CROSS: loCation pROof techniqueS for consumer mobile applicationS

Gabriel A. Maia
Instituto Superior Técnico, Universidade de Lisboa, Portugal
gabriel.maia@tecnico.ulisboa.pt

***Abstract*—The ubiquitousness of smartphones, wearables and other mobile devices, coupled with the increasing amount of communications infrastructure present in urban environments, has led to the rise of location-aware applications. Many of these applications do not verify the location information they consume, making them vulnerable to location spoofing attacks. Location proof systems aim to solve this problem by allowing devices to interact with location-specific resources and issue proof that they have been at specific locations on specific times.**

In this work we introduce CROSS, a system that performs location verification using techniques compatible with off-the-shelf Android smartphones. We present three strategies for the production of location proofs with increasing tamper-resistance, two of which are based on Wi-Fi and a third based on physical interaction with kiosk-like devices. Our techniques were designed with user privacy and security in mind, minimizing the amount of connections between devices. A prototype application was implemented to assess the feasibility and reliability of the architecture and location proof strategies. The application allows rewarding users who complete a touristic route, with proofs of visit collected along the way. We use smart tourism as the demonstrative use case for the usefulness and feasibility of location proofs, in the context of a mobile application. Our evaluation, which included experiments with 30 users, showed that the system can be feasibly used in real-world scenarios, providing adequate security guarantees for the intended use case.

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

I. Introduction

In the coming years, the amount of Internet-connected devices will increase by orders of magnitude. These sensors and actuators will connect the physical and virtual worlds constituting the Internet of Things (IoT). Smartphones will play an important role as user interfaces between people and the IoT devices.

Many mobile IoT applications use the location context to provide their core functionality or to augment their capabilities [1]. These systems typically do not verify the location information they use, and are susceptible to *location spoofing attacks*. Developing the means to validate location information is, therefore, of high importance. *Location proof systems* differ from location systems in that they focus on countering location spoofing, by providing verifiable location information. The methods that can be used to produce location proofs depend on the available information sources and on the intended use case. One of the possible use cases for location proofs is in *Smart Tourism* which provides tourists with rich experiences supported by mobile technology [2]. Personal devices of tourists can interact with existing or newly-added infrastructure in emblematic city locations. These interactions can then be used to verify location information allowing, for example, the implementation of reward schemes.

Wi-Fi can be used as infrastructure for location because most urban environments in populated areas tend to have many Wi-Fi networks. Some of these are for private or institutional use, while others are open for the general public to use. Nevertheless, the overwhelming majority of these networks announce their presence and can be detected using virtually all smartphones in the market.

In this paper we propose CROSS, a system that uses the Wi-Fi networks present in a predefined set of locations, to both detect the presence of the user in these locations, and to verify that the user is not spoofing his location. This information is used, in the example application, to ascertain whether the user completed any tourism circuits from a predefined set of routes. The smart tourism application runs on the smartphones of tourists. The system uses Wi-Fi to determine whether the user is present at a location, using techniques that allow the implementation of location proofs without degrading the user experience.

The rest of this paper is organized as follows. Section II presents a brief overview of existing works on location proofs. An overview of our system and its operation is presented in Section III. In Section IV, we propose three different location proof strategies. The evaluation is presented in section V. Finally, Section VI presents the conclusion.

II. Related work

Wi-Fi technology is widely used in mobile location systems, typically to complement GNSS[1]. Wi-Fi is also used for microlocation, in systems such as Google Indoor [3], where GNSS tend to perform poorly. SAIL [4] is an example of a microlocation system which works by combining the Time-of-Flight of Wi-Fi packets with motion sensor data. SurroundSense [5] uses fingerprinting techniques encompassing Wi-Fi, motion sensors, microphones and cameras, to identify the location of the user.

Most works in the field of location proofs focus on providing strong guarantees, often using complex cryptography schemes for proof production and verification. These can be used, for example, to implement authentication schemes, to

[1] Global Navigation Satellite Systems, such as GPS, Galileo or BeiDou.

limit the geographical availability of services, to aid in identity verification or to combat tax evasion. However, these systems can sometimes be obtrusive, requiring the user to perform unnatural actions when using their software and hardware. This is undesirable in a smart tourism application, which should be able to work using the platforms available today, without impairing the user experience.

Witness-based systems such as APPLAUS [6], LINK [7] and SureThing [8] typically use peer-to-peer communication between witnesses. Peer-to-peer communication is increasingly hampered by current consumer-oriented mobile operating systems (iOS and Android), which are heavily oriented towards client-server communication models, as they place few restrictions on Internet access while forbidding or requiring special permissions to access the peer-to-peer features of Wi-Fi and Bluetooth radios, ultimately resulting in a poor user experience if one wishes to use these capabilities.

Systems which rely solely on mobile witnesses, without fixed infrastructure, require a minimum amount of diverse users at each location to work. The CREPUSCOLO [9] 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 [10], a system that locates users and models their movement through the use of IoT devices and smart environments, uses homomorphic encryption for processing data on third-party servers, that can process but not learn the location of the users.

III. System overview

CROSS has four main components, represented in Figure 1: client application, server accessed through API, Wi-Fi Access Point (for proof strategy described in IV-B), and Kiosk (for proof strategy in IV-C). CROSS stands for "loCation pROof techniqueS for consumer mobile applicationS". The system uses a client-server model with no peer-to-peer communication between clients. This has advantages from a security and user experience standpoint, which we will detail later.

The system operation from the point of view of a tourist is represented in Figure 2. A tourist installs the smartphone application and signs up for an account on the system. Before starting the trip, the tourist starts the application, and downloads the catalog of locations. The application logs visits to locations, illustrated in the figure as points P1 through P4. The location sensing relies on Wi-Fi exclusively and takes advantage of the scans regularly 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.

The *catalog* is stored on the smartphone to allow offline operation. It contains information about the registered locations, tourism routes and respective rewards. It also contains the BSSIDs[2] for a subset of the Wi-Fi networks that can be found at each location. The application uses this subset, which we call *triggers*, to identify at which location it is, and set off the collection of Wi-Fi-based 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 server is responsible for computing which rewards the user is eligible to receive based on his route, after 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 at each one. 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.

The client communicates with the server through a REST API over HTTPS. This API is used to manage user sessions, to submit trip logs with the respective location proofs, and to check for rewards.

IV. Location proof strategies

We propose three different strategies for location proof production and verification, with increasingly stronger guarantees. The first strategy, *scavenging*, relies solely on existing Wi-Fi networks, deployed by third-parties. The second strategy, *TOTP* (Time-based One-time Password), relies on first-party Wi-Fi infrastructure deployed and configured specifically for use with CROSS. The third strategy, *Kiosk*, provides the strongest guarantees and requires users to physically interact with an electronic kiosk.

A. Scavenging strategy

The idea behind this approach is to harness the large amount of Wi-Fi networks installed by unrelated third parties in urban environments. These networks may appear and disappear at any time. In this strategy, location proofs are produced simply by storing Wi-Fi scan results with associated timestamps. These results are then submitted as part of the trip log.

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 main advantage of the scavenging strategy is its simplicity and reduced setup cost, as it just uses existing infrastructure. However, it is also the strategy that provides the weakest guarantees: as soon as the list of networks at a certain location is known, an attacker can forge trip logs.

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

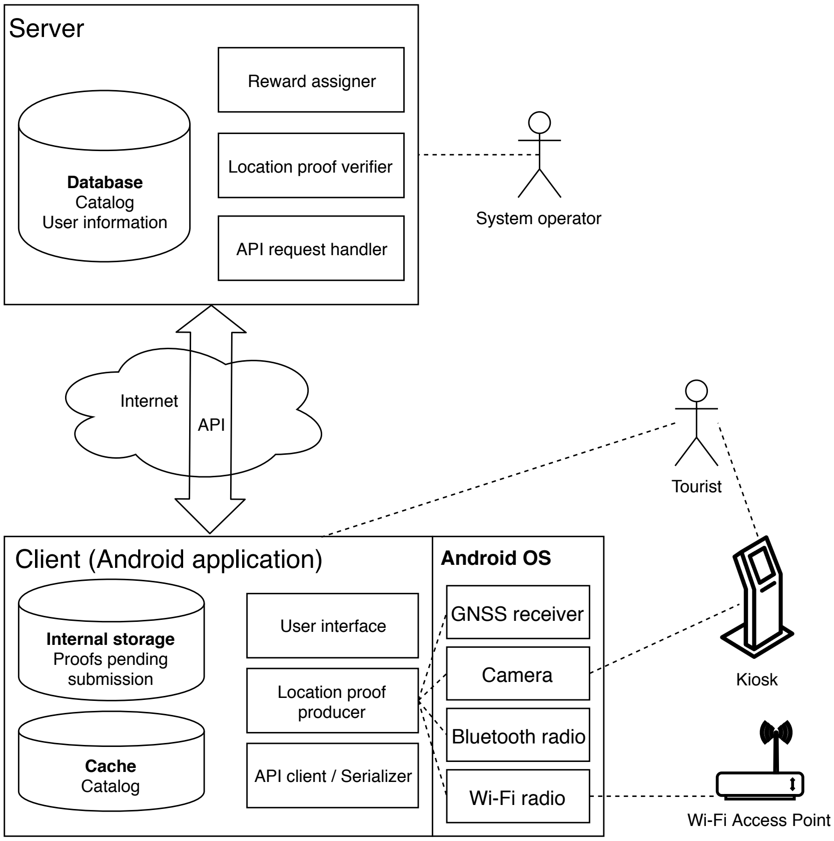


Fig. 1: Overview of the architecture of the developed solution.

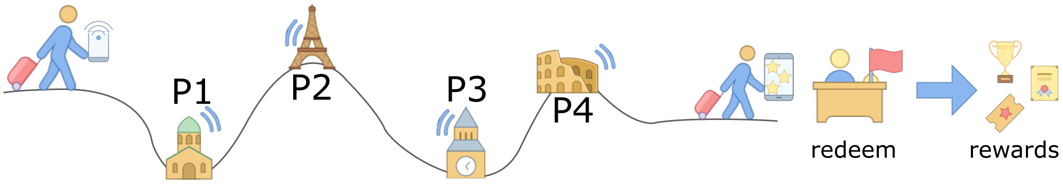
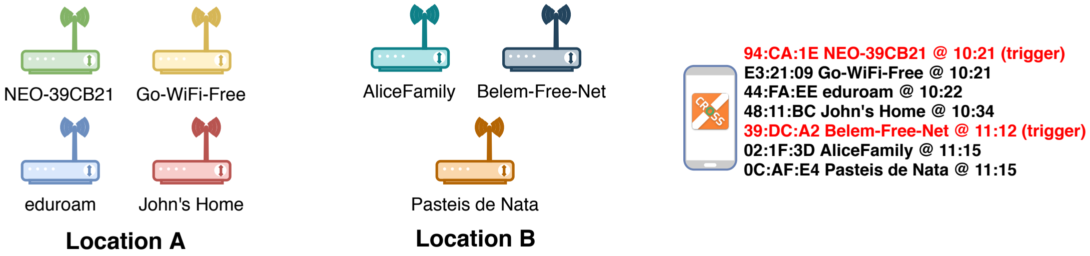


Fig. 2: User flow throughout a tourism route with four locations.



```
94:CA:1E NEO-39CB21 @ 10:21 (trigger)
E3:21:09 Go-WiFi-Free @ 10:21
44:FA:EE eduroam @ 10:22
48:11:BC John's Home @ 10:34
39:DC:A2 Belem-Free-Net @ 11:12 (trigger)
02:1F:3D AliceFamily @ 11:15
0C:AF:E4 Pasteis de Nata @ 11:15
```

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.

B. TOTP strategy

This strategy allows for stronger proofs by deploying a customized Wi-Fi access point that is dynamically changing the broadcast SSID[3]. 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.

1) 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 Time-based One-Time Password (TOTP), similar to the proposed in RFC 6238. Only the Wi-Fi AP and the CROSS server know the TOTP secret, to produce and validate OTPs, respectively. 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.

Our solution uses a carefully selected time-step size and hash algorithm, that are different from those recommended in RFC 6238, as our use case is different from the typical TOTP use case where the one-time password acts as a second authentication factor.

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 against SHA-512 HMAC is yet to be conceived [11], unlike HMAC using weaker hash algorithms [12].

2) 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. For example, if the client claims to have been present at a location for 20 minutes, but only 7 OTPs could be verified, corresponding to a total of 14 minutes within the claimed 20 minutes period, the strength score will be 70%. Whenever the TOTP strategy is set up at a location, it supersedes the scavenging one, as it provides stronger guarantees. This way, updating the list of networks is not a concern for locations where custom APs are installed.

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 [6], 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.

C. Kiosk strategy

The TOTP strategy prevents the attacker from creating new proofs on the fly, but not from replaying proofs from a legitimate visit under a different user account, or tunnelling the information to a distant user. The kiosk strategy counters the possibility of claiming multiple rewards for a single trip, by preventing variants of Sybil attacks, where a malicious visitor creates multiple user accounts and runs them in parallel using one or more smartphones.

This strategy requires the tourist to interact with a machine present at the location - the *kiosk* - in order to prove his presence. In CROSS, the main function of the kiosk is to sign a message for the CROSS client, logged on the account of the user and running on his smartphone. The kiosk can have other functionality unrelated to CROSS, including showing advertising or information about the location. Existing tourism information kiosks can be adapted for this purpose.

1) 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[4] 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

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

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

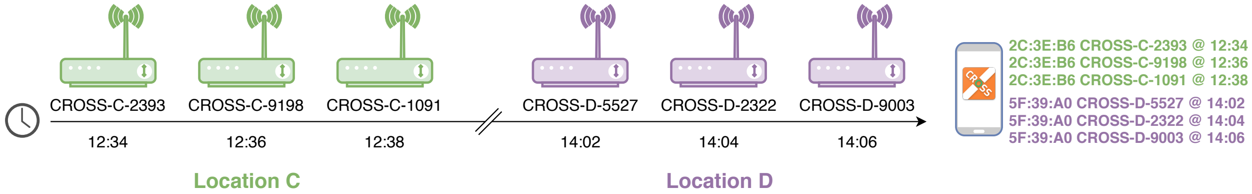

CROSS-C-2393
CROSS-C-9198
CROSS-C-1091
CROSS-D-5527
CROSS-D-2322
CROSS-D-9003
12:34
12:36
12:38
14:02
14:04
14:06
Location C
Location D
2C:3E:B6 CROSS-C-2393 @ 12:34
2C:3E:B6 CROSS-C-9198 @ 12:36
2C:3E:B6 CROSS-C-1091 @ 12:38
5F:39:A0 CROSS-D-5527 @ 14:02
5F:39:A0 CROSS-D-2322 @ 14:04
5F:39:A0 CROSS-D-9003 @ 14:06

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.

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, excluding attacks based on amplification of wireless signals that would be possible with Bluetooth or NFC, for example. This physical interaction is essential to prevent Sybil attacks [13]. It can easily be inspected by a bystander, e.g. a tourist attraction staff member, who can check the behavior of the users for any suspicious activity, e.g. attempting to check-in with more than one device, or using the same device to check in multiple times, using different user accounts.

The inconvenience for the user, and the kiosk setup cost for the system operators, can be greatly minimized by only using this strategy in a few locations per trip, where there are already tourist support infrastructures, and resorting to the previous strategies in other locations.

V. Evaluation

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

The client prototype is an Android application written in Java. It is a simplified version of a smart tourism application, compatible with off-the-shelf smartphones running Android 4.4 and up. The device must have a Wi-Fi radio, which is very common. The client uses a SQLite database to store the catalog for offline operation, and to store trip logs and respective location proofs, for opportunistic submission on the server.

The server is written in Go and uses a PostgreSQL database to store information about locations, tourism routes, rewards, and the Wi-Fi networks present at each location, including TOTP secrets for trusted APs. Most importantly, the database is used to store user credentials and trip logs including the respective location proofs, for auditing. The server exposes a REST API, with JSON payloads, which the client uses to obtain the catalog, and to submit trip logs.

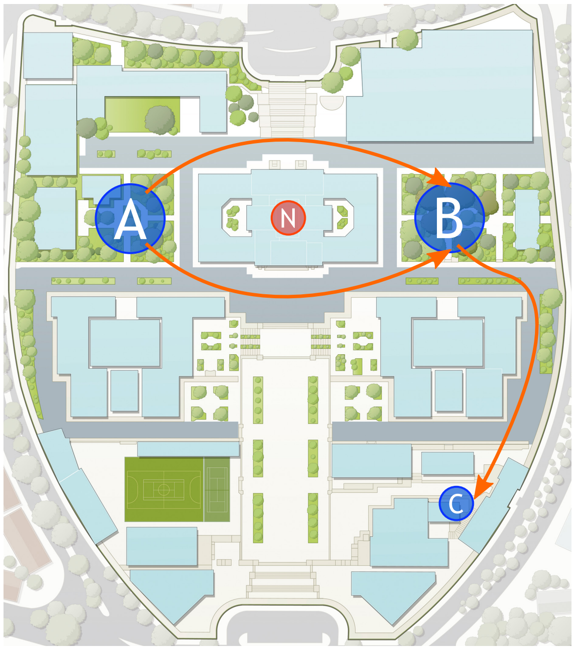


Fig. 5: Alameda campus route used in the evaluation.

The Wi-Fi AP component 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.

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. Participants were asked not to visit this location. 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.

A. *Location detection performance*

Because our system exclusively uses Wi-Fi to detect its proximity to each location, it is limited by the ability of the devices to accurately detect Wi-Fi networks. A plethora of other factors reduce the accuracy of the system, among them: AP transmit power, receiver sensitivity, amount of networks and interference sources in an area, signal propagation patterns,

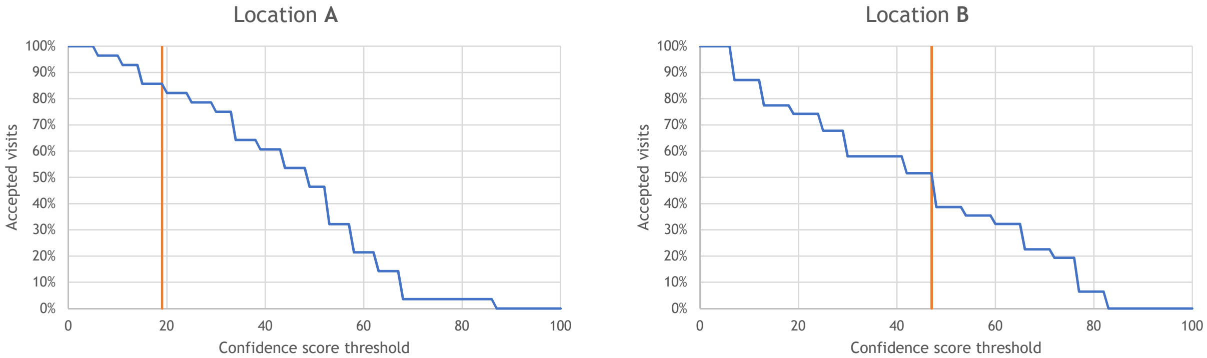


Fig. 6: Percentage of accepted visits in function of the confidence score threshold configured at locations **A** and **B**.

and the ability of the Wi-Fi station to display scan results in real-time (a minority of phones have delays presenting updated scan results).

In this experiment, the expected result is that each device should be able to detect all locations except **N**. The results presented in Table I 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**.

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

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

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 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.

B. Location proof performance

In CROSS, location proof elements are analyzed by the server, therefore here we will only consider visits that were submitted to the server. A minority of users experienced submission failures, due to e.g. Internet connectivity issues. The total number of trips analyzed per location is shown in Table II.

Location	Total visits	Total submissions
A	34	28
B	34	31
C	34	32

TABLE II: Visits submitted to the server for analysis, per location.

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.

Figure 6 shows the percentage of accepted visits for locations **A** and **B**, in 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).

C. 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

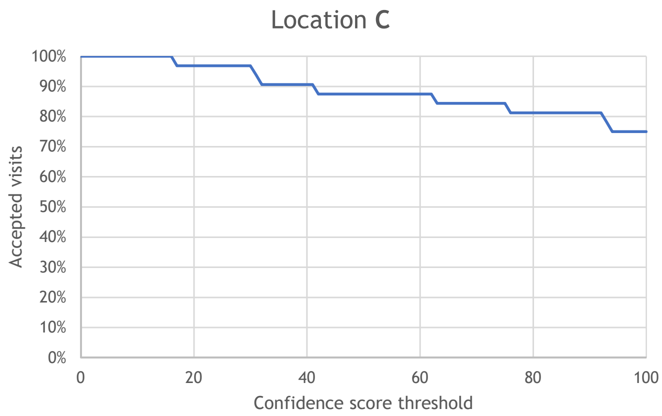


Fig. 7: Percentage of accepted visits in function of the confidence score threshold configured at location **C**.

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 III presents the results.

Method	Polling rate	Total test duration	Average battery drain
CROSS using GNSS and Wi-Fi	30 s	8 h	1.25 p.p.[5] / hour
CROSS using Wi-Fi	30 s	39 h 30 min	0.61 p.p. / hour
No collection	N/A	29 h 5 min	0.58 p.p. / hour

TABLE III: 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. This is not the case when GNSS is used. This increase, of 0.03 percentage points per hour, can be attributed to random variations and to deficiencies in the analysis method.

D. Scavenging feasibility

One of the concerns with the scavenging strategy, presented in Section IV-B, 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.

If the sets of networks at a location change too frequently, the scavenging strategy may prove to be inadequate for that location: at a limit, the server database would need to be updated almost daily. 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, that serves as an example of a location where there could be an interest in using the system, despite not being a recognized tourist attraction.

The application used to collect the information was a modified version of CROSS, which registered location information and Wi-Fi scan results every 30 seconds. Visits to locations lasted for 15 minutes each, and data was simultaneously collected by three different smartphones, in order to always collect the largest amount of networks possible and minimize random variations where a device may not see a network for unknown reasons.

We are interested in knowing, for each location, how many Wi-Fi networks are still *Present*, and how many *New* networks became available, after ten days and after a month. The results, presented in Table IV, correspond to the deduplicated 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 IV: Wi-Fi networks present at each location after ten days and after a month.

In the general case, the scavenging strategy appears to be viable. A large number of networks is present at each location. In certain locations, notably, Comércio and Jerónimos, most networks appear to be temporary. For the scavenging strategy we propose, temporary or mobile Wi-Fi networks should not be considered. 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.

In terms of 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, where 10, or 30%, of the permanent networks seem to have been disabled after twenty days.

[5] Percentage points

The suggestions given by the server on what networks to add and remove, can be used to update the lists without revisiting the locations, if one is willing to trust the trip logs of the users. At the very least, they can indicate that a new thorough survey of the networks is necessary.

E. Limitations

The scavenging and TOTP strategies are limited by the Wi-Fi capabilities of each device, as detailed in section V-A. The scavenging strategy provides weak security guarantees, as its proofs can be easily forged. The TOTP strategy is stronger, but still allows proofs for each time period to be reused by different user accounts. It is also vulnerable to denial of service attacks, where clients collect invalid SSIDs broadcast by impostor Access Points. The kiosk strategy overcomes these limitations and provides much stronger guarantees, but it is still vulnerable to denial of service attacks, even if they will require much more effort from the attacker with no clear benefit for him.

VI. Conclusion and future work

In this paper we presented CROSS, a system that implements location proof techniques in consumer mobile applications. Location proofs allow for the verification of the location information provided by smartphone sensors, increasing the dependability of this information. We used smart tourism as the demonstrative use case, developing a smartphone application where location proofs are used to implement a reward scheme. Users unlock rewards by verifiably completing predefined tourism circuits.

CROSS uses three different location proof strategies, with increasing tamper-resistance. The evaluation, performed in a realistic setting using a diverse sample of devices, demonstrates the feasibility of location proofs in consumer-oriented mobile applications, running in current mobile operating systems and hardware without special privileges or configurations. Our contribution allows trade-offs between strong security guarantees and easier user experience.

In terms of future work, the scavenging strategy would benefit from the implementation of a method for identification of the optimal threshold setting and trigger networks. Regarding user privacy, the proposed techniques are already privacy-protecting, by not broadcasting the location of the users, not disclosing their presence to others, and not collecting location information outside of predetermined locations. CROSS could benefit from further work in this area, e.g. by making it impossible to associate a trip log with a specific user, while still assigning rewards.

Acknowledgements

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2019 (INESC-ID) and through project with reference PTDC/CCI-COM/31440/2017 (SureThing).

References

[1] 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, p. 263, 2007.

[2] U. Gretzel, M. Sigala, Z. Xiang, and C. Koo, "Smart tourism: foundations and developments," *Electronic Markets*, vol. 25, no. 3, pp. 179–188, aug 2015.

[3] Google LLC. Indoor Maps – About. [Online]. Available: https://www.google.com/maps/about/partners/indoormaps/

[4] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "SAIL," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services - MobiSys '14*. ACM Press, 2014.

[5] M. Azizyan, I. Constandache, and R. R. Choudhury, "SurroundSense," in *Proceedings of the 15th annual international conference on Mobile computing and networking - MobiCom '09*. ACM Press, 2009.

[6] Z. Zhu and G. Cao, "APPLAUS: A privacy-preserving location proof updating system for location-based services," in *2011 Proceedings IEEE INFOCOM*. IEEE, apr 2011.

[7] M. Talasila, R. Curtmola, and C. Borcea, "LINK: Location verification through immediate neighbors knowledge," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Springer Berlin Heidelberg, 2012, pp. 210–223.

[8] J. Ferreira and M. L. Pardal, "Witness-based location proofs for mobile devices," in *17th IEEE International Symposium on Network Computing and Applications (NCA)*, Nov. 2018.

[9] E. S. Canlar, M. Conti, B. Crispo, and R. D. Pietro, "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.

[10] I. Agadakos, P. Hallgren, D. Damopoulos, A. Sabelfeld, and G. Portokalidis, "Location-enhanced authentication using the IoT," in *Proceedings of the 32nd Annual Conference on Computer Security Applications - ACSAC '16*. ACM Press, 2016.

[11] C. Dobraunig, M. Eichlseder, and F. Mendel, "Security evaluation report on SHA-224, SHA-512/224, SHA-512/256, and the six SHA-3 functions," CRYPTREC, Tech. Rep., 2015.

[12] S. Contini and Y. L. Yin, "Forgery and partial key-recovery attacks on HMAC and NMAC using hash collisions," in *Advances in Cryptology – ASIACRYPT 2006*. Springer Berlin Heidelberg, 2006, pp. 37–53.

[13] Douceur and J. R., "The sybil attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. London, UK, UK: Springer-Verlag, 2002, pp. 251–260. [Online]. Available: http://dl.acm.org/citation.cfm?id=646334.687813