ISSN: 2321-7782 (Online)
Volume 4, Issue 4, April 2016
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Secure Verification using LBS covered by WIFI

Prachi P. Sadawarte[1]
Computer Science & Technology
Sant Gadge Baba University
Amravati – India

P. A. Tijare[2]
Computer Science & Technology
Sant Gadge Baba University
Amravati – India

Abstract: ***The location-based social networking services allow users to share their location information. The location-based services use the geographical position to enrich user experiences in a number of contexts, including location-based searching and location-based mobile based advertising. In an LBS system, the users check in at different venues at different point to acquire rewards such as virtual points or real-world coupons/discounts, and easily share with their friends or family the recent activities. As an example, we use foursquare to introduce a novel location cheating attack by attacker, which can easily pass the current location verification mechanism (e.g., cheater code of foursquare).***

While these rewards benefit benign users a lot, they are incentives for malicious users to cheat on their locations. For these purpose in this paper, we propose a novel verification system named WiLoVe, which employs the WiFi coverage to verify LBS check-ins. It maps the physical area of a venue to the local WiFi coverage and involves the venue owner as the verifier, hence utilizes the user's capability of one-hop communication with the verifier to verify the user's presence at the venue. A good location verification system must meet the many requirements like accuracy, cost efficiency, transparency and scalability.

Keywords: location-based services, check-ins, one-hop communication, verifier, scalability.

I. INTRODUCTION

Computer scientists are used to studying for access control mechanisms where one's identity shows what one is authorized to do. However, identity is not the only thing that matters in the physical world: often, the exact physical location of the requester also plays an important role in determining access rights. This suggests studying location-based access control. Location-based access control in the physical world is natural and familiar. For example, being able to switch on or the lights in a room it requires having a physical presence in the room. To en- force location-based access control policies on information resources, we need a way to perform location verification, where a principal's location is securely verified to meet certain criteria: e.g., being inside a specific building or a particular room.

Once a principal's location has been verified using a protocol for location verification, the principal can be granted to access a particular resource according to the desired policy. This approach is naturally combined with physical security; guards or locks that provide security which might be used to determine who is allowed to enter a building or room, then location verification employed to allow wireless access to all those inside. A recent report determines that 70% of smartphone users use their phones to get location-based information, and 30% use LBSes to check in to certain locations or share their locations with friends. LBSes not only bring convenience to the user's daily life, but also it provide real world benefits such as coupons and discounts. Unfortunately, these benefits such as discounts and coupons are also incentives for malicious users to abuse LBS and cheat on their locations. For example, in Foursquare, the most popular check-in app up to date, a user can easily check in to a venue (e.g., book stores, restaurants, shopping malls, etc.) to acquire the coupon without physically visiting on that place, or impress their friends by frequently. Check in to any desired place (e.g. gyms) while relaxing at home. These behaviors claiming a fake location is called *location cheating*. It goes against both the spirit of LBS systems and rules, and leads to unfairness in the

LBS community. Without a proper scheme to detect or prevent location cheating, the LBS system may be used other than the way they are supposed to. In practice, to protect against location cheating relies on the server side solution.

Foursquare adopts the "cheater code" to detect location cheating by examining if a user's movement (according to the user's check-in record) violates the human speed limitation. For example, it is not possible for a user to visit two places which are hundreds of miles apart within minutes. However, The "cheater code" has been proven to be insufficient. Moreover, it has been revealed that attackers can easily conduct bot check-ins to target locations and bypass the server's inspection. Therefore, a practical and efficient location verification system is in great demand to securely verify the user's real location and prevent bot check-ins from attackers.

II. Related Work

Many efforts have been made to realize practical and efficient location verification. However, most of them require additional hardware. For example, N. Sastry et al. apply ultrasonic techniques to decide the distance [6]; Jack Brassil et al. propose to authenticate the user's location with voice signatures, which needs additional femtocell to monitor the traffic variation during a voice call[7]; B. Carbunar and R. Potharaju[4] implement FES (Feed-back enabled embedded system) which requires extra LCD screen to show QR code for verification. Another part of researches can not deal with proxy attacks. For example, Wanying Luo and Urs Hengartner propose six design goals for location verification and demonstrate with cryptographic techniques, but none of the goals consider defending proxy attacks [8]. Some existing studies impose extra operations on the users, thus are less practical. K. Zhang et al. use challenge response to verify users [9].

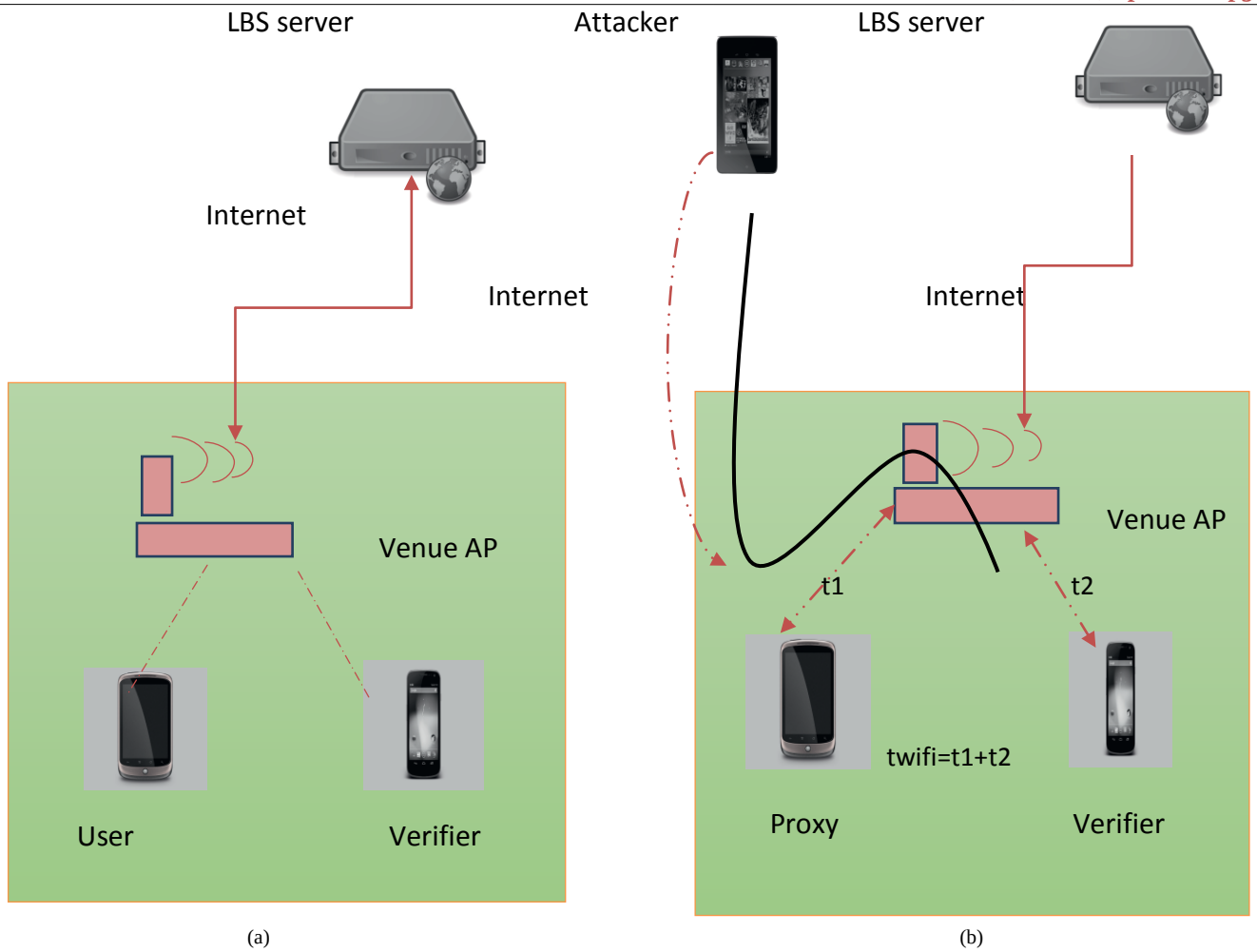
Verification systems based on the communication latency exist before this paper. B. Carbunar and R. Potharaju adopt similar idea as us by detecting abnormal communication delay [4]. In their application, the round-trip time is required to be highly precise for accurate ranging. In contrast, our goal is only to verify whether the user device is inside the WiFi coverage, which is much more loose [10]. Thus, our dynamic proxy detecting algorithm fits better to the problem. S. Capkun et al. also propose the idea of measuring round-trip latency [11]. However, their solution requires at least one additional bas station (AP in our context) which is invisible to the user. Such invisible AP can only be used for verification, thus will become a burden for the venue owners.

III. Design of WiLove

A. System Design

By involving the venue owner as a local verifier, the system architecture shows that it transforms from a client-server structure in existing LBS system to the trilateral one in WiLoVe (Fig. 1a). The user operates through a smart phone while the verifier can use either a computer connected to local WiFi access point (AP) or a mobile device. Built on top of the basic check-in flow in the existing LBS, WiLoVe's work flow involves the following modifications.

1. The user and the verifier each generate an asymmetric key-pair, all the communications are done with encrypted using these keys. The server acts as a trusted entity to distribute public keys between them. 2. The verifier's have WiFi IP address and SSID are included in the venue info and updated to the server. 3. The user can acquire the IP and SSID of the verifier from the server, and now it sends check-in requests to the verifier(venue owner) instead of the server; the verifier listens on a predefined port for the check-in requests. 4. The verifier verifies the user generated encrypted messages which are dynamically generated. 5. The time spent on the message decryption and communication during the verification is employed to detect proxy attack.



(a) (b)

Fig 1. Architecture of the verification system (a) Normal check-in scenario (b) Proxy attack scenario. *tWiFi* is the round trip delay in the local WiFi network. *t*1 and *t*2 indicate the round trip delay along the corresponding path respectively.

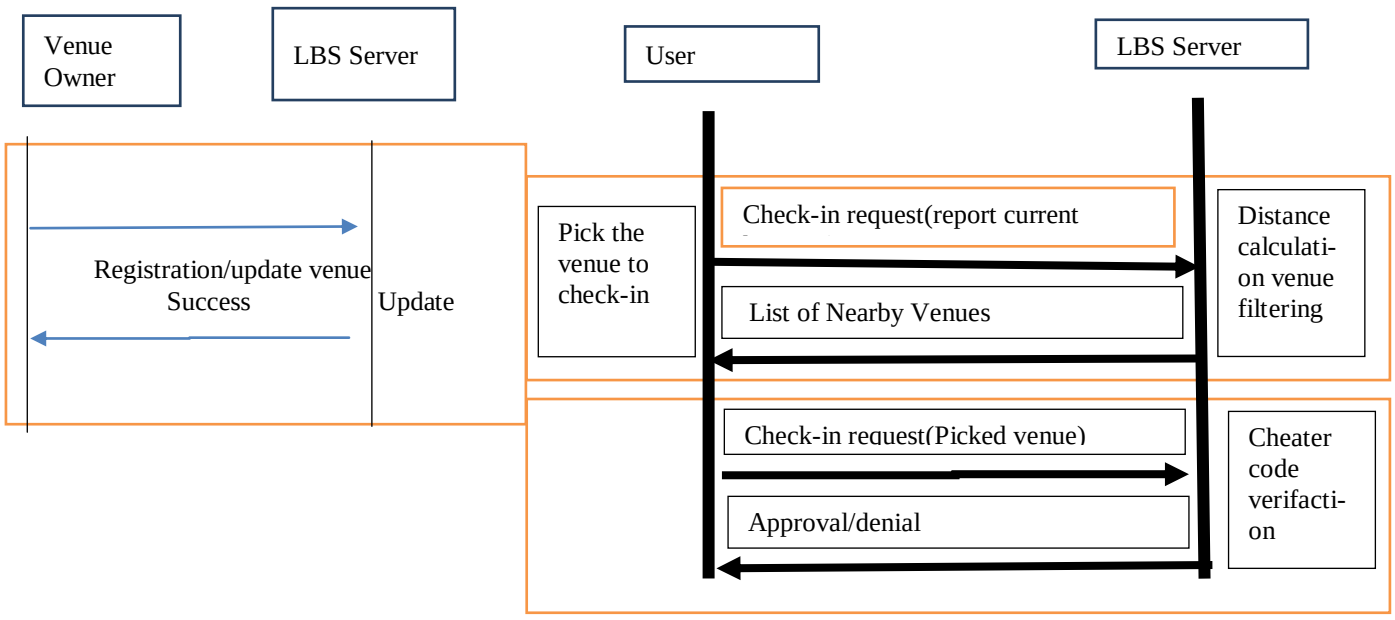


Fig 2. The check in process of a typical LBS system

B. Defence Strategies

Against Proxy Attack

In a proxy attack, a proxy device is placed inside the venue area to relay the communication between the verifier and attacker. In this case, the attacker can pretend to be in the coverage of the venue WiFi. One observation of the proxy attack is

that it inevitably brings in extra communication delay. WiLoVe employs such delay to distinguish the proxy attack and normal check-ins. For each check-in, the recorded verification delay is

$$\Delta t = Treceive - Tsend = tdecryption + tWiFi \quad (1)$$

Where *Tsend* and *Treceive* are the timestamps of the verifier sending encrypted message to the user and receiving corresponding decrypted message from the user, respectively. *tWiFi* is the round-trip delay in the local WiFi network; *tdecryption* is the time used by the user to decrypt the verification message. In the proxy attack scenario (Fig. 1b), the overall delay is

$$\Delta tproxy = tdecryption + tWiFi + tp$$

Here, *tp* is the round-trip time between the attacker and the proxy. In the work by B. Carbunar and R. Potharaju4, they claim that $\Delta tproxy$ is 12 times higher than Δt. In their case, *tdecryption* is at an ignorable level, thus *tp* should be about 11 times higher than *tWiFi*. However, our tests show that such distinct difference does not necessarily exist. For example, in one of our tests, the attacker is in the same city with the venue, and the average *tp* is only 1.25 times of *tWiFi*. Additionally, *tp* has a large variance, and in many cases the range of *tp* overlaps with that of *tWiFi*. Thus it is not easy to detect if a small *tp* is added to *tWiFi*. Based on this observation, we propose an adaptive proxy detecting algorithm which is simple yet effective. It is based on moving average, as follows:

1. After recording *Tsend* and *Treceive*, compute the delay Δt of current check-in using Eq.1.

2. Each time a new check-in's delay is recorded, it is compared to current upper bound *U*, calculated by

$$U = \mu + k * \sigma \quad (2)$$

Here, μ is the mean value and σ is the standard deviation of measurements in *History*; *k* is a predefined model parameter, which we will discuss later. If the delay of the newly recorded check-in is larger than *U*, the checkin is treated as a potential proxy attack, and the verifier app will generate and encrypt another random message, send it back and record the delay again. This is called a recheck-in. *MaxCount* sets a limit on the number of continuous recheck-ins allowed. If the recheck-in still fails even when the maximum limit is reached, this whole check-in session is treated as a proxy attack. If the delay of the newly recorded check-in/recheck-in is smaller than the upper bound *U*, it will be accepted.

3. A FIFO queue called *History* stores the delays of *len* recent check-ins. When a check-in session ends, even if the check-in is accepted, only the delay of the last recheck-in in this session is inserted into *History*. This is based on the following two reasons. First, the last recheck-in is the newest one, thus taking it will keep the freshness of the *History* queue. Second, for those rejected recheck-ins, if they are not attacks, then their high delays reflect the network fluctuation and should not be recorded; else if they are attacks, they should not be recorded either, otherwise they will pull up the average the *History*.

The proxy detecting algorithm is adaptive to the dynamic changes of the network delay for two reasons. First, *History* is a FIFO queue that updates over time, thus the mean value μ in Eq. 2 is actually computing the moving average, which dynamically reflects the trend of network delay. Second, the standard deviation σ plugs the latest network delay information into the upper bound, thus makes it tolerant of short-term fluctuation. In Section 3, we will show how the parameters, i.e. the length *len* of *History*, the maximal allowed recheck-in number *MaxCount* and the bound factor *k*, will affect the system performance against proxy attacks.

IV. Conclusion

In this paper, we analyzed the solutions currently available intensive research and development in mobile based application and services, location of users and their interaction has evolved drastically. Wireless communications has given rise to information services that can predict, identify and adapt to the location of the mobile user. We use Wifi coverage area for

WiLoVe to verify LBS check-ins. Every time venue owner act as the verifier , which is implemented on his WiFi connected device, such as laptop or smart phone or computer devices.

When one of the user wants to share the information to their friends, this LBS system is very helpful. LBSes not only bring convenience to the user's daily life, but also provide real world benefits such as coupons and discounts. Unfortunately, these benefits are also incentives for malicious users to abuse LBS and cheat on their locations. Foursquare adopts the "cheater code" to detect location cheating by examining if a user's movement (according to the user's check-in record) violates the human speed limitation. For example, it is impossible for a user to visit two places which are thousands of miles apart within minutes. WiLoVe has its deficiency. It can only be applied to venues with WiFi. Thus it will act more like an optional security-enhanced strategy in WiFi enabled venues. Furthermore, they have strong needs of check-in security. How to seamlessly integrate WiLoVe with the existing check-in system and make them work accordingly to different needs remain as our future work.

ACKNOWLEDGEMENT

It is me great pleasures on bringing out paper entitled "Secure verification using LBS covered by WIFI". The author would like to thank the anonymous referees for their constructive comments.

References

1. Zickuhr,K..Three-quarters of smart phone owners use location-based services.2012URL:http://pewinternet.org /Reports/2012/Location-baseservices.aspx.com.
2. Foursquare,Foursquare.2013.URL:https://play.google.com/store/apps/details?id=com.joelapenna.foursquared.
3. He, W., Liu, X., Ren, M.. Location cheating: A security challenge to location-based social network services. In: Distributed Computing Systems (ICDCS), 2011 31st International Conference on. IEEE; 2011, p. 740–749..
4. Carbunar, B., Potharaju, R.. You unlocked the mt. everest badge on foursquare! countering location fraud in geosocial networks. In:Mobile Adhoc and Sensor Systems (MASS), 2012 IEEE 9th International Conference on. IEEE; 2012, p. 182–190.
5. FoursquareSuggestions for shopping in new york city. 2013. URL:https://foursquare.com/explore?cat=shops&near=New20York%2C%20NY%2C%20US
6. Sastry, N., Shankar, U., Wagner, D.. Secure verification of location claims.In: Proceedings of the 2nd ACM workshop on Wireless security.ACM; 2003,p1-10.
7. Brassil, J., Netravali, R., Haber, S., Manadhata, P., Rao, P.. Authenticating a mobile device's location using voice signatures. In: Wireless and Mobile Computing, Networking and Communications (WiMob), 2012 IEEE 8th International Conference on. IEEE; 2012,p. 458–465.
8. Luo, W., Hengartner, U.. Proving your location without giving up your privacy. In: Proceedings of the Eleventh Workshop on Mobile Computing Systems and Applications. ACM; 2010, p. 7–12.
9. Zhang, K., Pelechrinis, K., Krishnamurthy, P.. Detecting fake check-ins in location-based social networks through honeypot venues 2013;.
10. Waters, B., Felten, E.. Secure, private proofs of location. Department of Computer Science, Princeton University, Tech Rep TR-667- 032003;.
11. Capkun, S., Bonne Rasmussen, K., Cagalj, M., Srivastava, M.. Secure location verification with hidden and mobile base stations. Mobile Computing, IEEE Transactions on 2008;7(4):470–483.

AUTHOR(S) PROFILE

Ms. Prachi P. Sadawarte, received the B.E. degree in Information Technology from the P.R. Patil college of institute, Amravati of Sant Gadge Baba university, Amravati in 2015; the M.E degree in Computer Engineering pursuing in Sipna college of engineering in Sant Gadge Baba University, Amravati. Her Interested is in Location verification using various Applications.

