Preserving QoI in Participatory Sensing by Tackling Location-Spoofing through Mobile WiFi Hotspots

Francesco Restuccia, Andrea Saracino, Sajal K. Das and Fabio Martinelli

***Abstract*— The Quality of Information (QoI) in Participatory Sensing (PS) systems largely depends on the location accuracy of participating users. However, users could easily provide false information through Location Spoofing Attacks (LSA). Existing PS systems are not able to efficiently validate the position of users in large-scale outdoor environments, thus being prone to reduced QoI. In this paper we present an efficient scheme to secure PS systems from LSAs. In particular, the user location is verified with the help of mobile WiFi hot spots (MHSs), which are users activating WiFi interface on their smartphones and waiting connections from nearby users, and thereby validating their position inside the sensing area. A reputation-based algorithm is proposed to rule out sensing reports of location-spoofing users, thereby increasing the reliability of the PS system. The effectiveness of our scheme is analyzed by real-world experiments and simulation study.**

***Index Terms*—Participatory Sensing, Smartphones, Security, WiFi Hotspots, Location Spoofing.**

I. Introduction

Undoubtedly, smartphones have become one of the most powerful and pervasive technologies today. Due to their popularity, smartphones are ideally suited for a novel and tremendously potential sensing paradigm, known as *participatory sensing* (PS) [1]. The basic idea behind PS is to allow the citizens to participate in large-scale sensing surveys with the help of user-friendly applications installed in their smartphones. This not only reduces the deployment costs of fixed infrastructures, but also provides fine-grained spatio-temporal coverage of the sensing area. Significant research and development from industry and academia has been devoted to design PS systems to improve life experience of users. Indeed an abundance of real-life applications, which take advantage of both low-level sensor data and high-level user activities, range from real-time traffic monitoring [2] [3] to air pollution or garbage monitoring [4]–[6] to social networking [7], to name a few.

Participatory sensing applications mostly depend on the user location; for example, it is not meaningful to communicate the presence of a traffic jam to a PS system without also providing a location for the user or the jam. Even worse, if a user provides, intentionally or by mistake, a wrong (false) position, the injected data to the PS system will be false as well. This misbehavior, known as *Location Spoofing Attack (LSA)*, seriously affects the quality of information (QoI) in a PS system. Unfortunately, providing a false location is a simple action for average smartphone users. In fact, smartphone applications (apps) like *FakeLocation*[1] make it extremely easy for the user to spoof her current global position system (GPS) location. Such software provides the users with easy-to-use interfaces to manually set the GPS coordinates of their device. This location is read and used by all applications on the device, including PS apps. A user of such software may thus send wrong information to the PS system by mistake (e.g. she is not aware of the interaction between *FakeLocation* and the PS app), or by purpose. For example, if the PS system uses an incentive mechanism [8] to stimulate users' participation, a malicious user may exploit *FakeLocation* or similar software to locate herself in the area and obtain unfair credits for her data.

The above discussion implies that solving LSAs for PS systems is of paramount importance. However, given the extremely large scale of real-world PS systems, verifying the location of users becomes remarkably challenging. This motivates our work.

In this paper, we propose a scheme which efficiently and effectively tackles LSAs in PS systems. The proposed scheme exploits the ad-hoc WiFi capability of modern smartphones to validate the position of other users. In fact, two smartphones directly connected through ad-hoc WiFi practically share the same location, due to the limited WiFi range. Thus, these two users can mutually validate their locations inside the sensing area. By exploiting this technique on a large scale, our approach implements an effective, scalable and distributed anti location-spoofing system. A reputation-based algorithm is also proposed to filter out reports coming from malicious users. The efficiency and effectiveness of the proposed scheme against LSAs is demonstrated through simulation study. Simulation results show that our scheme is resilient to high percentages of attackers (up to 40%) and scales well with the number of users. The viability of the approach is also demonstrated through (preliminary) real-world experiments performed at National Research Council (CNR), Pisa, Italy.

F. Restuccia and S. K. Das are with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, 65401 USA (e-mail: {frthf, sdas}@mst.edu).

A. Saracino and F. Martinelli are with the Istituto di Informatica e Telematica del Consiglio Nazionale delle Ricerche, Via G. Moruzzi n.1, 56124, Pisa, Italy (e-mail: name.surname@iit.cnr.it).

[1]Available on play.google.com

978-1-4799-8425-1/15/$31.00 ©2015 IEEE

The rest of the paper is organized as follows. Section II introduces the threat model. Section III describes in depth the proposed scheme, while Section IV presents experimental and simulation results of the scheme considering practical PS scenarios. Section V discusses the related work. Finally, Section VI draws conclusions with directions of future work.

II. Threat Model

We will assume that the communication between the users and the PS server leverages reliable and protected channel, where data cannot be lost, eavesdropped, modified or substituted. We also assume the PS server is totally reliable and trustworthy (root of trust), in particular, in terms of user registration, key management, issuing credentials, trust assessment and reputation management. Users are uniquely identified inside the network through an identifier which exploits a digest of the smartphone IMEI (International Mobile Equipment Identifier), which is unique for any device worldwide [9]. Thus, it is sound to assume that the system is protected from sybil attacks.

A user u_i performs a location-spoofing attack (LSA) when she declares to the PS server a location different from the real location. We define such users as *spoofers*. This attack can be caused both by malicious users, willing to appear in a location different from the real one, and also unintentionally by users that have a location privacy mechanism active on the smartphone and are not aware of the side effects on the PS system. The target of malicious users is dependent on the specific PS application considered. For example, malicious users could exploit LSA to obtain reward from incentivization systems [8] and/or acquire reputation inside the system. The LSA is said to be *solved* when the PS system is able to detect the spoofers and therefore exclude unreliable reports from the PS system.

Henceforth, we will focus our attention to solving the LSA only. In particular, attacks via the communication channels (e.g., eavesdropping, traffic jamming, etc.) are out of the scope of this paper.

III. Proposed location validation scheme

In this section we describe our approach to tackle the location-spoofing attack (LSA). We describe the system model and the WiFi-based location validation algorithm. Finally, we describe the reputation-based algorithm used to filter out unreliable reports.

A. System Model

Hereafter, we will suppose the smartphone sensing area is logically divided into W *location areas* of size $S \times S$, in which N users can move without restrictions (we do not assume any particular user mobility pattern and model). Specifically, users are free to move from one location area to another, and a given location area may contain any number of users (from 0 to N). However, users cannot be in two different location areas at the same time. We assume S and W are parameters depending on the specific PS application and its required accuracy of user location. Figure 1 illustrates the sensing area system model, where the users are depicted as black dots. We assume the location area L_k^t of user u_k at time t is identified by a pair of numerical coordinates representing a point in the two-dimensional Cartesian coordinate system $C \triangleq \{\mathcal{O}, \mathcal{X}, \mathcal{Y}\}$.

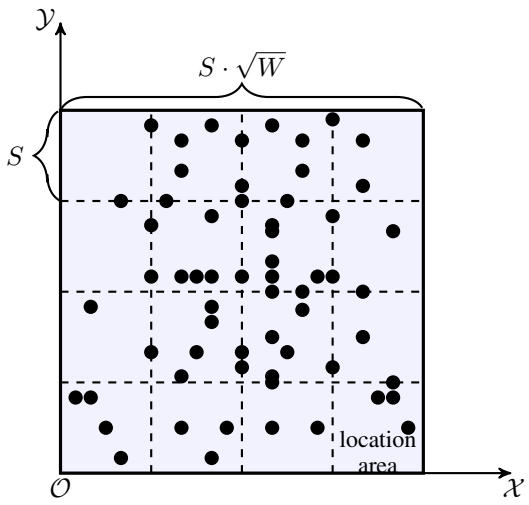


Fig. 1: Sensing area model.

Let us now formally define the Location Spoofing Attack (LSA). Let a PS system have N active users $U^j = \{u_1, \ldots, u_N\}$ at time t_j and W location areas $\mathcal{A} = \{A_1, ..., A_W\}$. A location-spoofing attack (LSA) is performed when one or more users u_s (called spoofers) belonging to the set $\mathcal{U}_s \subseteq U^j$ advertise to the PS server a position in a location area A_l^f (fake location area) while their real location is in the location area A_l^r (real location area), where $A_l^f \neq A_l^r$. The location in A_l^f is provided continuously by the spoofer. We assume that during the attack, spoofers can move from one location area to another, but the condition $A_l^f = A_l^r$ is never met.

B. Location Validation Algorithm

Mobile hot-spots (MHSs) are defined as the subset of selected users who activate the built-in WiFi hotspot feature of their smartphones and wait for other users to connect. Users that reside in the WiFi range of MHSs are called *neighbors*. The neighbors and the MHS mutually validate their locations and therefore, users who are spoofing their location are identified.

The location validation algorithm divides time into validation *rounds*, occurring every T_r time units; henceforth, we will refer to $t_j = j \cdot T_r$ as the time of the j-th validation round. During a validation round, the MHSs and their neighbors mutually validate their locations. A set of consecutive validation rounds is called validation *epoch* (Figure 2). The number of rounds composing a validation epoch and hence, its duration T_e, is variable and will be detailed later in the subsection.

Let N_i^j denote the number of users physically present in the i-th location area A_i at time t_j. Also, let $D_i^{t_j}$ define

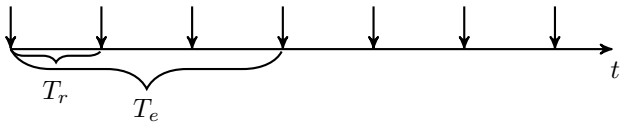


Fig. 2: Validation algorithm timeline.

the number of users advertising their position to be inside the location area A_i. During every validation round, the validation algorithm performs the following three steps.

S1. Each user transmits her current location position to the PS server. For each i-th location area A_i, the PS server selects a subset of users among $D_i^{t_j}$ users that appear to be in the i-th location area.

S2. Each of the selected users will act as MHS and validate the position of its neighbors through the WiFi connection. At the same time, the neighbors also validate the position of the MHS.

S3. Each user transmits the information acquired in the current validation round to the PS server.

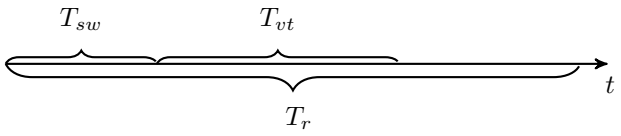
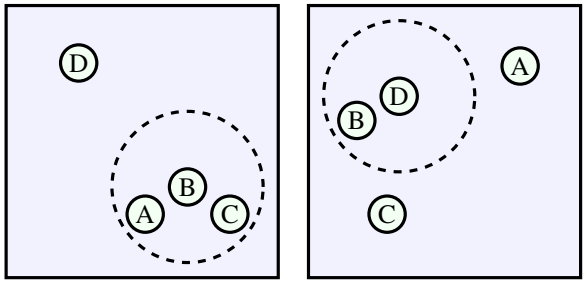


Fig. 3: Validation round timeline.



(a) (b)

Fig. 4: Position of users at rounds j and $j+1$.

More in detail, the operations performed by each user during each validation round (timeline in Figure 3) are summarized as follows.

- Each MHS turns on the WiFi hotspot capability, and after WiFi setup time T_{sw}, starts accepting connections from nearby users for a maximum validation time T_{vt}. After a user connects to an MHS they exchange their IDs for mutual validation.
- After the validation time T_{vt} elapses, each MHS turns off the WiFi connection (if not active before the validation phase).
- Each user reports to the PS server the IDs of the users verified in the current validation round (if any).

The selection of MHSs aims at maximizing the area coverage. The description of such an algorithm has been reported in [10] due to space limitations.

As an illustrative example, let us consider location area A_i containing four users A, B, C and D at the validation round j (Figure 4.a). During this round, the PS server chooses B to be MHS since she is close to users A and C. Users A and C are within the WiFi range of B, while D is in a different zone of A_i. Therefore, A validates the location of B and C, while both B and C validate the location of A. During round $j+1$ (Figure 4.b), D and B validate the location of each other.

As mentioned earlier, the PS server evaluates the reputation of each user once a validation epoch is finished. In particular, a *validation epoch* ends when the position of all users in a given location area has been validated by at least q users, where q is a system parameter. More formally, the duration of j-th validation epoch for location area A_i is defined as $\min(e_M, e_{max})$, where e_M is the number of validation rounds required to validate $M\%$ of the $D_i^{t_j}$ users by at least q users, and e_{max} is a system parameter.

C. Reputation Model

Let's now introduce the reputation model used by the system to rule out the reports submitted by users spoofing their location. The system assigns to each user u_i reputation value ρ_i^m, which is updated at the end of the m-th validation epoch. In particular, the reputation ρ_i^m of each user u_i is updated after the end of the m-th validation epoch according to the following relation, inspired to the Jøsang model [11]:

$$\rho_i^m = b_i^m - d_i^m - u_i^m$$

where $0 \leq \rho_i^m, b_i^m, d_i^m, u_i^m \leq 1$. In detail, b_i^m, d_i^m and u_i^m are respectively the *belief*, *disbelief* and *uncertainty* level associated to the reputation of user u_i after the m-th validation epoch. These three values are updated at the end of the $m-1$-th validation epoch according to Algorithm 1.

Algorithm 1 Updating ρ_l

$\rho_l = b_l - d_l - u_l$
$b_l + d_l + u_l = 1$
for all $u \in U^j$ **do**
  if u location is verified **then**
    $b_l = b_l + \Delta_b$
    $u_l = u_l - \frac{\Delta_b}{2}$
    $d_l = d_l - \frac{\Delta_b}{2}$
  else
    if u location is not verified **then**
      $u_l = u_l + \Delta_u$
      $b_l = b_l - \Delta_u$
    end if
  else
    if u location is fake $\wedge$ u is malicious **then**
      $d_l = d_l + \Delta_d$
      $b_l = b_l - \frac{\Delta_d}{2}$
      $u_l = u_l - \frac{\Delta_d}{2}$
    end if
  end if
end for

Let us now explain the algorithm in detail. By defining A_i^d as the location area advertised by user u_i, the location of user u_i is *verified* when at the end of a validation epoch her position has been validated by at least q users. The location of user u_i is *not verified* when, at the end of a validation epoch, less than q users have validated the position of u_i to be in the location area A_l^d. Finally, the location of u_i is considered *fake* when her position has been validated by q_e users in a location area $A_l^e \neq A_l^d$ and $q_e > q$.

Since the condition $b_l + d_l + u_l = 1$ must always hold, after each update the three components are normalized. We point out that Δ_b,Δ_d and Δ_u are configurable parameters of the framework and can be varied to best fit to different configurations with different values of T_r, e_{max}, user density and number of location areas.

IV. Results

In this section we present both the experimental and simulative evaluation of our scheme.

A. Experimental evaluation

The target of the experimental evaluation is to evaluate the viability of the WiFi-based approach of the system in real scenarios. This is done by measuring the amount of successful connections between two devices in normal usage conditions, for example, when users are moving in real environments, with physical obstacles which may impair the connection procedure.

The experiments have been performed using two `Nexus 4` running Android 4.4. Experiments have been performed with different configurations of users' speeds, distances and movement patterns. Specifically, in each experiment the two users (hereafter referred to as U1 and U2) perform the following operations. At the beginning of each experiment, both U1 and U2 have their WiFi interface off. As soon as the experiment starts, U1 becomes an MHS and activates the built-in WiFi hot spot feature, while U2 simply turns on the WiFi interface and attemps to connect to U1. After 30 seconds from the beginning of each experiment, U1 and U2 shut down their WiFi interfaces, ending the experiment. We developed a simple Android application which implements the authentication protocol of the scheme.

The experiments (results summarized in Table I) have been performed in a building of the National Research Council (CNR) in Pisa, Italy. The experimental setup is depicted in Figure 5, with the following configurations. Each test has been performed with the same conditions for 15 times.

Experiment 1 (E1). In this experiment, U1 moves on a linear pattern while U2 stands still. The two users are physically separated by a wall, as can be seen in Figure 5. The experiment has been performed with U1 moving at two different speeds, namely 6 km/h and 15 km/h, to evaluate the effectiveness of our approach with different walking speeds.

Experiment 2 (E2). Both U1 and U2 are moving on straight and parallel linear patterns but in opposite directions. As shown in Figure 5, this experiment is performed

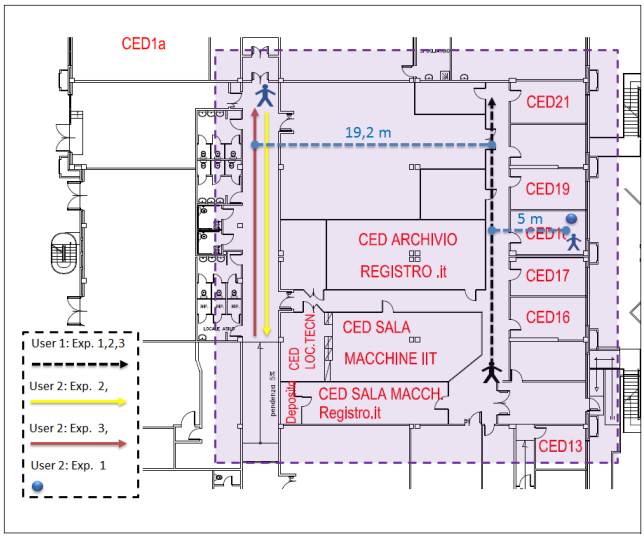


Fig. 5: Experimental setup.

Exp.	Distance	6 km/h	15 km/h
1	5 m	15/15	15/15
2	19.2 m	6/15	4/15
3	19.2 m	15/15	14/15

TABLE I: Details of Experiments.

with several obstacles between U1 and U2. The two users move on parallel trajectories which are 19,2 meters far. The presence of obstacles and the moving speed caused the authentication protocol to fail 9 times on 15 for the slow speed experiments, and 11 times on 15 for the fast speed experiments.

Experiment 3 (E3). Same configuration as E2, but with the users moving in the same direction. The experiment has been performed with the users moving at the same speed.

Table I concludes that if users are in the same room or in nearby rooms, it is almost guaranteed the mutual verification will be successful. However, walls and interference caused by other electronic devices may affect the mutual verification, as E2 shows. The improvement in E3 is due to the reduction of the variance of the perceived signal strength between U1 and U2.

B. Simulation Results

In this section, we evaluate through simulation experiments the performance of the system in terms of resilience from attackers and efficiency. To simulate a realistic environment, we modeled the sensing area as a single location area large 4 square kilometers (size of a small city or city block). As far as user mobility is concerned, we assumed users move about the location area following the Truncated Lévi Walk (TLW) mobility model [12], which has been shown to best represent the mobility of humans [13].

For the sake of simplicity, we modeled the WiFi range of the smartphones devices as circles centered on the user with radius 50 meters. As default system parameters, we chose as reputation parameters $\Delta_b = 0.25$, $\Delta_d = 0.6$, and $\Delta_u = 0.15$. The setup time T_{sw} has been set to 7 seconds

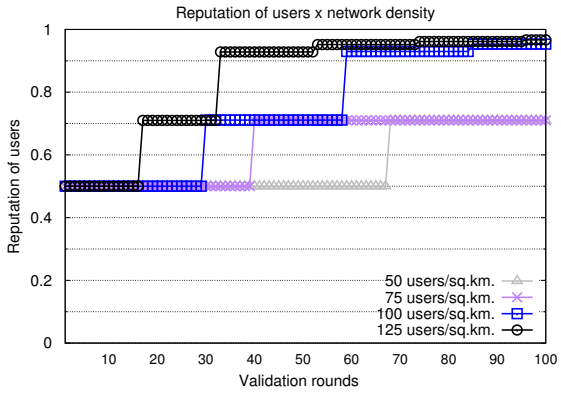


Fig. 6: Reputation of users (users density).

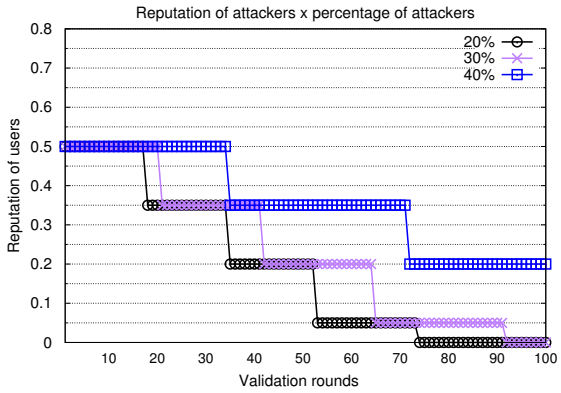


Fig. 7: Reputation of attackers (125 users/sq.km).

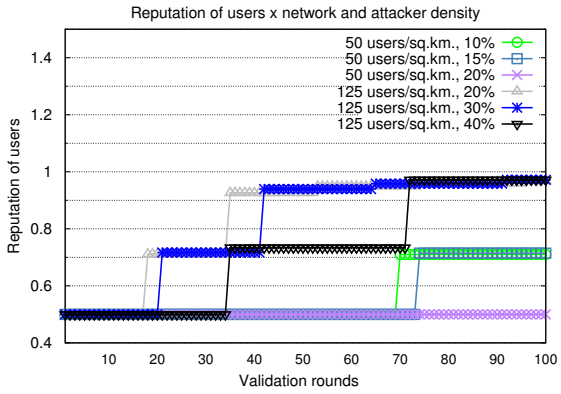


Fig. 8: Reputation of users (network and attacker density).

according to the experimental evaluation of Section II, while the validation round time T_{vr} has been set to 15s. The validation epoch threshold M and θ have been respectively set to 0.9 and 0.8, while the q parameter has been set to 2 in all experiments. The confidence intervals are set to 95%. For the sake of graphical clarity, the confidence intervals are not shown when less than 1% of the average. In the following, we will refer to as "users" the participants not faking their position, and to "attackers" as participants who fake their position and implement the LSA described in Section 2. For the sake of simplicity, and without losing in generality, we also assumed that users remain active inside the same location area for an entire validation epoch.

First, we evaluate the impact of the user density on the users' reputation and the efficiency of LVS. Specifically, Figure 6 shows the average reputation opinion of users as function of user density, supposing no attackers are present in the location area. As expected, from Figure 6 we observe that to greater user density corresponds faster increase of user reputation level over time, which is given by the faster termination of each validation epoch of LVS.

Users x sq.km.	% of MHSs	C.I.
50	14.5	2.51
75	17.33	3.12
100	21.75	3.56
125	24.25	4.11

TABLE II: Number of users selected as MHSs.

To further validate the scalability of the approach, Table II reports the percentage of users selected as MHSs in function of user density. Table II concludes that the percentage of users selected as MHS is less than the total number of users, even when the density becomes relatively high.

Let us now evaluate the resilience of the system to attackers with Figure 7, which shows the average reputation of all the attackers in function of the percentage of attackers in the system. Specifically, the attack has been simulated by setting the position of all attackers outside the location area, and by making them advertise a random position inside the location area to the PS platform. We recall that we do not consider in this analysis colluding attackers. As anticipated earlier, note that the system does **not** increase the reputation level of attackers in any circumstance, given the location of the attackers will never be validated by any MHS. Also, note that the reputation of attackers never reaches the θ threshold necessary to accept their reports inside the PS system. Therefore, we conclude the system is able to exclude unreliable reports from the PS system and therefore **protects** the PS system from the location-spoofing attack defined in Section II, without compromising the functionality of the PS application. We would like to point out that the security parameters of the reputation algorithm, as well as the validation round time T_{vt}, may be tuned by the PS application deployer according to the desired tradeoff between efficiency and security.

To gain further insights on the impact of the attackers on the reputation of users, Figure 8 show the reputation of users as function of the percentage of attackers and the users density. Figure 8 shows that when the density of users is relatively low (50 users/sq.km.), more validation epochs are needed to increase the reputation of users. Simply enough, this is due to the fact that in this case the users density on the location area becomes very low, and therefore the system takes additional time to validate the users' locations. However, Figure 8 also remarks that when the density of users is relatively high (125 users/sq.km.), our approach is able to tolerate a very high percentage of attackers (40%) without hindering the reputation of users. This is because in this case the system will still maintain

enough users to validate each user's location and therefore will tolerate a higher number of attackers.

V. Related Work

Over the years, several techniques have been proposed to estimate and verify the actual position of smartphone users. Starting from 2004, the main techniques used to identify the location of a mobile device have exploited both GPS and GSM (Global System for Mobile Communications) cell triangulation [14]. Alongside, approaches based on fixed, WiFi-based connectivity have also been proposed [15], [16], followed by techniques based on ambience-based fingerprints [17]. In particular, the latter exploit the sensors equipped on the devices to collect ambience elements such sounds or pictures, and then establish the user location based on the similarity of readings.

Given location-spoofing software like `Fake Locator` is able to hijack both GPS and GSM location services, approaches such the one presented in [14] are prone to the LSA and therefore not suitable to validate user location in PS systems. In addition, the user location obtained through GSM cell triangulation is known by the telephone service providers only, and may not be shared with external parties due to privacy issues. Conversely, the framework does not require any piece of information that cannot be retrieved on smartphones, which is essential for easy deployment. Existing WiFi-based solutions [15], [16] were specifically designed for indoor environment only, and are therefore not appliable to large-scale outdoor PS systems. Instead, the system leverages a technique that is valid for both indoor *and* outdoor PS systems. Although [15], [16] and similar solutions yield a greater accuracy than our system, we point out here that our approach is not aimed at calculating the precise location of users. Instead, the goal is to *verify* the user location provided by other localization services and thus solve LSAs. Finally, approaches based on ambient-based fingerprints [17] are not suitable in PS scenarios in which users are not able to observe the same phenomenon (e.g., users located in different floors/rooms of a building). The proposed framework, instead, is *independent* of the collected data type and relies only on WiFi to verify user position.

VI. Conclusions and Future Work

In this paper, we have proposed a location validation scheme which verifies user location in participatory sensing (PS) systems and solves the discussed location-spoofing attack (LSA), thus preventing loss of QoI in the PS system. We have proposed an approach which authenticates user location in a distributed and scalable way through the use of the mobile WiFi hotspot capability of modern smartphones. We have also proposed a reputation-based system based which rules out reports coming from users spoofing their location. Finally, we have tested the proposed approach with real experiments and we have shown its effectiveness against the LSA through simulations. Results conclude that LVS is applicable in almost every practical PS scenarios, and effectively solves LSA-based attacks. As future works we plan to strengthen the threat model also considering attacks with malicious users colluding. A more extensive testbed for real experiments, including outdoor environments, has also been planned as a future extension.

References

[1] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. *In: Workshop on World-Sensor-Web (WSW06): Mobile Device Centric Sensor Networks and Applications*, pages 117–134, 2006.

[2] P. Mohan, V.N. Padmanabhan, and R. Ramjee. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. *SenSys 2008 - Proceedings of 6th ACM Conference on Embedded Networked Sensor Systems*, pages 323–336, 2008.

[3] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson. Cooperative transit tracking using smart-phones. *SenSys 2010 - Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 85–98, 2010.

[4] 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. *MobiSys'09 - Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services*, pages 55–68, 2009.

[5] J. Carrapetta, N. Youdale, A. Chow, and V. Sivaraman. Haze watch project. online: http://www.pollution.ee.unsw.edu.au. 2010.

[6] E. Kanjo, J. Bacon, D. Roberts, and P. Landshoff. Mobsens: Making smart phones smarter. *Pervasive Computing, IEEE*, 8(4):50–57, 2009.

[7] 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. *SenSys 2008 - Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, pages 337–350, 2008.

[8] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, pages 173–184, 2012.

[9] How to write or change your imei on android, http://www.protechlover.com/2013/09/android-secret-codes-tips-and-tricks.html?showComment=1385454574906.

[10] Extended version available at https://www.dropbox.com/s/41702i14koo6tzs/percom2015.pdf.

[11] A. Josang. An algebra for assessing trust in certification chains. In *Proceedings of the Network and Distributed System Security Symposium*, pages 89–99, 1999.

[12] S. Hachem, A. Pathak, and V. Issarny. Probabilistic registration for large-scale mobile participatory sensing. In *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on*, pages 132–140, March 2013.

[13] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong. On the levy-walk nature of human mobility. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1597–1606, April 2008.

[14] F. Alcala, J. Beel, A. Frenkel, B. Gipp, J. Lülf, and H. Höpfner. Ubiloc: A system for locating mobile devices using mobile devices. In *Proceedings of 1st Workshop on Positioning, Navigation and Communication*, pages 43–48, 2004.

[15] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. Ieee, 2000.

[16] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye. Push the limit of wifi based localization for smartphones. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12, pages 305–316, New York, NY, USA, 2012. ACM.

[17] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 155–168, New York, NY, USA, 2011. ACM.