On Cooperative Obfuscation for Privacy-Preserving Task Recommendation in Mobile CrowdSensing

Christine Bassem
Computer Science Department
Wellesley College
Wellesley, MA, USA
cbassem@wellesley.edu

***Abstract*—With the increased popularity of mobile crowdsensing, personal trajectory information has become easily attainable, compromising the privacy of participants. The focus of this work is to obfuscate the participants' trajectory information during the task recommendation process within crowdsensing. In this paper, we present a cooperative peer-to-peer crowdsensing model, in which peers assist each other in obfuscating their trajectory information. We define a cooperative recommendation mechanism, coupled with an efficient trip segmentation algorithm, which together can preserve the privacy of participants, without sacrificing the performance of task recommendation in the system. Finally, since privacy achieved via dummy-based obfuscation cannot be theoretically guaranteed, we evaluate the privacy and efficiency of the proposed mechanism via simulations.**

***Index Terms*—recommendation, privacy, cooperative, obfuscation, peer-to-peer, crowdsensing**

I. Introduction

With Mobile Crowdsensing (MCS), the sensing capabilities of mobile devices controlled by already roaming crowds are leveraged to improve the sensing process. By involving the humans in-the-loop of sensing, more data can be collected with less cost, when compared to traditional sensor networks [1], allowing for the development of new smart services and applications [2]–[4].

In typical MCS platforms, participating crowds have access to the sensing tasks through a central service provider, which has various degrees of involvement in the process of task recommendation[1]. No matter the task allocation model, an individual's participation will inadvertently reveal sensitive private information about their daily trajectories and routines. The possibility of this information falling in hands of a malicious entity may discourage users from fully participating in the system. Thus, negatively affecting the overall performance of the system. The main premise of this work is the design of a privacy-preserving allocation paradigm that provides added incentive for crowds to participate.

The crowdsensing paradigm holds some semblance to location-based services (LBS), especially in terms of the crowd participant initiating the communication with the server to participate in the crowdsensing process, by requesting information of nearby available sensing tasks. However, its core difference is that the performance of an MCS system depends on the quality of the task recommendation process, which depends greatly on the awareness of the service provider of the location and availability of the participants in the system. Thus, although the problem of preserving the participants' privacy has long been studied in the context of location-based services [5], [6], many of the existing privacy-preserving mechanisms in LBS cannot be directly applied to MCS [7], [8].

Moreover, most of the existing works on privacy-preserving within crowdsensing mainly consider the privacy issues related to data collection and submission [9], [10], as well as spatial privacy issues related to task allocation [8], [11]. In our work, the goal is to define privacy-preserving mechanisms within MCS, which would address the privacy concerns within the platform as a whole; from initial contact, to task recommendation, to data collection and submission, and all the way to compensation and rewards.

In this paper, we present the first step towards achieving that goal, with the definition of a cooperative privacy-preserving task recommendation mechanism without sacrificing the quality of task allocation within the MCS system. The proposed mechanism is designed to obfuscate the information of complete trajectories of participants, which allows for better task recommendation and mobility coordination [12]. Moreover, it is designed with adjustable obfuscation parameters, to allow for the adjustment of the overhead in the system based on the privacy needs of the participants.

Paper Outline. We define the preliminaries of a crowdsensing platform and its associated threat model in Section 2. Our proposed cooperative privacy-preserving task recommendation mechanism is defined in Section 3, followed by a discussion of how the rest of the sensing process can be managed while preserving the participants' privacy in Section 4. In Section 5, we evaluate the privacy and efficiency of the proposed mechanism via simulations, since privacy achieved via dummy-based obfuscation cannot be theoretically guaranteed, and we conclude the paper with a discussion related works and our future work in Sections 6 and 7 respectively.

This work is supported by NSF grant #1755788.

[1]In this work, we use the term task recommendation to refer to task assignment or allocation, since the participant has a choice on whether to follow these allocations or not.

II. Preliminaries

In this section, we define the components of a Coordinated MCS system and its corresponding adversarial model.

A. Coordinated Mobile CrowdSensing

In MCS, the sensing process is coordinated by a central service provider, which acts as a broker between the tasks to be completed and the mobile participants who can complete them, as depicted in Figure 1.

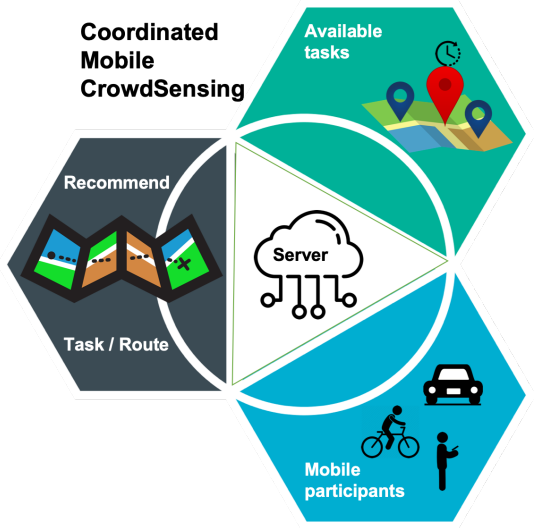

Fig. 1. In MCS, tasks are distributed over the mobility field, be it a city, campus, or within a smart building, and participants roam freely over that mobility field, according to their personal schedules. Tasks are recommended for participants via a central task allocation service, which takes as input the availability information of spatio-temporal tasks and mobile participants.

System Model. The server represents the authority in the platform, which is always available and reachable though a basic Internet connection. A participant represents a platform user or the software agent acting on their behalf, which is assumed to have direct and non-interrupted access to the server. Accordingly, a participant becomes an active peer in the system once they join the MCS service, and the server is aware of all active peers at all times.

Sensing tasks are spatio-temporal, *i.e.* they need to be completed at different locations throughout time. A task is completed when its corresponding sensing action is performed, which can vary from taking pictures [13], to recording audio [14], to answering questions [15]. In the scope of this work, we assume that no matter what action is involved in task completion, it can be completed within a single unit of time. Moreover, we make the typical assumption that time is split into discrete steps that are granular enough to capture the dynamic nature of the system, while allowing amble time for task completion.

Sensing Process. As an active peer in the system, the participant can also engage in the sensing process. The sensing process is composed of three phases; task discovery, data collection and submission, and compensation.

During task discovery, the participant shares some trajectory information with the server and receives recommendations of tasks to complete on their way. The quality of these recommendations depends on the level of granularity of the trajectory information shared. In this work, we assume that a participant shares information about their upcoming trip, by sharing the two spatio-temporal endpoints representing their current location and requested destination. With this level of granularity, the mobility of participants in the MCS system can be coordinated to obtain a higher quality of sensing service.

After receiving a recommended set of tasks, the participant decides on the subset of these tasks to complete and plans their exact route accordingly. While on their route, the user completes the sensing tasks chosen, by collecting the corresponding data and submitting it to the server. Finally, the participant is compensated for the tasks completed and submitted.

B. Adversarial Model

Private Information. A participant's mobility behavior, in the form of the locations they visit at certain times, is private to them. The goal of an adversary is to reveal any private information about a participant, *i.e.*, revealing any true spatio-temporal properties of the participant's route, including its origin and destination.

Adversary Identity. We define an adversary as an entity that aims to learn a participant's private information, regardless of their intent. We consider both the active and passive adversarial models. An active adversary can compromise the server, and obtain all information about the participants in the system, A passive adversary can perform eavesdropping attacks to learn more information about a certain participant. In this work, we consider the server itself to be an active adversary, and the other active peers in the system to be passive adversaries.

privacy

III. Privacy-preserving Task Recommendation

To allow for task recommendation without revealing much information, a participant shares information about their upcoming trajectory [2] with the server to obtain a tailored set of tasks. To protect against both active and passive adversaries, we define a cooperative obfuscation mechanism, in which the participant's private information is obfuscated before sharing it with the server.

The proposed mechanism, as summarized in Figure 2, is composed of two series of communication; from the participant to the server, and then back from the server to the participant. In this section, we define our privacy-preserving mechanism with its associated algorithms.

[2]The model proposed can be easily expanded to allow for multiple trips per participant, but we present it with a single trip for the sake of simplicity.

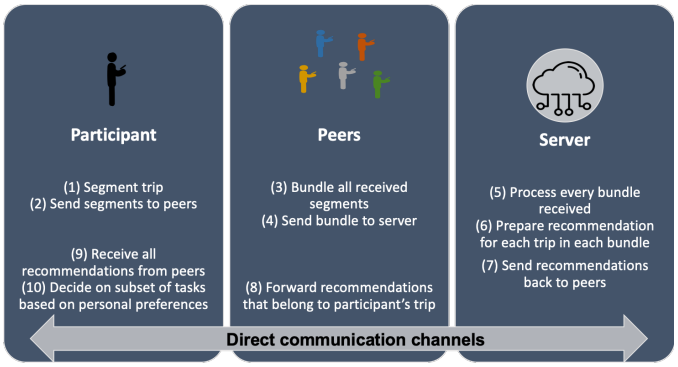

Fig. 2. Privacy-preserving mechanism for requesting recommendations from the server in cooperative coordinated MCS.

A. Request to the Server

A request to the server can be in the form of a single trip or multiple trips (bundle). A trip is an abstraction of the participant's trajectory, in the form of their location at the time of the request and their destination location along with the latest arrival time at that destination. It is the responsibility of the server to process the trips received and return set of recommendations, one set per trip received, regardless of origin of these trips.

To achieve an adequate level of privacy, we require the participant to first segment their trip, to obfuscate its true spatio-temporal information using the dummy location approach [16], [17]. Then, peers from the system are recruited to help with the forwarding of these trip segments, to avoid communicating the trip information directly with the server. Thus, preventing an adversarial server from associating the dummy trip segments to a single participant.

1) Trip segmentation: We define Algorithm 1 to efficiently partition a single trip into a set of k dummy trip segments. The algorithm is designed to disassociate the spatio-temporal properties of the original trip from each individual segment, while preserving these properties in the set as an aggregate.

Algorithm 1 Segmentation heuristic to split a single trip into its corresponding k dummy-trips.

Input: Trip: (v_1, t_1) and (v_2, t_2); k
Output: A set of k dummy trips
1: Create $R[k]$
2: Let seg_time = Min(2, $\frac{t_2 - t_1}{k}$)
3: Let $start_loc$ = v_1
4: Let $start_time$ = t_1
5: **for** $counter$ = 1 to $k-1$ **do**
6: Let end_time = $start_time$ + seg_time
7: Let end_loc = ConstrainedRandom($start_loc$, $start_time$, end_time, (v_2, t_2))
8: $R[counter]$ = ($start_loc$, $start_time$) and (end_loc, end_time)
9: $start_loc$ = end_loc
10: $start_time$ = end_time
11: **end for**
12: $R[k]$ = ($start_loc$, $start_time$) and (v_2, t_2)
13: *return* R

In Algorithm 1, segmentation is achieved by splitting the temporal properties of the trip. For every trip segment, the segment duration is pre-determined, and a random walk algorithm is used to determine the endpoint of each segment. Albeit being random, the spatio-temporal properties of the original trip are maintained by constraining the random walk within a feasible region around that trip.

2) Cooperation with Peers: After segmentation, the participant selects peers from the set of active participants in the system. It can be assumed that the server provides an anonymized list of active participants, to maintain an adequate level of participant privacy. At least one segment is shared with each peer, allowing the participant to share all k generated dummy trip segments. After receiving a dummy trip from a participant, the peer associates the dummy trip to the sending participant in a local cache, and then forwards only the trip to the server, obfuscating the identity of the participant who initiated the process. Moreover, if the peer received multiple dummy trips from different participants, these trips are grouped together and sent as a single bundle to the server.

B. At the Server

The server continuously receives recommendation requests; either in the form of a single trip or a bundle or trips. Its responsibility is to allocate the available tasks to the trips received, optimizing a system-wide objective. After an allocation decision is made, the tasks chosen for each received trip are associated with their corresponding trip, as its recommendation, and sent back to the peer that shared that trip with the server.

In this work, we assume that the system's objective is to maximize the total revenue collected from tasks allocated. Even with the unrealistic assumption of all participants completing the tasks recommended to them, the sub-modular welfare maximization problem [18] can be reduced to such an allocation problem, which renders it NP-hard. Thus, we define a greedy algorithm to efficiently solve the problem, since greedy is one of the most efficient techniques known to approximate this type of problems [19].

Algorithm 2 Greedy task recommendation mechanism for a bundle of trips

Input: Bundle of trips
Output: Recommendations for each trip in the bundle
1: Let $R[i] = \{\phi\}$, for each trip in bundle
2: Sort tasks in decreasing order by their payoff
3: **for** each task t **do**
4: **for** each trip p in the bundle **do**
5: **if** t is feasible within p **then**
6: Calculate the benefit of p if t is added to it
7: **end if**
8: **end for**
9: Find the trip $p*$ with highest utility
10: $R[p*] = R[p*] \cup t$
11: **end for**
12: *return* R

In Algorithm 2, we define a greedy algorithm that prioritizes assigning the most valuable tasks first. After sorting, the

benefit of adding the task to every trip is calculated. We define the benefit as the ratio between the collected revenue along a trip and the length of that trip. Finally, the task is assigned to the trip that gains the maximum benefit from including it.

The running time complexity of Algorithm 2 depends on the number of trips in the bundle, n, the number of available tasks, m, and the maximum duration of the largest trip in the bundle, t_{max}, and in the worst case it is $\Theta(n \times m \times t_{max})$. The average running time of the algorithm can be improved by considering only the set of tasks that are valid within the duration of the trips of the bundle.

C. Back to the Participant

1) Cooperation with Peers: After the server sends the sets of recommendations back to the intermediate peer, it becomes their responsibility to forward these trip recommendations to their owners. Since the server associates each recommendation set to its corresponding trip, the peer can simply extract the trip information and use it to lookup its local cache, to determine the participant who initiated the process. Finally, the peer extracts the set of recommendations and forwards them to the original participant.

2) Task Selection: After collecting task recommendations for their upcoming trip, the participant selects a subset of these tasks to complete on their way. That selection process depends on the participant's preferences, and can be designed to optimize various individual objectives; such as minimizing their total distance traveled, maximizing the number of tasks they select, maximizing their revenue collected from tasks completed, or a combination of any of these objectives. [3].

IV. Beyond Recommendation

As mentioned above, the communication process in MCS doesn't stop at the recommendation phase, but continues to the data submission and compensation phases. Although the focus of this paper is to present a privacy-preserving recommendation mechanism, we present in this section the set of existing mechanisms for the other phases, which would work best with our proposed mechanism.

A. Data Collection and Submission

After task selection, the participant is expected to collect the data required to complete these tasks, and send them to the server. Similar to the recommendation process, sending the data collected directly to the server would compromise the participant's privacy. For the purposes of this paper, we adopt the path jumbling approach [9] to cooperatively submit completed tasks.

In more details, after the participant completes some x number of tasks, they can recruit the help of a random set of k peers and exchange the data with them. We recommend adopting the random-fair exchange strategy [9], in which participants exchange an equal number of tasks between themselves. This strategy ensures that the number of tasks that the user submits at the end is the same as the number of tasks that they originally completed. Moreover, repeating this strategy with each peer leads to effective path jumbling. Finally, after the exchange is performed with every peer, the participant sends the data, which have been collected from the exchanges, directly to the server.

[3]Due to length limitation, we exclude the algorithms defined for task selection, as they are not the main contribution.

B. Compensation and Rewards

In this work, we assume that tasks have equal compensations, which is a typical assumption in existing MCS systems. Thus, the compensation process becomes that of paying the participant based on the number of tasks that they submit directly to the server. These transactions can be performed anonymously using micro-payments, which were first introduced in PayWord [20], or by using more efficient and practical anonymous payment mechanisms designed for Tor networks [21]. An interesting open problem that this cooperative model introduces is the investigation of fair and private payment models with heterogenous tasks, in which participants are also compensated for their cooperation in obfuscating their peers' identities.

V. Performance Evaluation

In this section, we evaluate the performance of our defined segmentation and recommendation algorithms, in the offline and online setting.

A. Simulation Setup

A Java discrete event simulator is developed to simulate the MCS system, and evaluate various recommendation and routing algorithms. All simulations were performed on a Mac machine with iOS 11.4 and RAM of 32GB.

All results shown below are averages of 20 executions of the simulation with varying random seeds, on a 10×10 Manhattan Grid, which is representative of a small campus or town setup, with a simulation time of 6 hours, or 360 time units, and 3000 spatio-temporal tasks uniformly distributed over the grid. Participants' spatio-temporal endpoints are generated using mobility traces from [22].

Evaluation Metrics. For the system-based metrics, we measure participation rate as the ratio between participants generating revenue and the total number of active participants, task coverage as the ratio between completed tasks and all available tasks, and revenue achieved is the ratio between the total payoff collected from completed tasks and maximum possible revenue.

B. Privacy experiments

In the first set of experiments, we evaluate the partitioning algorithm as defined in Algorithm 1. In all experiments, we generate 100 participants, each with a single trip that is partitioned into k segments, in which $k =$number of peers.

Initially, we measure the cosine distance between each dummy segment created and the trajectory of the original trip, as shown in Figure 3. A lower value indicates a similar trajectory (spatially), which is bound to happen with at least

dummy segment in the experiment, due to the grid-based graph used in the simulations. Although there exists at least one dummy segment with almost identical trajectory with every segmentation, there also exists at least one dummy segment with a distance close to 2, creating an average distance in the range of 1.

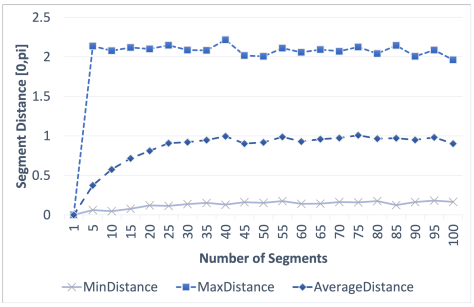

Fig. 3. Algorithm 1 succeeds in generating pseudo-random trip segments with varying similarity degrees between 0 and 2.5.

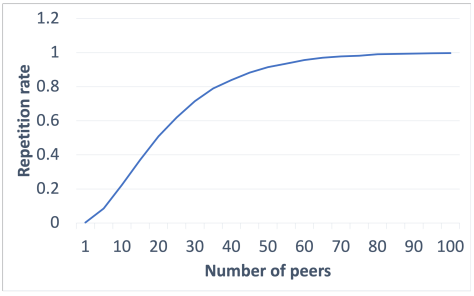

Fig. 4. Higher leakage probabilities are certain when half of the active participants are recruited as peers.

Then, we estimate information leakage with repeated queries, by measuring the probability that a peer would receive more than one dummy segment from the same participant originating from the same original segment, after being part of the system for a week. A higher probability indicates a higher risk from passive attackers. As indicated in Figure 4, as the number of peers increase dramatically, it is bound that repetitions will occur over time. These results indicate that it's more prudent to keep the number of peers below 30% of the total number of participants, even when sharing a single trip segment with each.

C. Online Experiments

In this second set of experiments, we evaluate the performance of the recommendation mechanism in a real-time setup, in which all participants are assumed to be active participants and they connect with their peers when they have a trip ready. We compare our private recommendation mechanism to the optimal choice that a participant would make if given information about all available tasks by implementing the Optimal reward-maximizing routing algorithm from [12].

Figure 5 represents the task coverage rate, when varying the number of peers within the mechanism while fixing the number of generated dummy segments (k). An increase in the number of segments leads to a slight deterioration in the performance of task recommendation, since the partitioning algorithm aims to hide the true spatio-temporal properties of the original trip, leading to bad recommendations. However, this deterioration can be improved by increasing the number of peers, since this allows the participant to get multiple recommendations from the service provider, increasing their chances of finding the perfect set of tasks.

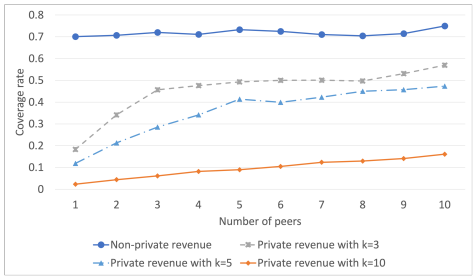

Fig. 5. Increasing the number of peers allows the participant to get more information about feasible tasks from the service provider. This increases the task coverage rate with an accompanied increase in communication overhead.

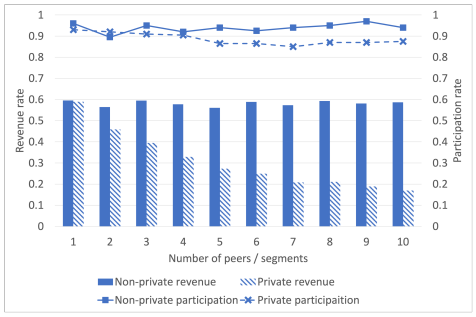

Fig. 6. The quality of recommendation decreases rapidly with each added segment/peer, while adding unnecessary overhead.

Finally, in Figure 6, we set the value of k to the number of peers, to verify our hypothesis that with more dummy segment generated, the performance of task allocation deteriorates, although the privacy is increased as discussed above. With the number of dummy segments set to be equal to the number of peers, the quality of recommendation decreases rapidly with each added segment/peer, adding unnecessary overhead. It's best to set to the number of segments to some fraction of the number of peers, as shown in Figure 5.

VI. Related Work

Privacy-related solutions can be generally categorized based on their purpose within the sensing process [5], [8].

For data collection and submission, jumbling triplets were proposed in [9] to hide participant's paths during the data collection and submission process. In [10], sensing records are transferred to k participants in the form of slices. These slices can then be sent in to the server who can piece together the sensing record without knowing who the original sender was.

For compensation and rewards, payment transactions can be performed anonymously using micro-payments, which were first introduced in PayWord [20], or by using more efficient and practical anonymous payment mechanisms designed for Tor networks [21].

For task recommendation, the general categories of solutions range from trusting explicit application privacy policies [5], to cryptography-based solutions such as differential privacy [11] and hashing encryption [23], to obfuscating spatial information via k-anonymity [24], spatial cloaking [16] and temporal cloaking [25].

Our work fits within the spatio-temporal cloaking, in which the true spatio-temporal information is hidden from the server. Cloaking can be done generating dummy information [8], [16], trusting third-party servers to provide cloaking mixed with k-anonymity [26], or cooperating with others in the system to generalize queries [27]. We adopt a hybrid approach of generating dummy trips with peer cooperation to achieve trajectory privacy, which has not been investigated in the context of MCS to the best of our knowledge.

VII. Conclusion and Future Work

In this paper, we defined a novel cooperative recommendation mechanism within MCS that preserves the privacy of participants' trajectory information, while not sacrificing the performance of task recommendation within the platform. Our experiments confirm that privacy and performance are diametric, as better obfuscation leads to worse task coverage, but an adequate balance can be reached. Our plan for future work is to investigate privacy-preserving payment models that compensate peers for their cooperation, as well as the aggregation of these mechanisms into a comprehensive private MCS platform.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] C. Leonardi, A. Cappellotto, M. Caraviello, B. Lepri, and F. Antonelli, "Secondnose: an air quality mobile crowdsensing system," in *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. ACM, 2014, pp. 1051–1054.

[3] H. R. Arkian, A. Diyanat, and A. Pourkhalili, "Mist: Fog-based data analytics scheme with cost-efficient resource provisioning for iot crowdsensing applications," *Journal of Network and Computer Applications*, vol. 82, pp. 152–165, 2017.

[4] R. Pryss, W. Schlee, B. Langguth, and M. Reichert, "Mobile crowdsensing services for tinnitus assessment and patient feedback," in *2017 IEEE International Conference on AI & Mobile Services (AIMS)*. IEEE, 2017, pp. 22–29.

[5] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, "Location privacy-preserving mechanisms in location-based services: A comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 1, pp. 1–36, 2021.

[6] L. Chen, S. Thombre, K. Järvinen, E. S. Lohan, A. Alén-Savikko, H. Leppäkoski, M. Z. H. Bhuiyan, S. Bu-Pasha, G. N. Ferrara, S. Honkala *et al.*, "Robustness, security and privacy in location-based services for future iot: A survey," *IEEE Access*, vol. 5, pp. 8956–8977, 2017.

[7] L. Kazemi and C. Shahabi, "A privacy-aware framework for participatory sensing," *ACM Sigkdd Explorations Newsletter*, vol. 13, no. 1, pp. 43–51, 2011.

[8] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *2014 IEEE 15th International Conference on Mobile Data Management*, vol. 1. IEEE, 2014, pp. 73–82.

[9] D. Christin, J. Guillemet, A. Reinhardt, M. Hollick, and S. S. Kanhere, "Privacy-preserving collaborative path hiding for participatory sensing applications," in *Proceedings - 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS 2011*, 2011, pp. 341–350.

[10] F. Qiu, F. Wu, and G. Chen, "Slicer: A slicing-based k-anonymous privacy preserving scheme for participatory sensing," in *2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems*. IEEE, 2013, pp. 113–121.

[11] Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang, and H. Qi, "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1330–1341, 2018.

[12] C. Bassem, "Mobility coordination of participants in mobile crowdsensing platforms with spatio-temporal tasks," in *Proceedings of the 17th ACM International Symposium on Mobility Management and Wireless Access*, 2019, pp. 33–40.

[13] L. Deng and L. P. Cox, "Livecompare: grocery bargain hunting through participatory sensing," in *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, 2009, pp. 1–6.

[14] Y. Chon, N. D. Lane, Y. Kim, F. Zhao, and H. Cha, "Understanding the coverage and scalability of place-centric crowdsensing," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 2013, pp. 3–12.

[15] M. Avvenuti, S. Bellomo, S. Cresci, M. N. La Polla, and M. Tesconi, "Hybrid crowdsensing: A novel paradigm to combine the strengths of opportunistic and participatory crowdsensing," in *Proceedings of the 26th international conference on World Wide Web companion*, 2017, pp. 1413–1421.

[16] T.-H. You, W.-C. Peng, and W.-C. Lee, "Protecting moving trajectories with dummies," in *2007 International Conference on Mobile Data Management*. IEEE, 2007, pp. 278–282.

[17] C.-Y. Chow and M. F. Mokbel, "Trajectory privacy in location-based services and data publication," *ACM Sigkdd Explorations Newsletter*, vol. 13, no. 1, pp. 19–29, 2011.

[18] J. Vondrák, "Optimal approximation for the submodular welfare problem in the value oracle model," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, 2008, pp. 67–74.

[19] M. Kapralov, I. Post, and J. Vondrák, "Online submodular welfare maximization: Greedy is optimal," in *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2013, pp. 1216–1225.

[20] R. L. Rivest and A. Shamir, "Payword and micromint: Two simple micropayment schemes," in *International workshop on security protocols*. Springer, 1996, pp. 69–87.

[21] M. Green and I. Miers, "Bolt: Anonymous payment channels for decentralized currencies," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 473–489.

[22] X. Li, G. Pan, Z. Wu, G. Qi, S. Li, D. Zhang, W. Zhang, and Z. Wang, "Prediction of urban human mobility using large-scale taxi traces and its applications," *Frontiers of Computer Science*, vol. 6, no. 1, pp. 111–121, 2012.

[23] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," 1998.

[24] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 754–762.

[25] A. Krause, E. Horvitz, A. Kansal, and F. Zhao, "Toward community sensing," in *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*. IEEE, 2008, pp. 481–492.

[26] J. Meyerowitz and R. Roy Choudhury, "Hiding stars with fireworks: location privacy through camouflage," in *Proceedings of the 15th annual international conference on Mobile computing and networking*, 2009, pp. 345–356.

[27] C.-Y. Chow, M. F. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, 2006, pp. 171–178.