CROSS City: Wi-Fi Location Proofs for Smart Tourism

Gabriel A. Maia[1 [0000−0002−8691−6110]], Rui L. Claro[1 [0000−0003−0176−2720]], and Miguel L. Pardal[1 [0000−0003−2872−7300]]

INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal
{gabriel.maia,rui.claro,miguel.pardal}@tecnico.ulisboa.pt

Abstract. The ubiquitousness of smartphones, wearables and other mobile devices, coupled with the increasing number of communications infrastructure present in smart cities, has led to the rise of location-based services. Many of these services do not verify the location information they consume and are vulnerable to spoofing attacks. Location proof systems aim to solve this by allowing devices to interact with location-specific resources and later prove that they were at the location.
In this paper we describe and evaluate CROSS, a system that performs location verification using techniques compatible with off-the-shelf Android smartphones. We present three strategies to produce location proofs with increasing tamper-resistance. We designed our system with user privacy and security in mind, minimizing the number of connections between devices. We implemented a prototype application to assess the feasibility and reliability of the proof strategies. The application allows rewarding users who complete a touristic route with proofs of visit collected along the way. Our evaluation, which included experiments with 30 users, showed that we can use the system in real-world scenarios, providing adequate security guarantees for the use case.

Keywords: Location Spoofing Prevention · Location Proof · Context-Awareness · Security · Internet of Things.

1 Introduction

Location is one of the most important pieces of contextual information for Smartphone applications, and is at the core of Location-Based Services (LBS) [3]. These services typically do not verify the location information they use, and are susceptible to *location spoofing attacks*. Developing the means to certify location information is, therefore, of high importance. *Location proof systems* counter location spoofing by providing verifiable location information. One of the use cases for location proofs is in *Smart Tourism* [10]. Tourists can interact with existing or newly-added infrastructure in emblematic city locations, using their personal devices, and record information that can later be used to verify location information.

Wi-Fi can be used as infrastructure for location because most urban environments in modern cities, or other densely populated areas, contain many Wi-Fi

networks. The overwhelming majority of these announce their presence and can be detected using commodity smartphones.

In this paper we describe and evaluate CROSS (loCation pROof techniqueS for consumer mobile applicationS) with an example application, to ascertain whether the user completed any tourism circuits from a predefined set of routes, as represented in Figure 1. This paper extends a presentation and security assessment made earlier [11], and adds experimental results and their discussion.

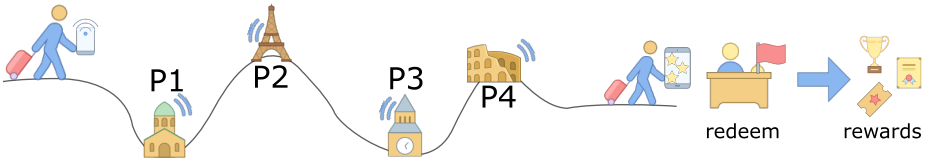

Fig. 1: User flow throughout a tourism route with four locations.

The paper contains the following sections: Section 2 presents a brief overview of existing works in the field of location proofs; Section 3 gives an overview of our system and its operation; Section 4 presents location proof strategies; Section 5 discusses experimental results using the prototype implementation; and Section 6 concludes the article.

2 Related work

Wi-Fi technology is widely used in mobile location systems, usually to complement GNSS (Global Navigation Satellite Systems), such as GPS, Galileo or BeiDou. Wi-Fi is also used for microlocation, in systems such as Google Indoor [9]. SAIL [12] is an example of a microlocation system which works by combining the Time-of-Flight of Wi-Fi packets with motion sensor data. SurroundSense [2] uses fingerprinting techniques encompassing Wi-Fi, motion sensors, microphones and cameras, to identify the location of the user. Witness-based systems such as APPLAUS [14], LINK [13] and SureThing [8] typically use peer-to-peer communication between witnesses. However, this type of communication is increasingly hampered by mobile operating systems, like iOS and Android, for security reasons. On the other hand, web server communication is usually not restricted. Systems which rely solely on mobile witnesses, without fixed infrastructure, require a minimum amount of diverse users at each location to work. The CREPUSCOLO [4] system solves this problem by introducing trusted witnesses that are installed on specific locations.

User privacy is a primary concern when dealing with exact and certifiable location information. Icelus [1] is a system that locates users and models their movement through IoT devices and uses homomorphic encryption for processing data on third-party servers, that can process but not learn the location of the users.

3 System overview

The main components of CROSS are represented in Figure 2: the client application, the server, the Wi-Fi Access Point (for proof strategy described in 4.2), and the Kiosk (for proof strategy described in 4.3). The system uses a client-server model with no peer-to-peer communication between clients.

The system operation starts when the tourist installs the smartphone application and signs up for an account. Before starting the trip, the application downloads the catalog of locations. During its use, the application logs visits to locations. The location sensing relies on Wi-Fi exclusively and leverages the scans regularly already performed by the mobile operating system. At the end of the trip, the logging stops, the application submits the collected information to the server, and rewards will be issued.

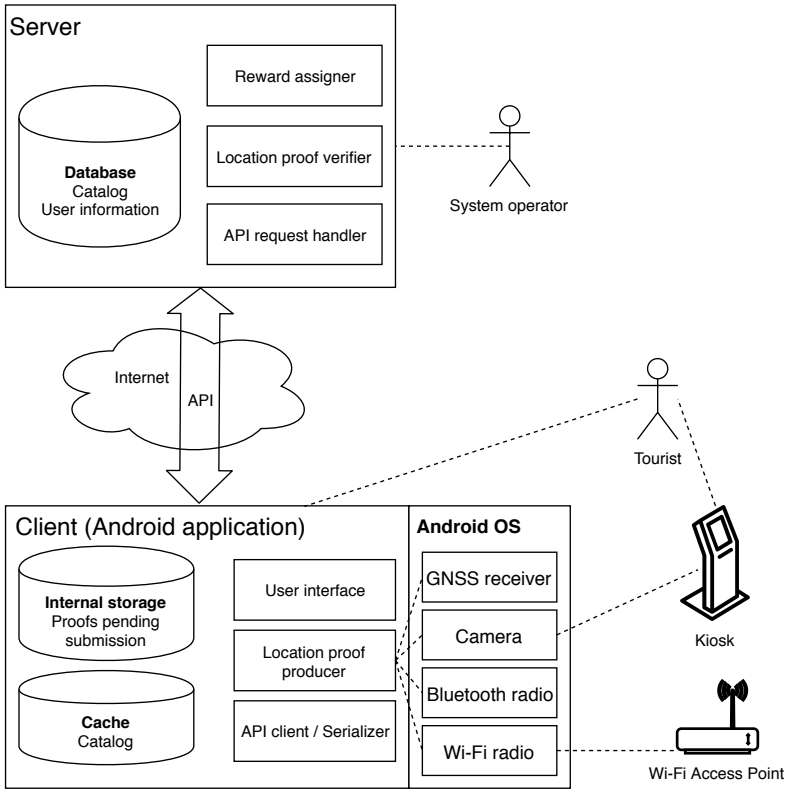

Fig. 2: Overview of the architecture of the developed solution.

The *catalog* stored on the smartphone contains information about the registered locations, tourism routes and respective rewards. It also contains the

BSSIDs[1] for a subset of the Wi-Fi networks that can be found at each location, that we call *triggers*, because they identify at which location it is, and set off the logging of observations for the location proofs. The ability to operate offline is important, as the intended users – tourists – may be roaming without a data plan, or the cellular coverage may not be available. The client communicates with the server, before and after the trip, through a REST API over HTTPS.

The server is responsible for validating the location proofs submitted by the client. For each claimed visit to a location, the server computes a *strength score* based on the set of proofs backing the visit. This value is calculated differently from location to location, depending on the proof strategy used. This score is also modified according to the characteristics of the movement of the user, i.e., it checks if the proofs were collected at a human-like pace.

In the definition of a route, each location is associated with a minimum strength score and a minimum visit duration. The user is eligible to receive the reward for a given route if the collected proofs match or exceed the minimum values acceptable for each point in the route. System operators handle these rewards, and the value of those are dependent of the location proof strategy used. Stronger proof strategies are then more suited for high value rewards.

4 Location proof strategies

We propose three different strategies for location proof production and verification, with increasingly stronger guarantees: *scavenging*, *TOTP*, and *Kiosk*.

4.1 Scavenging strategy

The scavenging strategy, represented in Figure 3, harnesses the large number of Wi-Fi networks installed by unrelated third parties in urban environments. Location proofs are produced simply by storing Wi-Fi scan results with associated timestamps. We store the SSIDs of networks in plaintext, since they are broadcasted by APs and therefore are public domain. If this were not case, an encryption algorithm would be used before storing the network SSIDs.

On the server side, the set of Wi-Fi networks present in the scan results is compared with the list of known networks for each location. This list is maintained by the system operators. To deal with the volatility of the network list and assist system operators in curating these lists, the server can analyze past location proofs to suggest the addition and removal of certain Wi-Fi networks from the database. The *strength score* is the fraction of client-presented networks over the total number of server-known networks.

The scavenging strategy is simple and has a reduced setup cost, as it just uses existing infrastructure. However, it provides weak guarantee: as soon as the list of networks at a certain location is known, an attacker can forge trip logs.

[1] Basic Service Set Identifiers, normally the address of the radio of the Access Point

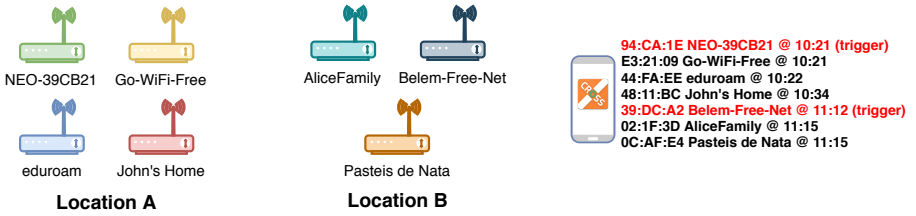

Fig. 3: Representation of the networks and logged information in a visit to two locations, A and B, where the scavenging strategy is used. At each location, one of the networks is known beforehand to trigger the identification.

4.2 TOTP strategy

The TOTP (Time-based One-Time Password) strategy is illustrated in Figure 4. This strategy allows for stronger proofs by deploying a customized Wi-Fi access point that is dynamically changing the broadcast SSID[2]. The SSID is used as a low-bandwidth, unidirectional communication channel to transmit a changing value. This strategy is standards-compliant and compatible with existing devices. Note that the device is observing the changing SSID values and does not need to connect to the network.

Time-based SSID setting The SSID should change in a way that is unpredictable to an observer, but which can be verified by the server. We achieve this by including in the SSID a TOTP, similar to the proposed in RFC 6238. Only the Wi-Fi AP and the CROSS server know the secret, to produce and validate the codes. Each AP should use a different secret key, and only the server should know the keys used by all APs. The APs and server must have synchronized clocks with minute granularity, but both components do not need to communicate, which means APs can function as stand-alone beacons in locations without Internet access. Since we are using minute granularity, clock deviation can happen, but does not impact our solution. We expect users to be in range of the APs for longer periods of time and therefore observe multiple changes in the SSID the AP.

We use a time-step size of 120 seconds, sufficient to provide enough resolution during proof verification, while still fitting within the constraints of most Wi-Fi Stations when it comes to updating scan results. We chose SHA-512 HMAC as the TOTP hash algorithm, with keys as long as the HMAC output, instead of the typically used SHA-1 HMAC. This allows the use of longer keys. These settings were selected to make it computationally complex to infer the secret TOTP key by continuously observing the different SSIDs assumed by the AP. This would amount to a key-recovery attack, where the key is recovered by observing the cipher output for known inputs. To the best of our knowledge, such an attack

[2] Service Set Identifier, the user-facing name for a Wi-Fi network

against SHA-512 HMAC is yet to be conceived [6], unlike HMAC using weaker hash algorithms [5].

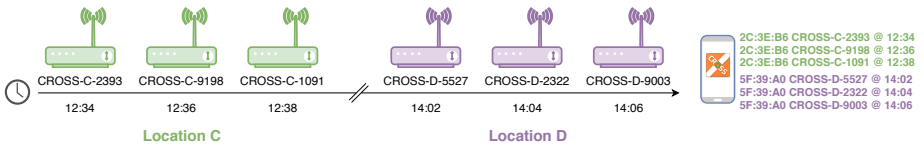

Fig. 4: Representation of the networks and logged information in a visit to two locations, C and D, where the TOTP strategy is used. There is one AP at each location.

Proof collection and validation Clients are programmed to log all the different SSIDs a Wi-Fi network assumes during their visit to a location, along with the timestamps at which each SSID was observed. Clients do not know whether each Wi-Fi network is part of the infrastructure for this strategy, as that is irrelevant to how they collect proofs; only the server needs to know this, to select the correct proof validation strategy. In other words, as far as the client implementation is concerned, the scavenging strategy and the TOTP strategy are the same.

The TOTP strategy, unlike the scavenging one, allows for attesting not just that the user was present at a certain location, but also that he did so at a certain point in time. Therefore, this strategy allows for verifying the visit duration. Here, the strength score corresponds to the fraction of visit time that could be verified, in relation to the total time the client claims to have been present at the location.

Validating the authenticity of Wi-Fi and Bluetooth devices is complex as the hardware identifiers can be trivially spoofed. Because this solution does not involve bi-directional communication with other devices or networks, as in many witness-based proof strategies [14], it minimizes user exposure to attacks. This also protects their privacy, as only the entity operating the CROSS server will be able to know which locations each user visited.

4.3 Kiosk strategy

The kiosk strategy counters the possibility of claiming multiple rewards for a single trip, by preventing variants of Sybil attacks [7], where a malicious visitor creates multiple user accounts and runs them in parallel using one or more smartphones. This strategy requires interaction with a kiosk device present at the location. The device can have other functionality, including showing information about the location or advertising. Existing tourism information kiosks can be adapted for this purpose. This approach can be an inconvenience for tourists.

To mitigate that, we can take advantage of existing ticket machines, so that the process of interaction with a kiosk is done while acquiring tickets for attractions.

Proof production and validation Similarly to Wi-Fi APs in the TOTP strategy, kiosks are required to have their clocks synchronized with the server, also with minute granularity. Each kiosk keeps a private key, which they will use to sign information. The server has the corresponding public key. Kiosks do not need to have a connection to the server.

Location proofs are produced as follows. The client application sends the username of the logged in user to the kiosk, by displaying a QR code[3] that is scanned by the kiosk. The latter, using its private key, signs a message containing the kiosk ID, the username of the user, the current date and time, and a randomly generated large number (a nonce). This message and respective signature is sent back to the client, again using a QR code, which is scanned by the latter.

The smartphone stores this data as a visit proof, part of the trip log. When the trip log is submitted to the server, it verifies this proof by checking the signed message using the public key associated with the kiosk and also that the kiosk ID matches that of a kiosk available at the visit location; the username matches the user account submitting the proof; the date and time is contained within the period of the visit; the nonce was not reused from any other visit proof submitted in the past.

By eliminating the remote network connection to the kiosk, an attacker must be physically present at the location to interact with it. Using QR codes for communication between the kiosk and the smartphone requires physical interaction. This physical interaction can also be inspected by a bystander, e.g., a tourist attraction staff member, to check for suspicious activity like attempting to check-in with more than one device.

This strategy is more inconvenient for the user but it boosts security. It should be used where there are already tourist support kiosks in place, and use the previous strategies in other locations.

5 Evaluation

To validate our solution, we developed prototypes of the client, server and Wi-Fi AP components. This allowed us to evaluate the scavenging and TOTP strategies.

The client prototype is an Android application written in Java, compatible with off-the-shelf smartphones running Android 4.4 and up. The client uses a SQLite database to store the catalog for offline operation, and to store trip logs and respective location proofs. The server exposes a REST API, with JSON payloads, which the client uses to obtain the catalog, and to submit trip logs. The server is written in Go and uses a PostgreSQL database to store information

[3] A QR (Quick Response) code is a type of barcode that can be scanned by a smartphone built-in camera.

about locations, tourism routes, rewards, and the Wi-Fi networks present at each location, including TOTP secrets. The database is also used to store user credentials and trip logs including the respective location proofs, for auditing. The Wi-Fi AP component for TOTP was implemented using a ESP8266 board, a low-cost Wi-Fi microchip with full TCP/IP stack. The firmware was written in C++ using the Arduino environment for this microchip.

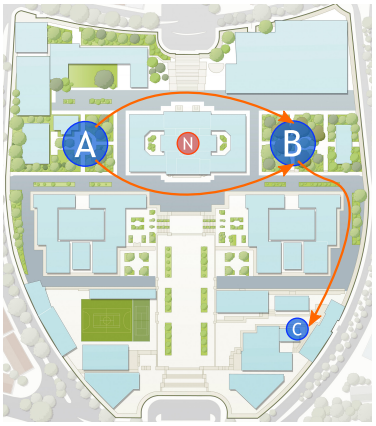

Fig. 5: Campus route used in the experiments.

An evaluation scenario was set up in the Alameda campus of Instituto Superior Técnico, where voluntary participants completed a simulated tourism route, shown in Figure 5, composed of three locations **A**, **B** and **C**. Additionally, a control location, **N**, was selected to serve as off-limits, and the participants were asked not to visit it.

The simulated route made use of both the scavenging and TOTP strategies. Participants brought their own personal Android phones, which let us reach a large and diverse sample size. A total of 34 Android smartphones were used in the experiment.

5.1 Location detection performance

Some factors that reduce the accuracy of the system include: AP transmit power, receiver sensitivity, number of networks and interference sources in an area, and signal propagation patterns. Despite these factors, the expected result in this experiment is that each device should be able to detect all locations except **N**. The results presented in Table 1 correspond to the results after the devices were present for three minutes at each location, except for location **N**, near which every device passed on the way between **A** and **B**.

As expected, no devices detected control location **N**. For other locations, results are satisfactory as well. The lower detection rate of location **A** in comparison with **B** may be explained by the lower number of trigger networks configured

Location	Total visits	Total detections	Success rate
A	34	30	88%
B	34	33	97%
C	34	34	100%
N	0	0	100%

Table 1: Location detection performance after three minutes at each location (except for **N**, not visited).

for **A**. All devices detected location **C** within three minutes, which may be explained by the fact that the single AP was in the same room as the participants, therefore its signal was much stronger and easier to detect than the signals of the APs at **A** and **B**, which were installed in the nearby buildings, at distances between 20 and 80 meters from the users.

5.2 Location proof performance

In locations **A** and **B**, the Scavenging strategy was used. In this strategy, the confidence score corresponds to the percentage of networks found by the client, compared to the total number of APs registered in the server for each location. In this experiment, we previously registered 21 known APs for location **A**, and 17 known APs for location **B**.

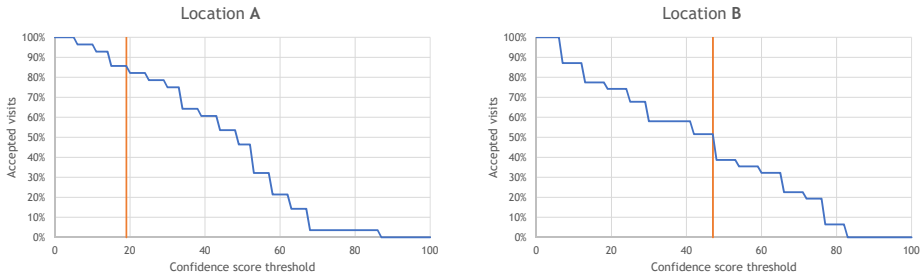

Fig. 6: Percentage of accepted visits in function of the confidence score threshold configured at locations **A** and **B**.

Figure 6 shows the percentage of accepted visits for locations **A** and **B**, as a function of the confidence score threshold that is set for those locations. When deciding whether to reward an user, all visits must be accepted for the trip to count, but here, each location is being analyzed individually. The vertical orange line in the charts corresponds to the percentage of known networks that are triggers, at each location. We consider that it represents the minimum confidence score threshold acceptable, as only visits proofs with a higher score are guaranteed to contain a non-trigger (secret) network.

Results for this strategy fell short of expectations, as the confidence score threshold has to be set very low – lower than recommended – for a large percentage of visits to be accepted. These results show that most devices did not see a majority of the networks associated to each location, in part certainly due to the short visit duration (three minutes) and the weak network signal levels, whose APs were relatively distant.

In location **C**, the TOTP strategy was used. In this strategy, the confidence score corresponds to the percentage of visit time that could be verified by the TOTP codes present in the scan results collected by the client. Figure 7 shows the relation between the threshold and the accepted visits, for this location.

Results for this strategy were positive. Most devices successfully captured the SSID changes every two minutes; 24 devices (75%) were even able to capture TOTP codes attesting the entirety of the visit period (10 minutes).

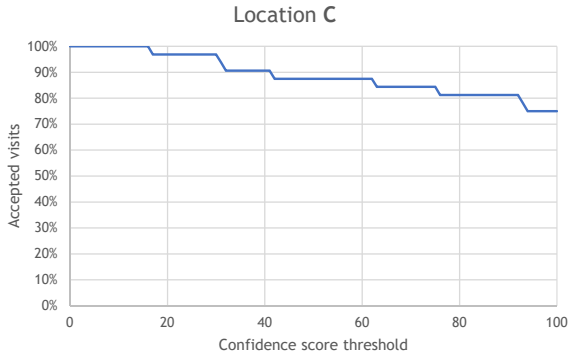

Fig. 7: Percentage of accepted visits in function of the confidence score threshold configured at location **C**.

5.3 Power consumption

To assess the power consumption of our techniques and compare their consumption with that of alternative solutions, we collected battery usage data on a LG V40 ThinQ smartphone, running Android 9.0.

We compared three different situations: location using both Wi-Fi and GNSS, location using exclusively Wi-Fi scanning, and no location collection at all. For the first case, a modified CROSS application, that also used GNSS to collect location information, was used. In the second case, the unmodified CROSS application was used. In both cases, data was requested every 30 seconds. In the third case, no applications were used - the phone was left turned on, with Wi-Fi enabled, without explicitly using any applications. Table 2 presents the results. *p.p.* stands for *percentage points*.

Method	Polling rate	Total test duration	Average battery drain
No collection	N/A	29 h 05 min	0.58 p.p. / hour
Collection using Wi-Fi	30 s	39 h 30 min	0.61 p.p. / hour
Collection using GNSS and Wi-Fi	30 s	08 h 00 min	1.25 p.p. / hour

Table 2: Battery drain depending on the location collection method.

CROSS, which exclusively uses Wi-Fi, presents a negligible increase in power consumption relative to no location collection.

5.4 Scavenging feasibility

One of the concerns with the scavenging strategy, presented in Section 4.2, is the need to maintain the lists of Wi-Fi networks for each location where this strategy is used. As time passes, some of the networks may disappear, and new, different networks may appear. Even though the server suggests the addition and removal of networks based on the submitted visit proofs, these suggestions need to be manually vetted. Therefore, it is important to understand how frequently Wi-Fi networks appear and disappear in the real world, to assess whether the current implementation is adequate.

We collected data on the Wi-Fi networks in range, at six locations in Lisbon, in three dates. The second date was ten days after the first, and the third date was 31 days after the first. Five of the locations are well-known tourist attractions and one is a residential area, for comparison with a less busy location. The results, presented in Table 3, correspond to the duplicated network counts after merging the data from the three devices. Across devices and visits, APs were identified by their BSSID to avoid counting renamed networks (such as in our own TOTP strategy) as separate networks. Values for both periods are always relative to the first visit.

		After ten days		After one month	
Location	Initial total	Present	New	Present	New
Alvalade	86	74 (86%)	13	73 (85%)	31
Comércio	133	8 (6%)	60	7 (5%)	43
Gulbenkian	80	54 (68%)	92	54 (68%)	55
Jerónimos	148	34 (23%)	100	24 (16%)	62
Oceanário	39	22 (56%)	41	24 (62%)	40
Sé	61	25 (41%)	43	22 (36%)	44

Table 3: Wi-Fi networks present at each tourist attraction.

The number of networks still present ten days after the first visit is a good indicator of the number of networks that can be considered in the scavenging technique, at each location. Most locations have a sufficiently large set of usable networks, with the notable exception of Comércio, where just 8 APs appear to be permanently installed.

To assess the frequency at which the lists of networks must be updated, we can look at the number of permanent networks that disappeared between the second visit (after ten days) and the third visit (after one month). In most cases, there is only a minor reduction from one visit to another, with Jerónimos being the worst case, but still with a sufficient number of permanent networks.

6 Conclusion

In this paper we presented CROSS, a system that implements location proof techniques for consumer mobile applications. We used smart tourism as a use case, developing a smartphone application where location proofs are used to implement a reward scheme. CROSS includes three different location proof strategies, with trade-offs between strong security guarantees and easier user experience. The system was evaluated in a realistic setting using a diverse sample of devices. The results show the feasibility of location proofs running in current mobile operating systems and hardware without special privileges or configurations.

Acknowledgements

This work was supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under project UIDB/50021/2020 and through project with reference PTDC/CCI-COM/31440/2017 (SureThing).

References

1. Agadakos, I., Hallgren, P., Damopoulos, D., Sabelfeld, A., Portokalidis, G.: Location-enhanced authentication using the IoT. In: Proceedings of the 32nd Annual Conference on Computer Security Applications - ACSAC '16. ACM Press (2016). https://doi.org/10.1145/2991079.2991090
2. Azizyan, M., Constandache, I., Choudhury, R.R.: SurroundSense. In: Proceedings of the 15th annual international conference on Mobile computing and networking - MobiCom '09. ACM Press (2009). https://doi.org/10.1145/1614320.1614350
3. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing **2**(4), 263 (2007). https://doi.org/10.1504/ijahuc.2007.014070
4. Canlar, E.S., Conti, M., Crispo, B., Pietro, R.D.: CREPUSCOLO: A collusion resistant privacy preserving location verification system. In: 2013 International Conference on Risks and Security of Internet and Systems (CRiSIS). IEEE (oct 2013). https://doi.org/10.1109/crisis.2013.6766357

5. Contini, S., Yin, Y.L.: Forgery and partial key-recovery attacks on HMAC and NMAC using hash collisions. In: Advances in Cryptology – ASIACRYPT 2006, pp. 37–53. Springer Berlin Heidelberg (2006). https://doi.org/10.1007/11935230_3
6. Dobraunig, C., Eichlseder, M., Mendel, F.: Security evaluation report on SHA-224, SHA-512/224, SHA-512/256, and the six SHA-3 functions. Tech. rep., CRYPTREC (2015)
7. Douceur, R., J.: The Sybil attack. In: Revised Papers from the First International Workshop on Peer-to-Peer Systems. pp. 251–260. IPTPS '01, Springer-Verlag, London, UK, UK (2002), http://dl.acm.org/citation.cfm?id=646334.687813, accessed November 30, 2019
8. Ferreira, J., Pardal, M.L.: Witness-based location proofs for mobile devices. In: 17th IEEE International Symposium on Network Computing and Applications (NCA) (Nov 2018)
9. Google LLC: Indoor Maps – About, https://www.google.com/maps/about/partners/indoormaps/, accessed November 30, 2019
10. Gretzel, U., Sigala, M., Xiang, Z., Koo, C.: Smart tourism: foundations and developments. Electronic Markets **25**(3), 179–188 (aug 2015). https://doi.org/10.1007/s12525-015-0196-8
11. Maia, G.A., Pardal, M.L.: CROSS: loCation pROof techniqueS for consumer mobile applicationS. In: INForum. Guimarães, Portugal (Sep 2019)
12. Mariakakis, A.T., Sen, S., Lee, J., Kim, K.H.: SAIL. In: Proceedings of the 12th annual international conference on Mobile systems, applications, and services - MobiSys '14. ACM Press (2014). https://doi.org/10.1145/2594368.2594393
13. Talasila, M., Curtmola, R., Borcea, C.: LINK: Location verification through immediate neighbors knowledge. In: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 210–223. Springer Berlin Heidelberg (2012). https://doi.org/10.1007/978-3-642-29154-8_18
14. Zhu, Z., Cao, G.: APPLAUS: A privacy-preserving location proof updating system for location-based services. In: 2011 Proceedings IEEE INFOCOM. IEEE (apr 2011). https://doi.org/10.1109/infcom.2011.5934991