Witness-based Location Proofs for Mobile Devices

João Ferreira, Miguel L. Pardal
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal
{joao.ricardo.pais.ferreira,miguel.pardal}@tecnico.ulisboa.pt

***Abstract*—Location-aware mobile applications are gaining popularity. This growth has caused the emergence of services that are offered to the users only when they are at specific locations. To implement valuable services, like a product sale, it is necessary to verify the presence of the user's device in a way which can be reliably trusted by the providers.**

This paper presents a system to support the creation of proofs that the user's device is at a claimed location. The system relies on different techniques for location estimation and on witness devices to testify to the presence of the user's device.

A prototype was implemented and evaluated in regard to response times, accuracy of location estimates, and feasibility of proof exchanges. The results show that the solution is both practical and useful.

***Index Terms*—Mobile Security, Context-Awareness, Location Estimation, Location Proof, Internet of Things**

I. Introduction

The use of multiple sensors and actuators, embedded in the environment and connected to computers, enables *context-aware* systems that gather information about the real world for use in the Internet of Things (IoT) [1]. The smartphone is the device that can better take advantage of this environment because of its numerous sensors, Internet connection and close bound with a human user. *Location* is one of the most used types of context [2] in smartphone applications.

This paper presents a location proof system for mobile devices that uses multiple location estimation techniques and relies on witnesses to testify the presence of a user at a given place [3], [4]. A prototype was implemented and evaluated.

II. Related Work

Regarding *location estimation*, there are alternative ways for a user to measure location. GPS [5] is highly available but does not work indoors. Other solutions use wireless networks based on cell towers [6] or Wi-Fi access points [7], [8]. The Android Network Location Provider (ANLP)[1] uses both cell tower and Wi-Fi to determine the location. This method responds faster and uses less battery than GPS and achieves more precise results in areas with more Wi-Fi access points. Bluetooth location systems are based on *beacons* spread over the area of interest [9]. Since range is limited (below 10 meters), it is assumed that, if a user can detect a beacon, then she is near the location. To cover a large area, many beacons are needed which imposes high hardware and installation costs.

Proximity systems [10] can verify, for example, maximum communication latency to assert proximity. However, they are vulnerable to relay and signal amplification attacks.

Regarding *location proofing*, the main concept is the location proof [11]. A proof states that a user is at a given place, at a given time. It contains the following attributes: prover identifier and location, witness identifier and location, random number and/or timestamp to ensure freshness, and signature to assure authenticity of data. The APPLAUS system [12] works without depending on any infrastructure from the place where a proof is generated. Instead, it uses a peer-to-peer approach between devices. Each device acts as a witness. To prevent abuses, there is also a collusion detection mechanism. The CREPUSCOLO system [13] also uses a peer-to-peer but it adds infrastructure, namely, token providers that act like trusted witnesses, fixed to a specific location.

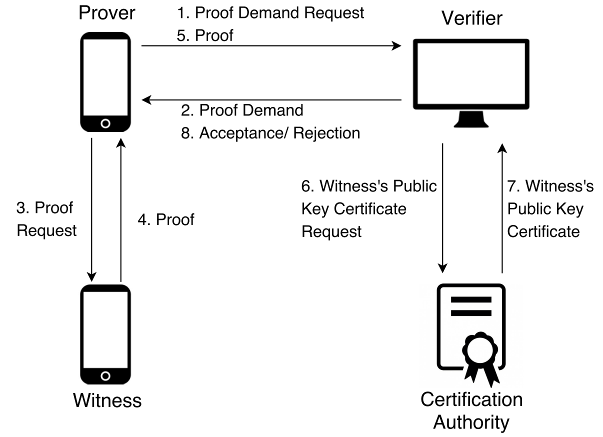

Fig. 1: Communication between entities.

III. Solution

In our system design we want to take advantage of the diversity of user presence in the physical space and use it as a security information source. We assume that participating users are willing to share some device network and power resources and their personal identification and location. There are four roles played by devices in our system, similar to the ones defined in APPLAUS [12]: the *prover* that needs to prove its location; the *witness* that agrees to give a location proof; the *verifier* that specifies techniques to use and later validates the proof; and the *certification authority* that is trustful and binds identities and public keys. Figure 1 presents how the entities communicate when a location proof is requested.

[1] Android Network Location Provider (ANLP): https://developer.android.com/guide/topics/location/strategies.html

978-1-5386-7659-2/18/$31.00 ©2018 IEEE

	Additional Proof Fields	*Setup*
Geo Proof	Geographic Location (obtained from GPS or ANLP)	Collect geographic coordinates with associated radius
Wi-Fi Proof	Wi-Fi Fingerprint	Collect Wi-Fi fingerprints for each zone
Beacon Proof	Closest beacon ID detected	Associate beacon values with their corresponding place

TABLE I: Proof Techniques Fields and Setup Phase.

A. Location Proof Techniques

The location proof techniques are presented in Table I. Each one requires a *setup* stage to be performed before location proofs can be made. After a proof is issued, it has to be verified. The *verifier* starts by checking the signature and its freshness. Table II presents the tasks required to complete the verification for each kind of proof.

B. Implementation

We implemented a prototype for mobile devices running Android OS, namely, Huawei P9 Lite devices. Bluetooth was chosen for the short-range communication, without pairing to allow ad hoc interactions. Security is added only in the application layer. Both the *verifier* and *certification authority* are implemented as RESTful web services, written in Java 1.8, with JSON payloads.

C. Collusion Avoidance Mechanisms

To prevent abuse, there is *witness redundancy* and *decay*. In the *redundancy* protection, proofs have to be gathered from multiple witnesses instead of only one. In the *decay* mechanism, proofs from a same witness gradually lose their value.

$$V_{\text{xy}} = \begin{cases} V & if\ \text{N}_{\text{xy}} = 0 \\ V - \frac{N_{\text{xy}}{}^{k}}{U} & if\ \text{N}_{\text{xy}} > 1 \end{cases} \tag{1}$$

The proof value is calculated with Equation 1 where V_{xy} represents the proof value given to user x by witness y and V represents the maximum proof value. N_{xy} represents the number of times that y testified the presence of x and U is the total number of users in the place.

IV. Evaluation

A. Location estimation accuracy

As a *scenario* for evaluation, we used a real building standing in as *shopping center* where a loyalty application is deployed to reward frequent visitors. Figure 2 presents the map of the location. Figure 3 represents the areas that were defined for use in Geo proofs, one for each zone. We tested the location estimation accuracy for each technique.

1) Geo and Wi-Fi: ANLP was used to identify in which area from Figure 2 the user is in. The results are presented in Figure 4. For the majority of places, Wi-Fi fingerprinting with 10 readings was the best option. Geo has, in most cases, lower accuracy than Wi-Fi fingerprint with 5 and 10 readings.

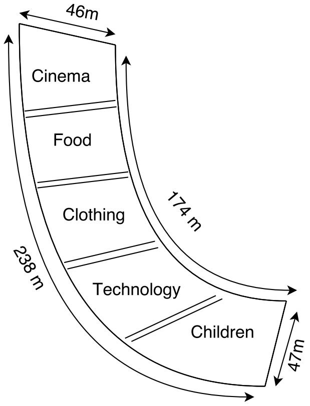

Fig. 2: Shopping center.

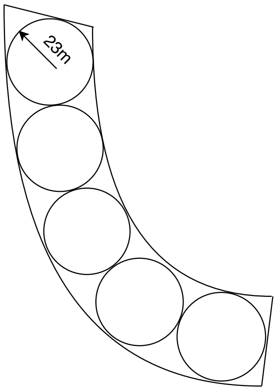

Fig. 3: Geo proof setup.

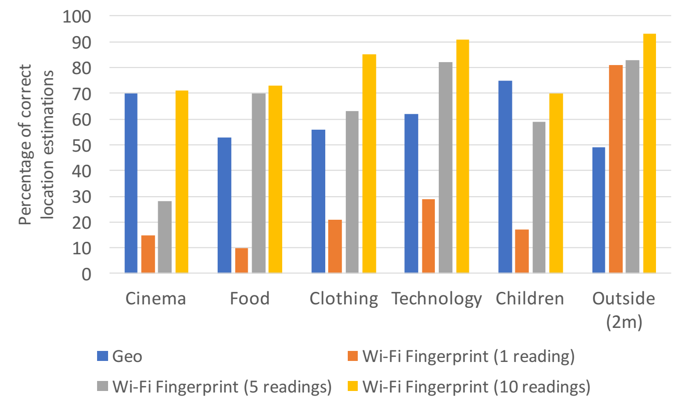

Fig. 4: Location accuracy for Geo and Wi-Fi.

2) Beacon: To obtain the location estimation accuracy with *Estimote* beacons, we prepared the setup shown in Figure 5. Tests were done with the user standing at 1 meter. Table III summarizes the results. The beacons provide a correct location 80% of the times. The middle beacon (*Vegan restaurant*) has the lowest accuracy.

B. Proof time

Figure 6 represents the total time for proof processing. Geo was the fastest, however, the mean value is far from the median value, as half of the measurements were done below 1.61 seconds. The Wi-Fi technique is the one with the most regular readings regarding time spent, as mean and median values were practically the same. Beacon proofs can sometimes take

	Verifier tasks performed upon receiving proof
Geo Proof	1. Verify if geographic locations of the prover and witness are close to each other (threshold is configurable) 2. Check if the location of the prover and witness are inside any of the previously saved areas. 3. If both are inside the same area, accepts the proof. Otherwise, rejects it.
Wi-Fi Proof	1. Retriev places with closest fingerprint to the one given by the prover and witness. 2. Compare places. If they are different, rejects proof. 3. Verify if fingerprints of the prover and witness have a minimum amount of access points in common with the saved fingerprint for that place. If they have, the proof is accepted. Otherwise, it is rejected.
Beacon Proof	1. Compare beacon values provided by the prover and witness. 2. If they are the same and there is a place associated with that beacon, accepts proof. Otherwise, rejects it.

TABLE II: Verifier tasks for each proof technique.

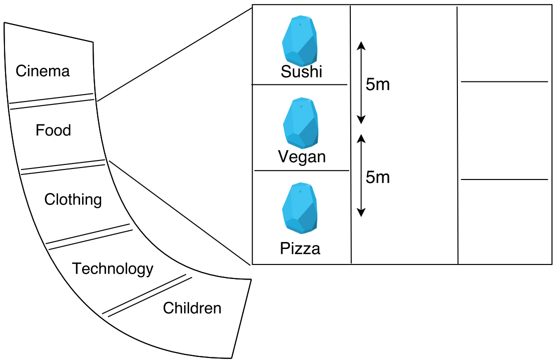

Fig. 5: Beacon proof setup.

	Sushi	Vegan	Pizza
Correct Claims	85%	72%	81%
Wrong Claims	15%	28%	19%

TABLE III: Location estimation accuracy with beacons.

longer to connect or to discover the closest beacon, which increases the time needed to obtain the location.

Figure 7 demonstrates the total time, from proof request until rejection or acceptance. Most of the time is spent on obtaining location data and this is what differentiates the total time between techniques. For example, in the Wi-Fi fingerprint, 10.76ms are spent in obtaining location information (5.38ms in each device). Each technique spends approximately between 4.5 and 5 seconds in processes other than obtaining location data.

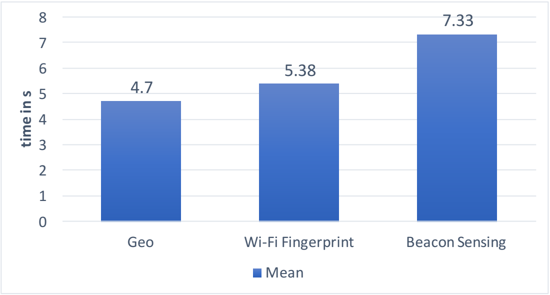

Fig. 6: Time required to get location data.

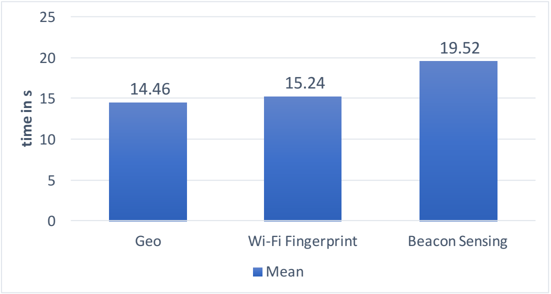

Fig. 7: Time spent from requesting proof demand until the verifier informs the prover about the acceptance.

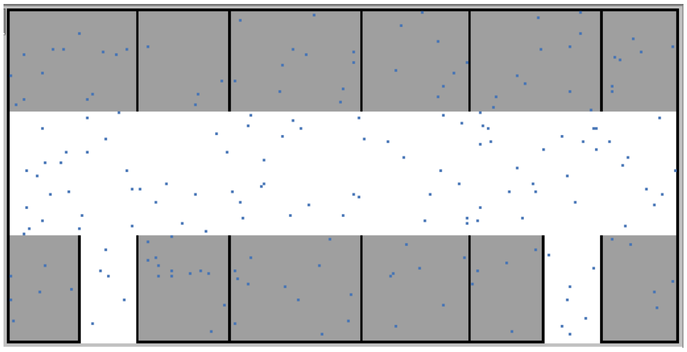

Fig. 8: Simulated shopping center in *Netlogo*.

C. *Collusion Avoidance Simulation*

To assess the impact of the collusion avoidance mechanisms, we simulated the system using *Netlogo*[2], a multi-agent programmable simulating environment.

1) Setup: The shopping center is represented as a grid, shown in Figure 8. Grey areas represent the stores, white areas represent the corridors, and the blue dots are the users.

Users move randomly around the shopping center. At each tick of the simulation, each user has a 1% probability of requesting a proof. The witnesses can be at a maximum distance of 10 cells from the requesting user. After 10 ticks, the witnesses that remained near the prover will respond to him. This simulates the time that prover and witness take when exchanging proof data via Bluetooth.

[2] Netlogo: https://ccl.northwestern.edu/netlogo/

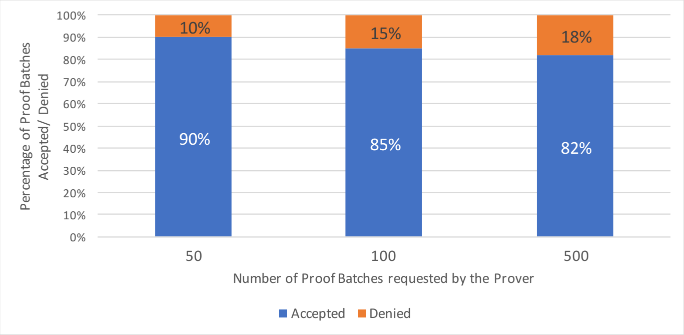

Fig. 9: Acceptance and denial rate of the proof batches. A proof batch is a group of proofs provided by the witnesses that the prover contacted.

2) Results: The number of cases where no witnesses are found remains near to 20%, as shown in Figure 9.

We also analyzed the average number of witnesses found per request. With our described setup, with 250 users, there was an average of only 2 witnesses found. If the population becomes 1000 users, the number of witnesses increases to 7.

If a malicious user wants to deceive the system, he will have to collude with false witnesses. If the verifier asks for *witness redundancy*, the user will have to gather proofs from N dishonest witnesses. As we have shown before, in a setup with 250 users, finding just 2 witnesses may not be too difficult for the malicious user. The *witness decay* mechanism, defined by Equation 1, enforces that the dishonest prover cannot reuse the same witnesses too often.

The decay of a proof value given by the same witness to the same prover will eventually deny access to the service. Our proposal is able to prevent collusion by taking advantage of the diversity of witnesses. The location proof system can also adapt to the number of users in the location and take advantage of it. It uses the total number of users in the space to calculate the probability of a prover encountering the same witness and adjusts the proof decay accordingly.

V. Conclusion

In this paper we presented a location proof system for mobile devices, that allows users to prove their location. The system allows the use of geographical coordinates, Wi-Fi fingerprinting and Bluetooth beacons. The evaluation with performance measurements and a simulation scenario, demonstrates that our proposal can be useful and adaptable. The solution is most effective for crowded locations, where a user can obtain location proofs with a diversity of witnesses. As future work, we will address the privacy protection of users.

Acknowledgements

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013 (INESC-ID) and through project with reference PTDC/CCI-COM/31440/2017 (SureThing).

References

[1] D. Uckelmann, M. Harrison, and F. Michahelles, “An architectural approach towards the future internet of things,” in *Architecting the internet of things*. Springer, 2011, pp. 1–24.

[2] M. Baldauf, S. Dustdar, and F. Rosenberg, “A survey on context-aware systems,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.

[3] R. Khan, S. Zawoad, M. M. Haque, and R. Hasan, “‘who, when, and where?’location proof assertion for mobile devices,” in *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 2014, pp. 146–162.

[4] X. Wang, A. Pande, J. Zhu, and P. Mohapatra, “Stamp: enabling privacy-preserving location proofs for mobile users,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3276–3289, 2016.

[5] H. Koyuncu and S. H. Yang, “A survey of indoor positioning and object locating systems,” *IJCSNS International Journal of Computer Science and Network Security*, vol. 10, no. 5, pp. 121–128, 2010.

[6] M. Ibrahim and M. Youssef, “Cellsense: A probabilistic rssi-based gsm positioning system,” in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–5.

[7] R. Hansen and B. Thomsen, “Efficient and accurate wlan positioning with weighted graphs,” in *International Conference on Mobile Lightweight Wireless Systems*. Springer, 2009, pp. 372–386.

[8] L. Chen, B. Li, K. Zhao, C. Rizos, and Z. Zheng, “An improved algorithm to generate a wi-fi fingerprint database for indoor positioning,” *Sensors*, vol. 13, no. 8, pp. 11 085–11 096, 2013.

[9] G. Anastasi, R. Bandelloni, M. Conti, F. Delmastro, E. Gregori, and G. Mainetto, “Experimenting an indoor bluetooth-based positioning service,” in *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*. IEEE, 2003, pp. 480–483.

[10] A. Ranganathan and S. Capkun, “Are we really close? verifying proximity in wireless systems,” *IEEE Security Privacy*, vol. 15, no. 3, pp. 52–58, 2017.

[11] S. Saroiu and A. Wolman, “Enabling new mobile applications with location proofs,” in *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*. ACM, 2009, p. 3.

[12] Z. Zhu and G. Cao, “Applaus: A privacy-preserving location proof updating system for location-based services,” in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1889–1897.

[13] E. S. Canlar, M. Conti, B. Crispo, and R. Di Pietro, “Crepuscolo: A collusion resistant privacy preserving location verification system,” in *Risks and Security of Internet and Systems (CRiSIS), 2013 International Conference on*. IEEE, 2013, pp. 1–9.