Mobile Crowd Sensing

Manoop Talasila, Reza Curtmola, and Cristian Borcea
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ, USA
Email: mt57@njit.edu, crix@njit.edu, borcea@njit.edu

***Abstract*—People-centric sensing with smart phones can be used for large scale sensing of the physical world at low cost by leveraging the available sensors on the phones. Despite its benefits, mobile people-centric sensing has two main issues: (i) incentivizing the participants, and (ii) reliability of the sensed data. Unfortunately, the existing solutions to solve these issue either requires infrastructure support or adds significant overhead on user phones. We believe that mobile crowd sensing will become a widespread method for collecting sensing data from the physical world once the data reliability issues are properly addressed.**

We present the concept of mobile crowd sensing and its applications to everyday life. We describe the design and implementation of McSense, our mobile crowd sensing platform, which was used to run a user study at the university campus for a period of two months. We also discuss the data reliability issues in mobile crowd sensing by presenting several scenarios involving malicious behavior. We present a protocol for location reliability as a step toward achieving data reliability in sensed data, namely, ILR (Improving Location Reliability). ILR also detects false location claims associated with the sensed data. Based on our security analysis and simulation results, we argue that ILR works well at various node densities. The analysis of the sensed data collected from the users in our field study demonstrate that ILR can efficiently achieve location data reliability and detect a significant percentage of false location claims.

I. Introduction

Mobile sensors such as smart phones and vehicular systems represent a new type of geographically distributed sensing infrastructure that enables mobile people-centric sensing [1]–[3]. According to a forecast for global smart phone shipments from 2010 to 2017, more than 1.5 billion phones are expected to be shipped worldwide [4]. Smart phones already have several sensors: camera, microphone, GPS, accelerometer, digital compass, light sensor, Bluetooth as proximity sensor [5], [6], and in the near future they are envisioned to include health and pollution monitoring sensors [7]–[9]. Vehicular systems have access to several hundred sensors embedded in cars, and recent vehicles come equipped with new types of sensors such as radar and camera. Compared to the tiny, energy constrained sensors of static sensor networks, smart phones and vehicular systems can support more complex computations, have significant memory and storage, and offer direct access to the Internet. Therefore, mobile people-centric sensing can be a scalable and cost-effective alternative to deploying static wireless sensor networks for dense sensing coverage across large areas.

Smart phones have already enabled a plethora of mobile sensing applications [10]–[13] in gaming, smart environments, surveillance, emergency response and social networks. Specially, activity recognition through mobile sensing and wearable sensors has led to many healthcare applications, such as fitness monitoring, elder-care support and cognitive assistance [14]. The expanding sensing capabilities of mobile phones have gone beyond the sensor networks' focus on environmental and infrastructure monitoring where people are now the carriers of sensing devices, the sources, and the consumers of sensed events [15]–[19].

Despite its benefits, mobile people-centric sensing has two main issues: (i) incentivizing the participants, and (ii) reliability of the sensed data. Mobile crowd sensing has been proposed as a solution for the first issue. A mobile crowd sensing platform plays a similar role with the one played by Amazon's Mechanical Turk (MTurk) [20] or ChaCha [21] in crowdsourcing [22], [23]: it allows individuals and organizations (clients) to access a sheer number of people (providers) willing to execute simple sensing tasks for which they are paid. Unlike the MTurk's tasks which are executed on personal computers and always require human work, mobile sensing tasks are executed on mobile devices that satisfy certain context/sensing requirements (e.g., location, time, specific sensors) and many times do not require human work (i.e., automatic sensing tasks). Many organizations and individuals could act as crowd sensing clients. For example, local, state, and federal agencies could greatly benefit from this new sensing infrastructure as they will have access to valuable data from the physical world. Commercial organizations may be very interested in collecting mobile sensing data to learn more about customer behaviour. Researchers in many fields of science and engineering could collect large amounts of sensed data for various experiments. Ultimately, all of us could act as clients through many mobile apps (e.g., find out the traffic conditions ahead on the highway).

Regarding the data reliability issue, the sensed data submitted by participants in crowd sensing is not always reliable as they can submit false data to earn money without executing the actual task. This problem will be illustrated in section I.C, and our solution will be discussed extensively in this chapter.

A. Mobile Crowd Sensing Applications

In the following, we present several application domains that can benefit from mobile crowd sensing as well as a number

of applications (some of them already prototyped) for each domain:

Smart Cities: Worldwide, cities with high population density and a very large number of interconnected issues make effective city management a challenging task. Therefore, several significant government and industrial research efforts are currently underway to exploit the full potential of the sensing data by initiating smart city systems to improve city efficiency by deploying smarter grids, water management systems [24] and ultimately the social progress [25]. Around the world, the government of South Korea is building the Songdo Business District, a green low-carbon area that aims at becoming the first full-scale realization of a smart city [26]. Despite their potential benefits, many of these efforts could be costly. Crowd sensing can reduce the costs associated with large scale sensing and, at the same time, provide additional human-related data. For example, our recent work on ParticipAction [27] proposes to leverage crowd sensing to directly engage citizens in the management of smart cities; people can actively participate in sensing campaigns to make their cities safer and cleaner.

Road Transportation: Departments of transportation can collect fine grain and large scale data about traffic patterns in the country/state using location and speed data provided by GPS sensors embedded in cars. These data can then be used for traffic engineering, construction of new roads, etc. Drivers can receive real-time traffic information based on the same type of data collected from smart phones [28]. Drivers can also benefit from real-time parking data collected from cars equipped with ultrasonic sensors [29]. Transportation agencies or municipalities can efficiently collect pothole data using GPS and accelerometer sensors [30] in order to quickly repair the roads. Similarly, photos (i.e., camera sensor data) taken by people during/after snowstorms can be analyzed automatically to prioritize snow cleaning and removal.

Healthcare & Wellbeing: Wireless sensors worn by people for heart rate monitoring [7] and blood pressure monitoring [9] can communicate their information to the owners' smart phones. Typically, this is done for both real-time and long-term health monitoring of individuals. Mobile sensing can leverage these existing data into large scale healthcare studies that seamlessly collect data from various groups of people, which can be selected based on location, age, etc. A specific example involves collecting data from people who eat regularly fast food. The phones can perform activity recognition and determine the level of physical exercise done by people, which was proven to directly influence people's health. As a result of such a study in a city, the municipality may decide to create more bike lanes to encourage people to do more physical activities. Similarly, the phones can determine the level of social interaction of certain groups of people (e.g., using Bluetooth scanning, GPS, or audio sensor). For example, a university may discover that students (or students from certain departments) are not interacting with each other enough; consequently, it may decide to organize more social events on campus. The same mechanism coupled with information from "human sensors" can be used to monitor the spreading of epidemic diseases.

Marketing/Advertising: Real-time location or mobility traces/patterns can be used by vendors/advertisers to target certain categories of people [31], [32]. Similarly, they can run context-aware surveys (function of location, time, etc.). For example, one question in such a survey could ask people attending a concert what artists they would like to see in the future.

B. Mobile Crowd Sensing Applications and Platforms

Recently, several mobile crowdsourcing projects tried to leverage traditional crowdsourcing platforms for mass adoption of people-centric sensing: Twitter [33] has been used as a publish/subscribe medium to build a crowdsourced weather radar and a participatory noise-mapping application [34]; mCrowd [35] is an iPhone based platform that was used to build an image search system for mobile phones which relies on Amazon MTurk [20] for real-time human validation [35]. This has the advantage of leveraging the popularity of existing crowdsourcing platforms (tens of thousands of available workers), but does not allow for truly mobile sensing tasks to be performed by workers (i.e., tasks which can only be performed using sensors on mobile phones). PEIR is an application for participatory sensing that exploits mobile phones to evaluate if users have been exposed to airborne pollution, enables data sharing to encourage community participation, and estimates the impact of individual user/community behaviors on the surrounding environment [36]. Medusa is a mobile crowd sensing framework that uses a high-level domain-specific programming language to define sensing tasks and workflows that are promoted with monetary incentives to encourage user participation [37]. So far, none of these existing applications and platforms has addressed the reliability of the sensed data.

C. Data Reliability Issues in Sensed Data

By leveraging smart phones, we can seamlessly collect sensing data from various groups of people at different locations using mobile crowd sensing. As the sensing tasks are associated with monetary incentives, participants may try to fool the mobile crowd sensing system to earn money. Therefore, there is a need for mechanisms to efficiently validate the collected data. In the following, we motivate the need for such a mechanism by presenting several scenarios involving malicious behavior.

Traffic jam alerts [38], [39]: Suppose that the Department of Transportation uses a mobile crowd sensing system to collect alerts from people driving on congested roads and then distributes the alerts to other drivers. In this way, drivers on the other roads can benefit from real-time traffic information. However, the system has to ensure the alert validity because malicious users may try to pro-actively divert the traffic on roads ahead in order to empty these roads for themselves.

Citizen-journalism [40], [41]: Citizens can report real-time data in the form of photos, video, and text from public events or disaster areas. In this way, real-time information from

anywhere across the globe can be shared with the public as soon as the event happens. But, malicious users may try to earn easy money by claiming that an event is happening at a certain location while being somewhere else.

Environment [8], [42]: Environment protection agencies can use pollution sensors installed in the phones to map with high accuracy the pollution zones around the country. The participants may claim "fake" pollution to hurt business competitors by submitting the sensed pollution data associated with false locations.

Ultimately, the validation of sensed data is important in a mobile crowd sensing system to provide confidence to its clients who use the sensed data. However, it is challenging to validate each and every sensed data point of each participant because sensing measurements are highly dependent on context. One approach to handle this issue is to validate the location associated with the sensed data point in order to achieve a certain degree of reliability on the sensed data. Still, we need to overcome a major challenge: how to validate the location of data points in a scalable and cost-effective way without help from the wireless carrier? Let us note that wireless carriers may not help with location validation for legal reasons related to user privacy or even commercial interests.

To achieve reliability on participants' location data, there are a few traditional solutions such as using Trusted Platform Modules (TPM) [43] on smart phones or duplicating the tasks among multiple participants. However, these solutions cannot be used directly for a variety of reasons. For example, it is not cost-effective to have TPM modules on every smart phone, while task replication may not be feasible at some locations due to a lack of additional users there. Another solution is to verify location through the use of secure location verification mechanisms [44]–[47] in real time when the participant is trying to submit the sensing data location. Unfortunately, this solution requires infrastructure support or adds significant overhead on user phones if it is applied for each sensed data point.

The rest of the chapter is organized as follows. Section II presents the overview of McSense, our mobile crowd sensing platform, and its prototype implementation. Section III describes our ILR scheme to achieve data reliability in McSense and analyzes ILR's security. The experimental evaluation and simulation results for ILR are presented in sections IV and V, respectively. In section VI, we discuss a number of lessons learned from our McSense field study as well as potential improvements for ILR. Finally, section VII concludes the chapter.

II. MCSENSE: A MOBILE CROWD SENSING PLATFORM

We have designed and implemented McSense [48], a mobile crowd sensing platform that allows clients to collect many types of sensing data from smart phones carried by mobile users. The interacting entities in our mobile crowd sensing architecture are:

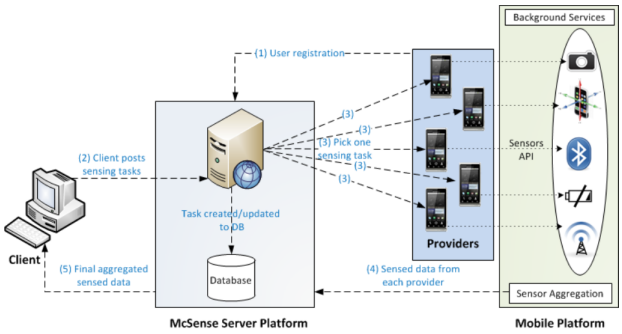

Fig. 1. McSense Architecture.

- *McSense:* A centralized mobile crowd sensing system which receives sensing requests from clients and delivers them to providers; these entities are defined next.
- *Client:* The organization or group who is interested in collecting sensing data from smart phones using the mobile crowd sensing system.
- *Provider:* A mobile user who participates in mobile crowd sensing to provide the sensing data requested by the client.

A. System Architecture and Processes Involved

The architecture of McSense, illustrated in Figure 1, has two main components: (1) the server platform that accepts tasks from clients and schedules the individual tasks for execution at mobile providers; and (2) the mobile platform (at the providers) that accepts individual tasks from the server, performs sensing, and submits the sensed data to the server. The communication among all these components takes place over the Internet. Next we discuss the overall process in more detail.

User Registration: The McSense application on the smart phones shows a registration screen for first time users, prompting them to enter an email address and a password. During the registration process, the user phone's MEID (Mobile Equipment IDentifier) is captured and saved in the server's database along with the user's email address and password. We chose to store the phone's MEID in order to restrict one user registration per device. In addition, the server also avoids duplicate registrations when users try registering with the same email address again.

Posting new sensing tasks: New sensing tasks can be posted by clients using a web interface running on the McSense server. The sensing task details are entered on this web page by the client and submitted to the server's database. Once a new task is posted, the background notification service running on the provider's phone identifies the new available tasks and notifies the provider with a vibrate action on the phone. Providers can check the notification and can open the McSense application to view the new available tasks. When the application is loaded, the providers can see four tabs (Available, Accepted, Completed and Earnings). The providers can view the list of tasks in the respective tabs (Figure 2, left)

and can click on each task from the list to view the entire task details (type, status, description, accepted time, elapsed time, completion time, expiration time, payment amount).

Life cycle of a task: The life cycle starts from the Available tasks tab. When a provider selects an available task and clicks on the Accept button, the task is moved to the Accepted tab. Once a task is accepted, then that task is not available to others anymore (Figure 2, right). When the accepted task is completed according to its requirements, the task is moved to the Completed tasks tab. Finally, the providers view their aggregated total dollars earned for successfully completed tasks under the Earnings tab. If the accepted task expires before completing successfully according to its requirements, it is moved to the Completed tasks tab and marked as unsuccessfully completed. The providers do not earn money for the tasks that are completed unsuccessfully.

Background services on phone: When the network is not available, a completed task is marked as pending upload. A background service on the phone periodically checks for the network connection. When the connection becomes available, the pending data is uploaded and finally these tasks are marked as successfully completed. If the provider phone is restarted manually or due to the mobile OS crash, then all the in-progress sensing tasks are automatically resumed by the Android's BroadcastReceiver service registered for the McSense application. Furthermore, the Accepted and the Completed tab's task lists are cached locally and are synchronized with the server. If the server is not reachable, the users can still see the tasks that were last cached locally.

B. Prototype Implementation

The McSense application, shown in Figure 2, has been implemented in Android and is compatible with smart phones running Android OS 2.2 or higher. The application was tested successfully using Motorola Droid 2 phones which have 512 MB RAM, 1 GHz processor, Bluetooth 2.1, Wi-Fi 802.11 b/g/n, 8 GB on board storage, and 8 GB microSD storage. The McSense [49] Android application was deployed to Google Play [50] to make it available for campus students. The server side of McSense is implemented in Java/J2EE using the MVC (Model View Controller) framework. The Derby database is used to store the registered user accounts and assigned task details. The server side Java code is deployed on the Glassfish Application Server, which is an open-source application server.

C. User Study and Tasks Developed for McSense

To evaluate McSense and its data reliability protocol (see section III), we ran a user study at our campus for approximately two months. Over 50 students have participated in this study. Participants have been asked to download the McSense application from the Android market and install it on their phones. On the application server, we periodically posted various tasks. Some tasks have a monetary value associated with the task, which is paid on the task's successful completion; a few other tasks do not offer monetary incentives just to observe the participation of providers when collecting free

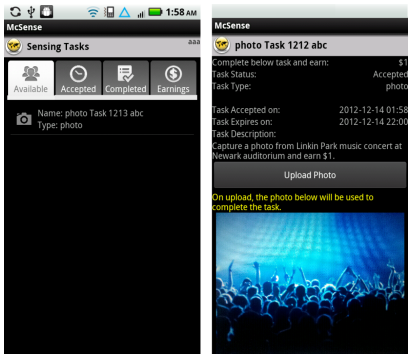

Fig. 2. McSense Android Application showing tabs (left) and task screen for a photo task (right).

sensing data. As tasks are submitted to the application server, they also appear on the phones where our application has been installed. Each task contains a task description, its duration, and a certain amount of money. The students use their phones to sign up to perform the task. Upon successful completion of the task, the students accumulate credits (payable in cash after the study terminated).

The sensing tasks that we choose to use for this study fall into two categories:

- Manual tasks, e.g., photo tasks.
- Automated tasks, e.g., sensing tasks using accelerometer and GPS sensors; sensing tasks using Bluetooth.

Manual Photo Sensing Task: Registered users are asked to take photos from events on campus. Once the user captures a photo, she needs to click on the "Complete Task" button to upload the photo and to complete the task. When the photo is successfully uploaded to the server, the task is considered successfully completed. These uploaded photos can be used by the university news department for current news articles.

Automated Sensing Task using Accelerometer and GPS Sensors: The accelerometer sensor readings and GPS location readings are collected at 1 minute intervals. The sensed data is collected along with the userID and timestamp, and it is stored into a file in the phone's internal storage which can be accessed only by the McSense application. This data is then uploaded to the application server on task completion (which consists of many data points). Using the collected sensed data of accelerometer readings and GPS readings, we can identify user activities such as walking, running, driving, or locations that are important to the user. By observing the daily activities, we could find out how much exercise each student is getting daily and derive interesting statistics such as which departments have the most active and healthy students.

Automated Sensing Task using Bluetooth Radio: In this automated sensing task, the user's Bluetooth radio is used to perform periodic Bluetooth scans (every 5 minutes) until the task expires; on its completion, the task reports the discovered Bluetooth devices with their location back to the McSense server. The sensed data from Bluetooth scans can provide interesting social information such as how often McSense

users are near to each other. Also, it can identify groups who are frequently together to determine the level of social interaction of certain people [5].

Automated Resources usage Sensing Task: In this automated sensing task, the usage of user's smart phone resources is sensed and reported back to the McSense server. Specifically, the report contains the mobile applications' usage, the network usage, the periodic WiFi scans, and the battery level of the smart phone. While logging the network usage details, this automated task also logs the overall device network traffic and per-application network traffic.

III. IMPROVING LOCATION RELIABILITY IN CROWD SENSED DATA

This section presents ILR, a scheme which improves the location reliability of mobile crowd sensed data with minimal human efforts. We also describe the validation process used by McSense to detect false location claims from malicious providers.

A. Assumptions

We assume that the sensed data is already collected by McSense from providers at different locations. However, this sensed data is awaiting validation before being sent to the clients who requested this data. We assume that every provider performs Bluetooth scans at each location where it is collecting sensing data. We also assume that the sensed data reported by providers for a given task always includes location, time, and a Bluetooth scan. Note that Bluetooth scans can have a much lower frequency than the sensor sampling frequency.

B. Adversarial Model

We assume all the mobile devices are capable of determining their location using GPS. We also assume McSense is trusted and the communication between mobile users and McSense is secure. In our threat model, we consider that any provider may act maliciously and may lie about their location.

A malicious provider can program the device to spoof a GPS location [51] and start providing wrong location data for all the crowd sensing data requested by clients. Regarding this, we consider three threat scenarios, where 1) The provider does not submit the location and Bluetooth scan with a sensing data point; 2) The provider submits a Bluetooth scan associated with a sensing task, but claims a false location; 3) The provider submits both a false location and a fake Bluetooth scan associated with a sensing data point. In section III.D, we will discuss how these scenarios are addressed by ILR.

We do not consider colluding attack scenarios, where a malicious provider colludes with other providers to show that she is present in the Bluetooth co-location data of others. In practice, it is not easy for a malicious provider to employ another colluding user at each sensing location. Additionally, these colluding attacks can be reduced by increasing the minimum node degree requirement in co-location data of each provider (i.e., a provider P must appear in the Bluetooth scans of at least a minimum number of other providers at her claimed location and time). Therefore, it becomes difficult for a malicious provider to create a false high node degree by colluding with real co-located people at a given location and time.

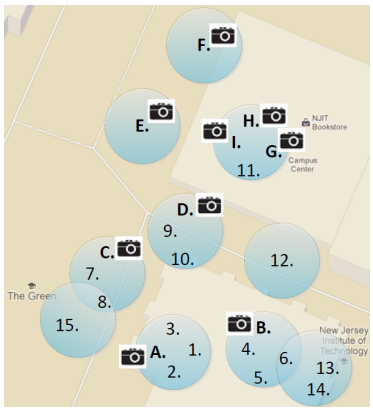

Fig. 3. Example of McSense collected Photo tasks [A-I] and Sensing tasks [1-15] on the campus map, grouped using Bluetooth discovery co-location data.

Finally, the other class of attacks that are out of scope for our current scheme are attacks in which a provider submits the right location and Bluetooth scan associated with this sensing task, but is able to fool the sensors to create false readings (e.g., using the flame of a lighter to create the false impression of a high temperature).

C. ILR Design

The main idea of our scheme is to corroborate data collected from manual (photo) tasks with co-location data from Bluetooth scans. We describe next an example of how ILR uses the photos and co-location data.

1) An example of ILR in action: Figure 3 maps the data collected by several different tasks in McSense. The figure shows 9 photo tasks [marked as A to I] and 15 sensing tasks [marked as 1 to 15] performed by different providers at different locations. For each of these tasks, providers also report neighbors discovered through Bluetooth scans. All these tasks are grouped into small circles using co-location data found in Bluetooth scans within a time interval t. For example, Photo task A and sensing tasks 1, 2, and 3 are identified as co-located and grouped into one circle because they are discovered in each others Bluetooth scans.

In this example, McSense does not need to validate all the photo tasks mapped in the figure. Instead, McSense will first consider the photo tasks with the highest node degree ($NodeDegree$) by examining the co-located groups for photo task providers who have seen the highest number of other providers in Bluetooth scans around them. In this example we consider $NodeDegree \geq 3$. Hence, we see that photo tasks A, B, C, D, and G have discovered the highest number of providers around their location. Therefore, McSense chooses these 5 photo tasks for validation. These selected photo tasks are validated either manually or automatically (we discuss this in detail in section III-C2). When validating these photo tasks, invalid photos are rejected and McSense ignores the

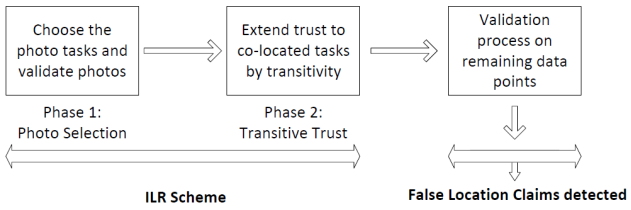

Fig. 4. The phases of the ILR scheme.

Bluetooth scans associated with them. If the photo is valid, then McSense considers the location of the validated photo as trusted because the validated photo is actually taken from the physical location requested in the task. However, it is not always possible to categorize every photo as a valid or a fake photo. Therefore some photos will be categorized as "unknown" when a decision cannot be made.

In this example, we assume that these 5 selected photos are successfully validated through manual verification. Next, using the transitivity property, McSense extends the location trust of validated photos to other co-located providers' tasks which are found in the Bluetooth scans of the A, B, C, D, and G photo tasks. For example, A extends the trust to the tasks 1, 2, and 3, while B extends the trust to tasks 4, 5, and 6. Then, task 6 extends its trust to tasks 13 and 14. Finally, after the end of this process, McSense has 21 successfully validated tasks out of a total of 24 tasks. In this example, McSense required manual validation for just 5 photo tasks, but using the transitive trust property it was able to extend the trust to 16 additional tasks automatically. Only 3 tasks (E, F, and 12) are not validated as they lack co-location data around them.

2) ILR Phases: The ILR scheme has two phases as shown in Figure 4. "Phase 1: Photo Selection" elects the photo tasks to be validated. And "Phase 2: Transitive Trust" extends the trust to data points co-located with the tasks elected in Phase 1:

Phase 1 - Photo Selection: Using collected data from Bluetooth scans of providers, ILR constructs a connected graph of co-located data points for a given location and within a time interval t (these are the same groups represented in circles in Figure 3). From these graphs, we elect the photo tasks that have node degree greater than a threshold (NodeDegree).

These selected photo tasks are validated either by humans or by applying computer vision techniques. For manual validation, McSense could rely on other users recruited from Amazon MTurk [20] for example. In order to apply computer vision techniques, first we need to collect ground truth photos to train image recognition algorithms. One alternative is to have trusted people collect the ground truth photos. However, if the ground truth photos are collected through crowd sensing, then they have to be manually validated as well. Thus, reducing the number of photos that require manual validation is an important goal for both manual and automatic photo recognition. Once the validation is performed, the location of the validated photo task is now considered to be reliable because the validated photos have been verified to be taken from the physical location requested in the task. For simplicity, we will refer to the participants who contributed valid photo tasks with reliable location and time as "Validators".

Phase 2 - Transitive Trust: In this phase, we rely on the transitive property and extend the trust established in the Validator's location to other co-located data points. In short, if the photo is valid, the trust is extended to co-located data points found in Bluetooth scan of the validated photo task. In the current scheme, trust is extended until all co-located tasks are trusted or no other task is found; alternately, McSense can set a TTL (Time To Live) on extended trust. The following two steps are performed in this phase:

- (Step 1) Mark co-located data points as trusted: For each task co-located with a validated photo task, mark the task's location as trusted.
- (Step 2) Repeat Step 1 for each newly validated task until all co-located tasks are trusted or no other task is found.

Algorithm 1 ILR Validation Pseudo-Code

Notation:

```
TList: Tasks List which are not yet marked trusted
after completing first two phases of ILR scheme.
T: Task submitted by a Provider.
L: Location of the Photo or Sensing Task (T).
t: Timestamp of the Photo or Sensing Task (T).
hasValidator(L, t): Function to check, if already
there exist any valid data point at task T's
location and time.
```

validationProcess():
run to validate the location of each task in TList

1: **for** each task T in TList **do**
2: **if** $hasValidator(L, t) == TRUE$ **then**
3: Update task T with false location claim at (L, t)

3) Validation Process: After executing the two phases of ILR scheme, all the co-located data points are validated successfully. If any malicious provider falsely claims one of the validated task's location at the same time, then the false claim will be detected in the validation step. Executing the validation process shown in algorithm 1 will help us to detect wrong location claims around the already validated location data points. For instance, if we consider task 12 from Figure 1 as a malicious provider claiming a false location exactly at photo task A's location and time, then task 12 will be detected in the validationProcess() as it does not appear in the Bluetooth scans of photo task A. In addition to the validation process, McSense also performs a basic spatiotemporal correlation check to ensure that the provider is not claiming a location at different places at same time.

D. Security Analysis

The goal of the ILR scheme is to establish the reliability of the sensed data by validating the claimed location of the data points. In addition, ILR seeks to detect false claims made by malicious participants.

ILR is able to handle all the three threat scenarios presented in our adversarial model section. In the first threat scenario, when there is no location and Bluetooth scan submitted along

with the sensed data, the sensed data of that task is rejected and the provider will not be paid by McSense.

In the second threat scenario, when a provider submits its Bluetooth discovery with a false location claim, ILR detects the provider in its neighbors' Bluetooth scans at a different location using the spatio-temporal correlation check and rejects the task's data.

Finally, when a provider submits a fake Bluetooth discovery with a false location claim, ILR looks for any validator around the claimed location and if it finds anyone, then the sensed data associated with the false location claim is rejected. But, if there is no validator around the claimed location, then the data point is categorized as "unknown".

As discussed in our adversarial model section, sensed data submitted by malicious colluding attackers could be filtered to a certain extent in McSense by setting the node degree threshold (NodeDegree) to the minimum node degree requirement requested by the client.

E. Related Work

Trusted hardware represented by the Trusted Platform Module (TPM) [52]–[54] to design new architectures for trustworthy software execution on mobile phones [55]–[57]. Recent work has also proposed architectures to ensure that the data sensed on mobile phones is trustworthy [58], [59]. When untrusted client applications perform transformations on the sensed data, YouProve [52] is a system that combines a mobile device's trusted hardware with software in order to ensure the trustworthiness of these transformations and that the meaning of the source data is preserved. YouProve describes three alternatives to combine the trusted hardware with software: The first two require to extend the trusted codebase to include either the code for the transformations or the entire application, whereas the third one requires building trust in the code that verifies that transformations preserve the meaning of the source data.

Relying completely on TPM is insufficient to deal with attacks in which a provider is able to "fool" the sensors (e.g., using the flame of a lighter to create the false impression of a high temperature). Recently, there have also been reports of successful spoofing of civilian GPS signals [51].

Orthogonal to the work in ILR, task pricing also helps in improving the data quality. A recent paper [60] presents pricing incentive mechanisms to achieve quality data in participatory sensing application. In this work, the participants are encouraged to participate in the sensing system through a reverse auction based on a dynamic pricing incentive mechanism in which users can sell their sensing data with their claimed bid price.

The LINK protocol [44], [61] was recently proposed for secure location verification without relying on location infrastructure support. LINK can provide stronger guarantees than ILR, but it has a number of drawbacks if used for mobile sensing. LINK requires a provider to establish Bluetooth connections with her co-located users at each sensing location, which increases latency and consumes more phone battery. In addition, LINK is executed in real-time to verify the users' locations, whereas ILR is executed on the collected data from mobile crowd sensing. Therefore, employing ILR helps providers in submitting sensed data quickly and also consumes less phone battery.

TABLE I
DEMOGRAPHIC INFORMATION OF THE STUDENTS

Total participants	58
Males	90%
Females	10%
Age 16-20	52%
Age 21-25	41%
Age 26-35	7%

TABLE II
PHOTO TASK RELIABILITY

	Number of photo tasks
Total photos	1784
Num of photos with Bluetooth scans (manually validated in ILR)	204
Trusted data points added by ILR	148

IV. EXPERIMENTAL EVALUATION: FIELD STUDY

The providers (students shown in Table I) registered with McSense and submitted data together with their userID. Both phases of ILR and the validation process are executed on data collected from the providers. In these experiments, we acted as the clients collecting the sensed data.

A. Evaluating the ILR Scheme

The location data is mostly collected from the university campus (0.5 miles radius). The main goal of these experiments is to determine how efficiently can the ILR scheme help McSense validate the location data and detect false location claims. ILR considers the Bluetooth scans found within 5min interval of measuring the sensor readings for a sensing task.

Table II shows the total photo tasks that are submitted by students; only 204 photo tasks have Bluetooth scans associated with them. In this data set, we considered the NodeDegree 1, therefore we used all these 204 photo tasks with Bluetooth scans in Phase-1 to perform manual validation, and then in Phase-2 we are able to automatically extend the trust to 148 new location data points through the transitive closure property of ILR.

To capture the ground truth, we manually validated all the photos collected by McSense in this study and identified that we have a total of 45 fake photos submitted to McSense from malicious providers, out of which only 16 fake photo tasks are having Bluetooth scans with false location claims. We then applied ILR to verify how many of these 16 fake photos can be detected.

We were able to catch 4 users who claimed wrong locations to make money with fake photos, as shown in Table III. Since the total number of malicious users involved in the 16 fake photo tasks is 10, ILR was able to detect 40% of them. Finally, ILR is able to achieve this result by validating only 11% of the photos (i.e., 204 out of 1784).

TABLE III
NUMBER OF FALSE LOCATION CLAIMS

	Detected by ILR scheme	Total	Percentage Detected
Tasks with False Location claim	4	16	25%
Cheating People	4	10	40%

TABLE IV
SIMULATION SETUP FOR THE ILR SCHEME

Parameter	Value
Number of nodes	200
% of tasks with false location claims	10, 15, 30, 45, 60
Bluetooth transmission range	10m
Simulation time	2hrs
User walking speed	1m/sec
Node Density	2, 3, 4, 5
Bluetooth scan rate	1/min

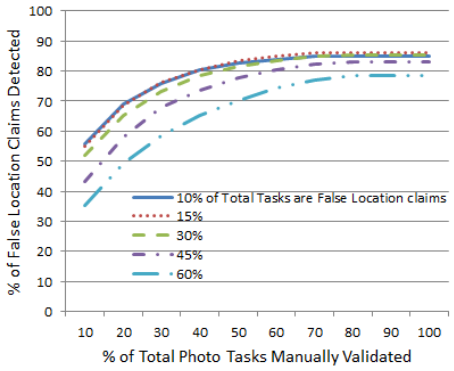

Fig. 5. ILR performance as function of the percentage of photos manually validated in phase 1. Each curve represents a different percentage of photos with fake locations.

B. The Influence of the Task Price on Data Quality

In the field study performed at NJIT, a few tasks are posted with a high price (ranging from $2 - $10) to observe the impact on the sensing tasks. We have noticed a 15% increase in the task completion success rate for the high priced sensing tasks compared to the low priced sensing tasks. In addition, we have noticed an improvement in data quality for the high priced photo tasks, with clear and focused photos compared to the low priced photo tasks (the task priced with $1 or lower are considered low priced tasks). Thus, our study confirms that task pricing influences the data quality. This result confirms that various task pricing strategies [60] can be employed by McSense in parallel to the ILR scheme to ensure data quality for the sensing tasks.

V. SIMULATIONS

This section presents the evaluation of the ILR scheme using the NS-2 network simulator. The two main goals of the evaluation are: (1) Estimate the right percentage of photo tasks needed in Phase 1 to bootstrap the ILR scheme, and (2) Quantify the ability of ILR to detect false location claims at various node densities.

A. Simulation Setup

The simulation setup parameters are presented in Table IV. Given a simulation area of 100m x 120m, the node degree (i.e., average number of neighbors per user) is slightly higher than 5. We varied the simulation area to achieve node degrees of 2, 3, and 4. We consider low walking speeds (i.e., 1m/sec) for collecting photos. In these simulations, we considered all tasks as photo tasks. A photo task is executed every minute by each node. Photo tasks are distributed evenly across all nodes. Photo tasks with false location claims are also distributed evenly across several malicious nodes. We assume the photo tasks in ILR's phase 1 are manually validated.

After executing the simulation scenarios described below, we collect each photo task's time, location, and Bluetooth scan. As per simulation settings, we will have 120 completed photo tasks per node at the end of the simulation (i.e 24,000 total photo tasks for 200 nodes). Over this collected data, we apply the ILR validation scheme to detect false location claims.

B. Simulation Results

Varying percentage of false location claims. In this set of experiments, we vary the percentage of photo tasks with false location claims. The resulting graph, plotted in Figure 5, has multiple curves as a function of the percentage of photo tasks submitting false location. This graph is plotted to gain insights on what will be the right percentage of photo tasks needed in Phase 1 to bootstrap the ILR scheme. Next, we analyze Figure 5;

- **Low count of malicious tasks submitted:** When 10% of total photo tasks are submitting false location, Figure 5 shows that the ILR scheme can detect 55% of the false location claims just by using 10% of the total photo tasks validated in Phase 1. This figure also shows that in order to detect more false claims, more photos need to be manually validated: for example, ILR uses up to 40% of the total photo tasks in Phase 1 to detect 80% of the false location tasks. Finally, Figure 5 shows that increasing the percentage of validated photo tasks above 40% does not help much as the percentage of detected false tasks remains the same;
- **High count of malicious tasks submitted:** When 60% of the total photo tasks are submitting false location, Figure 5 shows that ILR can still detect 35% of the false claims by using 10% of the total photo tasks in Phase 1. But in this case, ILR requires more validated photo tasks(70%) to catch 75% of the false claims. This is because by increasing the number of malicious tasks, the co-location data is reduced and therefore ILR cannot extend trust to more location claims in its Phase 2.

Therefore, we conclude that the right percentage of photo tasks needed to bootstrap the ILR is proportional to the expected false location claims (which can be predicted using the history of the users' participation).

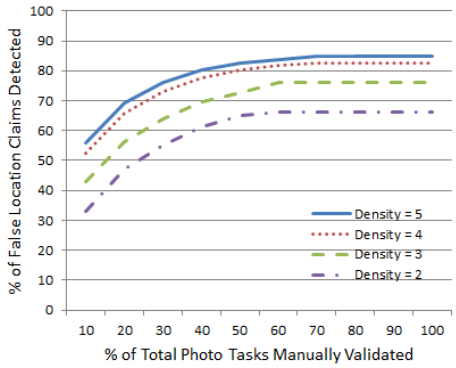

Fig. 6. ILR performance as function of the percentage of photos manually validated in phase 1. Each curve represents a different network density represented as average number of neighbors per node.

Node density impact on the ILR scheme. In this set of experiments, we assume that 10% of the total photo tasks are submitting false locations. In Figure 6 we analyze the impact of node density on the ILR scheme. We seek to estimate the minimum node density required to achieve highly connected graphs to extend the location trust transitively to more co-located nodes;

- **High Density:** When simulations are run with node density of 5, Figure 6 shows ILR can detect the highest percentage (85%) of the false location claims. The figure also shows similarly high results even for a node density of 4;
- **Low Density:** When simulations are run with node density of 2, we can see that ILR can still detect 65% of the false location tasks using 50% of the total photo tasks in Phase 1. For this node density, even after increasing the number of validated photo tasks in Phase 1, the percentage of detected false claims does not increase. This is because of there are fewer co-located users at low node densities.

Therefore, we conclude that ILR can efficiently detect false claims with a low number of manual validations, even for low node densities.

VI. Field Study Insights and Improving the ILR Scheme

In this section, we present our insights from the analysis of the data collected from the field study and discuss possible improvements of the ILR scheme based on these insights. In addition, we present observations of the survey that was collected from users at the end of the field study to understand the participants' opinion on location privacy and usage of phone resources.

A. Correlation of User Earnings and Fake Photos

To understand the correlation between the user earnings and the number of fake photos submitted, we plot the data collected from the McSense crowd sensing field study. The experimental results in Figure 7 show that the users who submitted most of

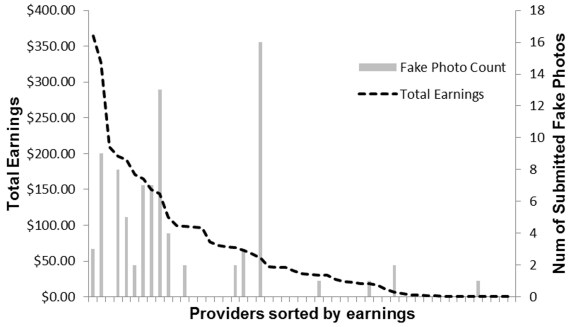

Fig. 7. Correlation of earnings and fake photos.

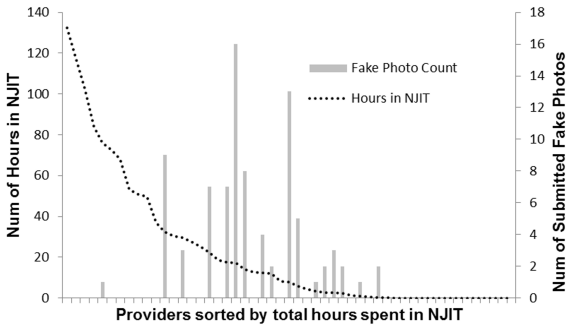

Fig. 8. Correlation of User Location and Fake Photos.

the fake photos are among the top 20 high earners (with an exception of 4 low earning users who submitted fake photos once or twice). This is an interesting observation that can be leveraged to improve the ILR scheme.

In the current ILR scheme, there are cases where the validation process cannot make a firm decision on some data points. Those data points fall under the "Unknown" category as described in the "ILR Design" section. If these "Unknown" data points are too many (in millions), then it becomes challenging to validate all of them manually. Therefore, to improve the ILR scheme, we propose that these "Unknown" cases of data points must go through an extra check to find whether the user is a high earner in the sensing system. If the user is a high earner, then there is a high probability that the user submitted data point is fake. Those photos should be manually validated. If the user is a mid/low range earner, then there is a low probability of his/her data point being faked and the data point should be considered as valid. This method will help in reducing the number of photos that require manual validation.

B. Correlation of Location and Fake Photos

We ask the question "Is there any correlation between the amount of time spent by users on campus and the number of submitted fake photos?" As suspected, the users who spent less time on campus have submitted more fake photos. This behavior can be observed in Figure 8.

Figure 8 shows the number of fake photos submitted by

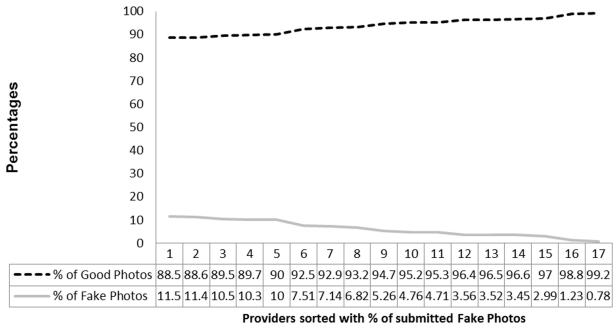

Fig. 9. Photo counts of 17 cheating people.

each user, with the users sorted by the total hours spent on the NJIT campus. The participants' total hours recorded at NJIT campus are the hours that are accumulated from the sensed data collected from "Automated Sensing task" described in the "Tasks Developed for McSense" section. The NJIT location is considered to be a circle with a radius of 0.5 miles. If the user is in circle, then she is considered to be at NJIT. For most of the submitted fake photos with the false location claim, the users claimed that they are at a campus location where the photo task is requested, but actually they are not frequent visitors on the campus.

This is an interesting observation, which can also be leveraged to improve the ILR scheme's validation process when there is a large number of data points classified as "Unknown". The intuition behind this argument is that users tend to fake the data mostly when they are not around the task's location. Therefore, to improve the ILR scheme, we propose to use a user's recorded location trail in the McSense system in order to identify whether the user is or is not a frequent visitor of the task's location. If the user is not a frequent visitor of the claimed location, then there is a high probability that location claim is false and the "Unknown" data point should be manually checked. On the other hand, if the user is a frequent visitor of the claimed location, then her claim can be trusted. By reducing the number of photos that require manual validation, McSense can improve ILR's validation process for "Unknown" data points.

C. Malicious User: Menace or Nuisance?

The photos submitted by malicious users are plotted in Figure 9. The data show that malicious users have submitted good photos at a very high rate compared to the fake photos. These malicious users are among the high earners, so they are submitting more data than the average user. Thus, it may not be a good idea to remove the malicious users from the system as soon as they are caught cheating.

Instead, it may be a better idea to identify the validity of the individual data points (which is exactly the same process done in the current ILR scheme discussed in "ILR Design" section). We conclude that the malicious users are not a significant menace, but may cause some confusion in the collected data. However, this can be filtered out by McSense through correlating the data with location and earnings as discussed earlier in the section.

D. Influence of Maintaining a Reputation Score

When a fake location claim is detected by ILR, McSense would benefit if the malicious user who submitted the fake claim receives a lower compensation upon completion of a task. In order to perform such a process, McSense should use a reputation module such as in [62], which maintains a trust score for each user. This is similar with many other systems which rely on the participation of users. The trust score varies between 0 and 1. Initially, the trust score is given a default value to every user, and it evolves depending on the user participation. The trust score is reduced when the user is caught providing fake data and is increased when the user submits good data.

We propose that the McSense system maintains a trust score for every user, and then it uses this score for calculating the user payment upon task completion. For example, for a completed task that is worth $5, a user with trust score 0.9 will be paid only $4.5. We envision that by maintaining a reputation score, the users providing fake data will eventually stop making false claims. We have seen earlier in the section that the malicious users also submit a significant amount of good data. But, if their trust score drops to 0, then the malicious users will not participate anymore as they do not earn the task amount and will eventually leave the system. As we argued earlier in this section, it is not a good idea to entirely remove the malicious users from the system. Therefore, to avoid eliminating malicious users from the system, the trust score will not decrease anymore after reaching a minimum threshold (e.g., 0.2). Hence, the malicious user will only get 20% of the task dollars until she improves her trust score. Therefore, the McSense system does not need to worry about discarding good data that is submitted by malicious users.

E. Users Survey Results and Observations

At the end of the field study, we requested each user to fill a survey in order to understand the participants' opinion on location privacy and usage of phone resources. The survey contains 16 questions with answers on a five-point Likert scale (1="Strongly disagree", 2="Disagree", 3="Neutral", 4="Agree", 5="Strongly agree"). Out of 58 participants, 27 filled in the survey. Based on the survey answers, we provide next a few interesting observations which are directly or indirectly relevant in the context of data reliability:

- One of the survey questions was: "I tried to fool the system by providing photos from other locations than those specified in the tasks (the answer does not influence the payment)". By analyzing the responses for this specific question, we observe that only 23.5% of the malicious users admitted that they submitted the fake photos (4 admitted out of 17 malicious). This shows that the problem stated in the article on data reliability is real and it is important to validate the sensed data;

- One survey question related to the user privacy was: "I was concerned about my privacy while participating in the user study". The survey results show that 78% of the users are not concerned about their privacy. This shows that many participants are willing to trade off their location privacy for paid tasks. The survey results are correlated with the collected McSense data points. We posted a few sensing tasks during weekends, which is considered to be private time for the participants who are mostly not in the campus at that time. We observe that 33% of the participants participated in the sensing and photo tasks, even when spending their personal time in the weekends. We conclude that the task price plays a crucial role (trading the user privacy) to collect quality sensing data from any location and time;
- Another two survey questions are related to the usage of phone resources (e.g., battery) by sensing tasks: 1) "Executing these tasks did not consume too much battery power (I did not need to re-charge the phone more often than once a day)"; 2) "I stopped the automatic tasks (resulting in incomplete tasks) when my battery was low". The responses to these questions are interesting. Most of the participants reported that they were carrying chargers to charge their phone battery as required while running the sensing tasks and were keeping their phone always ready to accept more sensing tasks. This proves that phone resources, such as battery, are not a big concern for continuously collecting sensing data from different users and locations. We describe next the battery consumption measurements in detail.

F. Battery Consumption

We try to determine the amount of energy consumed by the user's phone battery for collecting sensing data that is required for ILR. Basically, ILR is executed on the server side over the collected data. But the collected data such as Bluetooth scans at each location is crucial for ILR. Next, we provide measurements for the extra battery usage caused by keeping Bluetooth/Wi-Fi radios ON. We measured the readings using "Motorola Droid 2" smart phones running Android OS 2.2:

- With Bluetooth and Wi-Fi radios ON, the battery life of the "Droid 2" phone is over 2 days (2 days and 11 hours);
- With Bluetooth OFF and Wi-Fi radio ON the battery life of the "Droid 2" phone is over 3 days (3 days and 15 hours);
- For every Bluetooth discovery the energy consumed is 5.428 Joules. The total capacity of the "Droid 2" phone battery is: 18.5KJ. Hence, over 3000 Bluetooth discoveries can be collected from different locations using a fully charged phone.

VII. Conclusions

This chapter presented the concept of mobile crowd sensing and its applications to everyday life. We described the design and implementation of McSense, our mobile crowd sensing platform, which was used to run a user study with over 50 users at the NJIT campus for a period of 2 months. We also discussed the data reliability issues in mobile crowd sensing by presenting several scenarios involving malicious behavior. We presented a protocol for location reliability as a step toward achieving data reliability in sensed data, namely, ILR (Improving Location Reliability). ILR also detects false location claims associated with the sensed data. Based on our security analysis and simulation results, we argue that ILR works well at various node densities. The analysis of the sensed data collected from the users in our field study demonstrate that ILR can efficiently achieve location data reliability and detect a significant percentage of false location claims. Therefore, we conclude this chapter with our belief that mobile crowd sensing will become a widespread method for collecting sensing data from the physical world once the data reliability issues are properly addressed.

Acknowledgment

Parts of this chapter were published in the IJBDCN journal [63]. Reused with permission of the publisher.

References

[1] O. Riva and C. Borcea, "The urbanet revolution: Sensor power to the people!" *Pervasive Computing, IEEE*, vol. 6, no. 2, pp. 41–49, 2007.
[2] Smart phone sensing research @ dartmouth college. [Online]. Available: http://sensorlab.cs.dartmouth.edu/research.html
[3] Urban sensing research @ ucla. [Online]. Available: http://urban.cens.ucla.edu/
[4] Global smartphone shipments forecast. [Online]. Available: http://www.statista.com/statistics/263441/global-smartphone-shipments-forecast/
[5] S. Mardenfeld, D. Boston, S. J. Pan, Q. Jones, A. Iamntichi, and C. Borcea, "Gdc: Group discovery using co-location traces," in *Social computing (SocialCom), 2010 IEEE second international conference on*. IEEE, 2010, pp. 641–648.
[6] Reality mining project. [Online]. Available: http://reality.media.mit.edu/
[7] Garmin, edge 305. [Online]. Available: www.garmin.com/products/edge305/
[8] Intel labs, the mobile phone that breathes. [Online]. Available: http://scitech.blogs.cnn.com/2010/04/22/the-mobilephone-that-breathes/
[9] Mit news. [Online]. Available: http://web.mit.edu/newsoffice/2009/blood-pressure-tt0408.html
[10] T. Abdelzaher, Y. Anokwa, P. Boda, J. A. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich, "Mobiscopes for human spaces," *Center for Embedded Network Sensing*, 2007.
[11] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, "People-centric urban sensing," in *Proceedings of the 2nd annual international workshop on Wireless internet*. ACM, 2006, p. 18.
[12] R. Honicky, E. A. Brewer, E. Paulos, and R. White, "N-smarts: networked suite of mobile atmospheric real-time sensors," in *Proceedings of the second ACM SIGCOMM workshop on Networked systems for developing regions*. ACM, 2008, pp. 25–30.
[13] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 323–336.
[14] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. LeGrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway *et al.*, "The mobile sensing platform: An embedded activity recognition system," *Pervasive Computing, IEEE*, vol. 7, no. 2, pp. 32–41, 2008.
[15] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 261–272.
[16] A. Kansal, M. Goraczko, and F. Zhao, "Building a sensor network of mobile phones," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 547–548.

[17] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "Soundsense: scalable sound sensing for people-centric applications on mobile phones," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 2009, pp. 165–178.

[18] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 337–350.

[19] D. Siewiorek, A. Krause, N. Moraveji, A. Smailagic, J. Furukawa, K. Reiger, F. L. Wong, and J. Shaffer, "Sensay: A context-aware mobile phone," in *2012 16th International Symposium on Wearable Computers*. IEEE Computer Society, 2003, pp. 248–248.

[20] Amazon mechanical turk. [Online]. Available: http://www.mturk.com

[21] chacha: Your mobile bff. [Online]. Available: http://www.chacha.com

[22] A. Gupta, W. Thies, E. Cutrell, and R. Balakrishnan, "mclerk: enabling mobile crowdsourcing in developing regions," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*. ACM, 2012, pp. 1843–1852.

[23] P. Narula, P. Gutheim, D. Rolnitzky, A. Kulkarni, and B. Hartmann, "Mobileworks: A mobile crowdsourcing platform for workers at the bottom of the pyramid." in *Human Computation*, 2011.

[24] Londons water supply monitoring. [Online]. Available: http://www.ucl.ac.uk/news/news-articles/May2012/240512-

[25] Ibm smarter planet. [Online]. Available: http://\textbf{}www.ibm.com/smarterplanet/us/en/overview/ideas/

[26] Songdo smart city. [Online]. Available: http://www.songdo.com

[27] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, and R. Curtmola, "Fostering participaction in smart cities: a geo-social crowdsensing platform," *Communications Magazine, IEEE*, vol. 51, no. 6, 2013.

[28] Mobile millennium project. [Online]. Available: http://traffic.berkeley.edu/

[29] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, "Parknet: drive-by sensing of road-side parking statistics," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 123–136.

[30] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*. ACM, 2008, pp. 29–39.

[31] Google mobile ads. [Online]. Available: http://www.google.com/ads/mobile/

[32] Mobads. [Online]. Available: http://www.mobads.com/

[33] Twitter. [Online]. Available: http://twitter.com/

[34] M. Demirbas, M. A. Bayir, C. G. Akcora, Y. S. Yilmaz, and H. Ferhatosmanoglu, "Crowd-sourced sensing and collaboration using twitter," in *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*. IEEE, 2010, pp. 1–9.

[35] T. Yan, M. Marzilli, R. Holmes, D. Ganesan, and M. Corner, "mcrowd: a platform for mobile crowdsourcing," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*. ACM, 2009, pp. 347–348.

[36] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 2009, pp. 55–68.

[37] M. Ra, B. Liu, T. La Porta, and R. Govindan, "Medusa: A programming framework for crowd-sensing applications," in *Proceedings of the 10th international conference on Mobile systems, applications, and services (MobiSys'12)*. ACM, 2012, pp. 337–350.

[38] J. Herrera, D. Work, R. Herring, X. Ban, Q. Jacobson, and A. Bayen, "Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 4, pp. 568–583, 2010.

[39] J. White, C. Thompson, H. Turner, B. Dougherty, and D. Schmidt, "Wreckwatch: automatic traffic accident detection and notification with smartphones," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 285–303, 2011.

[40] K. Toyama, R. Logan, and A. Roseway, "Geographic location tags on digital images," in *Proceedings of the eleventh ACM international conference on Multimedia*. ACM, 2003, pp. 156–166.

[41] Photo journalism website. [Online]. Available: http://www.flickr.com/groups/photojournalism

[42] Sensordrone: The 6th sense of your smartphone. [Online]. Available: http://www.sensorcon.com/sensordrone

[43] Trusted platform module. [Online]. Available: http://www.trustedcomputinggroup.org/developers/trusted_platform_module

[44] M. Talasila, R. Curtmola, and C. Borcea, "Link: Location verification through immediate neighbors knowledge," in *Proceedings of the 7th International ICST Conference on Mobile and Ubiquitous Systems, (MobiQuitous'10)*. Springer, 2010, pp. 210–223.

[45] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proceedings of the 2nd ACM workshop on Wireless security (WiSe'03)*. ACM, 2003, pp. 1–10.

[46] S. Capkun and J. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *INFOCOM'05. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3. IEEE, 2005, pp. 1917–1928.

[47] S. Čapkun, M. Čagalj, and M. Srivastava, "Secure localization with hidden and mobile base stations," in *in Proceedings of IEEE INFOCOM*. Citeseer, 2006.

[48] M. Talasila, R. Curtmola, and C. Borcea, "Improving location reliability in crowd sensed data with minimal efforts," in *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*. IEEE, 2013, pp. 1–8.

[49] Mcsense android smartphone application. [Online]. Available: https://play.google.com/store/apps/details?id=com.mcsense.app

[50] Google play android app store. [Online]. Available: https://play.google.com/

[51] T. Humphreys, B. Ledvina, M. Psiaki, B. OHanlon, and P. Kintner Jr, "Assessing the spoofing threat: Development of a portable gps civilian spoofer," in *Proceedings of the ION GNSS International Technical Meeting of the Satellite Division*, 2008.

[52] P. Gilbert, J. Jung, K. Lee, H. Qin, D. Sharkey, A. Sheth, and L. Cox, "Youprove: authenticity and fidelity in mobile sensing," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys'11)*. ACM, 2011, pp. 176–189.

[53] A. Dua, N. Bulusu, W. Feng, and W. Hu, "Towards trustworthy participatory sensing," in *HotSec'09: Proceedings of the Usenix Workshop on Hot Topics in Security*, 2009.

[54] G. Xu, C. Borcea, and L. Iftode, "A policy enforcing mechanism for trusted ad hoc networks," *Dependable and Secure Computing, IEEE Transactions*, vol. 8, no. 3, pp. 321–336, 2011.

[55] J. McCune, B. Parno, A. Perrig, M. Reiter, and H. Isozaki, "Flicker: An execution infrastructure for tcb minimization," *SIGOPS Operating Systems Review*, vol. 42, no. 4, pp. 315–328, 2008.

[56] M. Nauman, S. Khan, X. Zhang, and J. Seifert, "Beyond kernel-level integrity measurement: enabling remote attestation for the android platform," *Trust and Trustworthy Computing*, pp. 1–15, 2010.

[57] F. B. Schneider, K. Walsh, and E. G. Sirer, "Nexus authorization logic (nal): Design rationale and applications," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 8:1–8:28, Jun. 2011.

[58] P. Gilbert, L. Cox, J. Jung, and D. Wetherall, "Toward trustworthy mobile sensing," in *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications (HotMobile'10)*. ACM, 2010, pp. 31–36.

[59] S. Saroiu and A. Wolman, "I am a sensor, and i approve this message," in *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications (HotMobile'10)*. ACM, 2010, pp. 37–42.

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

[61] M. Talasila, R. Curtmola, and C. Borcea, "Collaborative bluetooth-based location authentication on smart phones," *Pervasive and Mobile Computing*, 2014.

[62] X. Chu, X. Chen, K. Zhao, and J. Liu, "Reputation and trust management in heterogeneous peer-to-peer networks," *Telecommunication Systems*, vol. 44, no. 3-4, pp. 191–203, 2010.

[63] M. Talasila, R. Curtmola, and C. Borcea, "Ilr: Improving location reliability in mobile crowd sensing," *International Journal of Business Data Communications and Networking (IJBDCN)*, vol. 9, no. 4, pp. 65–85, 2013.