LVS: A WiFi-based System to Tackle Location Spoofing in Location-based Services

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

***Abstract*— The reliability of location-based services (LBS) is strongly dependent on the accuracy of the location of the users. However, existing LBS systems are not able to efficiently validate the position of users in large-scale outdoor environments, leading to possible location spoofing attacks by malicious users. To this end, we present an efficient and scalable *Location Validation System* (LVS) that secures LBS systems from location spoofing attacks. In particular, the user location is verified with the help of mobile WiFi hotspots (MHSs), who are users activating the WiFi hotspot capability of their smartphones and accept connections from nearby users, thereby validating their position inside the sensing area. The system also comprises a novel verification technique called *Chains of Sight*, which tackles collusion-based attacks effectively. LVS also includes a reputation-based algorithm that rules out sensing reports of location-spoofing users.**

***Index Terms*—Location-based Services, Smartphone sensing, Security, WiFi Hotspots, Location Spoofing.**

I. Introduction

Thanks to the availability of localization interfaces such as GPS, cell triangulation and Wi-Fi based localization, new generation mobile devices have become the main enablers of applications that use *location-based services* (LBS).

LBS applications (apps) provide a service that strongly depends from the current user position. Examples of extremely popular apps based on the LBS paradigm are *Foursquare* (`foursquare.com`) and *Waze* (`waze.com`). Foursquare is an LBS app where users share their experiences about visited place, stores and restaurants. Waze is a crowdsourced traffic monitoring service, which provides information about car traffic and driving times, exploiting user reports and monitored location. Users can also provide reports about traffic jams, speed traps, fuel stations' prices, etc. Users are rewarded with points for every driven mile and for provided reports. Finally, some LBS apps have users offering services to other users, where the best offer is chosen on the base of the provider location. An example is the *Uber* app, where registered car drivers offer rides to app users. The user chooses the driver and requests the service directly from the app, likely choosing the closest driver to reduce waiting time. A list of some popular LBS apps where a LSA gives unfair advantages to a user is shown in Table I.

Application	Type	LSA Effect
Foursquare	Location Review	Badges and points received without actually checking-in at locations.
Ingress	Social Game	Getting score and level unfairly claiming not visited locations (portals).
Shopkick	Store Review	Received rewards for reviews of stores not visited.
Uber	Car Trip Finder	Drivers can increase the odd of being called faking their location in points of interest (airport, train station, etc.)
Waze	Traffic Monitoring	Getting points through fake travels.

TABLE I
Examples of LBS apps.

It is clear that malicious users able to change their location programmatically (i.e. providing to the system a location different from the physical one), may take an unfair advantage of the rewards and generally of the specific LBS; such behavior is usually referred to as *Location Spoofing Attack* (LSA). Performing an LSA is all but difficult. In fact, smartphone applications (apps) like *LocationHolic* or *FakeLocation* [1] make extremely easy for users to spoof their current GPS location. LSAs are also extremely difficult to detect, given the PS system has no means to find out whether users are using apps such as *FakeLocator*. We point out that a successful LSA may not only bring an unfair revenue to the attacker, but also a *degradation* of the received service to the other users. Some examples are: **(i)** not objective reviews (Foursquare, Shopkick), **(ii)** not timely car trips (Uber), and **(iii)** false information on car traffic (Waze). This issue, will thus affect the reliability of the system causing a likely reduction of users, giving up for the unsatisfactory provided service.

The above discussions motivate our work and the following novel contributions. We propose a framework named *Location Validation System* (LVS), which efficiently and effectively tackles LSAs. LVS leverages the collaborative actions of users and the WiFi capability of smartphones to validate the position of other users. In fact, two smartphones directly connected through WiFi are practically sharing the same location, due to the limited WiFi range; thus, these two users can mutually validate their locations. The advantages of our approach will be discussed in details in Section II. A reputation-based algorithm is also proposed to filter out reports from malicious users.

The efficiency and effectiveness of LVS against location-spoofing attacks is proven through simulation experiments.

II. Location Validation System

In this section, we describe in detail the LVS framework. First, we describe the system model, then, we describe in details the algorithm used by LVS to select the users acting as mobile hot-spots (MHSs), as well as the WiFi-based location validation algorithm of LVS. Finally, we describe

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: {a.saracino, f.martinelli}@iit.cnr.it).

978-1-4799-7492-4/15/$31.00 ©2015 IEEE

the reputation-based algorithm used by LVS to filter out unreliable reports and therefore guarantee reliability of the LBS system.

A. System Model

In order to allow mathematical formalization of the problem, we assume the sensing area can be logically divided into W *location areas*, in which N users can move without restrictions. In particular, we do not assume any particular user mobility pattern and model. Thus, 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. An example of sensing area for the Uber app could be a metroplex, such as the Dallas-Forth Worth area, with the Location Area being squares of 1 sq. kilometer. In case of Waze, the location area could be as large as a city block. We assume that users are connected to the Internet through WiFi or 3/4G Internet connection.

Figure 1 reports a graphical description of the components of the LVS. Additional details on the architecture can be found in [2].

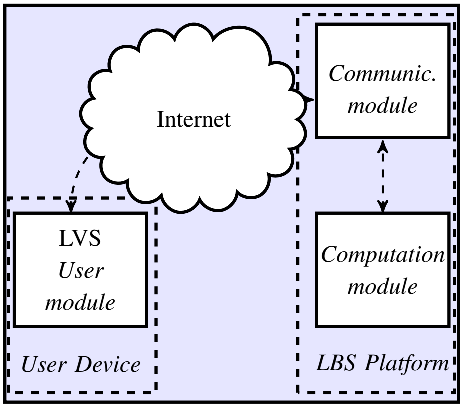

Fig. 1. Block diagram of LVS.

B. Location Validation Algorithm

The location validation algorithm divides time into validation *rounds* (Figure 2), 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*.

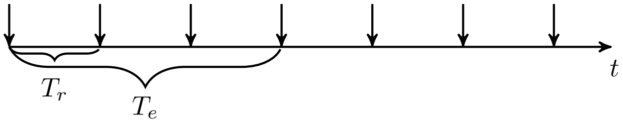

Fig. 2. Validation epoch timeline.

Let $N_i^{t_j}$ denote the number of users physically present in the i-th location area i at time t_j. Also, let $D_i^{t_j}$ define the number of users advertising their position to be inside the location area i at time t_j. During every validation round, the LVS validation algorithm performs the following steps.

S1. The user module transmits her current location $A_k^{t_j}$ to the LBS platform computation module through the LBS platform communication module. For each location area, the LBS platform selects a subset of users among $D_i^{t_j}$ users that appear to be in the i-th location area.

S2. The selected users receive a message request from the LBS platform to act as MHS and validate the position of their neighbors through WiFi connection. At the same time, the neighbors also validate the position of the MHS for additional security. This is when the location validation takes place, which we call *spotting* for brevity.

S3. Each user transmits the location validation information acquired in the current validation round to the LBS platform through the LVS user module. This information is used by the LBS platform computation module to compute users' reputation as detailed in III-A and III-B.

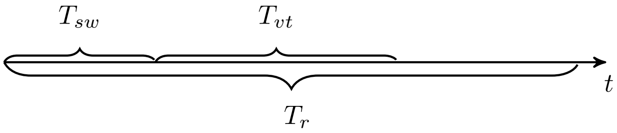

Fig. 3. Validation round timeline.

This operation of mutual validation between an MHS and a user in its WiFi range is also called *spotting*. If the user u_i is an MHS and the user u_j is in its WiFi area, we say that u_i *spots* u_j and u_j *spots* u_i.

III. Formal Analysis

After defining the main components of LVS, it is possible to extend the threat model by formally defining two attacks in addition to the LSA, arriving to a total of three attacks.

- *Location Spoofing Attack.* Let a LBS 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 LBS platform a position (*fake 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 into another, but the condition $A_l^f = A_l^r$ is never met.
- *Collusion Attack (CA).* The CA is performed when one or more sets of users $U_c = \{u_1, \ldots, u_c\}$ perform an LSA providing the location A_k when they are in location areas different from A_k, and at each validation round each user in U_c validate the fake position in

A_k of all other users in U_c. This attack can represent a situation in which a group of users is expected to be in a specific place, while they all are in different places. Thus, they collude mutually validating the fake position.

- *Fraud Covering Attack (FCA).* In the FCA, a user u_m performs an LSA providing a position in A_j, while located in A_k. At the same time, another user u_f really residing in A_j validates at each validation round the position of u_m in A_j. With this attack, a user can be located in a low density area, in order not to be spotted by other nodes residing in A_k and pretending to provide information on a different area.

A. Chains of Sights

Chains of Sight (CoS) represent the situation of a location area describing the series of direct and indirect spotting between users in a validation epoch. The CoSs have been designed to improve the performance of LVS and to effectively tackle the Collusion Attack (CA) and Fraud Covering Attack (FCA) described formerly. During each validation epoch, each user keeps track of the users spotted in the various rounds and shares this knowledge with the users spotted in the following rounds. As an example, if the user u_i spotted the user u_j in the round r, when at the round $r+1$ is spotted by the user u_k, u_i will tell to u_k about the presence of u_j in the area. Thus u_k indirectly spots u_j and also validates the u_j position. This information is expressed through a CoS in the following form: $u_j \rightarrow u_i/u_k$, read as "u_j sees u_k through u_i". It stems that through CoSs it is possible to reduce the number of validation round per epoch. A CoS has two main elements: the spotted node, which is the user identifier on the right end of the chain and the *length* which is the number of users in the chain. Notice that the chain length cannot be greater than ψ_{max}. At the end of each validation round the collection of CoS stored by a user u_i is defined as *user area knowledge* Ω_{u_i}, while the collection of all user area knowledges compose the *global area knowledge*.

Formally the CoSs are generated according to the following algorithm;

1) At each sensing round, the MHSs send their current area knowledge to the spotted users.
2) Spotted users send their current area knowledge only to their hot spot.
3) Each user including hot spots, update its area knowledge Ω_l with the useful information from the received area knowledge(s).

B. Reputation Algorithm

We now introduce a simple yet effective reputation algorithm used by LVS to rule out the reports submitted by users performing LSA, collusion attack, or fraud covering. LVS 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 reputation model:

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

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 [3]. These three values are updated at the end of the $m-1$-th validation epoch according to the following algorithm.

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. If it is verified, the belief is updated by a quantity Δ_b while disbelief and uncertainty are decreased by $\Delta_b/2$. 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. In this case, the uncertainty is updated by a quantity Δ_u while the belief is decreased by Δ_u. 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 $q_e > q$. In this case, the disbelief is updated by a quantity Δ_d while belief and uncertainty are decreased by $\Delta_d/2$.

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 LVS 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. Experimental Evaluation

In this section, we evaluate through simulation experiments the performance of LVS 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 kms (size of a 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 [4], which has been shown to best represent the human mobility [5]. Due to space limitations, we refer the reader to [5] for additional insights.

For the sake of simplicity, we modeled the WiFi range of the smartphones devices as circles centered on the user with radius 50 meters. The setup time T_{sw} has been set to 7 seconds, 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

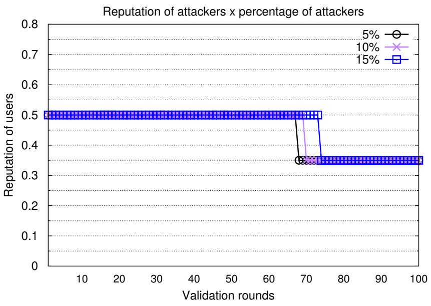

Fig. 4. Attacker Reputation (50 users/sq.km).

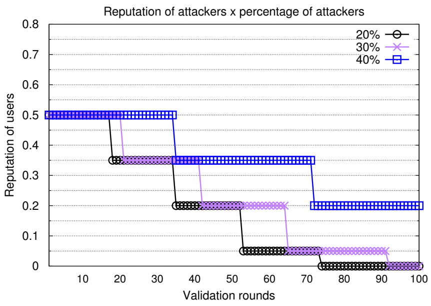

Fig. 5. Attackers Reputation (125 users/sq.km).

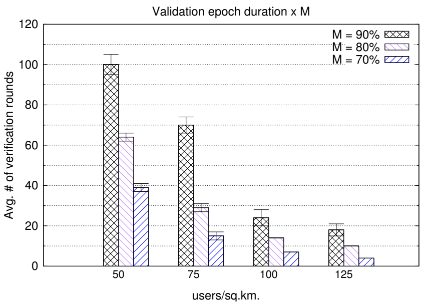

Fig. 6. Average time of validation epochs.

remain active inside the same location area for the whole validation epoch.

First, we evaluate the impact of the user density on the users' reputation and the efficiency of LVS. Figure 6 depicts the average duration, i.e. number of validation rounds, of the validation epochs of LVS as function of users density and the M parameter. Recall that in LVS, a validation epoch ends when M percent of users have their position validated by at least q users. As expected, the validation epoch duration decreases as M decreases and the users density increases.

1) Resilience from attackers: Let us now evaluate the resilience of LVS to attackers with Figures 4 and 5, which show 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 LBS platform. We recall that we do not consider in this analysis colluding attackers. Figures 4 and 5 conclude that the attackers' reputation decreases faster when the users density is higher, due to the shorter duration of validation rounds. Therefore, LVS is able to detect faster users not advertising their real position when the users density is higher. However, as anticipated earlier, note that LVS 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 LBS system. Therefore, we conclude LVS is able to exclude unreliable reports from the LBS system and therefore protects the LBS system from the location-spoofing attack, without compromising the functionality of the LBS app.

It results that when the density of users is relatively high (125 users/sq.km.), LVS is able to tolerate a very high percentage of attackers (40%) without hindering the reputation of users.

V. Related Work

Over the years, several techniques have been proposed to estimate and verify the actual position of smartphone users. Approaches based on fixed, WiFi-based connectivity have been proposed in [6], [7], followed by techniques based on ambience-based fingerprints [8], [9].

VI. Conclusions

In this paper, we have proposed LVS, a location validation system which verifies user location in Location Based Service systems. First, we have proposed LVS, which authenticates user location in a distributed and scalable way through the use of the mobilie WiFi hotspot capability of modern smartphones. Furthermore, we have introduced the formalism of the Chains of Sight, which are used to implement an algorithm to tackle collusion-based attacks.

References

[1] LocationHolic and FakeLocation. Available respectively on AppStore (iOS) and Google Play (Android) app markets.

[2] Francesco Restuccia, Andrea Saracino, Sajal K. Das, and Fabio Martinelli. Preserving qoi in participatory sensing by tackling location-spoofing through mobile wifi hotspots. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops 2015, St. Louis, MO, USA, March 23-27, 2015*, pages 81–86, 2015.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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*, pages 1–8, April 2013.