A Witness Protection Program for a privacy-preserving location proof system

João Paulo Nunes da Costa
Instituto Superior Técnico, Universidade de Lisboa, Portugal

***Abstract*—The widespread presence of smartphones in daily life has changed the experience and expectations of end-users. Mobile apps are routinely able to determine their location and present it to the users, but this information is vulnerable to location spoofing attacks. One of the approaches to thwart location spoofing is to collect unique sensor readings at the location in a specific time slot and then, later, respond to verification challenges and compare with readings made by crowd-sourced, ad-hoc witnesses. This work proposes a witness privacy protection to the SureThing location certification system. SureThing uses a combination of Wi-Fi, Bluetooth and other sources of signal, and ad-hoc witnesses at the same time and location, to produce a verifiable proof of location. The work was evaluated with detailed simulations for a use case, ticketless public transport, using a data set collected on actual public transports in a city. The simulation results show that the system is feasible and allows proofs to be successfully issued and verified with adequate privacy protection on 70% of requests made on a full bus.**

I. Introduction

The widespread presence of smartphones in daily life has changed the experience and expectations of end-users. Mobile apps are routinely able to determine their location and present it to the users, for applications such as ride-hailing, food delivery, parking meters and many forms of social networking. More and more applications rely on these location-based services [1] but the location information is vulnerable to spoofing attacks, as it is usually collected in a best-effort approach, using unauthenticated GPS, Wi-Fi or Bluetooth signals. In typical location spoofing attacks [2], the spoofer transmits a signal to the the receivers to deceive them. This deception can occur when the legitimate transmitter stops transmitting the signal. The spoofer can also transmit the deceiving signal with higher power to the receiver. Then, the receiver would accept the spoofing signal instead of the legitimate signal from the transmitter. In a specific example [3], GPS signals sent by the satellites for aircrafts and Unmanned Aerial Vehicles (UAVs) is not secure. A malicious transmitter can spoof the GPS signal by emitting similar signals with a higher power. The aircraft would accept the spoofed signal instead of the authentic signals and could be misdirected to an unwanted location.

Location spoofing attacks can have a severe impact on location-based services. One of the approaches to thwart location spoofing is to collect unique sensor readings at the location in a specific time slot and then, later, respond to verification challenges and compare with readings made by ad-hoc witnesses. This is the approach followed by the SureThing system [4]. Witnesses are other users that happen to be at the same location at the same time. These witnesses act as crowd-sourcing for a location certification system. They have the incentive to act as witnesses because they later need their own location proofs. However, for the system to be trusted and used by the witnesses it needs to be transparent about data use and include privacy protections. In other words, there must be a witness protection program in place.

This paper presents an extension of SureThing to make it a privacy-preserving location proof system for mobile devices that addresses attacks against the privacy of users and witnesses, and the reliability of the system. The privacy of the witnesses is protected by using a geo-indistinguishability mechanism adapted from the differential privacy mechanism for geo-location systems. It injects noise/error in the reported locations and the number of proof responses is limited to a threshold.

A. Use case

There many examples of use cases for location certification systems, such as ride-hailing, food delivery and vehicle navigation. For example, the well known Uber ridesharing app allows a user to request a ride and pay with a credit/debit card. This app uses two-way LBS technology so the driver knows where to pick up the user, and the user is able to see a live view of the driver's location on a map. Our system can add more credibility to the pick up location of the user by proving its claimed location with the help of other users in nearby. For food delivery is the same idea, this system can help the user to prove its delivery location with the help of other users in nearby locations.

For the evaluation of the privacy mechanism for SureThing, we used a *Ticketless urban mobility* scenario, for public transportation using provable location to enable efficient boarding and accurate billing. The user does not need to explicitly present a ticket at entry or exit. The user just needs to carry a mobile phone and the entries and exits from public transports, like buses, are detected and the user is billed accordingly. The system was simulated with a discrete event simulator tool[1] with source data collected from actual experiments in a bus network on a city with 500 000 inhabitants.

B. Outline

The remainder of the document is structured as follows. In Section 2, we present the background and related work.

[1] AnyLogic simulation software.
Available: https://www.anylogic.com/downloads/

In Section 3, we explain the high-level approach. In Section 4, we describe the system architecture. In Section 5, we present the Ticketless transports scenario and, in Section 6, we evaluate our system from the previous scenario. Finally, in Section 7, we conclude the paper and we discuss future work.

II. Background and Related Work

In this section, we introduce location proofs and the SureThing system, then we present privacy mechanisms, namely, the Differential Privacy and the Geo-Indistinguishability. We also introduce existing privacy-preserving location systems.

A. Location Proof

A location proof (LP) is a claim of the presence of someone at a specific time and place, and with digital evidence that allows the verification of the claim. The location proof contains the prover location and identifier, the witness location and identifier, a random number or timestamp to ensure freshness, and a digital signature to assure the authenticity of data. A location proof system allows proving the location of untrusted mobile devices and a history of locations visited by the user to third-parties applications and services [5].

B. Location Proof Systems

There are several systems that provide locations proofs. We highlight APPLAUS, CREPUSCOLO, and SureThing.

1) APPLAUS: (A Privacy-Preserving LocAtion proof Updating System) allows a device to prove its location by requesting location proofs from nearby mobile devices using Bluetooth. The location proofs are then updated to an untrusted Location Proof Server that verifies the trustworthiness level of each location proof [6]. APPLAUS preserves the privacy of the source location information of mobile devices from each other and the untrusted location proof server by using pseudonyms for the Prover and Witnesses. Every mobile device is registered with the Certificate Authority (CA) that generates a public/private key pair. The public key is used as the pseudonym of the mobile device, and the private key is used to digitally sign messages. The digital certificate validates the authenticity of the keys used for the signatures. The privacy knowledge is separated: the Location Proof Server only knows the pseudonyms and locations, the Verifier only knows the real identity and its authorized locations. The CA only knows the mapping between the real entity and its pseudonyms (public keys) and makes a connection between the Verifier and Location Proof Server. For the attackers to learn the location information of a user, they have to integrate all these sources of information.

APPLAUS includes a user-centric location privacy model with the purpose of each user evaluating their location privacy levels in real-time and then deciding if a location proof exchange request is accepted based on their location privacy levels. When one of the nodes generates a fake location proof, and then colludes with another node, this is called a Collusion attack. These attacks can be detected by using a threshold-based solution or by looking into the location traces. A downside of APPLAUS is that is vulnerable to the wormhole attack. This attack consists of an attacker to record a packet at one location and then tunnels the packet to another location and replays it there.

2) CREPUSCULO: (Collusion resistant and privacy-preserving location verification system collects location proofs from co-located mobile devices) is a system that uses a token from a trusted Token Provider [7]. Token Providers (TPs) are trusted entities, which issue tokens to mobile devices. Having these tokens combined with location proofs will prove that a determined mobile device is at a determined location at that time, to provide resiliency against collusion attacks. All entities in the system have to register with the CA, similar to the one available in APPLAUS, which provides authentication and authorization services. Every entity registered in the system has assigned a pseudonym, and only the CA can link a pseudonym to identity.

The operation of CREPUSCOLO consists of two phases: the acquisition phase and the verification phase. In the location-proof acquisition phase, the mobile devices collect location-proofs and store them in the Location Server (LS). LS is a non-trusted device that provides services to mobile entities, storage their location-proofs and tokens. In the location-proof verification phase, the Verifier (V) uses the information stored in the LS to check if the Prover (P) is at a certain location or a certain historical trace of locations. CREPUSCOLO protects the source location privacy by using pseudonyms and changing them periodically. The mechanism of changing pseudonyms has the pseudonym unlinkability characteristic, which prevents the attackers to identify a set of pseudonyms as belonging to the same identity. For the attackers to learn how to link pseudonyms to their associated identity, they have to compromise the CA, however, the CA is assumed to be trusted.

3) SureThing: is a location proof system for mobile devices which can provide evidence of the presence of a user at a given location relying on witnesses using location estimation techniques, including GPS coordinates, Wi-Fi fingerprinting and Bluetooth beacons [4].

The SureThing system follows the design of APPLAUS and CREPUSCOLO, and has four entities: the Prover that needs to prove its location and asks location proofs from witnesses; The Witness is the entity that agrees to give a location proof to the Prover. There are three types of witnesses, the master, mobile and the self witness. The master is a certified witness that can be trusted by the Verifier. The mobile witness is an untrusted random witness. If there is no witnesses available, the prover can act as a self witness and generates a weak location proof. The Verifier validates the proof of the Prover and informs it. It is up to the Verifier to define the acceptance criteria of a proof, depending on the application needs and required trust level for location data.

The CA in SureThing is assumed to be trustful and is responsible for generating a public key certificate for each user. We assume that each user of the system has to have a unique identifier and it has its own public and private keys.

When a location proof is requested, SureThing operates in the following way: the Prover asks to the Verifier how it should obtain a location proof; the Verifier replies with a Proof Demand that specifies how the Prover and the Witness should obtain their location evidence; afterwards, the Prover sends a Proof Request to the witnesses nearby, this request contains the identification of the Prover and the demand previously received; the Witness generates the location proof and returns to the Prover; this location proof is signed with the private key of the Witness; then, the Prover forwards the location proof to the Verifier to be verified; the location proof contains a prover identifier and location, witness identifier and location, a signature from CA for authenticity of the proof, and a token (i.e., random number and/or timestamp) to ensure freshness; the Verifier needs to check the signature in the location proof, so it requests to the CA the public key of the Witness; after verifying the location proof, the Verifier decides to accept or reject it.

SureThing uses witness redundancy and decay mechanisms to avoid collusion attacks. For the system to ensure redundancy protection, the location proofs have to be collected from multiple witnesses instead of one. Also, the same witnesses cannot be used too many times. This is achieved by decreasing the value of the proofs from the same witnesses. If an attacker wants to deceive the system, he will have to collude with many false witnesses. Given this reliance on many and fresh witnesses, SureThing is most effective for crowded locations, where a user can obtain location proofs with a diversity of witnesses. In its earlier versions, SureThing lacked privacy protection for its witnesses.

C. Differential Privacy

Differential Privacy is a privacy protection mechanism. Its principle is to quantify and limit the maximum possible information gain by the attacker, as a way to reduce the risk of the privacy being compromised [8]. Differential Privacy consists of analyzing and sharing information with individual privacy protection according to the existing policy or legal requirements for disclosure limitation or de-identification. This mechanism guarantees that anyone observing a set of differential private analyses will make the same inference about any private information of the individual, whether or not that private information of the individual is included in the input to the analysis.

The private information is limited and quantified by a *privacy loss* parameter, usually designated epsilon ϵ. This parameter quantifies the maximum possible information gain by the attacker and determines how much noise needs to be introduced during the differential private computation. Using a smaller value of ϵ results in stronger privacy protection but less accuracy due to the deviation between the real analysis and each approximation output computed scenario.

Differential Privacy protects against a wide range of potential privacy attacks, including unknown attacks at the time of deployment. In a set of individuals, their data will be differential private even when multiple analyses are performed on that data, as long as each of the analyses satisfies differential privacy. Releasing too many accurate statistics will result in a considerable privacy loss. To avoid this, the number of analysis performed on a specific dataset is limited while providing an acceptable guarantee of privacy. An example supposes that differential private data was given to Alice and Bob. The privacy loss parameter of 1ϵ is used every time. If Alice and Bob decide to collude, the resulting data is still protected, only the privacy will be weaker, i.e., the privacy loss parameter will become 2ϵ. They will gain some data, but you still quantify how much information they can get, this is a property of the composition. The composition is a method to stay in control of the level of risk as new use cases appear and processes evolve. The more the information is intended to be queried, the more noise has to be introduced in order to minimize the privacy leakage. Once the data has been leaked, it will no longer keep the information of the users private, this means that there can be limits on the number of queries answered by a user.

Differential privacy techniques can be applied to geographic location data.

D. Geo-Indistinguishability

The application of Differential Privacy to geographic location data is called Geo-Indistinguishability. It is a user-centric Location Privacy-Preserving Mechanism (LPPM) that limits and quantifies the information gain by the attacker observing the reports with location data between users. Geo-Indistinguishability protects the location of the users reporting their location and guarantees that any two locations within a given radius around the user are statistically indistinguishable.

Cunha et al. [9] propose a new mechanism that can be used to report location data, sporadically or continuously, called Clustering Geo Indistinguishability. This mechanism considers two important factors, the frequency of updates and the distance between the reported locations. It generates obfuscation clusters for closer locations, and the same obfuscated point is reported to nearby locations. The frequency of reported locations can compromise the privacy of the user, since the attackers can correlate the information of the reports to attack. Clustering Geo-Indistinguishability is based on Planar Laplace (PL) Geo Indistinguishability mechanism for sporadic scenarios and Adaptive Geo Indistinguishability mechanism for continuous scenarios. PL geo-indistinguishability consists of adding 2-dimensional Laplacian noise centered at the exact user location x and reporting it as an obfuscated location. Adaptive geo-indistinguishability is a combination of PL and a computed variable ϵ correlated between the past locations and the new location. With this variable ϵ, the adaptive mechanism can adjust the amount of noise necessary to obfuscate the exact user location.

The correlation ϵ is the error between the exact location and an estimation obtained with a simple linear regression. If the correlation between reports is low, the mechanism increases the privacy level, this means the attacker observing the report of the location will have less probability of knowing the real location. And if the correlation between the reports is high, the mechanism decreases the privacy level.

E. Privacy-preserving location systems

We have presented some location proof systems and privacy mechanisms for location data. In this section we present systems that use location and protect the user privacy: Icelus, MATRIX and Olteanu's framework.

1) Icelus: is a privacy-preserving location proof system that allows estimating the user location and modeling the user movement by combining multiple observations from multiple devices [10]. The user can spoof its location by using only their smartphone to prove its location, Icelus takes leverage of the increasing number of Internet of Things (IoT) devices used by users and those smart environments to locate them. The Icelus system organizes the IoT devices in a hierarchy. On top of the hierarchy is the hub, which hosts the Icelus service. The Hub receives information from different sources to avoid the data be seen by third-parties, considering that only the user controls the Hub. Those different sources that send information to the Hub are smartphones and smartwatches, and Beacons, i.e. third-party devices that observe devices of the users. The attacks that Icelus is designed to prevent are the attempts to bypass user authentication with physical devices and terminals to gain unauthorized access to locations, properties of the user or from third-parties. These types of attacks can compromise passwords, biometrics, or security tokens, such as smart cards and swipe cards. Icelus do not assume that devices of the users have not been compromised, but that they can be physically stolen, tampered or even remotely compromised.

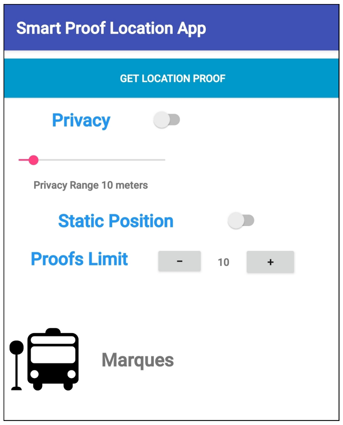
Using the Hub to get all geolocation information can bring privacy concerns for the users or third-parties. For Icelus to resolve this issue, the Hub and the Site can only learn the distances between the reported locations, and not the precise coordinates. Sites are entities that query the Hub about the possibility of the user to be physically present at a location. Icelus relies on proofs indicating that the user is not at a determined location. Instead of operating with the precise location of devices, it operates on their distances.

All the location reports arrive at the Hub are encrypted. Icelus uses Homomorphic Encryption (HE) to determine the distances without knowing the precise location of the devices. HE is a method of encryption that allows any data to remain encrypted while it is being processed and manipulated. This type of encryption is suitable for arithmetic computations, more specific, for Euclidean distances. To calculate their relative positions, the Hub needs to have a pairwise distance between three points.

2) MATRIX: Narai [11] proposes a system, called MATRIX, to allow end-users to control visibility of location and sensor accesses by mobile applications. This system implements a PrivoScope service with an user interface that verifies all locations and sensor accesses by mobile applications and gives real-time notifications, helping the users to make privacy aware decisions for the installed apps. And it uses a Synthetic Location service for users to provide obfuscated or synthetic location trajectories or sensor traces to mobile applications. Also, MATRIX implements a Location Provider that generates realistic privacy-preserving synthetic identities and trajectories for users by using traffic information from historical data of Google Maps Directions API, and accelerations from user driving experiments. A synthetic identity is a unique virtual identity for each mobile device user, and each one has a unique movement pattern. A synthetic identity does not have any specific attributes of the user location. These trajectories ensure location privacy because they are independent of users real locations, although, if the adversary detects that the trajectories are fake, the service is denied. The adversary is a mobile application that uses the location information of the user. The MATRIX protects against tracing attacks. To guarantee that the adversary does not detect that the trajectories are synthetic, they must emulate real movements by using routines of the users, their schedules, traffic information and driving behavior. These synthetic trajectories are important because they permit to reduce the privacy leaks and to understand how the user's location information is exploited by mobile applications.

3) Olteanu's co-location privacy framework: Most of the online social network providers, such as Facebook, give to the users the functionality of sharing their location jointly with their photos and posts. They also provide the ability to mention other users, to tag them on photos or in posts. In such cases, that information indicates that the users mentioned in a post or photo are co-located. Sharing this information brings social benefits but also location privacy concerns, for both the user who shared the information and for the tagged user.

Olteanu et al. [12] propose a framework to allow two users to make decisions about posting co-location information. Co-location information is location information that involves information from other users, i.e., there is a dependence between the users. This framework models the direct and indirect benefits, and the privacy concerns of location and co-location sharing, and permits the analysis of behaviour of users of sharing the location and co-location. This is important because can compromise the privacy of the users, the co-location information is related with all involved users. At any moment, an adversary, such as the service provider or the friends of these two users, has access to reported locations and co-locations. This framework is based on game theory and conjoint analysis. Game theory allows us to model and formalize the sharing behaviour of the users preferences. Conjoint analysis allows



Privacy

Privacy Range 10 meters

Static Position

Proofs Limit 10

Marques

Fig. 1: Ticketless transport mobile application screen design.

us to determine the benefits of sharing location and co-location information, and the associated location privacy concerns.

The authors conclude that because of conflicting preferences, one of the users can be forced into a situation that it does not desire, and also sharing co-location information can additionally encourage users to over-share their locations.

F. Summary

In this section we presented location proof systems, privacy mechanisms and their use. The discussed works are relevant for the privacy protections added to the SureThing system. Icelus illustrates how the use of distance instead of coordinates can still provide relative location. MATRIX shows the value of synthesized location and trajectory data to protect user's privacy. Olteanu's work shows that co-location, such as the one that happens when a witness testifies for a location proof, must be a core concern.

III. High-level approach

Our system extends a previous version of SureThing with important mechanisms to address the location privacy protection of the users. It allows the users to protect their privacy when they are helping other users, with their location data, to prove its claimed location. The system has two main functionalities to ensure the privacy of the location of the users. The geo-indistinguishability mechanism to inject noise in the location data of the user when is shared with other users to obfuscate its real location. The other functionality is to limit the number of replies to location proof requests from other users, this helps to limit the information shared because of the Differential Privacy principle, the more information is shared, the more noise needs to be injected. Also, if the witness is in the same area, it can respond to other users with a static position, to avoid to send other location of the same area. This functionality can reduce the leak of location information of the witness.

In terms of security, we want to secure the privacy of the users and specially the witnesses that help other users. For a malicious user interested in attacking the location privacy of other users, it has to be a user registered in the system. We consider an attacker, a malicious prover that requests location proofs from other users, i.e., witnesses to collect accurate location information of them. This attack can be proof stealing, i.e., stealing and using location proofs from other users. The malicious prover could want to obtain the identity of the witnesses by requesting location proofs to the witnesses. Also, we consider an external attacker from the system with the goal to break the anonymity of the users in the system. This attacker can intercept the communications between the prover and witnesses.

This system should reduce the leakage of location information of the witnesses by injecting noise in the location data and limiting the number of location proof responses. We assume that the communications between users are encrypted and the external attackers cannot break the encryption. Also, we assume that every user has a certificate and there is validation of the certificate using the CA, however to simplify the implementation and testing of this work, we do not use the certificates. In this work, we focus only in the location privacy protection, but the protection of the identity of the users is planned for future work.

In our system, the noise and the limit of replies, are presented to the users as the configuration of the *level of privacy*. They have the willingness to share resources and a risk appetite. This allows us to quantify how much noise they want to put in their location proofs and how many locations proofs they want to report to other users. The user does not have detailed control in the parameters, but can choose one of the available levels of privacy. Figure 1 illustrates the prototype of the ticketless transport mobile application, based on SureThing. This prototype have all the functionalities mentioned before. The user as a witness can activate or not the privacy mechanisms. It can activate or not the static position. And it can decide the number of location proof responses and the privacy range average distance. Pressing the blue button, it will request a location proof. In this case, the user location is at "Marques" bus stop.

The system has a fundamental trade-off between the levels of privacy protection and usability and accuracy of this system. We want a system capable of protecting the privacy of the users and at the same time, keep the system accurate and usable. Also, we want to evaluate the trade-off between few witnesses without privacy protection and more witnesses but less accurate due to privacy protection. This is meaningful for different use cases, where there can be a variable number of witnesses.

IV. Architecture

Figure 2 presents the main classes that are used by both the Prover and the Witness. Starting from the top, the Proof class is abstract to represent the Location Proof. A Geo Proof contains two Geo Location objects. One is for the location of the Witness and the other is for the Prover. Each Geo Location is defined by geographical coordinates, such as latitude and longitude, and accuracy. With this information, the area is

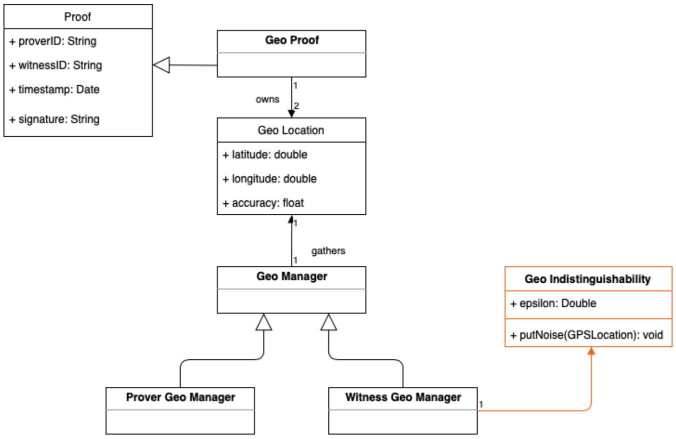


Fig. 2: Class diagram with the main classes used by both Prover and Witness.

divided in circles and the Verifier can check if the Prover and the Witness are inside of the same area. The Manager class is responsible for collecting geo location data. And it is divided in two because Prover and Witness will collect location data for different goals. The Prover Manager is to inform the Witness about the location of the Prover. so the Witness can add it to the proof. The Witness Manager obtain the location data and it will create a proof to send to the Prover. They share the same Manager because the process to obtain the location is the same. We extended SureThing with privacy protection for the location of the witnesses.

We created a new module called Privacy in the mobile application, we implement the geo-indistinguishability mechanism to increase the privacy of the location of the witnesses when reporting location proofs to other users. The geo-indistinguishability class is highlighted in Figure 2. The epsilon is the noise parameter that controls the quantity of noise is introduce in the location data. The function that introduces noise, it uses the Planar Laplace mechanism on the mobile application in order to report to the Prover the obfuscated location of the Witness rather than its real location.

A. Prover

In the beginning of the process of proving the user location, the prover will send a Proof Demand request to the Verifier to know what type of proof that the Verifier wants to be generated. Then, it will receive the Proof Demand with the following information: the witness model that is going to be used when gathering proofs; the proof technique that is going to be used by the Prover and Witness when collecting proofs, in our system it will be Geo proofs; a nonce to avoid replay attacks; and the number of witnesses for the collusion avoidance mechanism. With this information, the Prover starts the witness discovery process [13]. After a witness is detected, the Prover will send to it a Proof Request, that contains the proof technique and the nonce from the Proof Demand, and the identifier and the location data of the Prover. Then, the Prover will receive the location proofs necessary and it will send all at the same time to the Verifier.

B. Witness

When a Witness receives a location proof request from the Prover, the Witness can accept or reject the request according to the limit number of location proof responses sent to other users. If accepts it, the Witness will determine its location data, by using the proof technique in the proof request received from the Prover, in our system will be only the Geo technique. This technique collects geographic location information from the GPS receiver. Then, the Witness to protect its location privacy, it can define how much noise wants to inject in the location data. Then, the Witness signs the location proof with his private key to guarantee integrity and non-repudiation in the exchange of the proof. The proof is replied by the Witness to the Prover with the Prover ID, Witness ID, location data of the Prover, location data of the Witness, nonce and signature of the proof.

C. Verifier

To validate the proof of the Prover, the Verifier has to perform the validation of the digital signature of the proof made by the Witness, and must check if the nonce in the proof is the same that was sent to the Prover. After this, the Verifier will check the type of proof technique used to create the location proof. The location proofs were obtained by the Geo technique, the Verifier will make a comparison between the prover and witnesses locations to determine if their distance is smaller than the threshold defined by the developer, this threshold is the maximum valid distance between the prover and the witness. Then, the Verifier has to calculate the midpoint of all the location proofs of the witnesses, since the coordinates are close to each other, we can treat the Earth as being locally flat and simply find midpoint as thought they were planar coordinates. So, the Verifier calculates the average of the latitudes and the average of the longitudes to find the latitude and longitude of the midpoint. After the calculation of the midpoint, the Verifier will check if the midpoint is in the area previously defined, it will accept the proof, otherwise, will reject it. The threshold of that area should be adapted for the specific use. For wide areas, a high threshold would be acceptable, because the user is probably still inside that area. For small areas, the threshold should be lower.

V. Ticketless transports use case

In the last years, the population in urban areas has been increasing and it has tendency to increase more sharply. Consequently, the number of users of public transports has been increasing too. According to an International Association of Public Transport - UITP, currently 64% of the trips done in the world happened in urban areas. It is assumed that until 2050, the number of people moving per kilometer in urban environment triples [14]. A system of transport have an important paper in the large metropolitan areas, therefore is necessary to guarantee the best conditions of this service to offer better experience to the users.

The ticket services, consequently, have to be updated in order to speed up the process of acquiring and validating of tickets, reducing the waiting lines and saving time to the passengers. Initially, the tickets were made of paper, through the years, was evolving to systems with electronic support. Nowadays, there are proximity cards for electronic tickets, allows to save the tickets and offers guarantees of security to the operator as well to the users, and the users can acquire more than one ticket and save them in the same card. However, to acquire the tickets is necessary to go to a ticket office or to an automatic ticket machine and can cause waiting lines for the users.

Paradela [15] proposed a public transport system using a smartphone with NFC in smartphones to replace the proximity cards, in order to simplify the process of selling tickets. In this scenario, the smartphones are used to buy tickets through a mobile application, to save the ticket and to access the public transport service. In the validation, the smartphone works similar to the proximity cards, to validate the ticket it is necessary to approximate the smartphone to the validator. This is one example of the trend to use personal devices, like the smartphone, as the ticket for the public transport.

Considering these systems and the different ticket services, we propose a new system for public transports, a Ticketless transport system for public transportation using provable location to enable efficient boarding and accurate billing. The operator with this system encourage spontaneous use of transportation without being necessary to approximate from a validator, and can dematerialize the tickets, instead its used virtual tickets.

Most of the works are focused on the operator perspective to have a better system with reduced costs, and in this work we are focused on the user perspective and its use of location proofs. The user community is very important in a crowd sensing system, according to the survey results [16]. Crowd sensing is a data collection and sharing performed by a large number of regular users [17]. Calado [16] defines the user as a person that uses its Internet-connected smartphone to capture and share information, and defines community as a group of people that have a shared goal and that join together to share information related with the goal. This a cross-checking system, to make it more transparent, and involve the community to share location information with the user that pretends to prove its location.

The idea of this system is to allow public transport users to do small trips, without being necessary to pay a monthly pass or a bus ticket. Using a mobile application, the user can travel and the service charges only the route that it traveled. When the system detects the entry of the user in the bus, it starts to determine the route of the user that is doing through location proofs with witnesses near to the user. When the user leaves the bus, the system stop detecting the user and charges price of the trip. In the perspective of the user, it has interest in this system, it gives a flexible tariff in the public transports, there is no need for buying tickets in physical places, eliminating the waiting lines. And the user with the collected location proofs from the community can use it as complaint mechanism, in case the operator charges the user more than it should or for other possible scenarios.

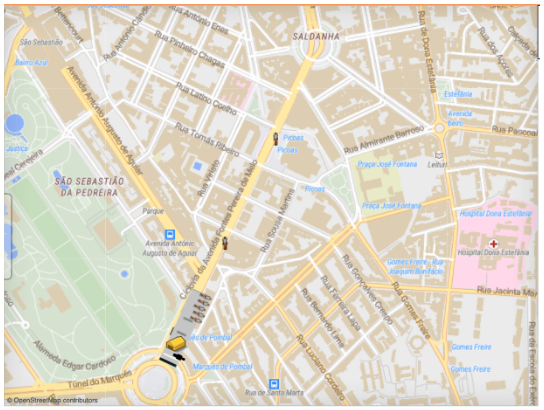


Fig. 3: The simulation route of the bus from Anylogic.

VI. Evaluation

Now, that we understand the system, how it works and what is the motivation, we have to evaluate the system to know if it is feasible and practical. For this purpose, we use the ticketless transport use case. Regarding the privacy concerns of the users, what is the proof acceptance rate considering that the users are using privacy protection on their location. And what type of attacks or adversaries this system can protect against the users.

We have a dataset of real GPS coordinates of smartphones during a bus trip between the Marquês de Pombal stop and the Campo Grande stop in Lisbon, collected in previous work by Santos [18]. We evaluate our system by simulating and analyzing the users using the Ticketless transport system in different experimental scenarios. The simulation was developed using AnyLogic software with a creation of a model to represent the real system. This model considers only the important details, therefore the model will be less complex than the original system. Anylogic is a multi-method simulation modeling, it develops simulation models using discrete events, agent-based and system dynamics. We choose Anylogic because it has GIS maps integration, i.e., it provides GIS maps in the simulation models. The elements of the simulation model can be placed on the map and can move from one point to another through existing roads and routes based on real spatial data. Anylogic has a built-in search, similar to Google Maps, that allows us to easily locate streets, roads, shops, and bus stops. This helps us to simulate a real route with GPS coordinates.

In our simulation, we create a model that represent the route of a public bus in the center of the city of Lisbon in Portugal. The route of the bus has four bus stops, it starts at the bus stop of Marquês de Pombal, and then goes to bus stop of Av. Fontes Pereira de Melo, Picoas and it ends at the bus stop of Saldanha. Figure 3 presents the route map of the bus, we decide this route because of the dataset available from Santos [18]. In the beginning of the simulation, the bus starts to do the route and the users will start appearing in the bus stops

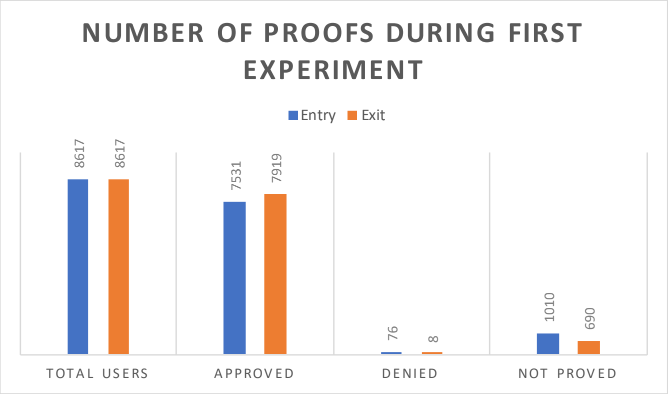


Fig. 4: Acceptance rate of proofs during the best case scenario.

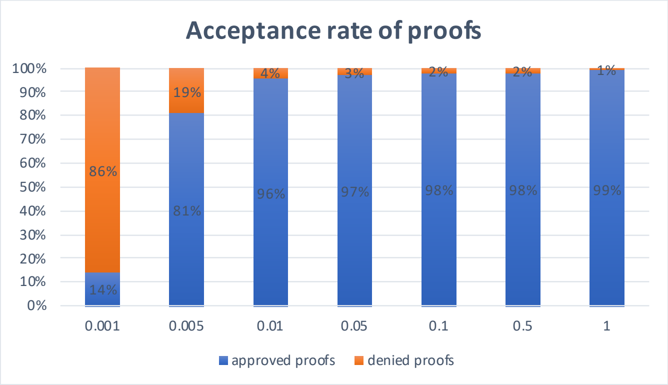


Fig. 5: Acceptance rate of proofs with different noise parameters.

waiting for the bus to pick them up. When the user enters the bus, it validates the beginning of the trip by proving it through location proofs. Then, when the user leaves the bus, it validates the end of the trip through location proofs.

In this simulator, the users to prove their location and to reply to location proof requests, they execute the adapted code from the SureThing project to simulate the system. For the privacy mechanism, we imported the Privacy module of the SureThing code, the Geo-indistinguishability class that permits to inject noise in the GPS coordinates of the witness. The location proof in the simulator is adapted from the SureThing code and only has the necessary information, it has witness location, prover location and the timestamp.

A. *Simulation setup*

For all the experiments that we are going to evaluate, we have to define parameters to setup the simulator. Each experiment has the duration of one hour due to the personal learning edition of Anylogic that allows only one hour of simulation. We define the capacity of the bus as the same number of a regular public bus, that is approximately 80 users. Every user has a unique name and surname to easily identify, and a Boolean parameter to know if the user has the mobile application or not to participate in the system. When the user goes to one of the stops of the route to wait for the bus, the user have a parameter for the destination stop determined by the probability of one of the stops. To verify in which bus stop the user is, we defined the threshold of the bus stops to 100 (one hundred) meters. It is high enough to be accurate to prove that is in that bus stop and not in other bus stop, since the distances between the bus stops is higher than 250 (two hundred and fifty) meters. The speed parameter of the bus is 30 (thirty) kilometers per hour, and the time that the bus waits in each stop it is between 1 (one) minute to 3 (three) minutes.

B. *Best-case scenario*

First, we evaluate the best case scenario, where all the users have this mobile application and none of them is using privacy protection. None of the users is using any privacy mechanism in this scenario, all the witnesses are sending their real location. This scenario evaluates if the system is feasible and practical for the users. These measures were obtained after 30 experiments. The average of users using the bus during one hour in each experiment is 290 users. Figure 4 represents the acceptance rate of proofs of the users during the entry and exit of the bus. We can observe that most of the claimed locations were approved during the experiments at the entry and at the exit of the bus. This is the best scenario possible, however, there were claimed locations denied, specially at the entry of the bus. Also, a significant number of claimed locations were not able to prove their location. A possible reason for these numbers is because of the simulator, when the users enter the bus to prove their location, they do it one at a time, the simulator cannot do multi-threading in this process.

C. *Geo-indistinguishability*

For the second experiment, we studied the effect of geo-indistinguishability error in the location of the witnesses to evaluate the proof acceptance rate. The witnesses will use different values of error to protect their location privacy.

We tested the following noise parameters: 1, 0.5, 0.1, 0.05, 0.01, 0.005 and 0.001, and for each noise parameter we ran 10 experiments and all the users have the same noise parameter. We wanted to compare if, by using more noise, we could have better location privacy for the witnesses while the system is usable. Our results are presented in the Figure 5. The noise parameter can variate between 0 and 1, and we can observe that when the noise parameter tends to 0, the location is more private. Setting the noise parameter to 1, it means there is no noise in the location data, that is why there is a high percentage of approved proofs. It is expected that decreasing the noise parameter, the percentage of approved proofs will decrease too. All the noise parameters that are equal and higher than 0.01 have high percentage of approved proofs, because the average distance between the bus stop and the witness location is smaller than threshold of the bus stop. Table I presents the average distance between the bus stop location and the

noise parameter	average distance (meters)
1	3.80
0.5	4.89
0.1	5.12
0.05	9.99
0.01	35.44
0.005	66.89
0.001	253.90

TABLE I: The noise parameters and the corresponding average distances between the witness and the bus stop.

witness position with the respective noise parameter. We can observe that the number of denied proofs start to increase when the average distance between the bus stop and the witness location gets closer to the threshold of the bus stop. Using the noise parameter 0.005, the percentage of denied proofs is 19% because the average distance is 66.89 meters that is very closer of the 100 meters of the threshold of the bus. Reducing the noise parameter to 0.001, the percentage of denied proofs is higher than the percentage of approved proofs because the average distance between the witness and the bus stop is higher than the threshold of the bus stop. Using this noise parameter will protect more the privacy of the witness but becomes useless for the system to verify the claimed location of the prover. So, if a witness wants to protect its privacy as maximum as possible, it should use lower values for the noise parameter. But if a witness wants to help other users and at the same time wants to protect its privacy, it should use values close to 0.005. The witness has full control of their privacy.

D. *Response throttling*

After studying the effect of geo-indistinguishability error, we studied the effect of response count, i.e., the number of replies to ad-hoc witness requests. We will use the noise parameter 0.005 tested in the previous simulation because it gives some relevant level of privacy and it has a good percentage of accepted proofs. Every user in this simulation will use the same noise parameter and the same limit of proof responses. During the previous simulations, we recorded the number of proof responses of all witnesses, and the average number of proof responses is 76 per each witness. The limit of proof responses will change in each experiment, first we will test the average number of proof responses and then, we will increase and decrease 50% of the average number to see the results of that effect. For each limit value of proof responses we will run 10 experiments for added statistical confidence in the results.

Figure 6 presents the results of testing different limits of proof responses from the witnesses and the effect on the acceptance rate of claimed proofs. We can observe that between the different limits of proof responses, there is a small variation of the acceptance rate of the claimed locations. The result shows that limiting the number of proof responses from the witnesses, it will decrease the percentage of approved proofs, but it will reduce the leakage of location data of the witness. Reducing 50% of the average number of proof responses, the difference of the results was 5% less of approved claimed locations, which has not big impact on the acceptance rate but has a good impact in protecting the privacy of the witnesses. Increasing 50% of the average number of proof responses, the percentage of approved claimed locations increased 7%, but the witnesses have more exposure of their location data. Also, we can observe from Table II that using the same noise parameter, and reducing the maximum of proof responses from the witnesses, it will result on a bigger distance between the witnesses and the bus stop, with less witnesses available it is less accurate to prove the claimed location.

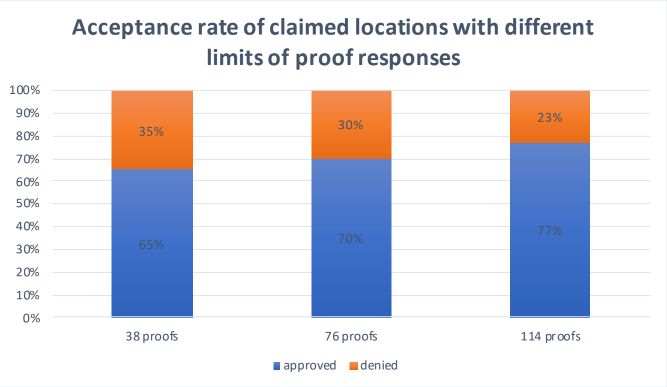


Fig. 6: Acceptance rate of proofs with different proof response limits.

limit of proof responses	average distance (meters)
38	76.85
76	76.40
114	68.52

TABLE II: The limit of proof responses and the corresponding average distances between the witnesses and the bus stop.

E. *Attack resistance*

We evaluate the defenses of the system against a malicious prover that intends to obtain the real location of a witness. The witnesses have privacy protection using the noise parameter 0.005 and a limit of 76 location proof responses, this values ensure a good privacy protection and, at the same time, the usability of the system. We create an attacker, i.e., a malicious prover, a user that is in the bus trying to collect the maximum possible information about the real location of the witnesses. The attacker will not leave the bus during the simulation, and it will try to gather location proofs from the witnesses. In this simulation, we assume that the attacker has prior knowledge about user's location. Because of the publicly available transportation information and road networks, the attacker knows there is a high probability that the target user is on the bus and it will try to know the path of the target user through the location proofs. We define the target users of the attacker, the users that entry on the bus in the first station and leave the bus in the last station. We decided the longest distance to evaluate how much information knows about the path of the target user.

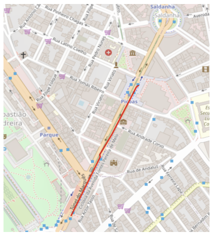


Fig. 7: Trace collected with high number of available proof responses from the target.

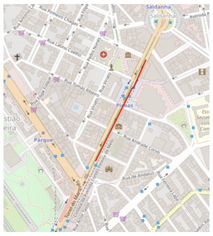


Fig. 8: Trace collected with medium number of available proof responses from the target.

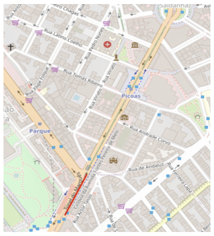


Fig. 9: Trace collected with low number of available proof responses from the target.

Our results are presented in Figures 7, 8 and 9. We can observe that the attacker could collect location information about the target users. In the figure 7, the attacker collected a high number of location proofs from the target user, which allows it to trace a significant path of the user. In this situation, the bus was almost empty and the target user had high availability to respond location proofs. In the figure 8, the attacker collected less number of location proofs compared to the previous one, because the bus had more users traveling and requesting location proofs, which reduce the availability of the target user to respond location proofs to the attacker. With the bus almost full, the target user had a very low availability to respond location proofs to the attacker, for this reason, the number of location proofs collected by the attacker is very low, the result is presented in the figure 9.

VII. Conclusion

In this paper we presented a privacy-preserving extension of SureThing for witness protection by using privacy mechanisms to protect the location data of the witnesses. The Geo-Indistinguishability mechanism protects the location privacy of the witness by injecting noise in the location data quantified by the noise parameter. We also proposed response throttling to limit the number of location proof responses by the witnesses. The witnesses have willingness to share resources and some risk appetite, so they can select a personal privacy setting in their mobile application that will quantify how much noise they want to put in their location proofs and how many locations proofs they want to report to other users. Our system has shown, through simulations, that it is practical for ticketless transport in a city, with a 70% proof acceptance rate, as long the noise parameter of each user is 0.005 and the limit of number of location proof responses is 76.

References

[1] K. W. Kolodziej and J. Hjelm, "Local positioning systems: Lbs applications and services," in *CRC press*, 2017.

[2] M. H. Yılmaz and H. Arslan, "A survey: Spoofing attacks in physical layer security," in *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*. IEEE, 2015, pp. 812–817.

[3] K. Jansen, M. Schäfer, D. Moser, V. Lenders, C. Pöpper, and J. Schmitt, "Crowd-gps-sec: Leveraging crowdsourcing to detect and localize gps spoofing attacks," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 1018–1031.

[4] J. ao 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.

[5] S. Saroiu and A. Wolman, "Enabling new mobile applications with location proofs," in *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, 2009, pp. 1–6.

[6] Z. Zhu and G. Cao, "APPLAUS: A Privacy-Preserving Location Proof Updating System for Location-based Services," in *IEEE INFOCOM*, 2011.

[7] E. S. Canlar, "CREPUSCOLO: a Collusion Resistant Privacy Preserving Location Verification System," in *International Conference on Risks and Security of Internet and Systems (CRiSIS)*, 2013.

[8] K. Nissim, T. Steinke, A. Wood, M. Altman, A. Bembenek, M. Bun, M. Gaboardi, D. R. O'Brien, and S. Vadhan, "Differential privacy: A primer for a non-technical audience," in *Privacy Law Scholars Conf*, vol. 3, 2017.

[9] M. Cunha, R. Mendes, and J. P. Vilela, "Clustering geo-indistinguishability for privacy of continuous location traces," in *2019 4th International Conference on Computing, Communications and Security (ICCCS)*, Oct 2019, pp. 1–8.

[10] I. Agadakos, P. Hallgren, D. Damopoulos, A. Sabelfeld, and G. Portokalidis, "Location-enhanced authentication using the iot: Because you cannot be in two places at once," in *Proceedings of the 32Nd Annual Conference on Computer Security Applications*, ser. ACSAC '16. New York, NY, USA: ACM, 2016, pp. 251–264. [Online]. Available: http://doi.acm.org/10.1145/2991079.2991090

[11] S. Narain and G. Noubir, "Mitigating location privacy attacks on mobile devices using dynamic app sandboxing," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 2, pp. 66–87, 2019.

[12] A.-M. Olteanu, M. Humbert, K. Huguenin, and J.-P. Hubaux, "The (co-)location sharing game," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 2, pp. 5–25, 2019.

[13] J. Ferreira, "Surething: User device location certification," Master's thesis, Instituto Superior Técnico, Lisbon, 2017.

[14] "The global public transport awards 2015."

[15] R. Paradela and M. Pardal, "Emulação de título de transporte seguro em telemóvel android," Master's thesis, Instituto Superior Técnico, Lisbon, 2015.

[16] D. Calado and M. L. Pardal, "Tamper-proof incentive scheme for mobile crowdsensing systems," in *17th IEEE International Symposium on Network Computing and Applications (NCA)*, November 2018.

[17] B. Guo, Z. Yu, X. Zhou, and D. Zhang, "From participatory sensing to mobile crowd sensing," in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*. IEEE, 2014, pp. 593–598.

[18] M. L. P. Henrique F. Santos, "Operation STOP: secure itinerary verification for smart vehicle inspections," in *INForum*, Guimarães, Portugal, Sep. 2019.