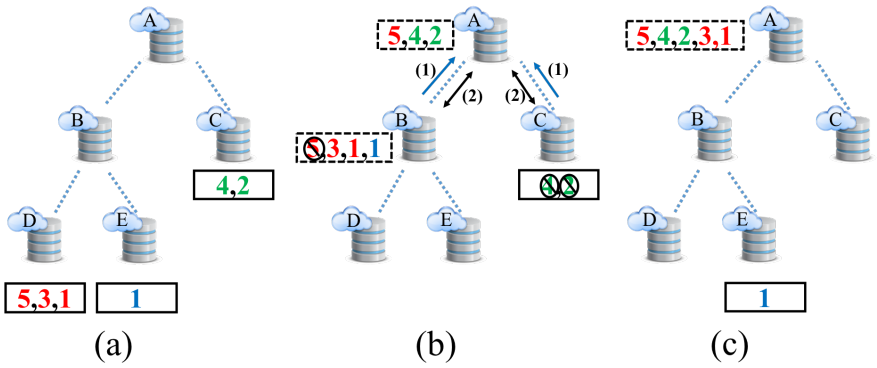


Figure 5.3: **Example of Top-Down Algorithm:** (a) 6 users on leaf nodes of 3-tier edge tree at beginning; each number represents the group size requirement of a user; (b) when processing all users at the root (Line 10-20 in Algorithm 5), 5, 4 and 2 cannot fit in the root's children groups, thus are left at the root in step 1 by blue arrow; when call Algorithm 6 at the root (as shown in Figure 5.4), 2 more users are brought to the root in step 2 by interaction black arrow; (c) the left user with requirement 1 will be remained to leaf when the TD is called at next level.

group (at the root) then everyone's privacy requirement can be met.

5.3 Grouping Algorithms for MCS-HEC

In this section, we propose several heuristic grouping algorithms to solve the optimization problems with feasible solutions effectively. *genetic* algorithms and *stimulated annealing* algorithms cannot be applied here since the feasible solutions are limited and the neighbor (or the mutation) of current solution is infeasible with high probability(with the assumption that the number of users on a server is less than the largest group size requirement). Hence, we propose two heuristic algorithms: *top-down* and *bottom-up*. While these are developed to optimize f_{II_3}, they can be used to get solutions for all of the objective functions. Further, we extend these two algorithms to get solutions that are geared for f_{II_2}. Similar methods can be developed for the other cases in Scenario II.

5.3.1 Top-Down Algorithm

The top-down algorithm is given in Algorithm 5. Its main idea is to arrange the users on the edge cloud server tree starting at the root and ending at the leaf nodes.

We begin at the root and consider all of its children. At each child we place all of the users such that that child is an ancestor of the leaf that the user starts on. If there are users at these children, whose requirements are not satisfied, they are moved back to the root. After this step the requirements of all users in groups on the children are satisfied. However, the requirements of some users in the root group may not be. If at least one user on the root has a requirement that is not satisfied, then we must move one or more users from the children to the root. The specific way that we do this is described by Algorithm 6. After this, all requirements of all users are satisfied and all users are either on the root or one of its children. We then repeat this procedure treating each child the same way that we treated the root. Figure 5.3 illustrates a simple example of the Top-Down algorithm, while Figure 5.4 shows the corresponding example of Algorithm 6. Note that the sorting in Algorithm 6 can be done via a global sorting in the beginning of Algorithm 5 at cost $O(N \log N)$. Then since each user is moved at least one level during the repeat procedure, the time complexity after sorting is only $O(N)$. Hence, the overall time complexity of Algorithm 5 is $O(N \log N)$.

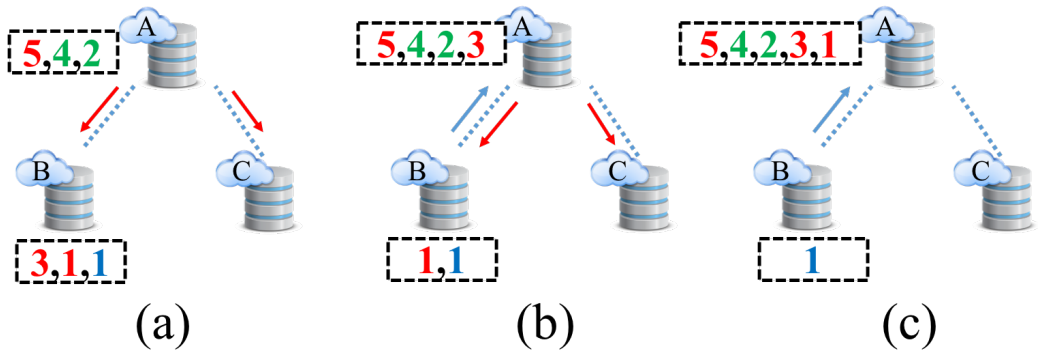


Figure 5.4: **Example of Algorithm 6:** (a) node A sends the request to node B and C to request more users; (b) nodes B offers his additional user with 3 to A and C has nothing to offer, and thus A chooses to accept the 3 from B. Since the requirement at A is not satisfied yet, A asks again; (c) finished after B contributes one more user from node B. Red arrow indicates the request and blue arrow indicates offloading.

Algorithm 5 Top-Down Algorithm for Scenario II

Input: each user's requirement $\gamma(u_i)$ and location $s_0(u_i)$
Output: each user's final group location $s(u_i)$ and group number $g(u_i)$

Each node v creates an empty group G_v
Place all users in the group at the root
for $l = 0$ to $I - 1$ **do**
  for all nodes v at level l **do**
    $\mathsf{TD}(v, l)$
return $s(u_i)$ and $g(u_i)$ for all users

Function $\mathsf{TD}(v, l)$
  if v is a leaf node **then**
    return
  for all $u \in G_v$ **do**
    place u in $G_{v'}$, where v' is the unique child of v that is an element $A_{s_0(u)}$
  for all v' that are children of v **do**
    while $\max_{u \in G_{v'}} \gamma(u) > |G_{v'}|$ **do**
      Move all $u \in G_{v'}$ with $\gamma(u) > |G_{v'}|$ from $G_{v'}$ to G_v
  if $|G_v| < \max_{u \in G_v} \gamma(u)$ **then**
    Call Algorithm 6 with node v and level $l + 1$
EndFunction

5.3.2 Bottom-Up Algorithm

The bottom-up algorithm is given in Algorithm 7. Its main idea is to first place all users into their lowest possible positions and then move them up as necessary to ensure that their requirements are satisfied. Specifically, we begin by placing all users into groups at their starting locations on the leaves. If a user cannot be satisfied at the leaf, that user is moved up to the parent node. In the end, a leaf node is either empty or it has a group, where all users have their requirements satisfied. After this we repeat the procedure at the next level, but with one important difference. This time we check if the user's requirements would be satisfied not just by all users on this node, but by all users on this node and any of its descendants. After we finish, we may have some users whose requirements are not met. Now, starting at the root, we modify the groups to ensure that all requirements are met. The details are given in Algorithm 7, and an example is illustrated in Figure 5.5. Similarly, the

Algorithm 6 Update G_v at node v

Input: Node v and level l' (along with all information about all users and groups)

while $|G_v| < \max_{u \in G_v} \gamma(u)$ **do**
  for all $v' \in D_{l',v}$ **do**
    Define the set $S_{v'} = \emptyset$
    Take any $u \in G_{v'}$ with $\gamma(u) = \max_{u' \in G_{v'}} \gamma(u')$ and place it into $S_{v'}$
    while $\max_{u \in G_{v'} \setminus S_{v'}} \gamma(u) > |G_{v'} \setminus S_{v'}|$ **do**
      Take any $u \in G_{v'} \setminus S_{v'}$ with $\gamma(u) = \max_{u' \in G_{v'} \setminus S_{v'}} \gamma(u')$ and place it into $S_{v'}$
  Sort the v' in order of increasing $|S_{v'}|$
  if $\max_{v'} |S_{v'}| \leq (\max_{u \in G_v} \gamma(u) - |G_v|)$ **then**
    Choose the last v' in the order and move all $u \in S_{v'}$ from $G_{v'}$ to G_v
  else
    Choose the first v' in the order such that $|S_{v'}| \geq (\max_{u \in G_v} \gamma(u) - |G_v|)$ and move all $u \in S_{v'}$ from $G_{v'}$ to G_v

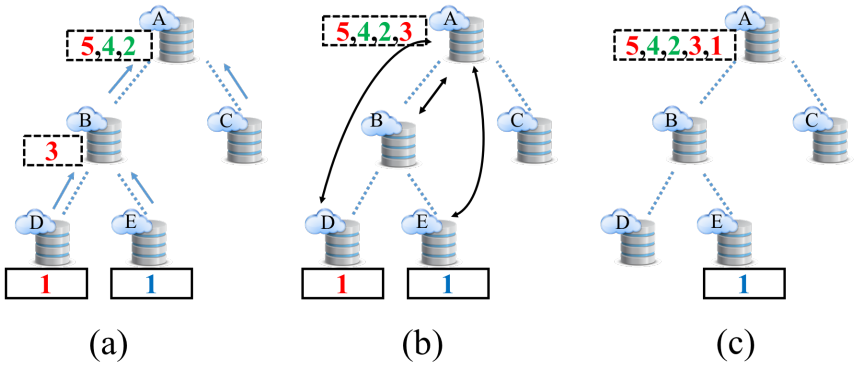


Figure 5.5: **Example of Bottom-Up Algorithm:** consider the same example in Figure 5.3(a); (a) first move all each user to the upper level node where the total users from descendants could satisfy its requirement; e.g., the user with 3 could be satisfied on node B since its descendants D and E have two more users; (b) then run Algorithm 6 for each node where the users are unsatisfied starting from root; here the root gets the users with 3 from B and 1 from D; (c) all the users are met their group size requirement.

time complexity of Bottom-Up algorithm is $O(N \log N)$ too.

5.3.3 Extended Algorithm for f_{II_2}

The outcomes of Algorithms 5 and 7 ensure that every user's requirements are satisfied. Further, they aim to minimize the total level differences. However, they can be improved upon when optimizing f_{II_2} (where the square of group size matters).

The algorithm is given in Algorithm 8. The idea is that we first run Algorithm 5 or

Algorithm 7 Bottom-Up Algorithm for Scenario II

Input: each user's requirement $\gamma(u_i)$ and location $s_0(u_i)$
Output: each user's final group location $s(u_i)$ and group number $g(u_i)$

1: Each node v creates an empty group G_v
2: **for all** users u **do**
3: Place u into $G_{s_0(u)}$
4: **for** $l = (I-1)$ to 0 **do**
5: **for** all nodes v at level l **do**
6: **if** $|G_v| = 0$ **then**
7: continue
8: Set $H_v = G_{v'} \cup \left(\bigcup_{v' \in D_v} G_{v'}\right)$
9: **while** $\max_{u \in G_v} \gamma(u) > |H_v|$ **do**
10: Move all $u \in G_v$ with $\gamma(u) > |H_v|$ from G_v to G_{v_p} where v_p is the parent of v
11: **for** $l = 0$ to $(I-1)$ **do**
12: **for all** v at level l **do**
13: **if** $G_v = \emptyset$ or $|G_v| \geq \max_{u \in G_v} \gamma(u)$ **then**
14: Continue
15: **for** $l' = l+1$ to $I-1$ **do**
16: **if** $\sum_{v' \in D_{l',v}} |G_{v'}| + |G_v| < \max_{u \in G_v} \gamma(u)$ **then**
17: move all users from $\bigcup_{v' \in D_{l',v}} G_{v'}$ to G_v
18: **else**
19: Call Algorithm 6 at v and l'
20: Break
21: **return** $s(u_i)$ and $g(u_i)$ for all users

Algorithm 7 to get a feasible solution. Then, to further reduce the objective function, we repeatedly find the node v whose corresponding group has the highest cost. We then reduce its cost by distributing some of its users to other nodes. We try all possible combinations of user subsets of v, which can satisfy the new requirement at v after partition (moving the subset to other nodes).Then we try to move the subset of users to one of their ancestor. As Line 6 in Algorithm 8, we can select the ancestor with least increasing cost (we call it Extended-LI). Or we can select the ancestor that can lead to the most decrease cost of current node v (such variation is called Extended-MD). We repeatedly decrease the maximal cost until no further reduction is possible.

Algorithm 8 Extended Algorithm for f_{II_2}

Input: each user's requirement $\gamma(u_i)$ and location $s_0(u_i)$
Output: each user's final group location $s(u_i)$ and group number $g(u_i)$

1: Perform Algorithm 5 or 7 to get a feasible solution
2: $max = \max_{i=1}^{x}(|G_i|^2 \max_{u_j \in G_i}(l_{s_0(u_i)} - l_{s(u_i)}))$ and $v = \arg\max_{i=1}^{x}(|G_i|^2 \max_{u_j \in G_i}(l_{s_0(u_i)} - l_{s(u_i)}))$
3: **repeat**
4: Put all users $u \in G_v$ (the group on node v) into list L and sort them based on $\gamma(u)$
5: Consider all possible user subsets from partitioning the users in L from its head or tail such that the remaining users in L can still satisfy the requirement
6: For each possible user subset, check whether moving them to their common ancestors can still satisfy the new requirement there. If so we pick the one whose cost increases least after serving these new users, and move these users to that ancestor's group
7: $max = \max_{i=1}^{x}(|G_i|^2 \max_{u_j \in G_i}(l_{s_0(u_i)} - l_{s(u_i)}))$ and $v = \arg\max_{i=1}^{x}(|G_i|^2 \max_{u_j \in G_i}(l_{s_0(u_i)} - l_{s(u_i)}))$
8: **until** max does not decrease anymore and all nodes v with max cost have been tried
9: **return** $s(u_i)$ and $g(u_i)$ for all users

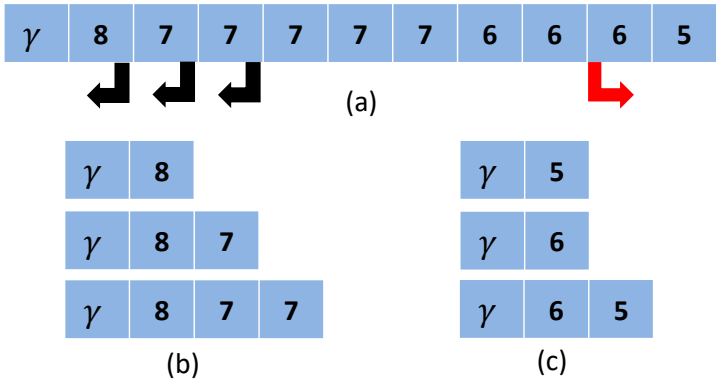


Figure 5.6: **User Partition in Algorithm 8:** (a) the original sorted user list L at v; (b) the subset of users by partitioning L from the head; (c) the subset of users by partitioning L from the tail. Algorithm 8 will try all possible subsets.

5.4 Summary

In this work, we developed several novel participant grouping mechanisms for protecting the privacy of participants over hierarchical edge clouds in mobile crowd sensing. By introducing the privacy-preserving grouping, not only can the participants be hidden in groups, but also the secure sharing/bidding within a group can be

performed more efficiently. Our design goal is to minimize the communication cost during secure sharing or bidding while satisfying the user requirements for privacy preservation. For different scenarios and optimization functions, we propose a set of grouping schemes to fulfill the goal and guarantee users' privacy requirements.

CHAPTER 6: PERFORMANCE EVALUATION

6.1 Simulations for Participant Selection

6.1.1 Dataset and Configuration

D4D Dataset: D4D dataset is a mobile phone call tracking data, from the Orange for the *Data for Development (D4D) challenge* [109]. The data is anonymized call detailed records of phone calls between $50,000$ Orange mobile users in Ivory Coast between December 1, 2011 and April 28, 2012. We use a dataset of individual mobile phone call tracking trace with high spatial resolution (*SET2* in D4D datasets), which contains the access records of antenna (cellular tower) of each mobile user in every two weeks. Since the density of phone call is very sparse, we merge records from multiple weeks into a single week and use one week (7 days) as the whole sensing cycle T. There are $46,613$ records for the user showing up in all cellular towers without duplication, and the number of such users is $10,704$. We assign the location of each task randomly from the locations of 18 cellular towers, which are with the highest call records. Most of these towers are located in the downtown region of Abidjan. We treat the distance between a participant (her current tower) and the task (its location at one of the 18 towers) as the bid value[1] for each participant to that task. In other words, when a participant is far away from a task location, her cost to perform the sensing task is high. Since the records of mobile phone call (tower location) is not the exact position of participants, in our simulations, we add an additional random distance with range $[0, 1]$ to the estimated distance as the original bid value.

SFC Dataset: Although D4D dataset provides a real-life large scale traces for

[1]Note that the bid value can be others, e.g., users' ability to perform the task. Here we just use the distance as an example, which is easy to obtain from both datasets.

Table 6.1: Parameters of D4D and SFC Simulations used in privacy-preserving participant selection.

Parameter	Value or Range (D4D)
Unit of time/Task duration	1 day
Number of locations (towers)	18
Number of tasks M	$60, 80, 100, 120, 140$
Number of candidate participants N	$2000, 4000, 6000, 8000, 10000$
Group size K	$20, 40, 60, 80, 100$
Length of whole sensing cycle	one week (7 days)
Number of data records	46613
Total period of traces used	Dec 5, 2011 to Jan 8, 2012
Parameter	**Value or Range (SFC)**
Unit of time/Task duration	$10, 20, 30, 40, 50, 60$ Minutes
Number of tasks M	$60, 80, 100, 120, 140$
Number of candidate participants N	504
Group size K	$10, 20, 30, 40, 50$
Length of whole sensing cycle	one day
Number of data records	508979
Total period of traces used	May 17, 2008 to June 10, 2008

human mobility, it does not have high spatial resolution (still at cellular tower level). Therefore, we also use the San Francisco Cab (SFC) Dataset [110] for simulations, which includes the GPS traces (total $11,200,335$ data records) from 536 cabs in total 25 days from $May 17, 2008$ to $June 10, 2008$. We believe that SFC can provide complemental scenarios for our simulations. Here, we use a subset of all traces (tailored both on temporal and spacial domains), which has 504 participants with $508,979$ data records. Since the GPS records are accurate locations, we randomly generate the locations of sensing tasks and use the distance between the participant and the task as the true bid.

Table 6.1 summarizes the parameter settings. R is set to $1,000$ times the largest bid.

6.1.2 Compared Methods and Metrics

In our experiments, we compare our proposed participant selection method with three alternative mechanisms: PRIDE [111], the group mechanism with trusted third

party (TTP) and the location obfuscation method (Noise). PRIDE is a privacy-preserving and strategy-proof spectrum auction in cognitive radio networks, which leverages complex cryptographic techniques (such as secure multiparty computation, order-preserving encryption, and oblivious transfer) to obtain the lowest bid and preserve bid privacy. We have adopt it to our scenario and use RSA with modulus of 1024 bits for encryption/decryption. In TTP, we introduce a completely trusted third party to perform group bidding. All information about participants such as bid, ID and spatiotemporal matrix, are transparent to TTP. It could absolutely protect the participants' privacy from platform but rely on TTP entirely. Noise applies a standard privacy preserving technique, adding noises (range from 0 to 10) in the bids (i.e. the distance between the participant and the task) from each participants.

We test all these methods under different settings (with various number of participants, number of tasks, group size, and task period), and evaluate them with the following metrics.

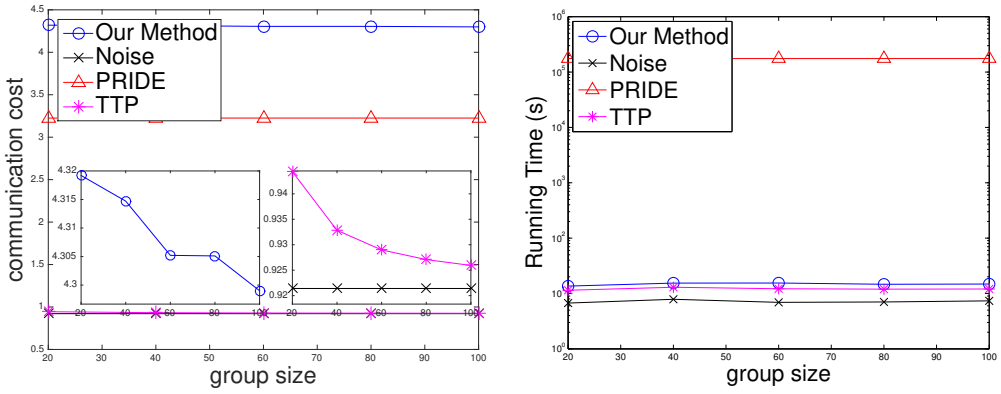
Running time: the time between the tasks is broadcast and all participants have been selected. Here we assume that the participant selection algorithm is the same for all method, picking the participant with the smallest bid as the winner.

Communication cost: the communication costs in all steps, including task broadcast, group formation, winner decision and second bid calculation in each group, and winner decision for tasks. It is measured as the average round of message exchanges from each participant per task.

Overpayment/Accuracy: since $b(t, l)$ acts as the bid of winner $w(t, l)$ and $p(t, l)$ as the related payment to her for this task, the overpayment for this task is defined as $p(t, l) - b(t, l)$. Then the total average overpayment is an average over all tasks.

6.1.3 Simulation Results

We first test the performance of all methods using both D4D and SFC datasets in terms of communication cost and running time. Simulation results are shown in

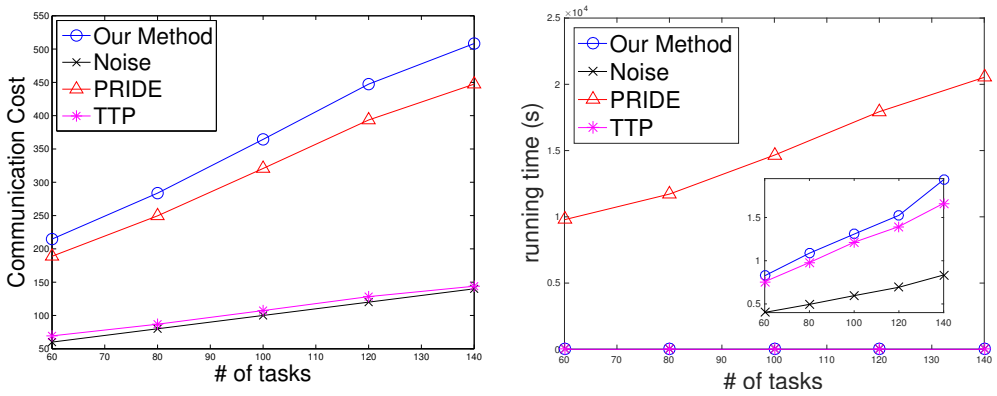


(a) communication cost (b) running time

Figure 6.1: **D4D simulation with different group sizes:** (a) average communication cost per participant per task and (b) running time per task with different group sizes over 100 tasks and 6,000 participants.

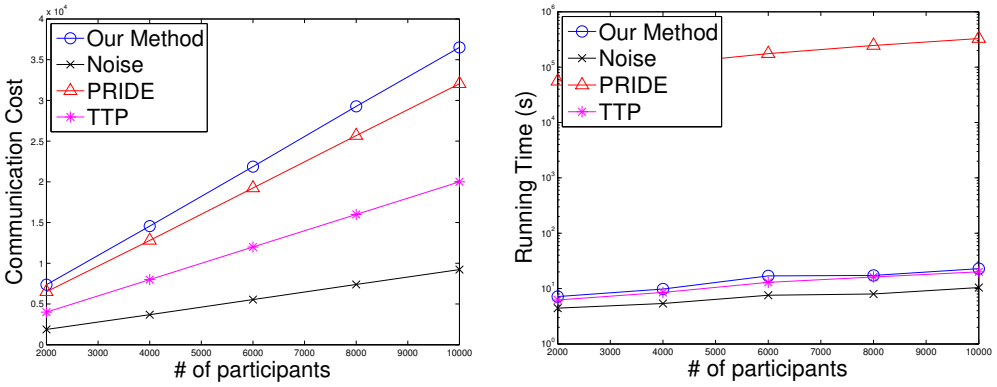
Fig. 6.1 ,Fig. 6.2, Fig. 6.3 ,Fig. 6.4 and Fig. 6.5, respectively. In Fig. 6.1(a) and Fig. 6.4 (a), we consider we consider the average number of message exchanges per task per participant with different group sizes, as shown in Fig. 6.1(a) and Fig. 6.4(a). First, the communication costs of Noise and PRIDE do not change with group size, while those of TTP and our method decrease with the growth of group size as the virtual participants on behalf of each group decease with the larger group size. Compared with TTP or PRIDE, our method needs more message exchanges to achieve privacy preserving. However, the larger group is, the more computation time for obtaining the polynomials value. Hence, for running time (Fig. 6.1(b) and Fig. 6.4(b)), our methods are similar with TTP which is increasing slightly longer than Noise (mainly for group creation and group bidding). However, PRIDE takes significantly more time than other methods because its encryption process is time consuming. In addition, with increasing number of tasks, more time is needed for all methods. Overall, our method can achieve privacy-preserving with similar running time but larger communication cost compared with TTP. The communication overhead is the price for privacy-protection.

We also measure the communication cost for different number of tasks and par-



(a) communication cost (b) running time

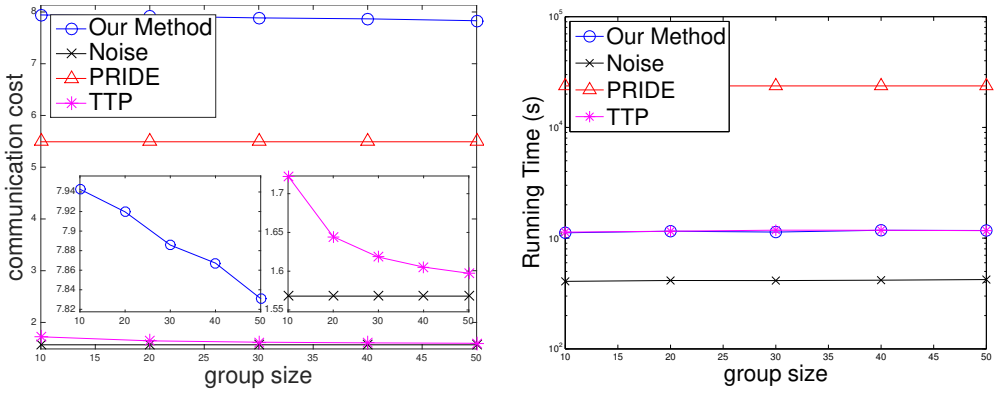
Figure 6.2: **D4D simulation with different number of tasks:** (a) average communication cost per participant and (b) total running time per participant per task with various number of tasks, 6,000 participants and group size at 60.



(a) (b)

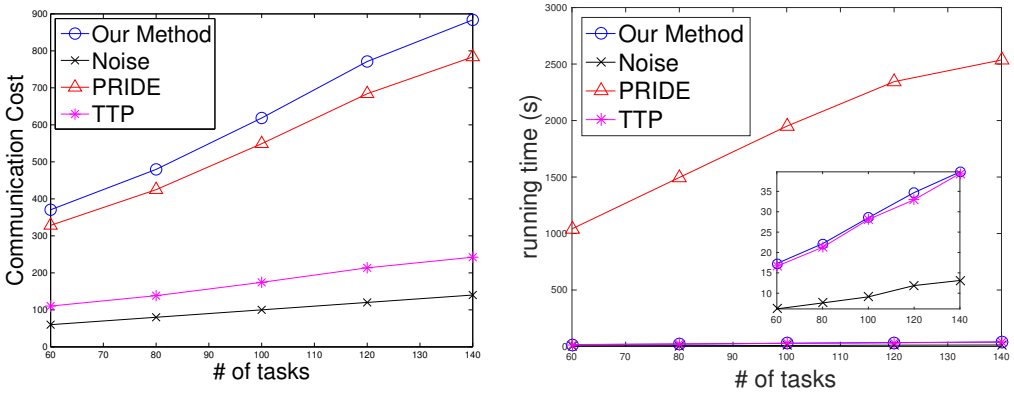
Figure 6.3: **D4D simulation with different number of participants:** (a) average communication cost per task and (b) total running time per task with different number of participants when group size is fixed at 60 with 100 tasks.

ticipants. In Fig 6.2 and Fig. 6.5, first, with the total tasks increasing, the total communication cost and running time for all methods rise for sure. Then the communication cost of our methods is higher than the others because the participants in our model need to exchange the message with TP or KG for privacy protection. As participants need to send their bids to third party first in TTP, hence the communication cost is slightly higher than Noise method. Nevertheless, among all the methods, running time of PRIDE is much longer than others for that the encryption and decryption is time consuming.



(a) communication cost (b) running time

Figure 6.4: **SFC simulation with different group sizes:** (a) average communication cost per task per participant and (b) total running time per participant with different group sizes over 100 tasks and 504 participants.

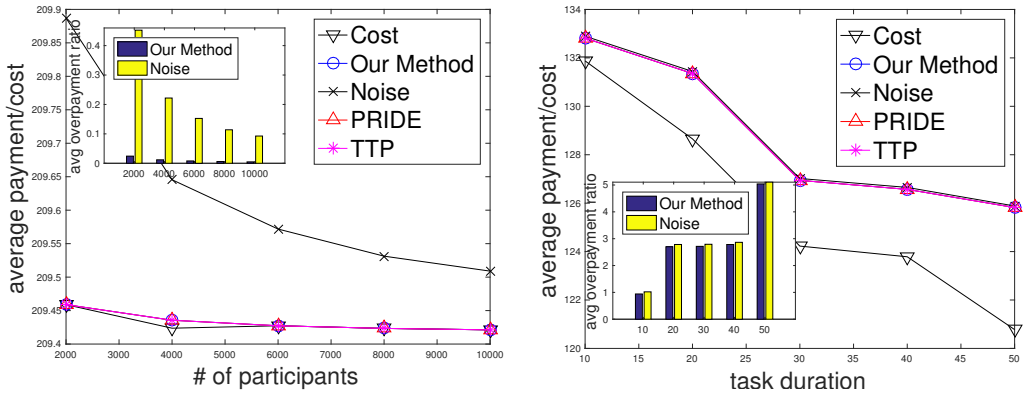


(a) (b)

Figure 6.5: **SFC simulation with different number of tasks:** (a) average communication cost per task and (b) total running time with different number of tasks when group size is fixed at 30 with 504 participants.

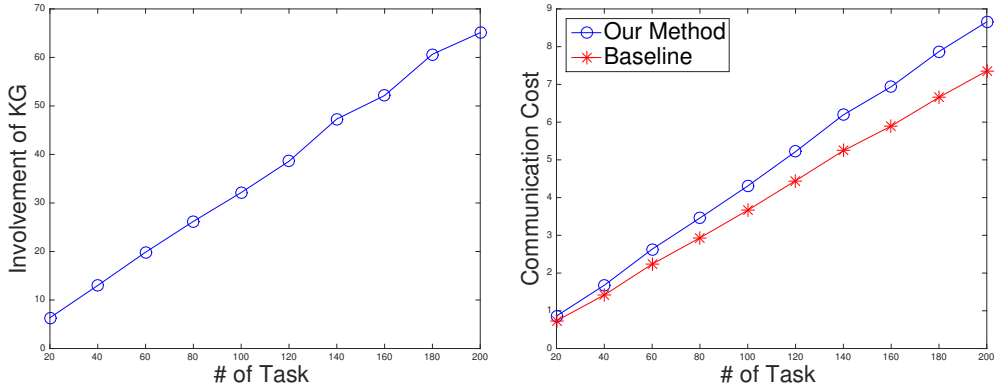
Beside, we test the communication cost and running time with different number of participants for D4D dataset when the group size and tasks number are fixed in Fig. 6.3. With more participants, more time is needed for more information exchanging. Even communication cost of our method is higher than the others (Fig. 6.3(a)), our running time is much lower than PRIDE (Fig. 6.3(b)).

We also measure the payments of different methods and compare them with the true cost. Results are shown in Fig. 6.6. First, both cost and payment decrease with the increase of number of participants (Fig. 6.6(a)) and the task duration (Fig. 6.6(b)).



(a) D4D (b) SFC

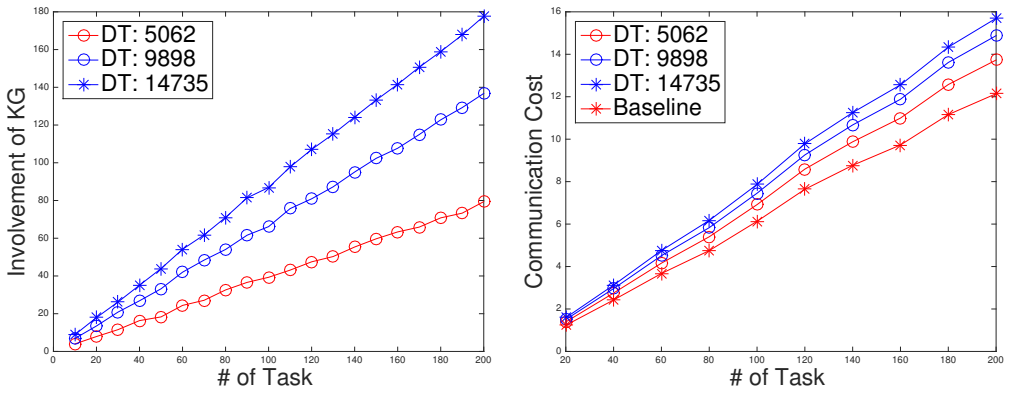
Figure 6.6: **Average cost and payment** for all the tasks by different methods in (a) D4D simulations and (b) SFC simulations. Smaller plots show the overpayment ratios of our method and Noise.



(a) KG involvement (b) communication cost

Figure 6.7: **D4D simulation:** KG involvement and communication cost with different number of tasks.

With more participants, the platform/group can choose lower minimum bid and pay less rewards to the winners. With longer task duration, participants have more chances to bid less. Further, our method, TTP and PRIDE pay the same amount, and Noise pays the most. This can be clearly seen in the smaller plots within the figures, which show the overpayment of our method and Noise. Obviously, Noise sacrifice the overpayment to protect the privacy. Note that the difference of overpayments between Noise and our method is not significant in SFC simulations. This may be due to that the range of random noises is much smaller than the distances (true bids)



(a) KG involvement (b) communication cost

Figure 6.8: **SFC simulation:** KG involvement and communication cost with various DT value and the number of tasks.

in SFC dataset.

Last, we consider the involvement of KG. Recall that when a participant wants to bid a task which has the same spatiotemporal requirement and desired group size with a previous task she bided, the participant needs to refresh her ID and polynomial values from KG. In the following experiments, we fix the group size and consider the effect of the number of tasks. Fig. 6.7 and Fig. 6.8 clearly show that, more involvements of KG (also extra communication cost) are needed when there are more tasks. This confirms the theoretical analysis we had in Section 3.4.2. Here, the baseline is the method without any refreshing of IDs and values. Since we do not have tower location as the task location for SFC dataset, we consider the distance among tasks instead for refreshing decision. We assume that a participant needs to refresh her ID and values from KG when the distance between the current task she wants to bid and any of her former tasks is less than the predefined distance threshold (DT). As shown in Fig. 6.8, with larger distance threshold, both KG involvement and communication cost become larger. This is reasonable, since the current task will interfere with more tasks in the larger range, which then results in more involvements of KG and more message exchanges.

6.2 Simulations for Participant Grouping

In this chapter, we evaluate the performance of the proposed algorithms on three different datasets (one synthetic and two real-world datasets). We begin by describing the datasets and simulation settings, then we compare and analyze the performance of our algorithms for both scenarios.

6.2.1 Dataset and Configuration

D4D Dataset: We use the D4D dataset for our experiment. We first pick 18 cellular towers with the highest call records as edge severs, and organize them into a 4-tier tree as shown in Figure 6.10. Hierarchical edge cloud could also be constructed with additional servers and we only use the towers in the dataset here. There are $6,880$ users access the edge cloud at the 9 leaf nodes with $50,898$ individual call records. Without loss of the generality, we average our simulation results by running on 34 different tasks with different time requirements and each of them has 300 users who are willing to participant. In addition, we assign the group size (privacy requirement) to each user with a uniform distribution with range of $[0, r]$, where $r = [30, 60, 90, 120, 150, 180]$.

6.2.2 Performance Metrics

We used the following metric to compare the performance of the proposed algorithms under different simulation settings. Note that we report the average values of these metrics over multiple rounds of simulations.

Objective Function: The cost of performing secure sharing/bidding is defined in Equation (4.1), Equation (4.2), and Equation (5.1) for different cases. For simplicity, when it is clear from the context we write f instead of f_{I_1}, f_{I_2}, f_{I_3}, f_{I_4}, f_{II_1}, f_{II_2}, f_{II_3}, and f_{II_4}.

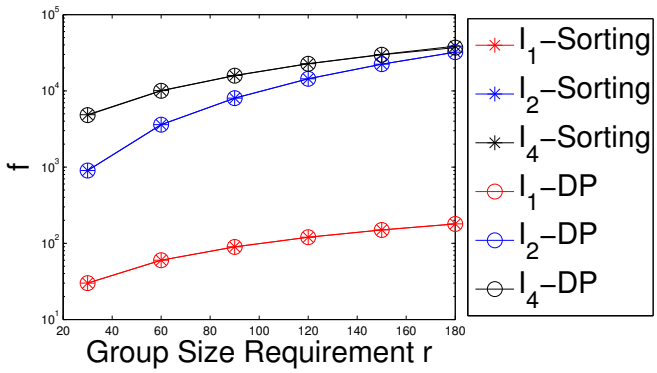


Figure 6.9: **Participant grouping with D4D:** simulation results of Algorithm 1 (Sorting) and Algorithm 2 (DP) with different maximal group size requirement r from 30 to 180.

6.2.3 Simulation Results

We test the performances of Algorithm 1 (Sorting) and Algorithm 2 (DP) for participant grouping. This set of simulations are performed on real-world D4D dataset. We use total 300 mobile users with the group size follows a uniform distribution with range of $[0, r]$, where r is from 30 to 180. Results are given in Fig. 6.9. As the maximal group size requirement r increases, the cost f for performing secure bidding also increases for the objective function. This is reasonable because with larger r, the users has probability to chose the larger group size requirement and more users are needed to form a group. Furthermore, the performances of these two algorithms is almost identical. Note that in chapter 4, we have showed that Sorting cannot guarantee to find the optimal solution while DP can. However, in this set of simulations, due to the uniform randomness of group size requirements, the cases where the optimal solution cannot achieved by Sorting (as those in Fig. 4.1) do not occur. Notice that when the maximal group size in optimal solution is the same with the maximal group size requirement, Sorting can indeed find the optimal solution. Last, in this set of simulations, we also observe that Algorithm 1 runs much faster than Algorithm 2 does, as theoretical analysis confirms.

6.3 Simulation for Grouping over Hierarchical Edge Clouds

6.3.1 Datesets and Configuration

Synthetic Dataset: In this dataset, 50 mobile users are uniformly distributed on 4 leaf nodes in a 3-tier balanced binary tree. This small dataset is mainly used for testing our methods for Scenario II against a Bruce Force method. The group size assigned to each user follows a uniform distribution with range of $[0, r]$, where $r = [10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30]$ respectively.

D4D Dataset: The part of the dataset that we used includes the records between December 5, 2011 and January 8, 2012. Further, we pick 18 cellular towers with the highest call records as edge severs, and organize them into a 4-tier tree as shown in Figure 6.10[2]. There are $6,880$ users access the edge cloud at the 9 leaf nodes with $50,898$ individual call records. Without loss of the generality, we average our simulation results by running on 34 different tasks with different time requirements and each of them has $100 - 500$ users who are willing to participant. In addition, we assign the group size (privacy requirement) to each user with a uniform distribution with range of $[0, r]$, where $r = [30, 60, 90, 120, 150, 180]$. The parameters for tasks and participants are summarized in Table 6.2.

SFC Dataset: We also use the San Francisco Cab (SFC) dataset [110] for our grouping over hierarchical edge cloud simulations. In particular, we used the data on June 6, 2008, which has the most users and data records, 504 and $508,979$ respectively. To generate the location of edge servers, we use a 16-grid map of San Francisco to generate a 3-tier balanced quad tree with 16 small grid cells as its leaves. We average all of our results by running 310 tasks with $50 - 300$ users. Since the number of users are much smaller than the one in D4D, hence, the group size for SFC follows the uniform distribution as well but with a smaller r value, where $r =$

[2]Hierarchical edge cloud could also be constructed with additional servers and we only use the towers in the dataset here.

Table 6.2: Parameters of D4D and SFC Simulations used in participant grouping over hierarchical edge clouds.

Dataset	Parameter	Value or Range
D4D	number of accessible towers	9 out 18 towers
	# of candidate N for f_{II_3}	$100, 200, 300, 400, 500$
	# of candidate N for f_{II_2}	50
	number of data records	$50,898$
	total number of tasks	34
	max group size req. r for f_{II_3}	$30, 60, 90, 120, 150, 180$
	max group size req. r for f_{II_2}	$2, 3, 4, 5, 6, 7$
	total period of traces used	Dec 5, 2011 to Jan 8, 2012
SFC	number of regions	16
	# of candidate N	$50, 100, 150, 200, 250, 300$
	number of data records	$508,979$
	total number of tasks	310
	max group size req. r for f_{II_3}	$20, 40, 60, 80, 100, 120$
	max group size req. r for f_{II_2}	$2, 3, 4, 5, 6, 7$
	total period of traces used	June 6, 2008

$[20, 40, 60, 80, 100, 120]$. Table6.2 shows the details of parameter settings.

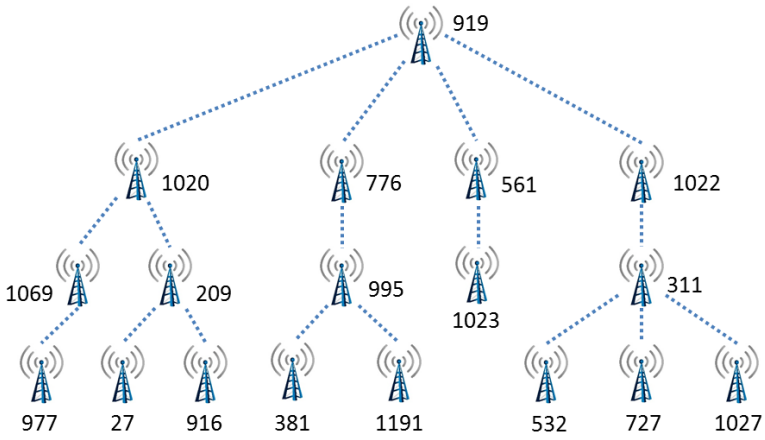
6.3.2 Performance Metrics

We used the following metrics to compare the performance of the proposed algorithms under different simulation settings. Note that we report the average values of these metrics over multiple rounds of simulations.

Grouping Ratio: This metric is used to measure how many levels for all the users (groups) are moved in the tree, and is denoted by:

$$\frac{f_{II_3}}{\sum_u l_{s_0(u)}} = \frac{\sum_u (l_{s_0(u)} - l_{s(u)})}{\sum_u l_{s_0(u)}} = 1 - \frac{\sum_u l_{s(u)}}{\sum_u l_{s_0(u)}}.$$

This metric is mainly used for f_{II_3} in Scenario II. Note that *group ratio* is ratio between f_{II_3} and its largest value (*i.e.* when all users are placed in the root's group and the leaf level is the largest) at beginning. Note that *group ratio* is a value between 0 and 1, smaller value means less moving (better and smaller cost f_{II_3} as well). When *group ratio* equals to 1, everyone moves to the root. When it equals to 0, everyone



D4D tree

Figure 6.10: **Hierarchical Edge Clouds for D4D dataset:** a 4-tier tree with 18 edge nodes, (i.e., towers with shown tower IDs).

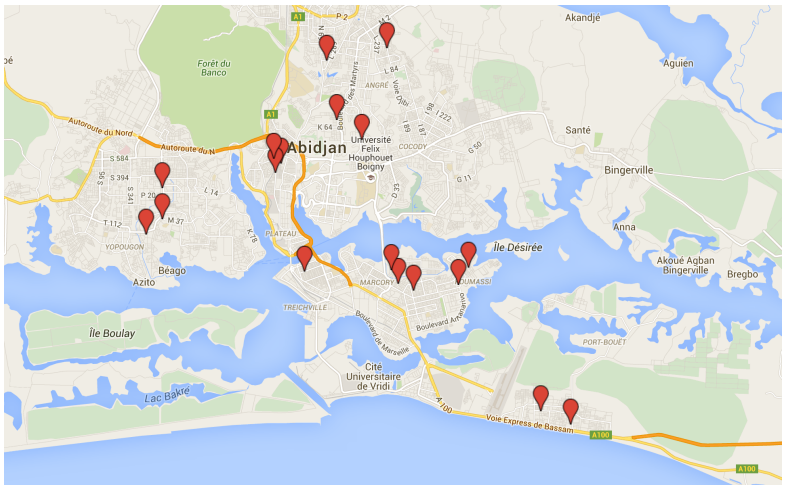
stays at the leaf nodes.

Objective Function: The cost of performing secure sharing/bidding is defined in Equation (4.1), Equation (4.2), and Equation (5.1) for different cases. For simplicity, when it is clear from the context we write f instead of f_{I_1}, f_{I_2}, f_{I_3}, f_{I_4}, f_{II_1}, f_{II_2}, f_{II_3}, and f_{II_4}.

Iteration Times of Algorithm 8: In Algorithm 8, we need to repeatedly distribute the participants from the edge server with the most costing group to other nodes (Lines 3-8) until no further improvement. Hence, we evaluate efficiency of the algorithm by measure the number of iterations needed to converge. A smaller number of iteration is preferred for Algorithm 8.

6.3.3 Simulation Results

Simulation Results for f_{II_3}: We first evaluate the performances of Algorithm 5 (Top-Down) and Algorithm 7 (Bottom-Up) by comparing it to a brute force algorithm on the small synthetic dataset for the objective f_{II_3}. Figure 6.13 shows the results of grouping ratios η of these three methods. Note for f_{II_3}, the grouping ratio η is just a scale of the cost f_{II_3}. As shown in Figure 6.13, η rises as the group size requirement

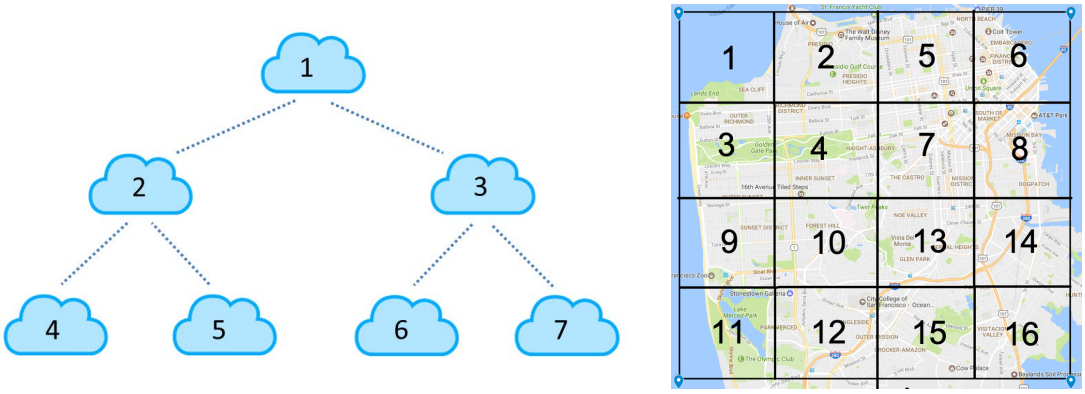


Egde nodes in D4D

Figure 6.11: Locations of cell towers near Abidjan used edge nodes in our simulations

r increases. This is because the users need to move towards the root when their group size requirements increase. Further, we can see that both proposed algorithms perform closely to the optimal solution obtained by the brute force algorithm. This demonstrates the efficiency of our proposed Top-Down and Bottom-Up methods.

Next, we evaluate the performance of these algorithms on two real-life datasets with larger number of users. Since Brute Force cannot perform on these datasets due to large search spaces, we also include results of Algorithm 1 (Sorting) for reference. Recall that Sorting ignores the level of edge structure during the grouping process. To make it suitable for Scenario II, we set each group to the common ancestor node of all users in the same group and merge groups on the same node to a single group. Simulation results are shown in Figure 6.14 and Figure 6.15 for D4D and SFC dataset, respectively. Since Sorting ignores the level of edge structure, it performs much worse than the other two algorithms on both datasets. Further, we can see that, all curves are analogous on these datasets. As the total number of users increases, the probability that a participant could be satisfied on a lower edge server node is higher, thus, the grouping ratio decreases. In addition, as the group size requirements increases, the users need to move towards the root to satisfy the requirements, as a result, the group-



(a)Synthetic tree (b) SFC quad tree

Figure 6.12: **Hierarchical Edge Clouds for Synthetic dataset and San Francisco dataset:** (a) a 3-tier binary synthetic tree with 4 edge nodes and (b) a 3-tier quad tree with 16 edge nodes (i.e, the 16 grid cells over maps).

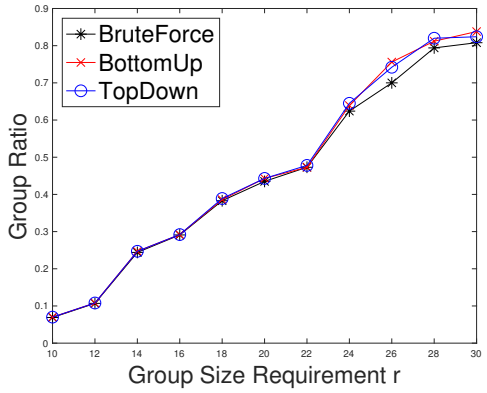
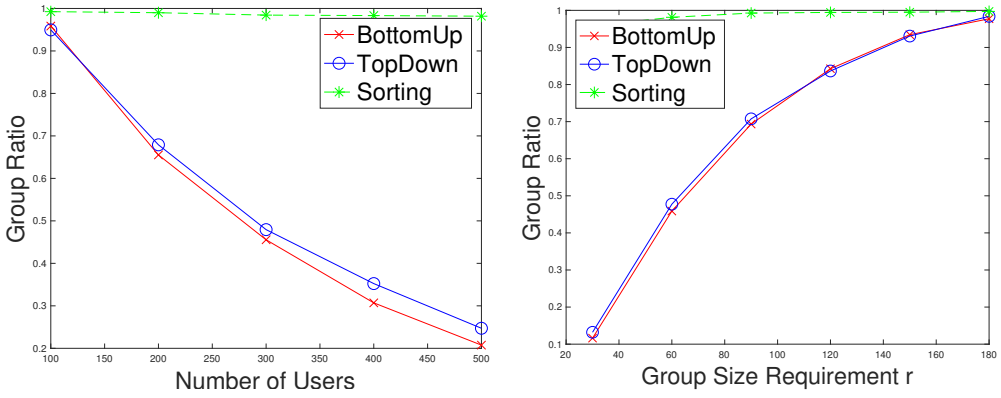


Figure 6.13: **Synthetic data simulation for f_{II_3}:** grouping ratios of Brute Force, Top-Down and Bottom-Up with 50 participants and maximal group size requirement r from 10 to 30.

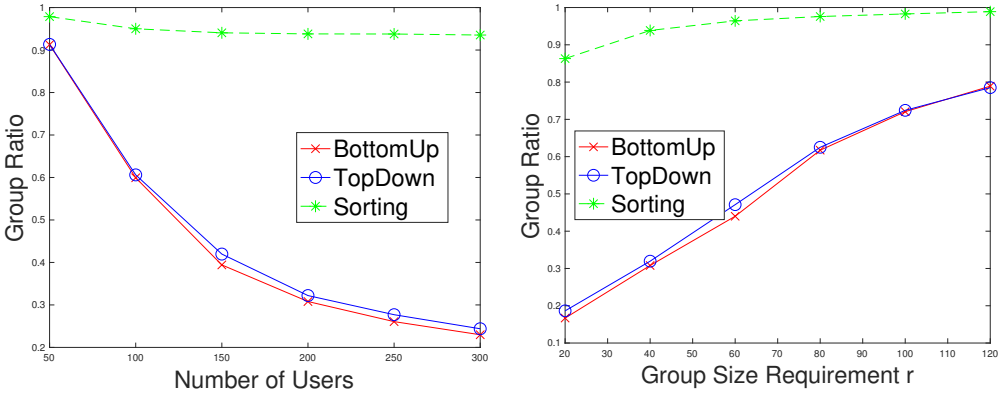
ing ratio also increases. There are no significant difference between performances of Top-Down and Bottom-Up, while Bottom-Up slightly performs better.

Simulation Results for f_{II_2}: Finally, we compare the performance of Algorithm 8 (two versions: Extended-LI and Extended-MD) to that of Algorithm 7 (Bottom-Up) on the two real-world datasets for f_{II_2}. Results are given in Figure 6.16 and Figure 6.17. Here we directly measure the cost f. As the group size bound increases, all of the cost curves rise. This is due to that we need larger group sizes and high levels to satisfy the users' requirements. Further, Extended-MD demonstrates better



(a) (b)

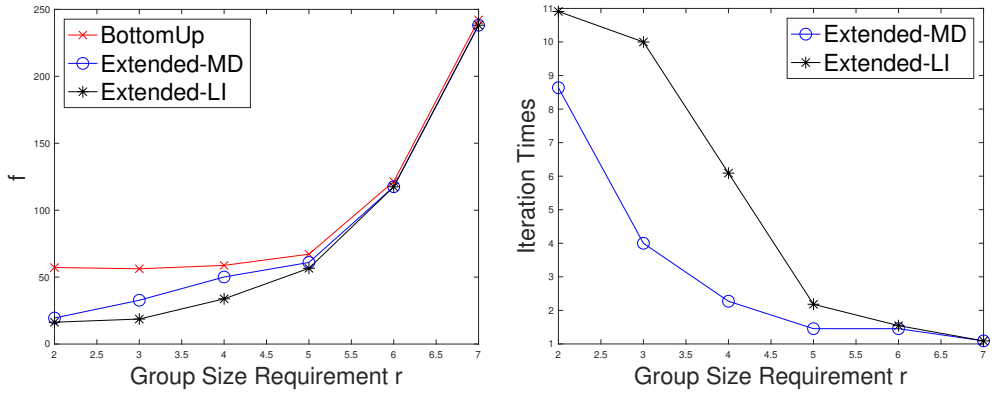
Figure 6.14: **D4D simulation for f_{II_3}:** (a) grouping ratios with maximal group size requirement $r = 60$ and various number of users; (b) grouping ratios with 300 participants and different maximal group size requirement r.



(a) (b)

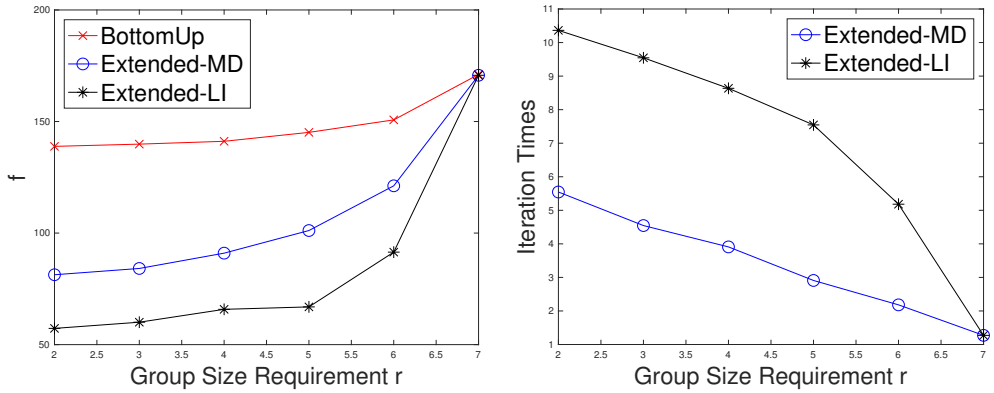
Figure 6.15: **SFC simulation for f_{II_3}:** (a) grouping ratios with maximal group size requirement $r = 40$ and various number of users; (b) grouping ratios with 200 participants and different maximal group size requirement r.

performance than Extended-LI. This is because the longer the participants list is, the less opportunity that the participants has common ancestor and then it is more likely that vibration happens, *i.e.*, the receiver node sends the same list of users to the sender node. Despite a better performance, Extended-MD also requires significant higher number of iterations than Extended-LI does. In addition, as the group size requirement increases, the iteration times decreases for both Extended-LI and Extended-MD, since there are less receiver nodes that have enough participants to satisfy the users requirements. Most importantly, both Extended-LI and Extended-



(a) (b)

Figure 6.16: **D4D simulation for f_{II_2}:** (a) cost f_{II_2} and (b) the iteration times of Algorithm 8 with different maximal group size requirement r.



(a) (b)

Figure 6.17: **SFC simulation for f_{II_2}:** (a) cost f_{II_2} and (b) the iteration times of Algorithm 8 with different maximal group size requirement r.

MD can achieve better performance than BottomUp.

6.4 Summary

In this section, we conduct the simulation for those different problems with real-life datasets, D4D dataset and San Francisco dataset. For experiment of privacy-preserving participant selection, we compare our method with other three existing methods and confirm the effectiveness and efficiency of our framework. For participant grouping, we analyze the two proposed algorithm with D4D dataset and confirm the theoretical analysis. At last, we extend and confirm our proposed algorithm for participant grouping over hierarchical edge clouds based on three different datasets.

We use a small size synthetic dataset to confirm the effectiveness of our heuristic solutions and then further implement it on the real life dataset with several different metrics and various hierarchical edge architectures. All of those simulation results testify and validate the scalable participant selection efficiency with privacy preserved.

CHAPTER 7: CONCLUSION AND FUTURE WORK

As we address, one of the key challenges in MCS is how to select the participant among the huge mobile users pool with considering the appropriate incentive, scalable system, and privacy preserved. Hence, in this work, we mainly focus on designing scalable privacy-preserving participant selection with appropriate incentive for mobile crowd sensing system.

Since auction base selection has been widely used for current MCS systems to achieve user incentive, following the classical VCG auction, we carefully design a scalable grouping based privacy-preserving participant selection scheme, where participants are grouped into multiple participant groups and then auctions are organized within groups via secure group bidding. By leveraging Lagrange polynomial interpolation to perturb participants' bids within groups, participants' bid privacy is preserved. In addition, we analyze the bidding game of our proposed solution with three implications to prove the security. Finally, we extend our scheme with consideration of sensing quality.

Besides, to address the participant grouping problem with the constraint of communication cost during participant bidding process, we propose two algorithms: sorting and dynamic programming (DP). We prove that sorting algorithm could efficiently achieve a feasible solution with certain approximation ratios. Dynamic programming algorithm is proved to provide the optimal solution with time complexity $O(N^2)$.

In addition, in order to avoid the high overheads and poor scalability suffering from cloud-based MCS platform and enhance the protection of user privacy, we further propose a set of novel privacy-preserving grouping methods, which place participants into small groups over hierarchical edge clouds. By doing this, not only can the

participants be hidden in groups, but also the overall privacy-preserving participant selection becomes more scalable. Our design goal is to group participants in a way that minimizes the communication cost during secure sharing/bidding, while satisfying each participant's requirement for privacy preservation. For different scenarios and optimization functions, we propose a set of heuristic grouping algorithms to fulfill this goal.

For all of above work, extensive simulations over both synthetic and real-life datasets are conducted to verify the efficiency and security, and confirm the effectiveness of proposed mechanisms.

With the developing of MCS, more and more applications or platforms are designed for different type of tasks. In the future, we will work on designed a scheme for distributed participant selection. Besides applying the caching to enhance the performance of the increasing and duplicated tasks, another methods is inferring the sensing value by leveraging the correlation between each task since sometimes they may be similar to each other on the spatial domain or temporal domain or even both. In order to enhance performance and save energy, we will work on integrating MCS system with some other applications, such as point-of-interest-tagging applications, about the participant recruitment and incentive scheme. At last, we will make efforts to apply other privacy definitions with measurable leakage degree (e.g. differential privacy) into the privacy preservation for MCS. By carefully designing the distortion noise, each participants' information about location, sensing data and sensing quality could be not distinguished.

REFERENCES

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *Communications Magazine IEEE*, 2011.

[2] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Yen, R. Huang, and X. Zhou, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Computing Surveys*, vol. 48, no. 1, 2015.

[3] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, (New York, NY, USA), 2008.

[4] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: An end-to-end participatory urban noise mapping system," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '10, (New York, NY, USA), pp. 105–116, ACM, 2010.

[5] L. Kong, L. He, X.-Y. Liu, Y. Gu, M.-Y. Wu, and X. Liu, "Privacy-preserving compressive sensing for crowdsensing based trajectory recovery," in *Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*, 2015.

[6] Y. Gao, W. Dong, K. Guo, X. Liu, Y. Chen, X. Liu, J. Bu, and C. Chen, "Mosaic: A low-cost mobile sensing system for urban air quality monitoring," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016.

[7] S. Nawaz, C. Efstratiou, and C. Mascolo, "Parksense: A smartphone based sensing system for on-street parking," in *Proceedings of the 19th ACM International Conference on Mobile Computing and Networking (MOBICOM 2013)*, 2013.

[8] M. Arab and T. Nadeem, "Magnopark - locating on-street parking spaces using magnetometer-based pedestrians' smartphones," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2017.

[9] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015.

[10] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *The 2014 ACM Conference on Ubiquitous Computing, UbiComp '14, Seattle, WA, USA, September 13-17, 2014*, 2014.

[11] H. Li, T. Li, and Y. Wang, "Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks," in *12th IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS 2015)*, 2015.

[12] H. Li, T. Li, F. Li, W. Wang, and Y. Wang, "Enhancing participant selection through caching in mobile crowd sensing," in *IEEE/ACM International Symposium on Quality of Service (IWQoS 2016)*, 2016.

[13] Y. Wang, H. Li, and T. Li, "Participant selection for data collection through device-to-device communications in mobile sensing," *Personal Ubiquitous Comput.*, vol. 21, pp. 31–41, Feb. 2017.

[14] B. Guo, H. Chen, Q. Han, Z. Yu, D. Zhang, and Y. Wang, "Worker-contributed data utility measurement for visual crowdsensing systems," *IEEE Transactions on Mobile Computing*, vol. 16, pp. 2379–2391, Aug 2017.

[15] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *The 18th Annual International Conference on Mobile Computing and Networking, Mobicom'12, Istanbul, Turkey, August 22-26, 2012*, 2012.

[16] K. Han, C. Zhang, and J. Luo, "Truthful scheduling mechanisms for powering mobile crowdsensing," *CoRR*, 2013.

[17] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Transactions on Networking*, 2016.

[18] D. Zhao, X.-Y. Li, and H. Ma, "Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully," *Networking, IEEE/ACM Transactions on*, 2014.

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

[20] H. Zhang, B. Liu, H. Susanto, G. Xue, and T. Sun, "Incentive mechanism for proximity-based mobile crowd service systems," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016.

[21] L. Pournajaf, L. Xiong, V. S. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *IEEE 15th International Conference on Mobile Data Management, MDM 2014, Brisbane, Australia, July 14-18, 2014 - Volume 1*, pp. 73–82, 2014.

[22] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling privacy-preserving incentives for mobile crowd sensing systems," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, 2016.

[23] L. Wang, D. Zhang, D. Yang, B. Y. Lim, and X. Ma, "Differential location privacy for sparse mobile crowdsensing," in *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, 2016.

[24] T. Li, T. Jung, Z. Qiu, H. Li, L. Cao, and Y. Wang, "Scalable privacy-preserving participant selection for mobile crowdsensing systems: Participant grouping and secure group bidding," *IEEE Transactions on Network Science and Engineering*, 2018.

[25] H. Xiong, D. Zhang, L. Wang, J. Gibson, and J. Zhu, "Eemc: Enabling energy-efficient mobile crowdsensing with anonymous participants," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2015.

[26] H. Xiong, D. Zhang, G. Chen, L. Wang, and V. Gauthier, "Crowdtasker: Maximizing coverage quality in piggyback crowdsensing under budget constraint," in *IEEE International Conference on Pervasive Computing and Communications(Percom'15)*, 2015.

[27] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment framework for participatory sensing data collections," in *Proceedings of the 8th International Conference on Pervasive Computing*, Pervasive'10, (Berlin, Heidelberg), pp. 138–155, Springer-Verlag, 2010.

[28] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi, "EMC^3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint," *Mobile Computing, IEEE Transactions on*, vol. pp, no. 99, pp. 1–1, 2014.

[29] D. Zhao, H. Ma, and L. Liu, "Energy-efficient opportunistic coverage for people-centric urban sensing," *Wireless Networks*, no. 6, pp. 1461–1476, 2014.

[30] Z. Yu, D. Zhang, Z. Yu, and D. Yang, "Participant selection for offline event marketing leveraging location-based social networks," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[31] L. Pournajaf, L. Xiong, D. A. Garcia-Ulloa, and V. Sunderam, "A survey on privacy in mobile crowd sensing task management," tech. rep., Department of Mathematics and Computer Science, Emory University, 2014.

[32] Q. Li and G. Cao, "Providing privacy-aware incentives for mobile sensing," in *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on*, 2013.

[33] Q. Li and G. Cao, "Providing efficient privacy-aware incentives for mobile sensing," in *2014 IEEE 34th International Conference on Distributed Computing Systems*, 2014.

[34] X. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "Artsense: Anonymous reputation and trust in participatory sensing," in *INFOCOM, 2013 Proceedings IEEE*, 2013.

[35] F. Restuccia and S. K. Das, "Fides: A trust-based framework for secure user incentivization in participatory sensing," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a*, pp. 1–10, IEEE, 2014.

[36] K. L. Huang, S. S. Kanhere, and W. Hu, "Preserving privacy in participatory sensing systems," *Comput. Commun.*, vol. 33, pp. 1266–1280, July 2010.

[37] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "Anonysense: Privacy-aware people-centric sensing," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, 2008.

[38] L. Becchetti, L. Filipponi, and A. Vitaletti, "privacy support in people-centric sensing," *journal of communications*, 2012.

[39] B. Agir, T. G. Papaioannou, R. Narendula, K. Aberer, and J. Hubaux, "User-side adaptive protection of location privacy in participatory sensing," *GeoInformatica*, vol. 18, no. 1, pp. 165–191, 2014.

[40] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "Prisense: Privacy-preserving data aggregation in people-centric urban sensing systems," in *Proceedings of the 29th Conference on Information Communications*, INFOCOM'10, (Piscataway, NJ, USA), pp. 758–766, IEEE Press, 2010.

[41] R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher, "Poolview: Stream privacy for grassroots participatory sensing," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, (New York, NY, USA), pp. 281–294, ACM, 2008.

[42] Z. Zhang, S. He, J. Chen, and J. Zhang, "Reap: An efficient incentive mechanism for reconciling aggregation accuracy and individual privacy in crowdsensing," *IEEE Transactions on Information Forensics and Security*, vol. 13, Dec 2018.

[43] X. Jin, R. Zhang, Y. Chen, T. Li, and Y. Zhang, "Dpsense: Differentially private crowdsourced spectrum sensing," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[44] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[45] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, and X. Ma, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation," in *Proceedings of the 26th International Conference on World*

Wide Web, WWW '17, pp. 627–636, International World Wide Web Conferences Steering Committee, 2017.

[46] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui, "Privacy preserving and cost optimal mobile crowdsensing using smart contracts on blockchain," in *Proceedings of 15th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2018.

[47] K. Han, C. Zhang, and J. Luo, "BLISS: budget limited robust crowdsensing through online learning," in *Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking, SECON 2014, Singapore, June 30 - July 3, 2014*, pp. 555–563, 2014.

[48] Y. Liu and M. Liu, "An online learning approach to improving the quality of crowd-sourcing," in *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 2015.

[49] X. Zhang, Y. Wu, L. Huang, H. Ji, and G. Cao, "Expertise-aware truth analysis and task allocation in mobile crowdsourcing," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017.

[50] H. Li, T. Li, F. Li, Y. Wu, and Y. Wang, "Cumulative participant selection with switch costs in large-scale mobile crowd sensing," in *27th IEEE International Conference on Computer Communications and Networks (ICCCN 2018)*, 2018.

[51] W. Alasmary, H. Sadeghi, and S. Valaee, "Crowdsensing in vehicular sensor networks with limited channel capacity," in *Proceedings of IEEE International Conference on Communications, ICC 2013, Budapest, Hungary, June 9-13, 2013*, 2013.

[52] 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 *2014 IEEE Conference on Computer Communications, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014*, pp. 1231–1239, 2014.

[53] W. Liu, Y. Yang, E. Wang, Z. Han, and X. Wang, "Prediction based user selection in time-sensitive mobile crowdsensing," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2017.

[54] L. Pournajaf, L. Xiong, and V. S. Sunderam, "Dynamic data driven crowd sensing task assignment," in *Proceedings of the International Conference on Computational Science, ICCS 2014, Cairns, Queensland, Australia, 10-12 June, 2014*, pp. 1314–1323, 2014.

[55] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015.

[56] L. Huang, Y. Zhu, J. Yu, and M. Y. Wu, "Group buying based incentive mechanism for mobile crowd sensing," in *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2016.

[57] H. To, L. Fan, L. Tran, and C. Shahabi, "Real-time task assignment in hyperlocal spatial crowdsourcing under budget constraints," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2016.

[58] J. Xie, S. Yang, F. Wu, X. Gao, and G. Chen, "A strategy-proof budget feasible online mechanism for crowdsensing with time-discounting values," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016.

[59] H. Jin, L. Su, and K. Nahrstedt, "Centurion: Incentivizing multi-requester mobile crowd sensing," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017.

[60] Z. Feng, Y. Zhu, H. Cai, and P. Luo, "Optimal distributed auction for mobile crowd sensing," *The Computer Journal*, 2017.

[61] Z. Duan, W. Li, and Z. Cai, "Distributed auctions for task assignment and scheduling in mobile crowdsensing systems," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017.

[62] M. H. Cheung, R. Southwell, F. Hou, and J. Huang, "Distributed time-sensitive task selection in mobile crowdsensing," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '15, (New York, NY, USA), pp. 157–166, ACM, 2015.

[63] L. Wang, D. Zhang, and H. Xiong, "effsense: energy-efficient and cost-effective data uploading in mobile crowdsensing," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pp. 1075–1086, ACM, 2013.

[64] H. Li, T. Li, X. Shi, and Y. Wang, "Data collection through device-to-device communications for mobile big data sensing," in *1st Workshop of Mission-Critical Big Data Analytics (MCBDA 2016)*, 2016.

[65] X. Zhang, Y. Wu, and G. Cao, "Resource-aware approaches for truth analysis in crowdsourcing," in *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2016.

[66] C. Meng, W. Jiang, Y. Li, J. Gao, L. Su, H. Ding, and Y. Cheng, "Truth discovery on crowd sensing of correlated entities," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, (New York, NY, USA), pp. 169–182, ACM, 2015.

[67] C. Meng, H. Xiao, L. Su, and Y. Cheng, "Tackling the redundancy and sparsity in crowd sensing applications," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, 2016.

[68] K. Han, H. Huang, and J. Luo, "Posted pricing for robust crowdsensing," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2016.

[69] F. Tian, B. Liu, X. Sun, X. Zhang, G. Cao, and L. Gui, "Movement-based incentive for crowdsourcing," *IEEE Transactions on Vehicular Technology*, 2017.

[70] K. Renaud and D. Gálvez-Cruz, "Privacy: aspects, definitions and a multi-faceted privacy preservation approach," in *Information Security for South Africa (ISSA), 2010*, pp. 1–8, IEEE, 2010.

[71] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1928–1946, 2011.

[72] M. Xiao, J. Wu, S. Zhang, and J. Wu, "Secret-sharing-based secure user recruitment protocol for mobile crowdsensing," in *INFOCOM, 2017 Proceedings IEEE*, 2017.

[73] T. Li, T. Jung, H. Li, L. Cao, W. Wang, X.-Y. Li, and Y. Wang, "Scalable privacy-preserving participant selection in mobile crowd sensing," in *IEEE 15th International Conference on Pervasive Computing and Communications (PerCom 2017)*, 2017.

[74] H. Li, T. Li, F. Li, , and Y. Wang, "Enhancing participant selection through caching in mobile crowd sensing," in *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, pp. 1–10, June 2016.

[75] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Inception: Incentivizing privacy-preserving data aggregation for mobile crowd sensing systems," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '16, (New York, NY, USA), pp. 341–350, ACM, 2016.

[76] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *Infocom, 2013 proceedings ieee*, pp. 1402–1410, IEEE, 2013.

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

[78] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '15, (New York, NY, USA), pp. 167–176, ACM, 2015.

[79] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press, 2007.

[80] M. Naor, B. Pinkas, and R. Sumner, "Privacy preserving auctions and mechanism design," in *Proceedings of the 1st ACM conference on Electronic commerce*, pp. 129–139, ACM, 1999.

[81] O. Goldreich, "Secure multi-party computation," *Manuscript. Preliminary version*, 1998.

[82] T. Jung and X.-Y. Li, "Enabling privacy-preserving auctions in big data," in *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, pp. 173–178, April 2015.

[83] Q. Huang, Y. Gui, F. Wu, G. Chen, and Q. Zhang, "A general privacy-preserving auction mechanism for secondary spectrum markets," *IEEE/ACM Transactions on Networking*, vol. 24, pp. 1881–1893, June 2016.

[84] H. Lipmaa, N. Asokan, and V. Niemi, "Secure vickrey auctions without threshold trust," in *Financial Cryptography*, pp. 87–101, Springer, 2003.

[85] M. Larson, R. Li, C. Hu, W. Li, X. Cheng, and R. Bie, "A bidder-oriented privacy-preserving vcg auction scheme," in *Wireless Algorithms, Systems, and Applications*, pp. 284–294, Springer, 2015.

[86] K. Suzuki and M. Yokoo, "Secure generalized vickrey auction using homomorphic encryption," in *Financial Cryptography*, pp. 239–249, Springer, 2003.

[87] M. Yokoo and K. Suzuki, "Secure generalized vickrey auction without third-party servers," in *Financial Cryptography*, pp. 132–146, Springer, 2004.

[88] F. Brandt, "Fully private auctions in a constant number of rounds," in *Financial Cryptography*, pp. 223–238, Springer, 2003.

[89] D.-H. Shih, H.-Y. Huang, and D. C. Yen, "A secure reverse vickrey auction scheme with bid privacy," *Information Sciences*, vol. 176, no. 5, pp. 550–564, 2006.

[90] F. Brandt, "Cryptographic protocols for secure second-price auctions," in *Cooperative Information Agents V*, pp. 154–165, Springer, 2001.

[91] F. Brandt, "Secure and private auctions without auctioneers," in *Technical Report FKI-245-02*, Institut fur Informatick, Technishce Universitat Munchen, 2002.

[92] J. Zeng, G. Telang, M. P. Johnson, R. Sarkar, J. Gao, E. M. Arkin, and J. S. B. Mitchell, "Mobile r-gather: Distributed and geographic clustering for location anonymity," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Mobihoc '17, (New York, NY, USA), pp. 7:1–7:10, ACM, 2017.

[93] A. Armon, "On min-max r-gatherings," *Theor. Comput. Sci.*, vol. 412, pp. 573–582, Feb. 2011.

[94] G. Aggarwal, R. Panigrahy, T. Feder, D. Thomas, K. Kenthapadi, S. Khuller, and A. Zhu, "Achieving anonymity via clustering," *ACM Trans. Algorithms*, vol. 6, pp. 49:1–49:19, July 2010.

[95] Y. Wang, , , , , and B. Xu, "L2p2: Location-aware location privacy protection for location-based services," in *2012 Proceedings IEEE INFOCOM*, pp. 1996–2004, March 2012.

[96] Y. Wang, D. Xu, and F. Li, "Providing location-aware location privacy protection for mobile location-based services," *Tsinghua Science and Technology*, vol. 21, pp. 243–259, June 2016.

[97] E. Shi, H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Annual Network & Distributed System Security Symposium (NDSS)*, Internet Society., 2011.

[98] M. Joye and B. Libert, "A scalable scheme for privacy-preserving aggregation of time-series data," in *International Conference on Financial Cryptography and Data Security*, pp. 111–125, Springer, 2013.

[99] T. Jung, X. Mao, X.-Y. Li, S.-J. Tang, W. Gong, and L. Zhang, "Privacy-preserving data aggregation without secure channel: Multivariate polynomial evaluation," in *INFOCOM, 2013 Proceedings IEEE*, pp. 2634–2642, IEEE, 2013.

[100] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, pp. 211–407, Aug. 2014.

[101] W. Vickrey, "Counterspeculation, Auctions and Competitive Sealed Tenders," *Journal of Finance*, pp. 8–37, 1961.

[102] E. Clarke, "Multipart pricing of public goods," *Public Choice*, vol. 11, pp. 17–33, 1971.

[103] T. Groves, "Incentives in teams," *Econometrica*, vol. 41, pp. 617–31, July 1973.

[104] T. Jung and X. Li, "Infinite choices of data aggregations with linear number of keys," *CoRR*, vol. abs/1308.6198, 2013.

[105] L. Pu, X. Chen, J. Xu, and X. Fu, "Crowdlet: optimal worker recruitment for self-organized mobile crowdsourcing," in *2016 IEEE Conference on Computer Communications, INFOCOM 2016, San Francisco, CA, April 10 - April 15, 2016*, 2016.

[106] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pp. 103–112, ACM, 1988.

[107] C. Huang, D. Wang, and N. Chawla, "Towards time-sensitive truth discovery in social sensing applications," in *Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on*, pp. 154–162, IEEE, 2015.

[108] H. Jin, L. Su, and K. Nahrstedt, "Theseus: Incentivizing truth discovery in mobile crowd sensing systems," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Mobihoc '17, (New York, NY, USA), pp. 1:1–1:10, ACM, 2017.

[109] The Data for Development (D4D) Challenge, http://www.d4d.orange.com.

[110] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD dataset epfl/mobility (v. 2009-02-24)." Downloaded from http://crawdad.org/epfl/mobility/20090224, Feb. 2009.

[111] F. Wu, Q. Huang, Y. Tao, and G. Chen, "Towards privacy preservation in strategy-proof spectrum auction mechanisms for noncooperative wireless networks," *IEEE/ACM Transactions on Networking*, vol. 23, pp. 1271–1285, Aug 2015.