A Witness Protection for a Privacy-preserving Location Proof System

João Paulo Nunes da Costa

Thesis to obtain the Master of Science Degree in

Information Systems and Software Engineering

Supervisor(s): Prof. Dr. Miguel Filipe Leitão Pardal

Examination Committee

Chairperson: Prof. Dr. Luís Manuel Antunes Veiga
Supervisor: Prof. Dr. Miguel Filipe Leitão Pardal
Member of the Committee: Prof. Dr. Alberto Manuel Ramos da Cunha

November 2020

This dissertation is dedicated to the memory of my mom who encouraged me to pursue my dreams and helped me in all things great and small. She was unable to see my graduation. This is for her.

Acknowledgments

I would like to start by thanking my supervisor, Professor Miguel Pardal, and the SureThing team. This last year was very difficult for me and all help, guidance, support, and comprehension from Professor Miguel were essential throughout this work. I want also to thank all the friends with whom I spent my university years. I enjoyed all the experiences and adventures that we had together. I am grateful for all the love and support of my girlfriend, Alexandra, throughout this journey. Finally, I would like to thank my family for their support throughout the years, and for giving me everything that I needed to complete this journey. To my father, João, thank you for showing me that life is not always easy. To my sister Inês and my brother Gonçalo, thank you for your support. Also thanks to my father-in-law Jorge that helped me with everything I needed. And finally, the one person who had made this all possible was my mom Isabel Nunes. She was a constant source of support and encouragement and had made an untold number of sacrifices for the entire family, and specifically for me to continue my schooling. She was and still is a great inspiration to me. I am so grateful and I am sure that without her I would never have reached this degree. Thank you.

Resumo

A presença de smartphones no dia-a-dia mudou a experiência e as expectativas dos utilizadores. As aplicações móveis são capazes de determinar a localização continuamente e apresentá-la aos utilizadores, mas essa informação é vulnerável a ataques de falsificação localização. Uma das abordagens para prevenir a falsificação de localização é recolher leituras únicas de sensores num determinado local e num intervalo de tempo específico e, mais tarde, responder aos desafios de verificação e comparar com as leituras feitas por um grande número de testemunhas ad-hoc. Este trabalho propõe uma proteção à privacidade das testemunhas no sistema de verificação de localização SureThing. SureThing usa uma combinação de Wi-Fi, Bluetooth e outras fontes de sinais e testemunhas ad-hoc num certo intervalo de tempo e num local específico para produzir uma prova de localização verificável. O trabalho foi avaliado com simulações detalhadas para um caso de uso de transportes públicos sem uso de bilhetes físicos, usando um conjunto de dados recolhidos em transportes públicos reais numa cidade. Os resultados da simulação mostram que o sistema é viável e permite que as provas sejam emitidas com sucesso e validadas com proteção de privacidade adequada em 70% dos pedidos feitos num autocarro com lotação elevada.

Palavras-chave: Serviços baseados em localização, Privacidade, Privacidade diferencial, Indistinguibilidade geográfica, Internet das coisas

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 witness privacy protection for 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 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.

Keywords: Location-based Services, Privacy, Differential Privacy, Geographic Indistinguishability, Internet of Things

Contents

List of Tables

List of Figures

Chapter 1

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, city trekking, 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 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 a higher power to the receiver, so that 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 aircraft and Unmanned Aerial Vehicles (UAVs) are 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. Another example of location spoofing occurs in one of the fastest-growing industries: autonomous vehicles [4]. Self-driving cars are expected to be cost-efficient and provide more safety compared to cars operated by humans, one of the most critical devices of an autonomous vehicle is GPS. The location spoofing attacks have a high risk in these systems, as the attacker can mislead a vehicle using incorrect GPS signals. These attacks are very hard to prevent by the owner of the vehicle.

As the examples above illustrated, 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 [5]. 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 is exactly what we propose in this work.

1.1 Contributions

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.

For the evaluation of the privacy mechanism for SureThing, we used a *Ticketless public transportation* scenario. 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.

1.2 Dissertation Outline

The remainder of the document is structured as follows: Chapter 2 presents the background and related work; Chapter 3 describes the design and the implementation of the system, including the architecture; Chapter 4 presents the use case scenario; Chapter 5 presents the evaluation performed to verify the design and system architecture; Finally, Chapter 6 presents our conclusion and future work.

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

Chapter 2

Background

In this chapter, we present relevant definitions and related work. We start with privacy protection techniques and then discuss location proof systems, exemplified by SureThing [5]. We conclude with a detailed analysis of existing privacy-preserving location proof systems.

2.1 Privacy-preserving mechanisms

Privacy is informally the protection of information about you that you do not want others to know about. A privacy-preserving mechanism preserves the privacy of shared data, e.g. location proofs that include the identity and location of users and witnesses in a location proof system. Next, we describe definitions and metrics and then the following privacy-preserving mechanisms: Dummy Location method, Cloaking, Mix zones, Zero-Knowledge Proofs, Differential Privacy, and Geo-Indistinguishability.

2.1.1 Definitions

The term *anonymization* represents the fact that an individual is not uniquely described inside of a group of individuals [6]. The concept of an individual refers to an active entity, such as a person or a computer. A group of individuals can be a group of persons or a computer network. A transaction or registration is considered anonymous when the data, individually or combined with other data, cannot be associated with a particular individual. Data anonymization is designed to make it more difficult to identify a particular individual from a set of related data, the purpose of it is to protect the privacy of the individual.

Generalization is one of the techniques to make data anonymous. This technique replaces the values of quasi-identifier attributes for less specific values, but with consistent semantics, that represent them. Quasi-identifiers are non-explicit identifiers, i.e. zip code, birth date, gender, and attributes are explicit identifiers like address, name. This technique categorizes the attributes, creating a taxonomy of values with levels of abstraction from the particular to the generic level. An example of generalization is replacing quasi-identifiers attributes, like birthdate and gender, with a less specific value, the first name of the person. All those quasi-identifiers are represented by the first name of the person.

The term *De-identification* refers to the process used to prevent someone's identity from being revealed. Identity disclosure occurs when an individual is linked to a particular record in the released database. Attribute disclosure occurs when new information about some individuals is revealed, i.e., the released data makes it possible to infer the characteristics of an individual more accurately than it was possible before the data release.

2.1.2 Metrics

The concept of *k-anonymity* privacy is the most known metric in anonymization [7]. K-anonymity ensures that for each combination of quasi-identifier values, exist at least k records in a set of records, forming an equivalence class. An equivalence class is a set of records that match on the quasi-identifiers. K-anonymity acts with the indistinguishability principle, each record in a set of k-anonym records is indistinguishable from each other considering the combination of quasi-identifier values. This way, k-anonymity can guarantee that each record cannot be linked to an individual by one attacker with a probability higher than $1/k$. K-anonymity can protect against identity disclosure, but cannot prevent attribute disclosure. Due to the limitation of k-anonymity, l-diversity was introduced as an alternative metric.

L-diversity requires that each equivalence class has at least l "well-represented" values for each sensitive attribute. This guarantees that the attacker, even with prior knowledge that allows him to discover an equivalence class of an individual, the attacker cannot infer the sensitive attribute of that individual with a higher probability than $1/l$. The idea of l-diversity is to protect against linkability attacks, in the cases where the attacker can infer sensitive information about the records without identifying them. The "well-represented" values mean that are at least l distinct values for the sensitive attribute in each equivalence class, and makes sure that the less frequent values do not appear too rarely, and the most frequent value does not appear too frequently. L-diversity is insufficient to prevent attribute disclosure and may be difficult and unnecessary to achieve because the l-diversity requirement ensures a diversity of sensitive values in each set, but it does not take into account the semantical closeness of these values.

L-diversity is vulnerable to a *Similarity attack*, this attack happens when one of the sensitive attributes gives a piece of sensitive information to the attacker, even that all sensitive attributes are distinct from each other. Another attack is the *Skewness attack*, it occurs when the distribution of values of the sensitive attribute within a given equivalence class is different from the distribution of the values of the same sensitive attribute over the whole database.

The *t-closeness* metric is a proposal to solve some limitations of l-diversity, such as protection against skewness attacks. For this purpose, t-closeness requires that the distribution of sensitive attributes in each equivalence class is close to the global distribution of the attribute. The maximum distance between the equivalence classes and global distribution is defined by the parameter t. There are some distance measurements to measure the difference between the equivalence class distribution and global distribution, such as variational distance and the Kullback-Leibler distance [8], but these distance measures do not reflect the semantic distance among values. Then, the Earth Mover Distance (EMD) is used to measure the distance that reflects the semantic, which results contain real values in the range [0,1], the higher the distance value, the weaker is the protection. EMD gives us a method for determining the distance between two distributions but does not tell us how to determine the distance between two elements in the distributions. This limits the amount of specific individual information that an attacker can learn.

T-closeness has some limitations like no flexibility to the specification of different levels of privacy for each sensitive attribute, and the EMD function is not adequate for linkability attacks when these are numeric, also to ensure t-closeness could compromise the utility to guarantee the same distribution in all equivalence classes.

Concluding this section, while k-anonymity protects against identifying disclosure, it does not protect against attribute disclosure. L-diversity solves this problem by requiring that each equivalence class has at least l "well-represented" values for each sensitive attribute. The t-closeness solves the protection against skewness attacks limitation of l-diversity by requiring that the distribution of a sensitive attribute in each equivalence class is close to the global distribution of the attribute. The goal of privacy preservation metrics is to measure the level of privacy and the amount of protection offered by privacy-preserving mechanisms.

2.1.3 Location Privacy mechanisms

Several privacy-preserving mechanisms are specific for location data, which include Dummy

Location method, Cloaking, and Mix zones.

The *Dummy Location* method preserves the privacy of the user by using a different location from the real location of the user, called the dummy location [9]. The higher the distance between the dummy location and the real location of the user, the higher is the privacy-preserving level, but the lower is the accuracy of the user's location. This technique is useful if the granularity of the location can be very coarse, e.g. knowing in which city the user is but not the actual street location.

Spatial and temporal cloaking uses the concept of k-anonymity. The idea of a cloaking approach is to compute a cloaked area that covers the user and at least k-1 other users. The real location of the user is not used, only a spatial range (cloaked area) in which the true location information is included. The benefit of using the spatio-temporal cloaking method is that can achieve a good balance between privacy and accuracy, however, it is insufficient to achieve optimal anonymity.

The idea of *Mix zones* consists in the definition of areas as a connected spatial region, where the precise position of a user is not known, is only known that the user is inside of that area [10]. The user identities are mixed using pseudonyms, it is not possible to distinguish between different users. All the pseudonyms of users from a certain mix zone are changed when a user enters that zone. With this method, an observer cannot link users which are going to the mixed zone with the users that are leaving it, however, if the movement profiles between zones are not equal, an attacker with statistical background knowledge can link user identities.

2.1.4 Zero Knowledge proofs

A Zero-Knowledge Proof (ZKP) is a mechanism that allows the sharing of validated data with a third-party without actually sharing the data itself [11]. A prover P can prove his identity to a verifier V without sharing any personally sensitive information, only cryptographic proof that does not leak any data is shared. Considering that the process of generating proof is trusted, ZKP provides privacy in sharing data.

There are three important properties of ZKP: Completeness, Soundness, and Zero-Knowledge [12]. Completeness means that everything true has proof. If the prover is telling the truth, he will eventually convince the Verifier. Soundness means that everything that is provable is true. A Prover can only convince a Verifier if they are telling the truth. And Zero-Knowledge means that only the statement being proven is revealed. The Verifier does not learn anything else about the information of the Prover.

There are two types of ZKP. The non-interactive ZKP does not require an interactive process but requires additional computation to determine the sequence of iterations to validate the proof. The interactive ZKP consists of an interactive process between the prover P and the verifier V with the purpose of V constantly asking questions about the knowledge of P.

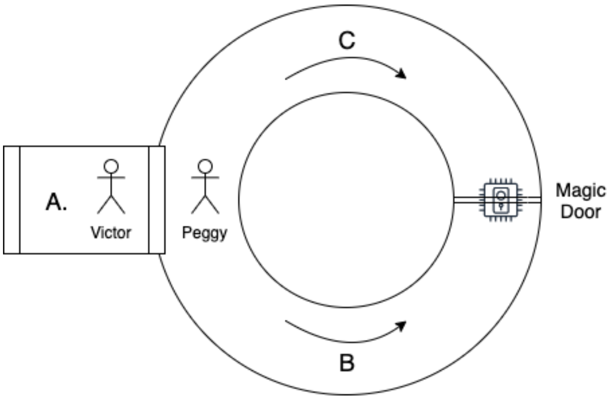


Figure 2.1: Illustrative example of Interactive ZKP with Peggy and Victor.

The ZKP concept of sharing sensitive information without really sharing sensitive information is not intuitive. To explain how the ZKP works we will illustrate with an example of an Interactive ZKP [13] with Peggy (the prover) and Victor (the verifier) using Figure 2.1 to present the scenario. Suppose that Peggy needs to prove to Victor that she can get through a magic door in the middle of a ring tunnel without revealing the secret code that opens the door. Victor and Peggy initially are in position A as shown in Figure 2.1. Then Peggy randomly chooses one path, B or C. Once Peggy is in the tunnel, Victor randomly declares a path, B or C, where Peggy has to go out of the tunnel. If Peggy enters from the same side Victor asks her to go out, then, she does not have to know the secret code. By repeating this test many times, the probability for Peggy to cheat is almost zero, thus Victor can verify that she knows the secret code without her revealing it.

This example of Zero-Knowledge authentication has the Completeness property, if Peggy exits the tunnel from the correct side several times, then Victor will be convinced that she knows the right secret code. Has the Soundness property, because the probability for Peggy to convince Victor that she knows the secret code without knowing it is 0.5^n where n is the number of test repetitions. And Zero-Knowledgeness property, where Victor must not learn the secret code of the magic door.

There are many ways to use Zero-Knowledge Proofs to protect the data. This is useful for any system where requires public verification of potentially private information.

2.1.5 Differential Privacy

Differential Privacy mechanisms quantify the maximum possible information gain by the attacker, which can reduce the risk of the privacy being compromised [14]. The private information is limited and quantified by a *privacy loss parameter*, usually designated epsilon ϵ. The privacy loss parameter consists of quantifying 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 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.

Differential Privacy protects against a wide range of potential privacy attacks, including unknown attacks at the time of deployment. In a given set of individuals, their data will be differentially private even when multiple analyses are performed on that data, as long as each of the analyses satisfies differential privacy. But releasing too many accurate statistics, will have a considerable privacy loss. To avoid this, the number of analysis performed on a specific dataset must be limited to provide an acceptable guarantee of privacy.

A specific example is the following: suppose that you gave differentially private data to Alice and Bob. You use the privacy loss parameter of 1ϵ every time. If they 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 to minimize privacy leakage. Once the data has been leaked, the user information will no longer be private.

2.1.6 Geo-Indistinguishability

The Geo-Indistinguishability mechanism, based on the concept of Differential Privacy, consists of 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 is an interesting privacy-preserving mechanism to protect the location of the users when reporting their location, guarantees that any two locations within a given radius around the user are statistically indistinguishable.

Cunha et al. [15] propose a new mechanism that can be used both in the sporadic scenario and in the continuous scenario, 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.

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 high, the mechanism increases the privacy level. And if the correlation between the reports is low, the mechanism decreases the privacy level.

```
Algorithm 1: Geo Indistinguishability Algorithm
Input: location
begin
    $angle \leftarrow generateRandomAngle()$;
    $radius \leftarrow getSampleRadius(epsilon)$;
    $noisyLat \leftarrow addSampleNoisyDistance(location.getLatitude())$;
    $noisyLon \leftarrow addSampleNoisyDistance(location.getLongitude())$;
    $location.setLatitude(radianToDegrees(noisyLat)$;
    $location.setLongitude(radianToDegrees(noisyLon)$;
    return $location$
```

Algorithm 1 describes the geo-indistinguishability mechanism. It receives a GPS location and injects the noise quantified by the epsilon parameter. Initially, it generates a random angle between 0 and 2*PI, to decide in which direction the location will change. Then, it will calculate the sample radius from the inverse cumulative polar Laplace distribution, using the epsilon parameter. With the random angle and with the calculated sample radius, it adds the sampled noise distance to the original location. To conclude the algorithm, the latitude and longitude need to be normalized to degrees from -180 to 180.

2.2 Location proof systems

There are several systems that provide location proofs. We highlight two early ones, APPLAUS, CREPUSCOLO, and a more recent one, SureThing.

2.2.1 APPLAUS

Privacy-Preserving LocAtion proof Updating System (APPLAUS) allows a device to prove its location by requesting location proofs from nearby mobile devices using Bluetooth. Then these location proofs are updated to an untrusted Location Proof Server that verifies the trustworthiness level of each location proof [16].

One of the benefits of APPLAUS is that it is easy to deploy in Bluetooth enabled mobile devices. It is not necessary to do any changes in the network infrastructure and mobile devices to implement it. In APPLAUS, mobile nodes communicate with an untrusted server through a cellular interface and communicate with other nodes nearby through a Bluetooth interface.

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 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, and the digital certificate validates the signature authenticity.

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 Certificate Authority (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 has to integrate all the information. This allows the system to have the properties of *statistically source location unobservability* and *pseudonym unlinkability.* Source location unobservability is a privacy property that can be satisfied if an attacker cannot determine the real identity of mobile nodes through full observation of the location proof records. And pseudonym unlinkability can be satisfied if any pseudonym of an identity presented in the location proofs records cannot be inferred from one to another.

2.2.2 CREPUSCOLO

CREPUSCOLO (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 [17]. 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. 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.

2.2.3 SureThing

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 [5].

The SureThing system follows the design of APPLAUS and CREPUSCOLO as shown in Figure 2.2, 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 are 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.

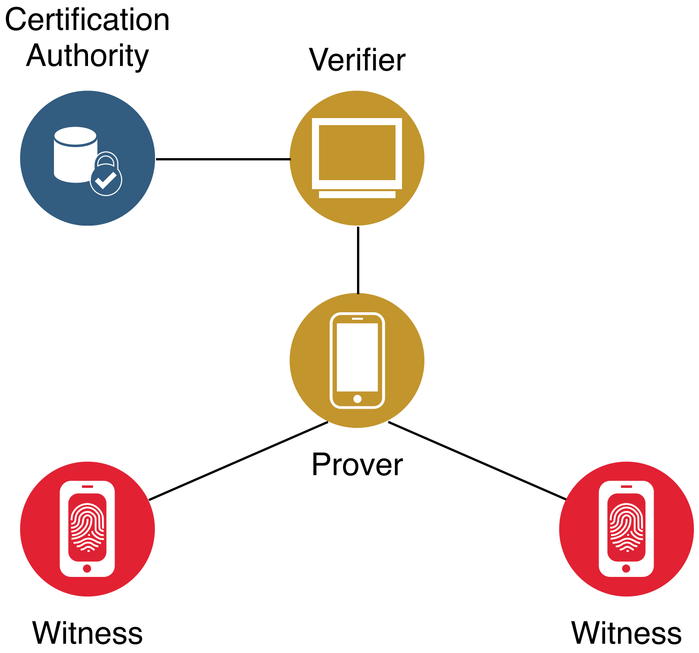


Figure 2.2: Surething architecture (adapted from [5]).

It is up to the Verifier to define the acceptance criteria of 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 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; afterward, 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 the 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.

CROSS (loCation pROof techniqueS for consumer mobile applicationS) is a system derived from SureThing that uses a set of predefined infrastructures to perform location verification with mobile devices [18]. An infrastructure can be a Wi-Fi network to detect whether the user is present or not in the locations and to verify that the user is not spoofing his location. In the Smart Tourism use case, the users can interact with representative city locations, using their mobile devices, using Wi-Fi as an infrastructure for location detection. The information gathered from the user is to check if the user completed any predefined tourism route. The privacy of users is preserved from Wi-Fi access points because the end-user device only collects information passively, as it moves around the itinerary locations. However, the privacy from the server is not preserved from the server, as it is assumed to be trusted and has access to the complete data.

STOP(Secure Transport lOcation Proofs) is another system derived from SureThing that uses mobile devices to improve road transportation inspection [19]. STOP relies on master witnesses for inspection of transportation of materials. STOP uses pseudonyms in the proximity BT protocol. The system uses GPS tracking combined with the location proofing mechanism. To prove the vehicle has been inspected, the location proofs are generated at each inspection. The STOP inspector role is based on the Token Provider entity presented by CREPUSCOLO [17].

2.3 Privacy-preserving location proof systems

In this section we present systems that use location and protect the user privacy: Icelus, MATRIX and Olteanu's framework.

2.3.1 ICELUS

The Icelus system allows estimating the user location and modeling the user movement by combining multiple observations from multiple devices. The location estimation of the user will be more robust than using only a smartphone, because of the scale and diversity of the Internet of Things (IoT) devices. The user can spoof its location by using only its smartphone to prove its location, Icelus takes leverage of the increasing number of IoT devices used by users and those smart environments to locate them [20].

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 does not assume that devices of the users have not been compromised, but that they can be physically stolen, tampered with, or even remotely compromised.

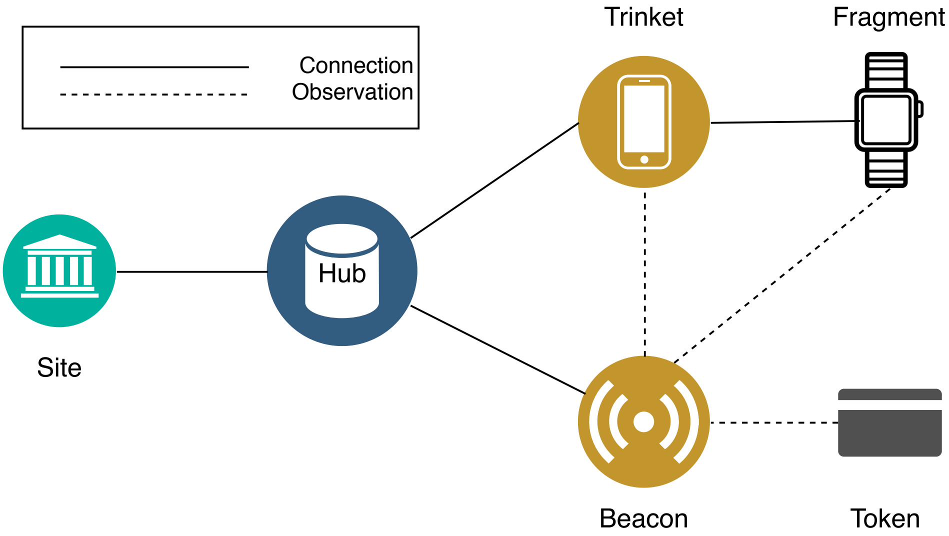


Figure 2.3: Icelus system architecture without privacy (adapted from [20]).

The Icelus system organizes the IoT devices in a hierarchy, as shown in the Figure 2.3. On top of the hierarchy is the hub, which hosts the Icelus service. The Hub receives information from different sources, considering that only the user controls the Hub, to avoid the data be seen by third-parties. Those different sources that send information to the Hub are Trinkets, i.e. smartphones and smartwatches, and Beacons, i.e. third-party devices that observe devices of the users. These devices collect data from smaller devices called Fragments and Tokens, i.e. smart wearables and smartcards. They have to rely on the Trinket to send their data to the Hub.

An ICELUS Avatar is a digital estimate of a user's location or a set of devices in the same area of a given radius. The probability that the user is physically near to that set of devices is called the Confidence Score. The higher the number of devices in a given area, the higher will be the Confidence Score.

When a Trinket reports geolocation information, it generates an Avatar at its location with all Fragments devices that Trinket collected data. Also, when a Token is observed through another device, it is linked to the Avatar at that location. If the token is observed far from the existing Avatars, it creates a new Avatar at the location where it was observed.

Sites are entities that query the Hub about the possibility of the user being physically present at a location. In the registration, it is defined as the places that the Sites can only send queries. The Hub is listening for queries from Sites. When the Hub receives a query, it asks Trinkets to send fresh data, even the devices are reporting fresh information periodically. After receiving the information, the Hub checks if exists an Avatar with confidence score higher in other location, than the rejection threshold configured for the Site.

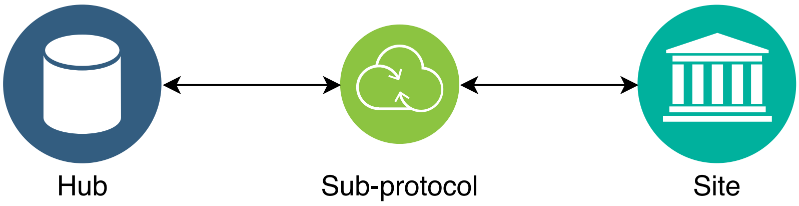


Figure 2.4: Icelus system architecture with privacy (adapted from [20]).

Using the ICELUS Hub to get all geolocation information can bring privacy concerns for the users or third-parties. To mitigate this issue, the Hub and the Site can only learn the distances between the reported locations, and not the precise coordinates. The idea is to check if there is no high probability of a user being in a different location. And to get the set of locations where the user can be, the Sites have to be pre-authorized to do queries about the location of the user. This avoids the leakage of the user's location unnecessarily. To prevent falsely rejecting users, in the case of the user forgets the device at home, and to prevent the stolen devices from being misused, 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.

Icelus uses Homomorphic encryption for a key-pair, and the public key is K and the private key is k. Homomorphic encryption 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. All the location reports arrive at the Hub are encrypted using K. When the Site performs a query, the Hub will start a sub-protocol with the Site, as shown in the Figure 2.4. Then the Hub computes the distances between any location. To calculate their relative positions, the Hub needs to have a pairwise distance between three points.

2.3.2 MATRIX

Narai and Noubir [21] proposed MATRIX, a system to allow end-users to control the visibility of the location and sensor accesses by mobile applications. The MATRIX is designed to protect 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 of how the user's location information is exploited by mobile applications. This system implements a service with a user interface that verifies all locations and sensor accesses by mobile applications and gives timely 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[1], and accelerations from user driving experiments. 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 the real locations of the users, 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.

2.3.3 Olteanu's co-location privacy framework

Most of the online social networks, such as Facebook, give users the functionality of sharing their location jointly with posts and photos. They also provide the ability to mention other users, i.e., 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. This co-location also happens when a location proof witness testifies to the presence of another user.

Olteanu et al.[22] 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 the behavior of users of sharing the location and co-location. 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 behavior of the preferences

[1] Google Maps Directions API. Available: https://developers.google.com/maps/documentation/directions/overview

of the users. The conjoint analysis allows 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.

2.4 Summary

We presented location proof systems, privacy mechanisms, and their use. The discussed works are relevant for the privacy protection. Icelus illustrates how the use of distance instead of coordinates can still provide a 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.

Table 2.1 summarizes the privacy-preserving mechanisms and the attacks they are designed to resist. All of these mechanisms have the objective of preserving identity, location, and time privacy. When an attacker aims to obtain information and to use it for its benefit, the attack can be defined by how it obtained the information, which *method* was used, what *knowledge* is obtained, and which its *target* is. The main targets are the identity, position, and timestamp at that position of the users.

The information can be collected from published or shared location information, history of locations, or distributions. Afterwards, the attacker collects some location information, and the power of this knowledge depends on whether the information has been processed or not. Observed location information is information that has been processed by the user, server, or service provider before being published or shared. Context knowledge is any information that can be linked to other information that the attacker has to reveal the location information of a user, e.g., the location restrictions of an area, relationships between different users, or the relationships between the identity of the users and their location.

As shown in table 2.1, *context linking* is one of the most important attacks for location privacy-preserving mechanisms. The context linking attack occurs in a system that protects the unlinkability of certain data (e.g. identifiers, pseudonyms, locations) that does not leak information that would enable an adversary to link these items, however, the attacker can take advantage of hints from the context in which the system operates. The contextual knowledge is easy to combine with other relevant information like with the observed location information for a localization attack or with precise location information for an identity attack. The *probability* method is based on collecting statistics about environment contexts, this is similar

Location Privacy Preservation Mechanism		**Adversary and attack**		
		Knowledge	*Attack method*	*Target*
Anonymization	*Mix-zone*	Observed; Context	Context linking	Identity
	Zero-Knowledge Proof	Observed	Probability	Identity
Obfuscation	*Dummy locations*	Observed	Not specified	Position
	Spatial obfuscation	Observed	Context linking	Position
	Time obfuscation	Observed	Context linking	Time
	Spatiotemporal obfuscation	Observed; Context	Context linking	Position; Time
	Geo-Indistinguishability	Observed	Probability	Position

Table 2.1: Comparison of location privacy preservation mechanisms (based on [23]).

to the contextual information, however, this method exploits probability theory to predict the locations.

Geo-Indistinguishability is a generic mechanism, independent of the specific user or the area where it is used, that allows a user to disclose enough location information while simultaneously preserving some privacy. The only parameter that matters is the level of privacy, i.e., the level of accuracy of the location data. In the next chapter we propose privacy protections, and one of them is based on geo-indistinguishability mechanism.

Chapter 3

Design & Implementation

There are many mechanisms to preserve identity and location privacy. In this chapter, we propose mechanisms for witness protection for the Surething location proof system. The main goal is to provide privacy protections to the witnesses that are volunteering to testify the presence of another user at a given place, using mobile devices. Section 3.1 presents the attacker model. Section 3.2 presents the high-level approach. Section 3.3 presents the architecture of the system, including the components and how they interact, and the privacy mechanisms. Finally, in Section 3.4, we present the software architecture and the implementation of the privacy module.

3.1 Attacker model

The *attacker model* considers the following attackers:

1. An *external attacker* that is not registered to SureThing, but can eavesdrop communications between the prover and witnesses and try to determine their identities and location, in personal area network (Bluetooth), local area network (Wi-Fi) and in the wide area network (Internet).

2. A *malicious user* registered on the system that performs proof stealing, i.e., steals and uses location proofs from other users.

3. A *malicious user* registered on the system and interested in attacking the location privacy of other users, requests location proofs from witnesses to collect accurate location information of them.

4. A *malicious user* with access to the proof history that can query and ask for sets of proofs.

Considering the attacker model presented, the system can mitigate and prevent some attacks. Regarding the attacker (1), the external eavesdropper, if the attacker tries to intercept the communications between the prover and witnesses, it is protected by a secure HTTPS channel. 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 a validation of the certificate using the CA.

The attacker (2) tries to steal proofs from other users. This is a special case of a *replay attack*. It can be detected by verifying the identity of the signature, the freshness, and the unique identifier.

Attacker (3) is a user that is requesting location proofs with the purpose, not of getting proof of locations, but to try and track the location of other users that are acting as witnesses. This attacker is the main concern in this work.

Finally, the attacker (4) that can query the ledger and ask for sets of past proofs, to track the location of users that requested proofs or that acted as witnesses. To mitigate this attack is necessary to use pseudonyms for the users in the system, as proposed in the next section.

3.2 Approach

In this work, we focus only on location privacy protection, but the protection of the identity of the users is planned for future work. Our system extends a previous version of SureThing with mechanisms to address the location privacy protection of the users.

3.2.1 Identity Protection

The idea consists of the users having pseudonyms to protect the identity of the witnesses. Based on the paper [24], the witnesses would use an ephemeral identifier (EphID), instead of using their real identity. This ephemeral identifier is generated pseudo-randomly by the smartphone, derived from the secret key (SK) of the smartphone, and it is rotated periodically. The CA is the only entity that can identify the real identity of the witness. So, when the prover requests location proofs to the witnesses, the witnesses will generate a new ephemeral identifier and it will send the location proof to the prover. When the witness generates the new ephemeral identifier, it will advertize it to the CA. For the prover, the witness identifier is a random ID that is changed frequently, so the prover cannot map it to the identity of the witnesses or to track them. After collecting all the location proofs from the witnesses, it will send to the Verifier to validate the claimed location of the prover. The Verifier will verify the authenticity and freshness of the location proofs of the witnesses, so it requests to the CA the public keys of the witnesses. The CA will linked the ephemeral identifier to the real identity of the witnesses and it will return the public keys of the witnesses.

3.2.2 Location Protection

It allows the users to protect their privacy when they are helping other users, acting as witnesses at a location. The system has three mechanisms to ensure the privacy of the location of the users: *geo-indistinguishability*, *response throttling* and *static position.*

The *geo-indistinguishability* mechanism injects noise in the location data of the user when it is shared with other users to obfuscate its real location. The *static position* mechanism allows the witness to reply to location proofs requested by other users with a static position, to avoid sending other locations of the same area. The *response throttling* mechanism limits the number of replies to location proof requests to other users when the user is acting as a witness following the Differential Privacy principle: the more information is shared, the more noise needs to be injected. These mechanisms are expected to reduce the leakage of location information of the witnesses.

The system has a fundamental trade-off between the levels of privacy protection and usability and accuracy of this system. The system must be capable of protecting the privacy of the users and at the same time, keep the system accurate and usable. There is also a trade-off between a 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.

3.3 Architecture

This section presents the redesigned SureThing architecture and its main components and important algorithms, as well as the interactions between the components. The main components of the system are still the Verifier, the Prover, and the Witness. The Figure 3.1 presents the redesigned Surething architecture with the main components, as well, the witness privacy protection.

3.3.1 Prover

At 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

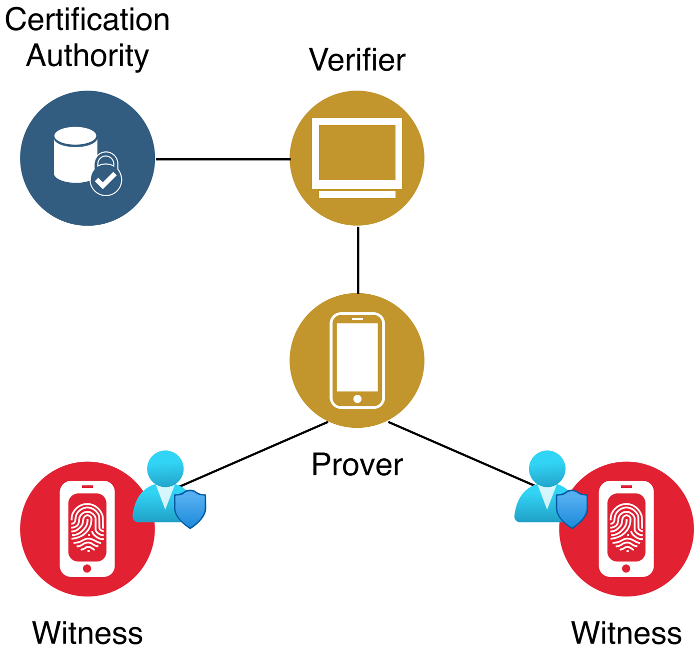


Figure 3.1: Redesigned Surething architecture.

information, the Prover starts the witness discovery process [25]. After a witness is detected, the Prover will send it 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.

3.3.2 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 location proof is replied by the Witness to the Prover with the following attributes:

- **Prover ID** - identifier of the user who wants to prove its location;
- **Witness ID** - identifier of the user who testifies the presence of the prover at a certain place;

- **Location data of the Prover** - latitude, longitude;
- **Location data of the Witness** - latitude, longitude;
- **Nonce** - a random number and/or timestamp used once to ensure freshness;
- **Signature of the proof** - a signature from the Witness to guarantee authenticity.

These location proofs generated, by the witnesses, are going to be validated by the Verifier, to prove the presence of the Prover at a certain place.

We created a new module called *Privacy* in the mobile application that implements 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 3.3. The epsilon is the noise parameter that controls the quantity of noise is introduced in the location data. The epsilon value can be defined between 0 and 1, where the smallest number of epsilon results in more noise injection than a higher epsilon. Section 5.2 shows results with variation of epsilon. The function that introduces noise, it uses the Planar Laplace mechanism on the mobile application to report to the Prover the obfuscated location of the Witness rather than its real location.

3.3.3 Verifier

The Verifier has the role of validating the claimed location of the Prover by checking the proofs of the witnesses. The following algorithm 2 describes the process of validation of the Witnesses proofs collected using the GPS technique.

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.

Algorithm 2: Verifier Tasks Algorithm

```
Input: A collection of Witness Proofs
begin
    foreach $witnessProof$ in the $witnessesProofs$ do
        $proof \leftarrow witnessProof.proof$;
        $witnessID \leftarrow witnessProof.witnessID$;
        $proofTechnique \leftarrow witnessProof.proofTechnique$;
        if proofIsCorrectlySigned(proof, witnessID) then
            if ValidNonce(proof) then
                if proofTechnique = GEO then
                    $AppendToWitnessesLocation(proof.getWitnessGeoLocation())$;
                if proofTechnique = WIFI then
                    /* Not relevant for this work */
                if proofTechnique = BEACON then
                    /* Not relevant for this work */
    if proofTechnique = GEO then
        $clientGeoLocation \leftarrow proof.getClientGeoLocation()$;
        $witnessGeoLocation \leftarrow CalculateAvgWitnessLocation(witnessesLocation)$;
        if locationsCloseToEachOther(clientGeoLocation, witnessGeoLocation) then
            $clientPlace \leftarrow GetSymbolicPlace(clientGeoLocation)$;
            $witnessPlace \leftarrow GetSymbolicPlace(witnessGeoLocation)$;
            if clientPlace = witnessPlace then
                return ACCEPT
            else
                return REJECT
    if proofTechnique = WIFI then
        /* Not relevant for this work */
    if proofTechnique = BEACON then
        /* Not relevant for this work */
```

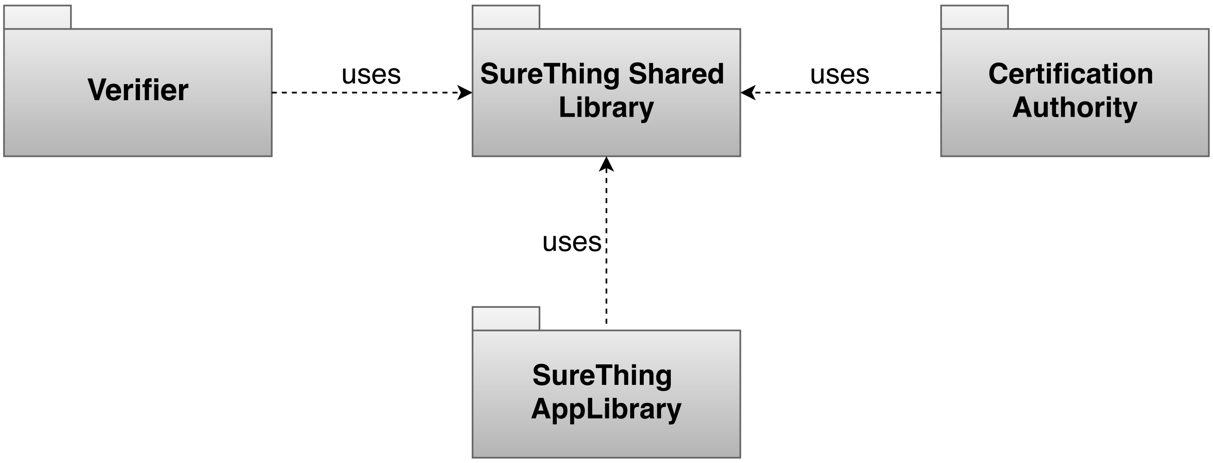


Figure 3.2: SureThing Project structure

3.4 Implementation

We developed our solution in the Surething location proof system. The solution contains one shared library for the mobile application (Prover and Witness) and the Verifier, and one library for mobile application, as shown in Figure 3.2. The SureThing Shared Library contains classes and methods that are used by all entities such as Location Proof objects. The SureThing App Library contains code to take advantage of SureThing's functionalities such as location retrieval or Bluetooth communication.

This system was developed using Java programming language with the IDE Android Studio[1] and was build using the Gradle[2] tool, because it is integrated with the Android Studio. For the prototype, we implemented the mobile application on mobile devices with the Android Operating System (OS). The communication between mobile devices uses Bluetooth without pairing.

The CA is used to register the users in the system and generates a public key certificate for the user. In our prototype, we deactivated the verification of the signatures of the location proofs, however, we assumed that are all signed and verified, and the CA is a trusted authority.

To implement the module of privacy, we had to change and to adapt the algorithm of the Verifier from the previous prototype to ensure that the location proofs are validated, properly.

In the SureThing App Library, we add the privacy module with the Geo-Indistinguishability class. This module is used only by the witness, and this class is used in the ReceiverGPSLocation class, which receives the GPS location and then verifies how much noise the witness wants to put in its location.

The Geo-Indistinguishability implementation was based on Location Guard[3]. It is a browser

[1] Android Studio (IDE). Available: https://developer.android.com/studio/index.html

[2] Gradle Build Tool. Available: https://gradle.org/

[3] Location Guard browser extension. Available: https://github.com/chatziko/location-guard

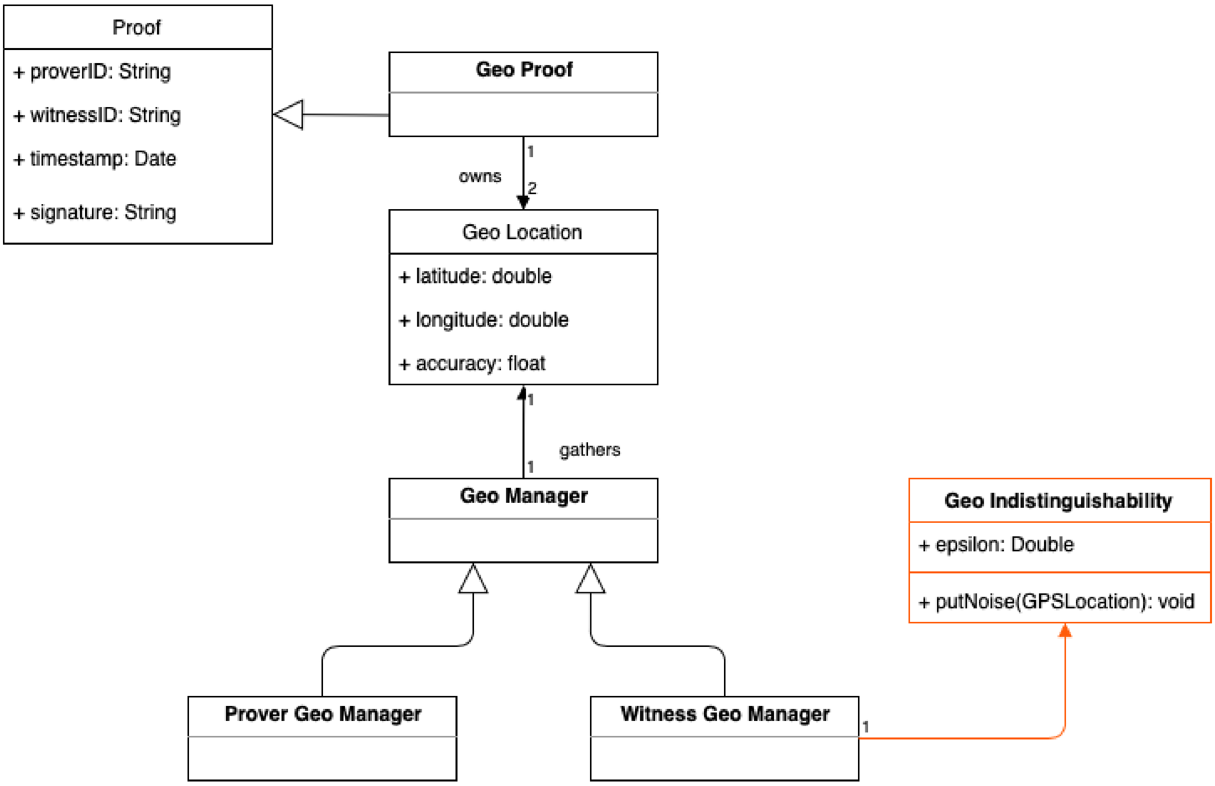


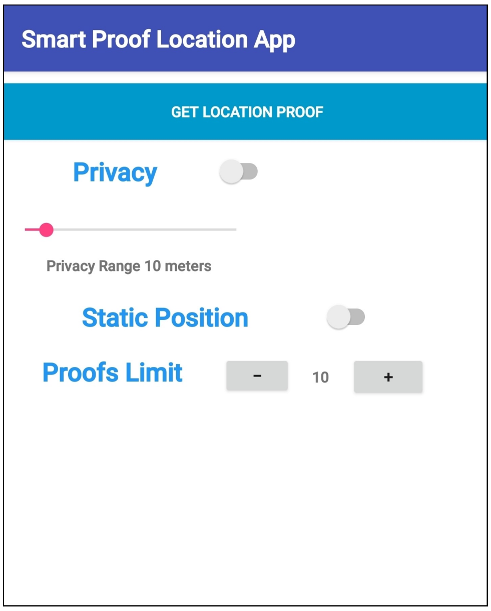
Figure 3.3: Class diagram with the main classes used by both Prover and Witness.

extension that allows protecting your location while using location-aware websites, by adding controlled noise to it. It implements a location obfuscation technique based on adding noise from a 2-dimensional Laplace distribution (Geo-indistinguishability). Their code was in the JavaScript programming language, we tested and it worked as it should, so we decided to apply it in our system in Java.

Figure 3.3 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 divided into 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 geolocation data. And it is divided into 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 obtains 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.

In this system, the mechanism for the users to use pseudonyms was designed, but was not implemented. To simplify the implementation and testing of this work, we do not use the certificates and we do not validate them.

The noise and the limit of replies are presented to the users as the configuration of the *level of privacy*. Different users have different willingness to share resources and risk appetite. The privacy options allow users 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.



Smart Proof Location App

Privacy

Privacy Range 10 meters

Static Position

Proofs Limit 10

Figure 3.4: Privacy options screen design.

Figure 3.4 illustrates the privacy options to be presented to users, with all the functionalities mentioned before. The user as a witness can configure 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. This prototype is for testing purposes, not for end users.

3.5 Summary

In this chapter we presented the approach followed to add privacy preserving mechanisms to SureThing. We described the three mechanisms - geo-indistinguishability, response throttling and static position - and also detailed the choices made for the software implementation. In the next chapter we present the use case used for evaluation of the implemented system.

Chapter 4

Use Case

In this chapter, we present the use case that we chose to test and evaluate our system. We also present the data set and the tool used to simulate the scenario in operation.

4.1 Public Transport

In the last years, the population in urban areas has been increasing and it has a 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 expected that until 2050, the number of people moving per kilometer in the urban environment will triple [1]. A system of transport plays an important role in large metropolitan areas, therefore is necessary to guarantee the best conditions of this service to offer a better experience to the users. The population can choose its type of transport, the car is the most common one, although choosing public transports instead of a private car will have an impact on the emissions of the CO_2 emissions [26]. The quality of public transports can help in the reduction of private cars in the cities, and the cities become more eco-friendly.

Over the last years, there has been a huge step in such services with the help of technology that is evolving even more (e.g. smartphones). Sharing and pooling services are the new ways to travel in the city, electrical car sharing, electrical bike sharing, and the implementation of multi-mode transportation services with buses and trains, where the customer is able to use multiple types of transportation when it is traveling. So, convenience for the users is very important, these transportation services should have easy access, attractive offers, good integration in the routine of the users. This multimodal service allows the users to share different vehicles for

[1] The global public transport awards, 2015. https://www.uitp.org/awards/

different purposes, like for short trips they can rent an electric scooter or bike, and for different destinations, the user can choose which transports wants to use according to the time, the cost, and ecological. The big challenge with multi-modal transport is the tracking of users for ticketing, i.e known when to charge fees and to whom the fees should be given.

Initially, the tickets were made of paper, but through the years, evolved to systems with electronic smartcard support and, more lately, for the use of the smartphone. Nowadays, there are proximity cards for electronic tickets, which allows saving 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 on 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 [27] 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 public transport.

4.2 Ticketless Transport

Considering these systems and the different ticket services, we consider a new system for public transports, a Ticketless transport system for public transportation using a provable location to enable efficient boarding and accurate billing. Most of the works are focused on the operator perspective to have a better system with reduced costs. In this work, we are focused on the user community perspective and its use of location proofs. The user community is very important in a crowdsensing system, according to the survey results [28]. Crowdsensing is a data collection and sharing performed by a large number of regular users [29]. Calado[28] 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 to the goal.

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. The operator with this system encourages spontaneous use of transportation without being necessary proximity to a validator and can dematerialize the tickets, instead of its used virtual tickets. The ticket services, using this system allows to speed up the process of acquiring and validating tickets, reducing the waiting lines and saving time to the passengers.

When the system detects the entry of the user on the bus, it starts to determine the route of the user. When the user leaves the bus, the system stops detecting the user and charges the price of the trip. In case, the bus is empty, the user can collect location proofs from a *witness master* that is the driver of the bus. From the perspective of the user, it has an 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. However, the user has to place full trust on the transport operator, and on the accurate detection of entries and exits from transports. To counter-balance the operator's detection system, we propose a user community-driven audit system. In it, the users periodically collect proofs of location as they walk around the city. These proofs are collected with the collaboration of other users that act as witnesses. The proofs of location are stored in a ledger that assures that they cannot be tampered. This is another component of the SureThing project, but outside the scope of this work. So, at the end of a operation period, let us say 1 day, for example, there are two records of the user activity. The record collected by the transports operator and the location proofs. Now, an audit can be performed for the two logs. Is the location proof log of the user consistent with the claimed trips made on the transport? Or can the user prove that he is being billed for trips he did not do? Hopefully, such a crowd-based system can keep the operators in check and let them continue to be honest as the trip interactions become more and more spontaneous. The ticket validation system by using location proofs is a complement, managed by users themselves, to allow them to have validation of the official ticket validation system. This is a cross-checking system, to make the overall system more transparent and involve the community to share location information with the user that pretends to prove its location. Using a mobile application, the user can configure the controls of privacy protection and getting context about the current trip, as shown in the Figure 4.1.

4.3 Simulator

Now, that we presented the concept of the ticketless transport system, we have to consider how to evaluate the system to know if it is feasible and practical, 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 set of bus trips between the Marquês de Pombal stop and the Campo Grande stop in Lisbon, Portugal; collected in

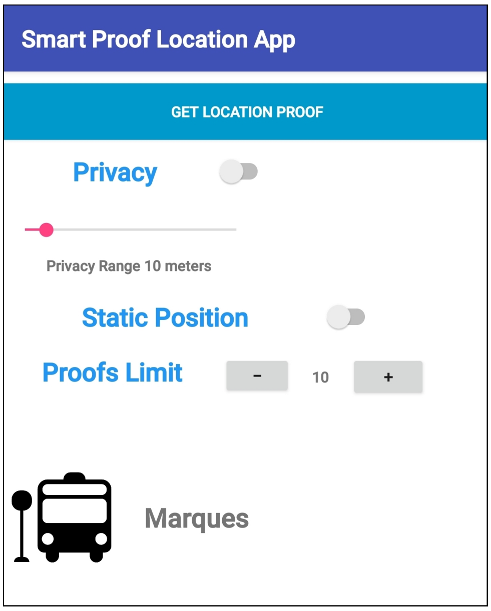


Figure 4.1: Ticketless transport mobile application screen design.

previous work by Santos [19]. We evaluate our system by simulating and analyzing the users using the Ticketless transport system in different experimental scenarios. We can interact with users on the bus but also on nearby streets or vehicles next to the bus.

The simulation was developed using AnyLogic software with the 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 chose 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 represents 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 the bus stop of Av. Fontes Pereira de Melo, Picoas and it ends at the bus stop of Saldanha. Figure 4.2 presents the route map of the bus, we decide on this route because of the dataset available from Santos[19]. At the beginning of the simulation, the bus starts to do the route and the users will start appearing in the bus stops waiting for the bus to pick them up. When the user enters the bus, the mobile application validates the beginning of the trip by proving it through location proofs. Then, when the user leaves the bus, the mobile application validates the end of the trip through location proofs.

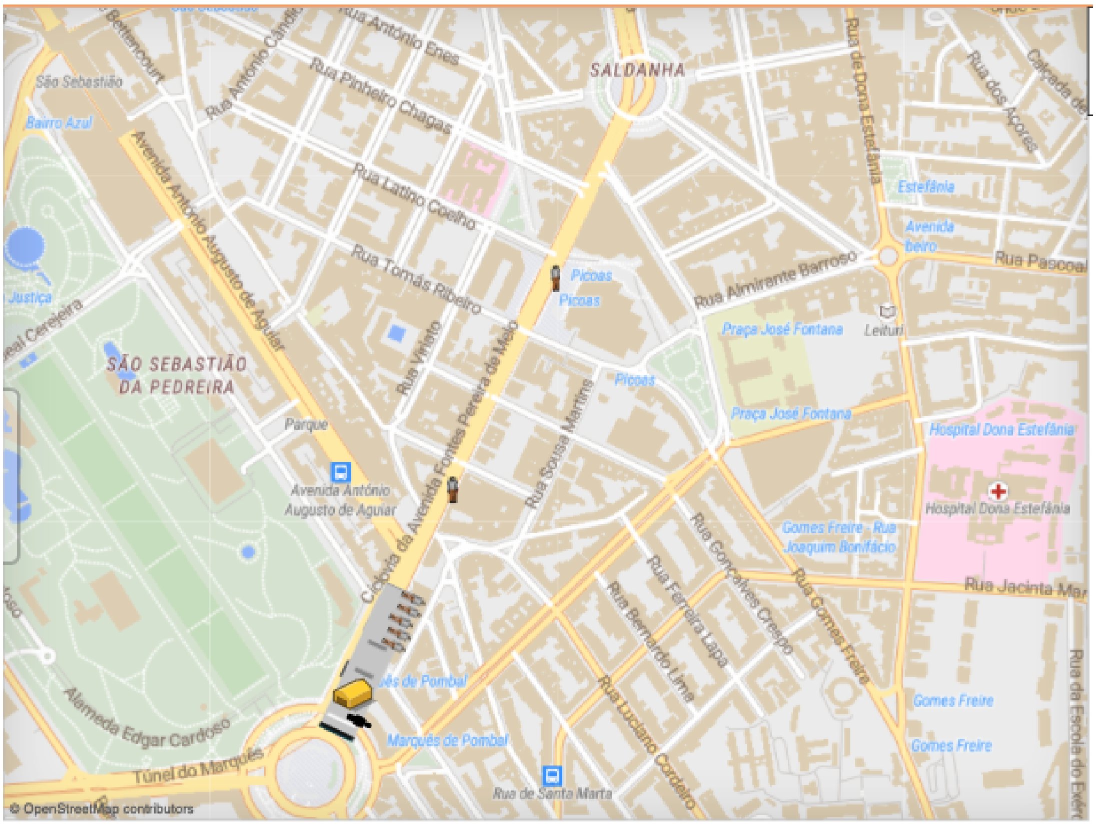


Figure 4.2: The simulation route of the bus in the Anylogic simulator.

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.

4.4 Simulation setup

For all the simulations to run, we have to define parameters to set up the simulator. Each experiment has a 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, which 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 has a parameter for the destination stop determined by the probability of one of the stops. To verify 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 at that bus stop and not in other bus stops since the distances between the bus stops are 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 at each stop is between 1 (one) minute to 3 (three) minutes.

4.5 Summary

In this chapter we presented the Ticketless public transport use case, and the dataset and simulation tools, that will allow us to evaluate the proposed system. In the next chapter we will present experiments made to find the ideal privacy parameters for this use case.

Chapter 5

Evaluation

Following the implementation and design of the SureThing witness protection mechanisms, we evaluate them with a set of experiments. We explain each experiment and present and discuss the evaluation results. We focused the evaluation on three important questions:

- How feasible and pratical is the system for the users?
- What are the ideal privacy parameters for the public transport use case?
- Are the privacy mechanisms suitable to protect the witnesses against adversaries?

We provide answers to these questions in the Discussion section, at the end of this chapter. First we present the experiments on the Baseline, with Geo-indistinguishability, with Response throttling, and with Attack resistance.

5.1 Baseline 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 in the best conditions possible. These measures were obtained after 30 experiments. The average of users using the bus during one hour in each experiment is 290 users. Figure 5.1 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 the exit of the bus. This is the best scenario possible, however, there were claimed locations denied, especially 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

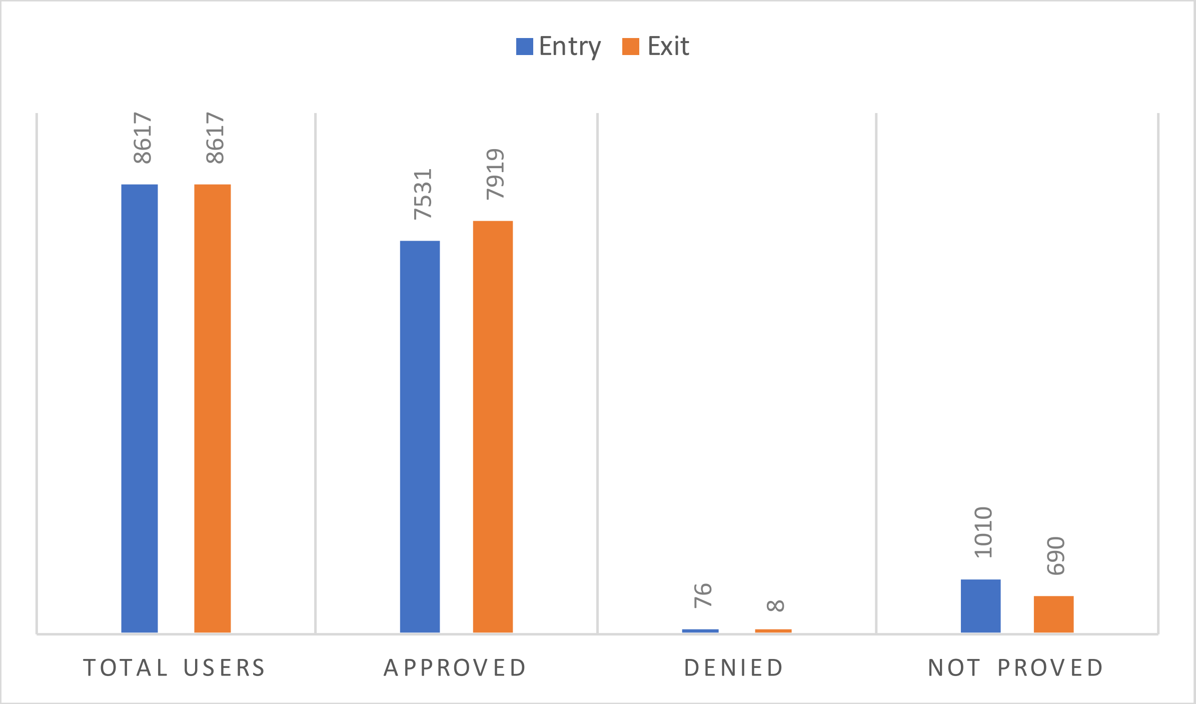


Figure 5.1: Absolute number of proofs performed for the baseline experiments.

location, they do it one at a time, the simulator cannot do multi-threading in this process. For example, when there are too many users to enter in the bus, and they are proving their start of the trip, the simulator can verify one at a time, and once all the users are inside of the bus, it will continue its route, and the users will not be able to prove it.

5.2 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 the 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. 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 still is usable. Our results are presented in the Figure 5.2.

The noise parameter can vary 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 by 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 a high percentage of approved proofs because the average distance between the bus stop and the witness location

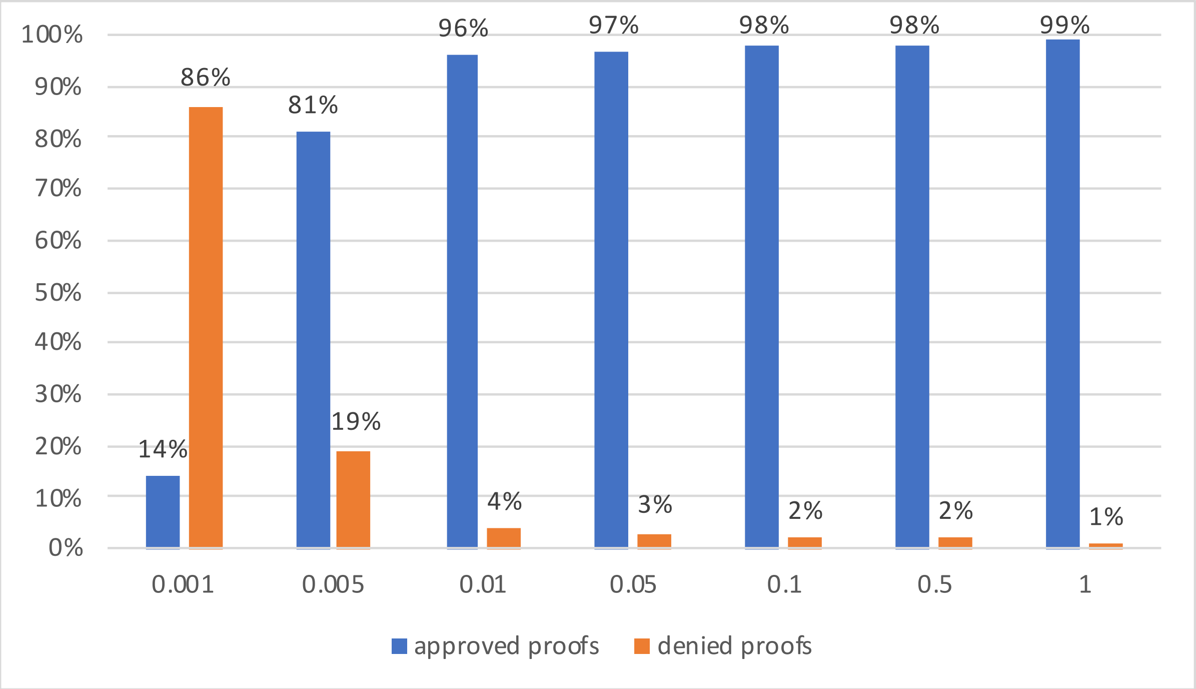


Figure 5.2: Acceptance rate of proofs with different noise parameters.

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

is smaller than the threshold of the bus stop. Table 5.1 presents the average distance between the bus stop location and the 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 to 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.

5.3 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 5.3 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 no big impact on the acceptance rate but has a good impact on 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 to their location data. Also, we can observe from Table 5.2 that using the same noise parameter, and reducing the maximum of proof responses from the witnesses, it will result in a bigger distance between the witnesses and the bus stop, with fewer witnesses available it is less accurate to prove the claimed location.

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

Table 5.2: The limit of proof responses and the corresponding average distances between the witnesses and the bus stop.

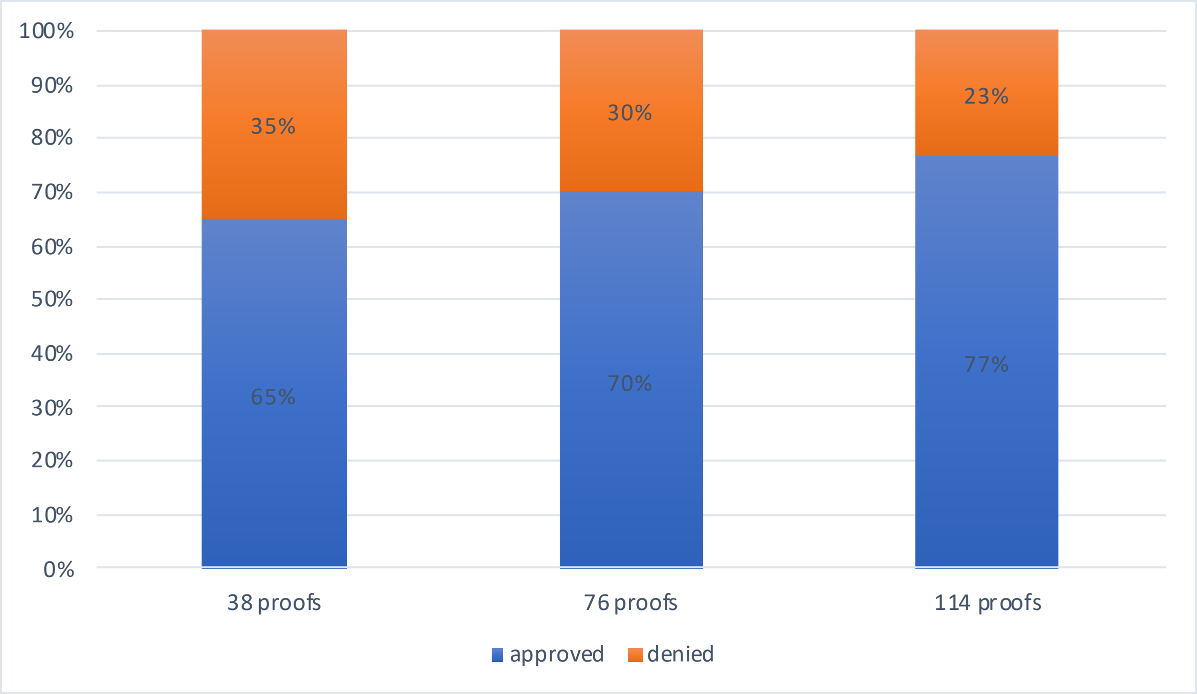


Figure 5.3: Acceptance rate of proofs with different proof response limits.

5.4 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, these values ensure 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 on the bus trying to figure out where the witness was by collecting the maximum possible information about the real location of the witness. 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 the location of the user. 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.

Our results are presented in Figures 5.4, 5.5 and 5.6. We can observe that the attacker could collect location information about the target users. In figure 5.4, 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 to location proofs. In the figure 5.5, the attacker collected less number of location

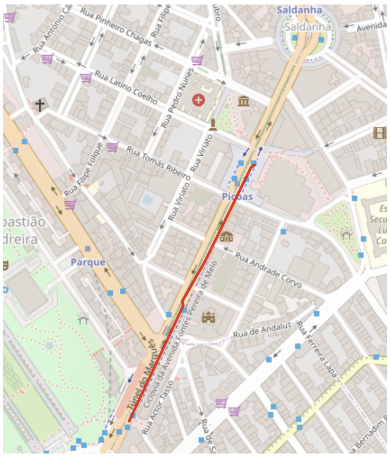


Figure 5.4: Trace collected with high number of available proof responses from the target.

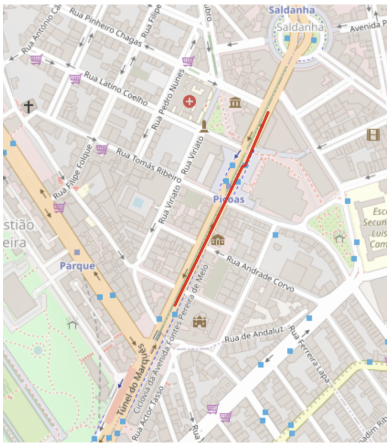


Figure 5.5: Trace collected with medium number of available proof responses from the target.

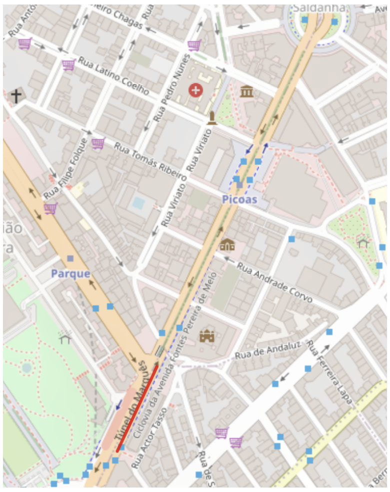


Figure 5.6: Trace collected with low number of available proof responses from the target.

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 5.6.

5.5 Discussion

We evaluated our system in a public transport use case, in the baseline scenario to use location proofs to cross-validate the entry and exit of the bus. As expected, the acceptance rate of location proofs at the entry and exit was high, but in a scenario where not all users have the application and the GPS signal error is more significant, which means that the total of the position error is the sum of the signal error and privacy error, therefore the acceptance rate of location proofs would be less than the scenario seen in experiment 5.1. The number of location proofs not proved could be the result of only one user who can prove its location one at a time. This limitation is in the simulator, using multi-threading would reduce the number of location proofs not proved.

Using the best scenario as a baseline, we evaluated the effect of injecting error in the location data using geo-indistinguishability. According to the algorithm in Section 3.3.2, we know that the closer the noise parameter gets to zero, the more private the location is. As seen in the experiment 5.2, the lowest noise parameter (0.001) in the experiment results in an average distance of 253.90 meters, which is higher than the threshold of the bus stops of the simulator

(100 meters), therefore the percentage of denied proofs is superior to the percentage of approved proofs. We conclude that from the different values of the noise parameter, the ideal noise parameter for the use case is 0.005. This value allows the system to be usable and gives protection to the witnesses. For each different situation or application, there is a specific ideal noise parameter.

With the ideal noise parameter, we evaluated the limit of location proof responses of the witnesses. And as expected in the experiment 5.3, the lower the number of location proofs, the higher is the average distance of the location proofs from the bus stop. This mechanism reduces the leakage of the witness location information and we conclude that the ideal parameter to limit the location proof responses of the witnesses is 76 proofs, because it still has a good acceptance rate of location proofs and it can reduce the leakage of information from the witness.

To define the ideal parameters, we need to understand what is the impact of compromising the accuracy of the system by using the privacy mechanisms for each situation. For some applications that are not critical to the need of having an accurate location, we can increase the privacy levels of the system to protect the witnesses. For the more critical applications, that are necessary to have a high accuracy of the location, we need to use weaker privacy levels. We can define the different privacy levels for the users to choose for each different application. In our situation, is very important to make sure that the users can collect location proofs to cross-validate the trips and at the same time protect the privacy of the witnesses. Also, to define the privacy parameters, we need to ensure that the distance range of the claimed location of the Prover is not higher than the distance range necessary to validate the proof.

Given the ideal parameters of the privacy mechanisms, we evaluated the behavior of the system against a possible attacker. A malicious prover attempted to obtain information about the real location of other users, knowing a *priori* that there is a high probability of the users being on the bus. As seen in the experiment 5.4, we conclude that the limitation of location proof responses of the witnesses had a big impact on protecting the location of the witnesses. The noise parameter did not have a significant impact on protecting the witness because, in the simulation, the malicious user can infer that the user is using the bus and so are the witnesses. If the user is not on the bus, then the location proofs are collected from other users, not on the bus, so the attacker would not have discovered information about the location of the witness. And the information that the attacker collected does not describe the complete path of the witness, it depends on the privacy levels of the witness to reduce the leakage of location information. The attacker can gain an advantage in retrieving witness locations when he knows for sure that the user is on the bus. The noise parameter can protect the witnesses in many different scenarios but it can be compromised, the attacker can exploit additional context and perform *probability* attack to predict the locations of the witness.

Chapter 6

Conclusion

In this dissertation, 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. And in some situations, the user can also opt for reporting a fixed location. The witnesses have a 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.

First, we researched the different privacy mechanisms, how they work, and which attacks they can defend against or not. Also, we researched the location proof systems that already have implemented privacy mechanisms. This information gave us an overview of what problems exist in privacy and what solutions exist to solve it.

6.1 Achievements

We implemented an improved prototype of Surething, by developing an Android mobile application, and a verifier server application. This system is focused on using the GPS location technique, and communicating between users with Bluetooth connections with no pairing. The Geo-Indistinguishability mechanism was implemented in the mobile application to allow the witness to inject noise in its location data to protect its real location. With this prototype, we chose public transportation as our use case to test and evaluate our system.

We developed a simulator to evaluate our system in terms of feasibility and accuracy, in a scenario of public transports. First, we evaluated the system in the best scenario where all users on the bus had the mobile application that could participate in the system and none of the witnesses was using privacy protection. Then, we compared the different values of noise injected in the location data of the witnesses and showed that the noise parameter the more close is to 0, the more noise is injected in the location data. Also, we compared the different number of replies to ad-hoc witness requests with the same noise parameter, and we concluded that limiting the number of proof responses from the witnesses, it will decrease the percentage of approved proofs. 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 the number of location proof responses is 76.

Finally, we evaluated the behavior of the system when a malicious prover attempted to obtain information about the real locations of the witness, and we conclude that the attacker can collect some location information of the witness, but only when the attacker knows for sure that the witness is on the bus. And the leaked location information depends on the privacy levels of the witness.

6.2 Future Work

Regarding future work, this prototype is very flexible and can be adapted for many different applications, beyond the public transportation use case. The parameters of privacy can be defined depending on the system and the level of privacy that wants to implement. Location-based services are applied in many applications but due to privacy concerns, the users do not feel safe to use it.

This system can be improved in different ways. In the evaluation, we could add experiments on actual deployment, to confirm the simulation results. Also, add other use cases, with different datasets, where the location of users and witnesses has more variability, and, as such, other values can be selected for the privacy protection parameters. The implemented privacy mechanisms help to decrease the leakage of private information, although it is not enough, it is possible to implement more privacy mechanisms to protect the witnesses. Also, it is important to protect the privacy of the identity of the users, by using pseudonyms and changing them after each interaction with other users. This system to work properly must have a balance between security and accuracy, and that is adapted for each situation. Co-location protection is a problem that this system does not solve and it has a big impact in the privacy of the users.

Another idea for the future is to implement a privacy mechanism in the Prover that can prove its claimed location without revealing it to the Verifier, this can be done by using a zero-knowledge proof mechanism and/or multi-party computation mechanism.

Bibliography

[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)*, pages 812–817. IEEE, 2015.

[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)*, pages 1018–1031. IEEE, 2018.

[4] S. Tayeb, M. Pirouz, G. Esguerra, K. Ghobadi, J. Huang, R. Hill, D. Lawson, S. Li, T. Zhan, J. Zhan, and S. Latifi. Securing the positioning signals of autonomous vehicles. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 4522–4528, 2017. doi: 10.1109/BigData.2017.8258493.

[5] 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.

[6] X. Wang, A. Pande, J. Zhu, and P. Mohapatra. Stamp: Enabling privacy-preserving location proofs for mobile users. *IEEE/ACM Transactions on Networking*, 24(6):3276–3289, December 2016. doi: 10.1109/TNET.2016.2515119.

[7] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115, April 2007. doi: 10.1109/ICDE.2007.367856.

[8] D. Johnson and S. Sinanovic. Symmetrizing the kullback-leibler distance. *IEEE Transactions on Information Theory*, 2001.

[9] Y. Wang, Z. Cai, Z. Chi, X. Tong, and L. Li. A differentially k-anonymity-based location privacy-preserving for mobile crowdsourcing systems. *Procedia Computer Science*, 129:28 –

34, 2018. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2018.03.040. URL `http://www.sciencedirect.com/science/article/pii/S1877050918302618`. 2017 Internation Conference on Identification, Information and Knowledge in the Internet of Things.

[10] I. J. Vergara-Laurens, L. G. Jaimes, and M. A. Labrador. Privacy-preserving mechanisms for crowdsensing: Survey and research challenges. *IEEE Internet of Things Journal*, 4(4): 855–869, Aug 2017. ISSN 2372-2541. doi: 10.1109/JIOT.2016.2594205.

[11] A. A. Rasheed, R. N. Mahapatra, and F. G. Hamza-Lup. Adaptive group-based zero knowledge proof-authentication protocol in vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–15, 2019. ISSN 1558-0016. doi: 10.1109/TITS.2019.2899321.

[12] F. Benhamouda, S. Krenn, V. Lyubashevsky, and K. Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In G. Pernul, P. Y A Ryan, and E. Weippl, editors, *Computer Security – ESORICS 2015*, pages 305–325, Cham, 2015. Springer International Publishing.

[13] D. Sudha and A. Usha. Zero knowledge protocol for authentication and key exchange – a survey. In *International Journal of Engineering Trends and Applications (IJETA)*, volume 5, 2019.

[14] 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*, 2017.

[15] 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)*, pages 1–8, Oct 2019. doi: 10.1109/CCCS.2019.8888111.

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

[17] 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.

[18] G. A. Maia, R. L. Claro, and M. L. Pardal. Cross city: Wi-fi location proofs for smart tourism. In L. A. Grieco, G. Boggia, G. Piro, Y. Jararweh, and C. Campolo, editors, *19th International Conference on Ad Hoc Networks and Wireless (AdHoc-*

Now), 2020. URL `http://web.tecnico.ulisboa.pt/miguel.pardal/www/pubs/2020_Maia_Pardal_AdHocNow_CROSSCity.pdf`.

[19] M. L. P. Henrique F. Santos. Operation STOP: secure itinerary verification for smart vehicle inspections. In *INForum*, Guimarães, Portugal, Sept. 2019.

[20] 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*, ACSAC '16, pages 251–264, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4771-6. doi: 10.1145/2991079.2991090. URL `http://doi.acm.org/10.1145/2991079.2991090`.

[21] S. Narain and G. Noubir. Mitigating location privacy attacks on mobile devices using dynamic app sandboxing. *Proceedings on Privacy Enhancing Technologies*, 2019(2):66–87, 2019.

[22] A.-M. Olteanu, M. Humbert, K. Huguenin, and J.-P. Hubaux. The (co-) location sharing game. *Proceedings on Privacy Enhancing Technologies*, 2019(2):5–25, 2019.

[23] B. Liu, W. Zhou, T. Zhu, Y. Xiang, and K. Wang. *Location Privacy-Preserving Mechanisms*, pages 17–31. Springer Singapore, Singapore, 2018. ISBN 978-981-13-1705-7. doi: 10.1007/978-981-13-1705-7_2. URL `https://doi.org/10.1007/978-981-13-1705-7_2`.

[24] Troncoso, Carmela, Payer, et al. Decentralized privacy-preserving proximity tracing, 2020.

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

[26] S. Melzer. *Integrated Affordable Mobility Solutions in a Smart Neighborhood*, pages 367–376. Springer International Publishing, Cham, 2016. ISBN 978-3-319-25715-0. doi: 10.1007/978-3-319-25715-0_21. URL `https://doi.org/10.1007/978-3-319-25715-0_21`.

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

[28] 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.

[29] 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)*, pages 593–598. IEEE, 2014.

Appendix A

Simulator Architecture and Design

A.1 Simulator Architecture

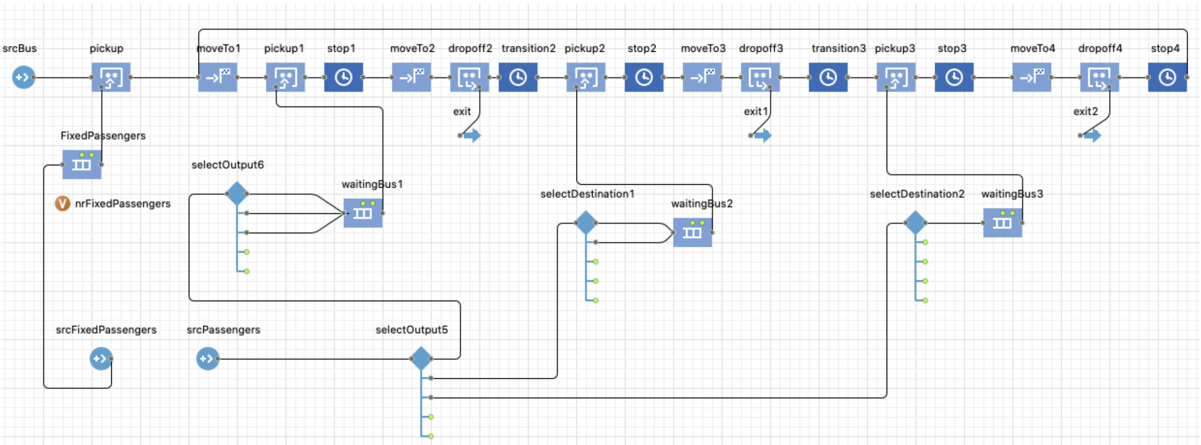


Figure A.1: Main simulator architecture of the ticketless public transportation.

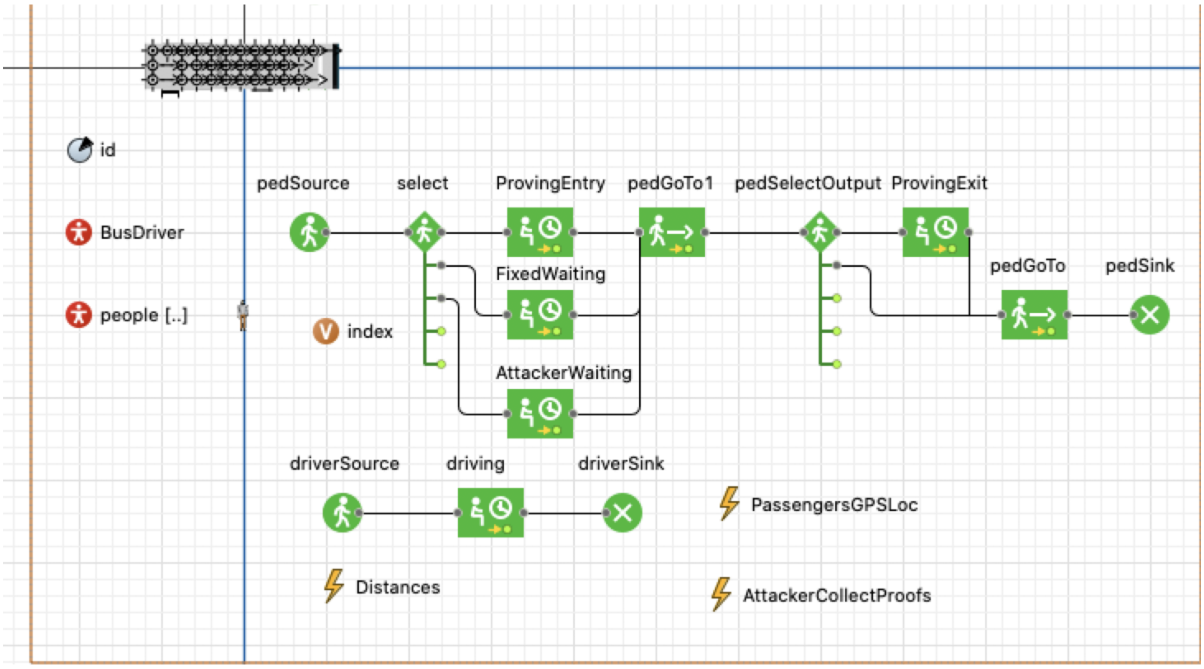


Figure A.2: Bus simulator architecture of the ticketless public transportation.