Technical Report

TR-2014-002

A Survey on Privacy in Mobile Crowd Sensing Task Management

by

Layla Pournajaf, Li Xiong, Daniel A. Garcia-Ulloa, Vaidy Sunderam

Mathematics and Computer Science

EMORY UNIVERSITY

A Survey on Privacy in Mobile Crowd Sensing Task Management

Layla Pournajaf, Li Xiong, Daniel A. Garcia-Ulloa, and Vaidy Sunderam, *Emory University*

***Abstract*—Mobile crowd sensing enables a broad range of novel applications by leveraging mobile devices and smartphone users worldwide. While this paradigm is immensely useful, it involves the collection of detailed information from sensors and their carriers (i.e. participants) during task management processes including sensor recruitment and task distribution. Such information might compromise participant privacy in various regards by identification or disclosure of sensitive attributes – thereby increasing vulnerability and subsequently reducing participation. In this survey, we identify different task management approaches in mobile crowd sensing, and assess the threats to participant privacy when personal information is disclosed. We also outline how privacy mechanisms are utilized in existing sensing applications to address these threats. Finally, we discuss continuing challenges facing participant privacy-preserving approaches during task management.**

I. Introduction

THE recent increase in the use of smart phones and other mobile devices has opened the opportunity to collectively sense and share information for common interests. *Mobile crowd sensing* (MCS) refers to the wide variety of sensing models in which individuals with sensing and computing devices are able to collect and contribute valuable data for different applications [1]. Examples of such applications are instant news coverage, finding parking spots, and monitoring traffic or crime mapping. In MCS, a participant or carrier is an individual who collects and contributes data using a sensing device (e.g. a smart phone) that she carries [1]. Collected data is consumed by end users directly or after processing by some applications [2]. Mobile crowd sensing can be categorized based on the involvement of participants in sensing actions as *participatory* or *opportunistic*. In a participatory sensing, participants agree to fulfill the requested sensing activities, and are thus actively involved in the sensing action (e.g. taking a picture or entering data). In an opportunistic sensing, data is collected with minimum or no involvement of the participants (e.g. reporting speed while driving). Opportunistic sensing could run as a background process, so collecting data requires no interaction with the individuals carrying the sensing devices.

To facilitate or coordinate the interaction between applications and participants a *task management* paradigm is needed to define tasks based on end user or application requirements to recruit qualified participants, distribute tasks, and possibly coordinate with participants until task completion. One of the major challenges in task management is to ensure certain degree of privacy for participants. Such an assurance of privacy would increase the disposition of the participants to engage in MCS activity, receive tasks and contribute data, and would ultimately lead to more effective end user applications.

In this paper, we discuss privacy issues and solutions in the context of task management in mobile crowd sensing. Our focus is participant privacy and we do not address privacy concerns of other entities in task management. In section II we review and categorize task management models in MCS. We then investigate privacy threats in different tasking schemes in section III which is followed by existing and applicable privacy solutions studied in section IV. We discuss limitations of participant privacy in task management and other challenges in section V. Finally, Section VI provides some concluding remarks.

II. Task Management in Mobile Crowd Sensing

We identify the following three entities in task management in mobile crowd sensing:

1) *Participants* are entities that use a sensor to obtain or measure the required data about a subject of interest.
2) *Applications* or end users are the entities that request data through tasks and then utilize the information acquired by participants.
3) *Tasking entities* are responsible for distribution of tasks to participants who meet the requirements of applications. In certain architectures, end users and participants can also act as tasking entities.

Figure 1 shows the general structure of task flow in MCS. Task management in crowd sensing can be studied from two perspectives: the distribution model and the nature of tasks.

A. Task Distribution Models

Task management models can be categorized according to the way tasks are distributed among participants. The three major categories for these association models are: centralized, decentralized, and hybrid.

1) Centralized: A central server or tasking entity provides the participants with different tasks to perform. For example, in a party thermometer application, a central server could choose a set of participants attending an event or party, and request that they rate it. These ratings could serve other users who are considering attending this event [2]. One major issue of a central model is having a single point of failure for interactions between participants and applications.

[1] In this paper we refer to these individuals as participants regardless of the sensing model (participatory or opportunistic)

[2] In this paper, we use the terms end user and application interchangeably

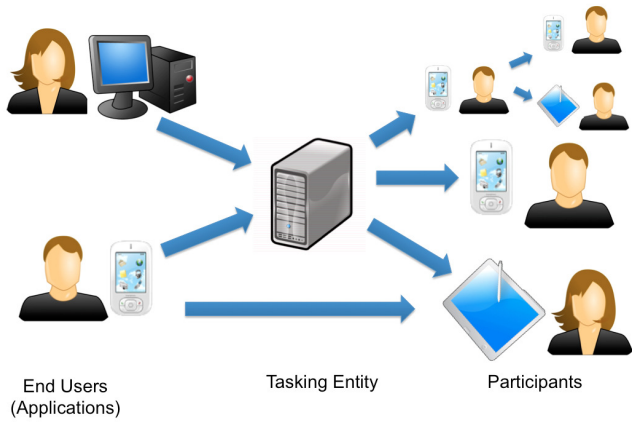

Fig. 1. General structure of the task flow in mobile crowd sensing. Note that end users and participants can also act as tasking entities.

2) Decentralized: In a decentralized model, each participant can become a tasking entity and decide either to perform a task or pass it forward to other participants who might be better-suited to fulfill the task. This decision would be based on certain attributes of other participants such as location, abilities, or the available hardware in her device. A decentralized recruitment model is proposed in [3] which notifies qualified participants of a forthcoming sensing activity. They build a task criteria which includes the locations and times of interest for the sensing activity and aim at recruiting those participants that are likely to fulfill the task, given its criteria. Some participants selected as recruiting nodes distribute the tasks in certain locations, then in a decentralized manner each participant passes the tasks to whoever matches the task criteria. When tasks are reached to a participant with matching similarity above a threshold, this participant keeps a copy of the task and also distributes it to others.

3) Hybrid: A hybrid model includes parts of the centralized and the decentralized models. In this scheme, a central server and a set of participants who act as tasking entities build the task management core. The bubble scheme [4], requires a central server to maintain control of the sensing tasks, which are allocated mostly in a decentralized way. In this model, a task is defined and broadcasted in a particular location of interest by a participant, called a bubble creator in this context. The task is registered in the server and other participants who move into the location of interest are signalled by the central server and become bubble carriers. These carriers can broadcast the task and can also fulfill them and report the sensed data to the server. To guarantee a prolonged existence of the bubbles, another set of participants who are less likely to leave the location of interest assume the role of bubble anchors. These bubble anchors keep broadcasting the tasks in case the bubble creator leaves the location of interest.

B. Task Schemes

Tasks can be classified into several categories based on features inherent to the tasks or the involved tasking entities. In this study, we classify tasks into four major categories: event based vs continuous, push vs pull based model, autonomous vs coordinated, and spatial vs non-spatial. We should note that all these scheme classifications are independent of each other and any combination is possible.

1) Event-based vs Continuous: One way to categorize different possible tasks is by the frequency with which the data is requested. The frequency could either be event-based or scheduled.

Event-based tasks are triggered when a particular situation occurs. This includes special circumstances such as the presence of a participant at a specific location or an ad hoc incident. For example, the tasking entity could ask participants to act as citizen journalists and submit images or other information from a scene of interest when an event occurs [2].

Continuous tasks request information from the participants periodically or frequently. For example, data could be requested every other minute to monitor the speed of cars on a specific highway, or vitals of a patient can be requested daily to track the development of an illness.

2) Push vs Pull model: A different classification for task management models is based on the entity that initiates the task. The categories for this type of partition would be *push* and *pull*.

Push model tasks are initiated by a tasking entity via pushing the tasks on the participants' devices. The platform proposed in [2] uses a push and centralized model where executable binaries of opportunistic tasks are pushed to an optimized set of participants based on a predefined criteria.

Pull model tasks are queried and downloaded by participants at an arbitrary time or location. A pull based task model can be found in [5], where a set of tasks are stored in a central tasking entity and the participants pull this information and decide which tasks to perform.

3) Autonomous vs Coordinated: Tasks can also be categorized based on the allocation scheme that is used to distribute the tasks among the participants. Two approaches that we consider are autonomous task selection and coordinated task assignment [6].

Autonomous task selection is an allocation method in which the participants have access to a set of tasks and they autonomously choose one or more tasks to perform. The participants do not necessarily need to inform the task distributing entity of their decision. The lack of an optimization algorithm for distributing the tasks can decrease the efficiency with respect to sensing cost or global utility. For example, the participants in general might prefer performing certain tasks rather than others, which would result in, on the one hand, unfulfilled or overdone tasks, and on the other, a waste of resources. Another major drawback of autonomous task selection is that it can generate bias in the obtained information. For example, people in urban areas might be more inclined to participate in a sensing task due to the greater presence of sensors or smart phones. This bias would directly affect the variables that are being studied, and will have an effect in the quality of the analysis [7].

Coordinated task assignment aims at improving the quality of the sensed data by optimizing the set of participants

recruited to perform tasks. This optimization is based on varied criteria including coverage, quality, sensing costs, and credibility of the sensed data [6]. Reddy et al. [8] proposed a recruitment process based on three stages. The first stage finds those participants that meet the minimum requirements, the second stage aims at maximizing the coverage over an area or time period, and the third stage checks the participant's reputation over coverage and data collection. Once the appropriate set of tasks and participants have been chosen, and the participants have performed the tasks, the task manager might review the participant's progress and evaluate them for future recruitment.

4) Spatial vs Non-Spatial tasks: In location-based tasks, the location of the participant plays an important role in determining task initiation, distribution, or assignment while non-spatial tasks can be triggered by time or other circumstances. Examples of spatial tasks are traffic control and coffee-shop table availability. An example of a non-spatial task the reporting of online retail pricing or sales volume by participants.

III. Privacy Threats in Task Management

In mobile crowd sensing, privacy concerns might discourage the participants from data contribution. Such concerns include a) disclosure of participant identity, b) disclosure of sensitive attributes including locations (e.g. home or work address) and other private information like personal activities or conditions (e.g. lifestyle or sickness). In MCS task management, participant privacy concerns can be aggravated either directly via sharing real IDs, IP addresses, exact locations, or other sensitive information, or indirectly by sharing insensitive locations (e.g. home address inference from trajectories of participants [9]). Designing a task management model that preserves the privacy of participants can be challenging due to the nature of crowd sensing tasks and task distribution models. In this section, we investigate the information that a participant shares with other tasking entities during task management process and discuss how this information can directly or indirectly breach her privacy. Our focus in this paper is on privacy and we do not address security issues in MCS task management.

Adversary Models

From the perspective of participant privacy, the adversaries in MCS task management may include some or all of end users (applications), tasking entities, and other participants based on their involvement in task management. Here, we study the privacy threats associated with each entity in different task management models.

A. Semi-honest Entities

These entities are assumed to follow the task management protocols and would not actively breach privacy of the participants. However, semi-honest entities may exploit any incidentally acquired information from participants to learn their private information. Privacy attacks that could be conducted by semi-honest entities are task tracing attacks and location-based inference attacks.

Task Tracing Attacks: When a participant pulls specific tasks during a tasking action, shares her preferences during a coordinated task assignment, or notifies server of accepting a pushed task, she may reveal some attributes such as location, time, the task types she is interested, or some attributes of the sensor she is carrying. These information alone might not breach her privacy, however, linking multiple tasking actions might allow an adversary to trace the selected tasks by participant and consequently reveal her identity or other sensitive attributes [5]. Some of the attributes that can be used to trace participants are user-ids, IP addresses, or other network information.

Location-based Inference Attacks: Spatial tasks which are conducted frequently by a participant (e.g. continuous tasks) might lead to disclosure of her sensitive attributes such as home address or eventually her identification through inference attacks [9]. In these types of tasks, even if the participant is using the application anonymously, her trajectory might reveal her sensitive locations.

B. Malicious entities

These entities may actively breach the privacy of participants. Privacy attacks associated with malicious task management entities includes both aforementioned attacks along with several active de-anonymization attacks such as narrow tasking, selective tasking, and collusion attacks. To prevent or stop these attacks, privacy countermeasure should be plugged into sensors or other trusted-parties.

Narrow Tasking: In the process of task definition, a malicious entity (either the application or a separate tasking entity) might create tasks that impose strict limitations on participant attributes or the device she is carrying (e.g., requiring a special lifestyle or a rare sensor type to qualify for the task). This attack, which is called narrow tasking [5] might result in disclosure of identity or other sensitive attributes of the participant who accepts such a strict task.

Selective Tasking: In the process of task distribution, a malicious entity (either the application, a separate tasking entity, or a participant) may share tasks to a limited set of participants to be able to learn their attributes or trace them [5] (e.g. pushing or assigning a task to only one participant).

Collusion Attack: Several applications (end users) might collude to link the information of the participants for de-anonymization. A malicious end user entity might create several applications in an attempt to collect more private data.

IV. Privacy Countermeasures in Task Management

We categorize privacy solutions in MCS task management based on state-of-the-art privacy mechanisms. These mechanisms can be adopted in MCS based on privacy threats relative to task schemes and distribution models. Table 1 summarizes privacy threats and countermeasures for different tasking schemes.

A. Anonymization

Anonymization techniques remove identification information from all the interactions between the participant and other

entities during task distribution. However, even anonymized participants might be prone to tracing attacks and inference attacks. We review some anonymization techniques here.

1) Pseudonyms: One of the basic methods to preserve the anonymity of the participants includes using pseudonyms by replacing the identification information with an alias [10].

2) Connection Anonymization: These methods can be used to avoid network-based tracing attacks using IP addresses. One such technique which is adopted in crowd sensing applications [5] is onion routing [11].

B. Spatio-Temporal Privacy Methods

With the growing advance of location-based services, several spatio-temporal privacy mechanisms have been developed recently. In MCS, location and time are two crucial pieces of information in several task management models, therefore, using the existing spatio-temporal privacy-preserving techniques can be challenging. Here, we study some of the applicable methods in MCS task management.

1) Spatio-Temporal Cloaking: In some spatial task schemes, a perturbed or cloaked location can be used when exact locations are not required (e.g. pollution or weather monitoring). Spatial cloaking or perturbation hides the participant location inside a cloaked region using spatial transformations, generalization, or a set of dummy locations in order to achieve location privacy [12]. In our recent work [6] participants of a coordinated spatial task assignment would share their cloaked location to obtain a set of closest tasks which are optimized for a global coverage goal.

Other approaches share exact locations for tasking; however, they avoid location-based inference attacks by controlling the timing of disclosures. For example, to avoid frequent revealing of location of participants in spatial tasks, Krause et al. [13] use a spatial obfuscation approach. In their solution, they divide the space into a set of regions, then with a certain probability distribution, a subset of participants is selected in each region to report their exact location. Such method can be used in traffic monitoring applications.

Another method to avoid inference attacks [13] assigns spatial tasks to participants in a way that the number of tasks for each participant is minimized. In such an approach, there will be longer intervals between each location disclosure, avoiding location-based inference attacks. This scheme can be further controlled by participants by setting expressive policies regarding the intervals in which they prefer to share their location. We discuss these methods in IV-C.

2) Private Information Retrieval: In autonomous pull-based tasking schemes, participants can retrieve the best suited tasks without providing their attributes using private information retrieval (PIR). PIR-based methods have been adopted for location-based services recently [12] since they guarantee cryptographic privacy by allowing data retrieval from a database without revealing any information to the database server about the retrieved item. In MCS, similar to other autonomous tasking schemes, this approach will suffer from overlapping task selection and bias since sharing entities would not learn which tasks are retrieved.

C. Policy-based Privacy Preferences

To avoid direct or inference-based privacy breaches, participants should be able to set fine-grained preferences to control sharing information in a way that a curious party can not learn or infer any private attributes. Such policies can include specifications to ignore location-based tasks when the participant is within specified range of some location (e.g. home or work), ignore narrow tasks, limit the number of tasks per time periods, or avoid sharing information that could be linked to previously disclosed data. Another solution to avoid inference attacks uses an incentive-based task assignment model in which participants are rewarded for fulfilling a task [14]. In such approaches, task costs can be defined based on the frequency of location disclosures, so the tasking entity will be reluctant to allocate tasks to a participant frequently.

V. Discussion

In this section we discuss further challenges related to participant privacy in MCS.

A. Private Tasking Limitations

1) Trust and Credibility: Privacy and trust generally follow conflicting goals since trust is gained by higher accuracy and exactness of provided data, but privacy aims at hiding or perturbing identifying data (which includes majority of exchanged data in MCS) to protect the participant [7]. Furthermore, trust issues become more challenging for anonymous tasking since they may result in tasking to untrustworthy or unqualified participants [10]. Anonymous participants are prone to provide falsified or faulty data and it would be challenging to evaluate their participation, especially if different task actions can not be linked due to privacy mechanisms. One approach to avoid trust issues in coordinated task management might be to assign a task to several participants to avoid the effect of malicious or faulty participation, however such method would result in a waste of resources.

2) Reward-based Tasking: The challenge for rewarding participants in the presence of privacy mechanisms is very similar to the trust challenges since both require participant evaluation. However, trust models need to trace and review participants progress while incentives can be handled per task completion without linking to other tasks. A delayed rewarding model is proposed in [5] which aims at preventing task-reward linkage.

3) Utility and Efficiency: Privacy mechanisms that obfuscate location, time, or other attributes challenge task management with uncertain or incomplete information. Therefore, the tasking entities may need to task a larger set of participants or conduct more computation to reach a certainty similar to the non-private models. In our recent work [6] we proposed a two-stage tasking model in which participants would share their cloaked locations rather than exact locations. Our model consists of a central tasking server which deals with location uncertainty and recommends globally optimized tasks to participants, and then each participant locally refines and further self-assigns tasks strictly following the global recommendation. Although, this model achieves a comparable utility as

TABLE I
SUMMARY OF PRIVACY THREATS AND COUNTERMEASURES FOR DIFFERENT TASKING SCENARIOS.

Privacy Threats	Tasking Scenarios	Countermeasures
Task Tracing	Pulling specific tasks Coordinated task assignment Push-based tasks with notification	Anonymization Temporal Cloaking
Location-based Inference	Spatial tasks	Spatio-temporal cloaking Private information retrieval
Narrow Tasking	All tasking schemes	Policy-based Privacy Preferences
Selective Tasking	Coordinated task assignment Push-based tasks	Policy-based Privacy Preferences
Collusion Attacks	All tasking schemes	Policy-based Privacy Preferences

the non-private method, the sensing and computational costs are higher due to uncertainty.

B. Task Context Privacy Concerns

In addition to how tasks are managed, task context (i.e. captured sensor data) might also lead to privacy issues for participants. For instance noise monitoring tasks might record participants' voices or if the participants continuously report their driving habits during a trip, the destination of the trip may still be inferred even without sharing specific locations [15]. Fine-grained privacy preferences can help participants to ignore the tasks requiring sensitive contexts. Other privacy-preserving data collection solutions can be also used to avoid sensitive information reporting.

C. Report-based Privacy Concerns

In most applications, captured sensor data contains time/location of collecting actions which might result in location-based inference attacks. Moreover, by linking reports to participants, other tracing attacks would arise. Privacy issues during reporting is extensively studied in literature, and several privacy-preserving data collection and aggregation methods have been proposed [10].

D. Privacy Mechanism Enforcement

We have discussed how suitable privacy mechanisms could be determined by the types of threats, but enforcing these mechanisms still remains as a challenge. In MCS, privacy mechanisms can be enforced on sensors (participants), semi-honest tasking entities, or trusted third-parties. Several mechanisms such as data perturbation or cloaking could use a centralized trusted third-party or could benefit from decentralized secure multiparty computation methods [1]. However, different models might introduce further security issues which needs to be considered in enforcement model decision making.

VI. CONCLUSION

Mobile crowd sensing is an emerging topic with a wide variety of possible applications. However, the functionality of MCS relies on the participation of individuals who might be concerned about their privacy. In particular, task management as a central part of crowd sensing structure poses several threats to participant privacy that needs to be identified and addressed. In this survey, we have classified different potential privacy risks and outlined their solutions for task management in MCS in an effort to raise awareness and preserve the privacy of the participants.

ACKNOWLEDGMENT

This research is supported by the Air Force Office of Scientific Research (AFOSR) DDDAS program under grant FA9550-12-1-0240.

REFERENCES

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

[2] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma, "Prism: Platform for remote sensing using smartphones." in Proceedings of the 8th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys), 2010, pp. 63–76.

[3] G. S. Tuncay, G. Benincasa, and A. Helmy, "Autonomous and distributed recruitment and data collection framework for opportunistic sensing," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, no. 4, pp. 50–53, 2013.

[4] H. Lu, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "Bubble-sensing: Binding sensing tasks to the physical world," *Pervasive and Mobile Computing*, vol. 6, no. 1, pp. 58–71, 2010.

[5] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos, "Anonysense: A system for anonymous opportunistic sensing," *Journal of Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 16–30, 2010.

[6] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations."

[7] C. C. Aggarwal and T. Abdelzaher, "Social sensing," in *Managing and Mining Sensor Data*. Springer, 2013, pp. 237–297.

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

[9] J. Krumm, "Inference attacks on location tracks," in *Pervasive Computing*. Springer, 2007, pp. 127–143.

[10] 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.

[11] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," DTIC Document, Tech. Rep., 2004.

[12] G. Ghinita, *Privacy for Location-Based Services*, ser. Synthesis Lectures on Information Security, Privacy, and Tru. Morgan & Claypool, 2013. [Online]. Available: http://books.google.com/books?id=RB-fcnlloxoC

[13] A. Krause, E. Horvitz, A. Kansal, and F. Zhao, "Toward community sensing," in *Proceedings of the 7th international conference on Information processing in sensor networks*. IEEE Computer Society, 2008, pp. 481–492.

[14] M. Riahi, T. G. Papaioannou, I. Trummer, and K. Aberer, "Utility-driven data acquisition in participatory sensing," EDBT/ICDT. ACM, March 2013.

[15] R. Dewri, P. Annadata, W. Eltarjaman, and R. Thurimella, "Inferring trip destinations from driving habits data," in *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. ACM, 2013, pp. 267–272.