Improving Location Reliability in Crowd Sensed Data with Minimal Efforts

Manoop Talasila, Reza Curtmola, and Cristian Borcea
Computer Science Department
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 by leveraging the cameras, microphones, GPSs, accelerometers, and other sensors on the phones. Ranging from manual photo tasks to automated sensing tasks for activity monitoring, any task can be crowd sourced to smart phones to sense data from different locations at reduced cost. However, the sensed data submitted by participants is not always reliable as they can submit false data to earn money without executing the actual task. Therefore, it is important to validate the sensed data. Validating the context of every sensed data point of each participant is not a scalable solution. One alternative is to first validate the location associated with the sensed data points in order to achieve a certain degree of reliability about the sensed data. However, location validation without support from the wireless carriers is difficult.**

To address this problem, we propose ILR, a scheme in which we Improve the Location Reliability of mobile crowd sensed data with minimal human efforts. In this scheme, we bootstrap the trust in the system by first manually or automatically using image processing techniques validating a small number of photos submitted by participants. Based on these validations, the location of these photos is assumed to be trusted. Second, we extend this location trust to co-located sensed data points found in the Bluetooth range of the devices that provided the validated photos. This transitive trust is extended until all the co-located tasks are trusted or no new data points are found. In addition, the scheme also helps to detect false location claims associated with sensed data. We applied ILR on data collected from our McSense prototype deployed on Android phones used by students on our campus and detected a significant percentage of the malicious users. Simulation results demonstrate that ILR works well at various densities and helps detect the false location claims based on a minimal number of validations.

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]. This new type of sensing can be a scalable and cost-effective alternative to deploying static wireless sensor networks for dense sensing coverage across large areas. Many clients can use this mobile people-centric sensing on demand and pay just for the actual usage (i.e., collected data). mCrowd [4] and Medusa [5] are some of the recent mobile crowd sensing platforms proposed to provide a common platform to perform any kind of sensing tasks supported by the smart phone sensors.

Mobile crowd sensing can be used to enable a broad spectrum of applications, ranging from monitoring pollution or traffic in cities to epidemic disease monitoring or real-time reporting from disaster situations. While all of us could directly take advantage of such applications (e.g., real-time traffic monitoring), we believe that researchers in many fields of science and engineering as well as 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. Additionally, commercial organizations may be very interested in collecting mobile sensing data to learn more about customer behavior.

A major challenge for broader adoption of these sensing systems is that the sensed data submitted by the participants is not always reliable [6] as they can submit false data to earn money without executing the actual task. Clients need guarantees from the mobile crowd sensing system that the collected data is valid. Hence, it is very important to validate 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 the 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. Therefore, in this paper we focus on validating the location data submitted by the participants. 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 [1]?

To achieve reliability on participant's location data, there are a few traditional solutions such as using Trusted Platform Modules (TPM) [7] on smart phones or duplicating the tasks among multiple participants. However, these 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 [8]–[11] in real time when the participant is trying to submit the sensing data location. Unfortunately, this solution requires infrastructure support and adds a very high overhead on users phone if it is applied for each sensed data point.

[1]Wireless carriers may not help with location validation for legal reasons related to user privacy or even commercial interests.

We propose ILR (**I**mproving **L**ocation **R**eliability), a scheme in which we utilize participatory sensing itself to achieve data reliability. This scheme is based on location validation using photo tasks and expanding the trust to nearby data points using periodic Bluetooth scanning. In this scheme, the system asks the participants to execute a number of photo tasks at known locations. Then, these photos are manually or automatically validated to ensure that they have been taken at the correct location [12]. The location and time details of these validated photo tasks will be our reference data points in the process of validating other data points collected nearby at the same time. The participants who generate reference data points become "Validators". Periodic Bluetooth scanning on each phone discovers other participants who are co-located with the Validators. ILR applies transitive closure and extends the trust from the Validators to data points collected by these participants. This transitive trust is extended until all co-located data points are trusted or no additional data points are found. ILR also detects false location claims by using these trusted data points. The major advantage of using the ILR compared to verifying location in real time is that our scheme does not require any overhead processing on the participant's phone when submitting sensed data, and thus it results in quicker data submission and has no impact on the phone's battery.

We evaluated ILR in the context of our McSense project. McSense is a crowd sensing system, and its sensing application is deployed in Google Play [13] to execute crowd sensing tasks for collecting photos, location, accelerometer, Bluetooth scans, and other phone sensing data. We collected data from over 50 users (students on our campus) during a two-month interval. Even though, Bluetooth scanning was executed rarely (as this was also a paid task), ILR was able to detect 40% of the users submitting photos from false locations; for ground truth validation, we manually inspected these photos.

These experimental results gave us confidence that ILR could work well in practice. Thus, to understand its behavior at scale, when Bluetooth scans are done periodically by all participants, we ran simulations. The results demonstrate that ILR works well at various node densities and helps detect false location claims based on a minimal number of validations.

In summary, we make the following contributions:

- We propose ILR, a scheme which Improves the Location Reliability of mobile crowd sensed data with minimal human efforts. The scheme also detects false location claims associated with the sensed data.
- We evaluate the proposed scheme on real-world data by developing McSense, a mobile crowd sensing system which is deployed on the Android market.
- Based on security analysis and simulation results, we show that ILR works well at various node densities.

The rest of the paper is organized as follows. Section II presents motivating real-world scenarios. Section III defines the assumptions and the adversarial model. Section IV describes the ILR scheme, and analyzes its security. The implementation and experimental evaluation are presented in Sections V and VI. Section VII presents the simulation results. The related work is discussed in Section VIII. The paper concludes in Section IX.

II. Motivation

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 validate the collected data efficiently. In the following, we motivate the need for such a mechanism by presenting several scenarios involving malicious behavior.

Traffic jam alerts [14], [15]: 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 [16], [17]: 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 [18], [19]: 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, location data validation is important in a mobile crowd sensing system to provide confidence to its clients who use the sensed data.

III. Preliminaries

This section defines the interacting entities in our environment, the assumptions we make about the system, and the adversarial model.

Interacting entities. The entities in the system are:

- *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.

Assumptions. We consider that McSense posts tasks to collect sensing data on behalf of clients. Providers execute any available task and report the sensed data back to McSense, which

delivers it to clients pending validation. 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. In the context of this paper, we use the terms "data point" and "task" interchangeably.

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 [20] 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 IV-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. It is not practically 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 be seen in at-least a minimum number of other providers' Bluetooth scans 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.

Finally, the other class of attacks that are out of scope for our current scheme are attacks in which a provider 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), but submits the right location and Bluetooth scan associated with this sensing task.

IV. Proposed Scheme

In this section we present the ILR 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.

Before going into the details of the scheme, we assume that the sensed data is already collected by the McSense system from providers at different locations. However, this sensed data is awaiting validation before being sent to the actual clients who requested this data.

For ILR, we will assume that the sensed data includes location, time and a Bluetooth scan performed at the task's location and time. 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 photo and co-location data.

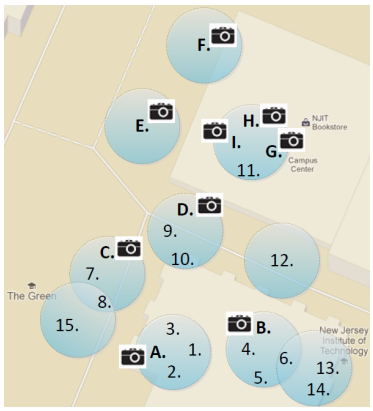

Fig. 1. Example of McSense collected Photo tasks [A-I] and Sensing tasks [1-15] on the campus map, grouped using Bluetooth discovery co-location data.

A. An example of ILR in action

Figure 1 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 (i.e., 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 will choose these 5 photo tasks for validation. These selected photo tasks are validated either manually or automatically (we discuss this in detail in section IV-B). When validating these photo tasks, if the photo is not valid then its photo is rejected and McSense ignores its Bluetooth scans. If the photo is valid then McSense will consider 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 very difficult 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 will extend the location trust

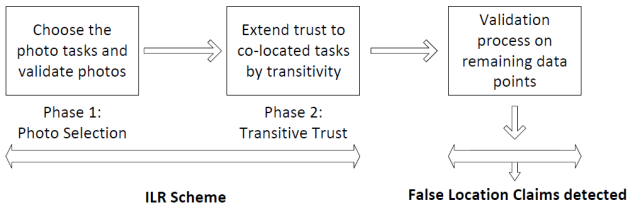

Fig. 2. The phases of the ILR scheme.

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 will extend trust to the tasks 1, 2, and 3, and B will extend trust to tasks 4, 5, and 6. Now task 6 will extend its trust to tasks 13 and 14. Finally, after the end of this process, McSense system will have 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 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.

B. The ILR Scheme

The ILR scheme has two phases as shown in Figure 2. "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.

1) Phase 1 - Photo Selection: Using collected data from Bluetooth scans of providers, we construct 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 as discussed in the above example). From these graphs, we elect the photo tasks that have node degree greater than a threshold (N_{th}).

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 [21] 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".

2) 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 discovery of the validated photo task. In 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)

C. 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 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 is not co-located in the Bluetooth scans of photo task A. In addition to the validation process, McSense will also do a basic spatio-temporal correlation check to ensure that the provider is not claiming 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 III). 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, then McSense will detect the provider in it neighbors' Bluetooth scans at a different location using the spatio-temporal correlation check and will reject the task's data.

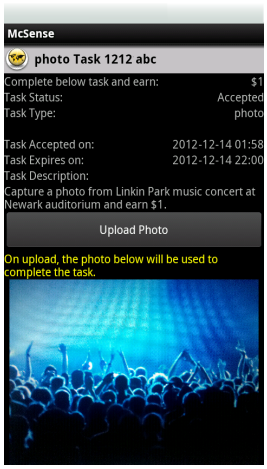

Fig. 3. McSense Android Application Task Screen for a Photo Task

Finally, when a provider submits fake Bluetooth discovery with a false location claim, then the scheme 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 III), sensed data submitted by malicious colluding attackers could be filtered to a certain extent in McSense by setting the node degree threshold (N_{th}) to the minimum node degree requirement requested by the client.

V. Overview of McSense Implementation and Deployment

We implemented McSense to create a platform to deploy participatory sensing tasks in real time to campus students and other participants. Among other tasks, we deployed: 1) Photo tasks, 2) Automated Accelerometer and GPS Sensing tasks, and 3) Automated Bluetooth Sensing tasks.

A. Prototype Implementation

The McSense application, as shown in Figure 3, has been implemented in Android and is compatible with smart phones having 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, WiFi 802.11 b/g/n, 8 GB on board storage, and 8 GB microSD storage. The McSense [22] Android application was deployed to Google Play 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.

B. Tasks Developed for McSense

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

1) Manual tasks, e.g., photo tasks
2) 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. Once 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 their current news articles. On click of "Complete Task" button, if network is not available, the photo task is marked as completed and waiting for upload. This task is shown with a pending icon under completed tasks tab. Then a background service takes care of uploading the pending photos when the network becomes available. If a photo is uploaded to server after the task expiration time, then the photo is useless for the client. Therefore, the task will be marked as "Unsuccessfully completed", and the user do not earn money for this task.

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 will be uploaded to the application server on completion of the task (which consists of many data points). Using the collected sensed data of accelerometer readings and GPS readings, we can identify users activities like walking, running, or driving. By observing the daily activities, we could find out how much exercise each student is getting daily and derive interesting statistics such as which department has 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 (*every 5mins*) Bluetooth scans until the task expires; the task reports the discovered Bluetooth devices with their location back to the McSense server on its completion. 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.

To participate in the study, students 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 provider's participation in collecting free 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

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

accumulate credits (payable in cash after the study terminated). We conducted the study for approximately 2 months.

VI. 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, and we acted as the clients collecting the sensed data in these experiments.

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 the ILR scheme can help McSense to 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 people; only 204 photo tasks are having Bluetooth scans associated with them. In this data set, we considered the $NodeDegree \geq 1$, 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).

VII. 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.

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

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 4, 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 4:

- **Low count of malicious tasks submitted:** When 10% of total photo tasks are submitting false location, Figure 4 shows that just by using 10% of the total photo tasks validated in Phase1, the ILR scheme can detect 55% of the false location claims. This figure also shows that in

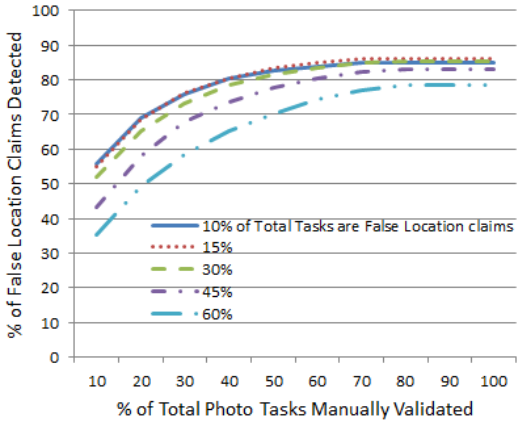

Fig. 4. ILR performance as function of the percentage of photos manually validated in phase 1. Each curve represents a different percentage of photos with fake locations.

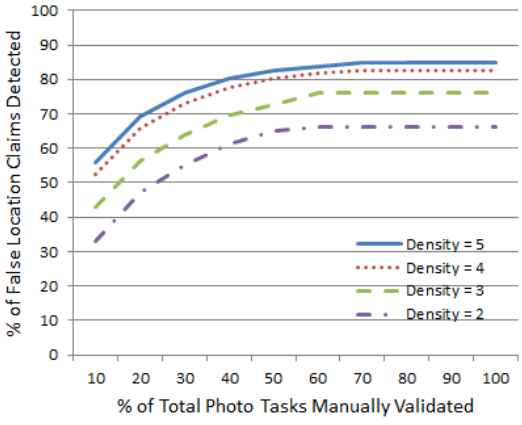

Fig. 5. ILR performance as function of the percentage of photos manually validated in phase 1. Each curve represents a different network density represented as average number of neighbors per node.

order to detect more false claims, we can use up to 40% of the total photo tasks in Phase 1 to detect 80% of the false location tasks. Finally, Figure 4 shows that increasing the percentage of validated photo tasks above 40% will 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 4 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 scheme 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 scheme is proportional to the expected false location claims (which can be predicted using the history of the users' participation).

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 5 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 5 shows the ILR scheme 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 the ILR scheme 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 the ILR scheme can efficiently detect false claims with a low number of manual validations, even for low node densities.

VIII. Related Work

The idea of mobile people-centric sensing was introduced fairly recently, but a lot of progress was made already. MetroSense [23], Participatory Sensing [24], and Urbanets [1] were among the first projects to demonstrate the feasibility of the idea. Recently, the Medusa framework [5] was proposed to provide a common platform to perform any type of sensing task supported by smart phone sensors. However, more research is still required in providing guarantees to clients that the collected data is reliable.

Trusted hardware represented by the Trusted Platform Module (TPM) [7], [25]–[27] has been leveraged to design new architectures for trustworthy software execution on mobile phones [28]–[30]. Recent work has also proposed architectures to ensure that the data sensed on mobile phones is trustworthy [31], [32]. When untrusted client applications perform transformations on the sensed data, YouProve [26] 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 [20].

In a recent work, we proposed the LINK protocol [8] for secure location verification, which does not require 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 location, while ILR is executed on the collected data from mobile crowd sensing. Therefore, employing ILR helps providers in submitting the sensing data quickly and also consumes less phone battery.

IX. Conclusions

This paper presented ILR, a scheme to increase the reliability of mobile crowd sensed data with minimal human efforts. ILR also detects false location claims associated with the sensed data. Based on security analysis and simulation results, we show that ILR works well at varying node densities. We evaluated the proposed scheme on real data, by developing McSense – a mobile crowd sensing system – which is deployed in the Android market. The analysis on sensed data collected from over 50 users during a two-month period demonstrated that ILR is efficient in attaining location data reliability and in detecting a significant percentage of false location claims.

Acknowledgment

This research was supported by the National Science Foundation under Grant No. CNS 0831753 and CNS 1054754. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

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] 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.

[5] 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.

[6] J. Downs, M. Holbrook, S. Sheng, and L. Cranor, "Are your participants gaming the system?: screening mechanical turk workers," in *Proceedings of the 28th international conference on Human factors in computing systems (CHI'10)*. ACM, 2010, pp. 2399–2402.

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

[8] 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.

[9] 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.

[10] 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.

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

[12] N. Ravi, P. Shankar, A. Frankel, A. Elgammal, and L. Iftode, "Indoor localization using camera phones," in *Proceedings of the Seventh IEEE Workshop on Mobile Computing Systems & Applications (WMCSA'06)*. IEEE, 2006, pp. 19–25.

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

[14] 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.

[15] 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.

[16] 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.

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

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

[19] Intel labs, the mobile phone that breathes. [Online]. Available: http://scitech.blogs.cnn.com/2010/04/22/the-mobilephone-that-breathes/

[20] 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.

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

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

[23] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, R. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G. Ahn, "The rise of people-centric sensing," *Internet Computing, IEEE*, vol. 12, no. 4, pp. 12–21, 2008.

[24] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," in *In: Workshop on World-Sensor-Web (WSW'06): Mobile Device Centric Sensor Networks and Applications*, 2006, pp. 117–134.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.